

В.П. Иванов  
А.С. Батраков

ТРЕХМЕРНАЯ  
КОМПЬЮТЕРНАЯ  
**ГРАФИКА**

*Под редакцией  
академика РАН  
Г.М.ПОЛИЩУКА*



МОСКВА  
**РАДИО И СВЯЗЬ**  
1995

ББК 32.98  
И 18  
УДК 681.3.06

Иванов В.П., Батраков А.С.  
И 18      Трехмерная компьютерная графика / Под ред. Г.М. Полищук. — М.: Радио и связь, 1995. — 224 с.: ил.  
              ISBN 5-256-01204-5.

Рассмотрены основные принципы формирования трехмерных изображений на компьютере и практическое применение трехмерной компьютерной графики: распознавание образов, моделирование трехмерных сцен, архитектурное проектирование, анимация и т.д. Приведены сведения о программной поддержке геометрического моделирования и компьютерного синтеза изображений, примеры программ на Турбо Паскале. Оригинальность и практическая значимость предложенных подходов подтверждена независимой конкурсной экспертизой компании Hewlett Packard.

Для программистов, специалистов в области компьютерной графики, может быть полезна студентам вузов.

2404040000-004  
И----- КБ-5-672-94      ББК 32.98  
046(01)-95

Издание для специалистов

Иванов Василий Петрович  
Батраков Анатолий Семенович

## ТРЕХМЕРНАЯ КОМПЬЮТЕРНАЯ ГРАФИКА

Заведующий редакционным отделом Ю.Г. Ивашов

Редактор Е.Н. Гарденина

Художественный редактор Т.Г. Родина

Обложка художника В.Г. Ситникова

Корректор Т.Г. Тертышная

ИБ № 2615  
ЛР № 010164 от 04.01.92

Подписано в печать с оригинал-макета 14.11.94.  
Формат 60x88/16. Бумага офс. № 2. Гарнитура таймс.  
Печать офсетная. Усл.печ. л. 13,72. Усл. кр.-отт. 14,09.  
Уч.-изд. л. 15,16. Тираж 20 000 экз. Изд. № 23865.  
Зак. № 984.      С-004.

Издательство "Радио и связь," 101000 Москва, Почтамт, а/я 693

Производственно-издательский комбинат ВИНИТИ  
Люберцы 10, Московской области, Октябрьский просп., 403

ISBN 5-256-01204-5

© Иванов В.П., Батраков А.С., 1995  
© Оригинал-макет.  
Иванов В.П., 1995

## ВВЕДЕНИЕ

Быстрое развитие вычислительных средств, расширение их возможностей являются главным фактором все более широкого внедрения их в различные сферы научной и практической деятельности. Исключительно интенсивно развивается направление компьютерного синтеза изображений. Можно считать, что в настоящее время в основном сформировалась новая отрасль информатики — машинная графика. Ее можно определить как науку о математическом моделировании геометрических форм и облика объектов, а также методов их визуализации.

Интерес к синтезу изображений объясняется высокой информативностью последних. Информация, содержащаяся в изображении, представлена в наиболее концентрированной форме, и эта информация, как правило, более доступна для анализа: для ее восприятия получателю достаточно иметь относительно небольшой объем специальных знаний.

Стремление визуализировать информацию наблюдается практически во всех сферах деятельности человека. И с начала использования ЭВМ возникла проблема представления получаемых данных в виде изображения. На начальном этапе программными средствами формировались различные символические изображения: диаграммы, графики, условные схемы, планы и т.п. Вывод такой информации осуществлялся непосредственно через алфавитно-цифровое печатающее устройство.

Формирование машинной графики как самостоятельного направления относится к началу 60-х годов, когда Сазерлендом был создан первый специализированный пакет программного обеспечения машинной графики. В 60-е годы были сформулированы принципы рисования отрезками, удаления невидимых линий, методы отображения сложных поверхностей, определены методы формирования теней, учета освещенности сюжета. Первые работы были в основном направлены на развитие векторной графики, т.е. рисования отрезками. Из первых работ этого направления наибольшее влияние на развитие векторной графики оказали алгоритмы Брезенхема.

В 70-е годы значительное число теоретических и прикладных работ было направлено на развитие методов отображения пространственных форм и объектов. Это направление принято называть *трехмерной машинной графикой*. Математическое моделирование трехмерных сюжетов требует учета трехмерности пространства предметов, расположения в нем источников освещения и наблюдателя. Потребностями трехмерной машинной графики обусловлено большое количество работ по методам аппроксимации и представления сложных поверхностей, отображению узоров из них, генерирации текстур, рельефа, моделированию условий освещения. Одновременно значительное внимание уделялось улучшению качества синтезированных изображений, повышению уровня их реалистичности; сглаживанию погрешностей, возникающих в результате аппроксимации геометрической формы реальных тел и пространственной дискретизации изображения. Методы трехмерной машинной графики позволяют визуализировать сложные функциональные зависимости, получить изображения проектируемых, еще не созданных объектов, оценить облик предмета из недоступной для наблюдения позиции и решить ряд других аналогичных задач.

В 80-е годы сфера применения машинной графики существенно расширилась. Появилась база для широкого использования методов машинной графики — персональные компьютеры. В результате машинная графика стала инструментом не только инженеров-исследователей, но и специалистов многих других отраслей, непосредственно не связанных ни с техникой, ни с программированием. Увеличение памяти и скорости обработки информации в персональных ЭВМ, создание видеокомплексов с широким набором программ машинной графики, возможность

управления ими в диалоговом режиме способствуют дальнейшему расширению применения машинной графики.

Одним из наиболее новых направлений в машинной графики является разработка принципов и методов формирования реалистичных изображений, т.е. изображений, которые могли бы наблюдаться визуально или регистрировались бы оптическими, фотографическими или оптико-электронными устройствами. Потребность в создании реалистичных изображений возникает в таких областях, как дизайн, машиностроительное и архитектурное проектирование, реклама и т.п. В ряде применений реалистичность используется как средство повысить эмоциональное воздействие изображения, как, например, в рекламе, в других же отраслях – как средство оценки качества выбранных решений (в архитектурном планировании, дизайне и др.), в третьих – как средство создания реалистичной обстановки (видеотренажеры, системы распознавания образов). В системах формирования реалистичных изображений должна обеспечиваться передача всей совокупности изобразительных свойств: объемность, расположение предметов в сюжете, полутона, цвет, текстура поверхности. Теоретические исследования последних лет в значительной мере были направлены на разработку методов и средств отображения всей совокупности характеристик сюжетов в синтезируемом изображении.

Быстрое расширение функциональных возможностей современной вычислительной техники создало базу для развития систем машинной графики, обеспечивающих отображение динамических сюжетов, в которых изображения последовательно сменяют друг друга. Из таких систем можно отметить три группы: системы графического моделирования для наглядного представления процессов в химии, медицине, астрономии и др.; системы имитации динамических ситуаций (например, тренажеры); системы получения двухмерных и трехмерных изображений для телевидения и кино. Именно в развитии этих систем наиболее рельефно проявляются трудности и проблемы трехмерной машинной графики. Для них требуется не только высокая точность моделей, но и чрезвычайно высокая производительность вычислительных средств. Создание и внедрение в практику транспьютерных вычислительных систем создают в настоящее время реальные перспективы дальнейшего развития возможностей машинной графики.

В трехмерной машинной графике реалистичных изображений ведущую роль занимает метод трассирования лучей, в основе которого лежит воспроизведение в математической форме хода лучей в реальных устройствах формирования изображений. Этот метод преимущественно рассматривается и в данной книге. Другим методам уделяется сравнительно небольшое внимание. Метод трассирования лучей иногда в литературе называют *силовым*. Это справедливо лишь в том, что при использовании этого метода можно не искать какие-либо специфические пути синтеза изображений, приспособливаться к специфике конкретного сюжета. Задача синтеза будет решена, но, возможно, некоторый объем вычислений будет избыточным. Отметим, однако, что если синтезируется реалистичное изображение, то избыточные затраты бывают весьма небольшими. Они велики лишь тогда, когда синтезируется условно-объемное, стилизованное изображение. Наряду с этим метод обратного трассирования лучей имеет несомненные достоинства: универсальность, простота его физической трактовки и, что весьма важно, – возможность распаралеливания вычислений. Это практически позволяет проводить синтез для каждой точки изображения независимо от других.

Глава 1 написана канд. техн. наук проф. А.С.Батраковым, гл. 2–6 и приложение – канд. техн. наук В.П. Ивановым.

# Глава 1. ФИЗИЧЕСКИЕ ПРОЦЕССЫ ФОРМИРОВАНИЯ ИЗОБРАЖЕНИЙ И ИХ МОДЕЛИРОВАНИЕ

## 1.1. МОДЕЛИ В МАШИННОЙ ГРАФИКЕ

Системы машинной графики отображают отработанную информацию о процессах или объектах в виде синтезированного изображения на экране дисплея. В отличие от фотографических, телевизионных, оптико-электронных и других аналогичных систем для систем машинной графики источником входной информации являются не сами физические процессы или объекты, а математические модели. Эти модели в общем случае представляют упорядоченную совокупность данных, числовых характеристик, параметров, математических и логических зависимостей, отображающих структуру, свойства, взаимосвязи и отношения между элементами объекта, а также между объектом и его окружением. Математические модели обычно являются обобщенными и предназначаются для описания определенного класса объектов. При вводе конкретных значений параметров система машинной графики на основе общей модели синтезирует изображение и визуализирует его.

Специфика моделей и обрабатывающих программ в трехмерной машинной графике обусловлена необходимостью передавать ощущение глубины пространства, пространственной формы и структуры объектов. Весьма существенное влияние на построение математической модели оказывает требуемый уровень соответствия (подобия) синтезируемого изображения визуально наблюдаемой картине или изображению, которое формируется съемочной системой. В дальнейшем изображение, которому ставится в соответствие синтезирование, будем называть *оригиналом*. При оценке степени соответствия синтезированного изображения оригиналу целесообразно использовать, как в кинематографии и телевидении, три уровня подобия: физическое, психофизическое (физиологическое) и психологическое.

В применении к машинной графике физическое подобие означает, что синтезированное изображение по основным физическим характеристикам повторяет оригинал. Физическое подобие устанавливается на уровне трех групп характеристик: геометрических (пространственных), яркостных (энергетических) и временных. При физически точном подобии определенные характеристики синтезированного изображения должны полностью соответствовать характеристикам оригинала либо быть пропорциональными им.

При психофизическом (физиологическом) подобии соответствие устанавливается на уровне зрительных ощущений. В силу ограниченных возможностей зрительного аппарата наблюдатель при некотором уровне искажений не может ощутить разницы между синтезированным изображением и оригиналом, так как зрительные ощущения идентичны, хотя яркость, форма и цвет одноименных участков неодинаковы.

Психологическое подобие предполагает, что по общему восприятию синтезированное изображение и оригинал являются схожими. Вследствие этого синтезированное изображение обеспечивает формирование у наблюдателя вполне определенного суждения о реальном объекте или сюжете, хотя синтезированное изображение и оригинал значительно различаются по физическим характеристикам.

Синтез изображений на уровне психологического подобия широко используется в машинной графике. Приемы передачи объемности и глубины пространства во многих элементах подобны используемым в изобразительном искусстве [3]. К таким приемам относятся формирование очертаний предмета и

характерных линий, передающих его объемность, загораживание одних предметов другими, и как следствие, наложение изображений переднего плана на изображения предметов в глубине (оверлэпинг), передача перспективы. Эти приемы находят в машинной графике свое математическое выражение: это алгоритмы формирования определенных сечений объемной фигуры, удаления невидимых линий изображения, формирования перспективной проекции. К числу приемов передачи объемности и глубины пространства относятся также яркостное или цветовое выделение некоторых элементов изображения, моделирование теней. Указывая на связь методов трехмерной машинной графики с изобразительным искусством, отмечаем, что современная машинная графика имеет в своем арсенале заметно меньше средств передачи объемности и глубины пространства, чем искусство, поскольку алгоритмическая и программная реализации приемов искусства связаны с серьезными затруднениями.

К числу задач синтеза, решаемого на уровне психологического подобия, можно отнести формирование трехмерных контурных изображений (чертежей), выполненных в выбранной аксонометрической проекции. Такие изображения используются в машиностроительном проектировании и позволяют конструктору (при наличии достаточного опыта) представить вид проектируемых изделий, провести анализ принятого технического решения и его корректировку.

В развитии современной трехмерной машинной графики значительное место занимают вопросы синтеза изображений на уровнях физического и психофизического подобия. Необходимость синтеза таких изображений возникает при решении ряда практических задач, в частности при имитации визуально наблюдаемой обстановки в видеотренажерах, создании банков эталонных изображений для автоматических распознающих систем, оценке внешнего вида и эстетических свойств проектируемых изделий или объектов и т.п.

В обобщенном виде процедуру синтеза изображения можно представить следующим образом:

$$G_{си} = A_{мг} G_{мод}, \quad (1.1.1)$$

где  $G_{мод}$  – математическая модель наблюдаемого объекта или сцены;  $A_{мг}$  – оператор преобразования (обрабатывающая программа) системы машинной графики;  $G_{си}$  – двухмерное изображение, сформированное системой машинной графики.

Синтезированное изображение должно сопоставляться с оригиналом при визуальном наблюдении или аппаратурной съемке:

$$G_{оп} = G_{виз} = A_{виз} G_o \text{ или } G_{оп} = G_{вых} = A_c G_o, \quad (1.1.2)$$

где  $G_o$  – поле излучения объекта;  $A_{виз}$ ,  $A_c$  – соответственно операторы преобразования при визуальном наблюдении или в съемочной аппаратуре;  $G_{виз}$ ,  $G_{вых}$  – соответственно изображения, полученные при визуальном наблюдении или съемке;  $G_{оп}$  – изображение-оригинал.

При физическом подобии синтезированное изображение и оригинал должны быть идентичны друг другу:

$$G_{си} \Leftrightarrow G_{оп}. \quad (1.1.3)$$

Это означает, что в синтезированном изображении и оригинал должны быть одинаковыми (или пропорциональными) ракурсы наблюдения, яркостные (или цветовые) соотношения, тени, текстура наблюдаемых поверхностей, масштабы и другие характеристики.

Условия физического подобия при синтезе принципиально могут быть выполнены, если оригинал получен съемочной аппаратурой. Для фотографической, телевизионной, оптико-электронной аппаратуры разработаны достаточно точ-

ные математические модели, описывающие оператор преобразования  $A_c$ . Для выполнения синтеза необходимо сформировать модель объекта (сцены)  $G_{\text{мод}}$ , в которой должны быть отражены источники освещения, трехмерная геометрическая форма и взаимное положение объектов, характеристики отраженного и собственного излучений. Кроме того, должно быть задано положение съемочной аппаратуры. Следует отметить, что при разработке модели сцены обычно учитывается процесс преобразования информации в съемочной аппаратуре. Если аппаратура не воспроизводит какие-либо особенности наблюдаемых объектов (мелкие детали, текстура, цвет), то учет их в модели является избыточным.

Если синтезируется изображение аналогично визуально наблюдаемому, то условия физического подобия практически не могут быть реализованы в полной мере. Системы машинной графики синтезируют двухмерное изображение, в котором теряется информация, определяемая бинокулярным зрением. Если объект представляет собой объемное тело, то у наблюдателя образуются два сетчаточных образа в правом и левом глазу. Оптическая система каждого глаза строит на сетчатке изображение. Центром проекции является центр зрачка глаза. Вследствие пространственного разноса глаз их оптические оси сходятся на объекте под некоторым углом, называемым углом конвергенции. Поэтому сетчаточные изображения в правом и левом глазу не тождественны. Совместное действие этих сетчаточных изображений формирует у наблюдателя зрительное восприятие объема и пространства.

Система машинной графики может синтезировать изображение, соответствующее только одному сетчаточному изображению, т.е. система способна воспроизвести лишь условия монокулярного наблюдения. Объемность изображения, их пространственное положение при этом воспринимаются на синтезированном изображении благодаря линейной перспективе, загораживанию одних объектов другими, характеру теней и изменению тона (или цветовых оттенков) по полю изображения. Существенное значение для восприятия объема и пространства имеет предшествующий опыт наблюдения, благодаря которому наблюдатель непроизвольно “достраивает” объемную структуру наблюдаемой сцены.

Таким образом, синтезированное компьютерное изображение может соответствовать оригиналу при визуальном наблюдении только на психофизическом уровне.

Максимальное приближение к оригиналу при этом, как и в случае аппаратурной съемки, можно обеспечить, если математическая модель сцены и обрабатывающая программа точно передают условия освещения, геометрическую форму объектов, их взаимное положение, обеспечивают условия энергетического подобия, передают размер и положение теней и другие особенности реальной сцены.

## 1.2. ИСТОЧНИКИ ОСВЕЩЕНИЯ

Для синтеза реалистических изображений необходимо и достаточно точно смоделировать условия освещения. Это предполагает задание координат условия освещения и их характеристик. К числу характеристик, требующихся для решения задач синтеза изображений, относятся лучистый поток, создаваемый источником излучения, спектральная характеристика, пространственное распределение излучения (индикатриса излучения). Для площадных излучателей необходимо также задавать поверхностную (габаритную) яркость.

Для источников, имеющих непрерывный спектр излучения, вместо

спектральной характеристики широко используется числовой параметр – *цветовая температура*. Это температура черного тела, при которой его излучение совпадает по цвету с излучением данного реального источника [37]. Наряду с этим очень часто цветовую температуру определяют как температуру черного тела, при которой относительные спектральные характеристики черного тела и реального излучателя совпадают в пределах заданного допуска.

Относительная спектральная характеристика источника излучения  $y(\lambda/\lambda_m)$  с цветовой температурой  $T_c$  описывается нормированной функцией Планка:

$$y(x=\lambda/\lambda_m) = 142,32 x^{-5} [\exp(4,9651/x)-1]^{-1}, \quad (1.2.1)$$

где  $x=\lambda/\lambda_m$  – безразмерный параметр, характеризующий в относительной форме длину волны излучения;  $\lambda_m$  – длина волны, соответствующая максимуму спектральной плотности излучения.

Величина  $\lambda_m$  определяется законом Вина:

$$\lambda_m [\text{мкм}] = 2898/T_c, \quad (1.2.2)$$

где  $T_c$  – цветовая температура по шкале Кельвина.

Спектральная плотность любой энергетической величины может быть выражена через  $y(x)$ . В частности, для лучистого потока

$$\Phi_\lambda = \Phi_{\lambda_{\max}} y(\lambda/\lambda_m), \quad (1.2.3)$$

где  $\Phi_{\lambda_{\max}}$  – максимальное значение спектральной плотности лучистого потока.

Используя соотношение (1.2.2), можно найти связь между лучистым потоком  $\Phi$  и величиной  $\Phi_{\lambda_{\max}}$ :

$$\begin{aligned} \Phi &= \int_0^\infty \Phi_{\lambda_{\max}} y(\lambda/\lambda_m) d\lambda = \Phi_{\lambda_{\max}} \lambda_m \int_0^\infty y(\lambda/\lambda_m) d(\lambda/\lambda_m) = \\ &= \Phi_{\lambda_{\max}} \lambda_m \int_0^\infty y(x) dx \approx 1,52 \Phi_{\lambda_{\max}} \lambda_m. \end{aligned}$$

Во многих случаях для источника излучения известен лучистый поток  $\Phi$ .

Тогда для  $\Phi_{\lambda_{\max}}$  можно получить

$$\Phi_{\lambda_{\max}} = 0,658 \Phi / \lambda_m. \quad (1.2.4)$$

При известных величинах  $\Phi_{\lambda_{\max}}$  и  $T_c$  несложно при необходимости определить лучистый поток  $F[\text{лм}]$  в светотехнической системе единиц:

$$F = 683 \int \Phi \lambda v_\lambda(\lambda) d\lambda = 683 \Phi_{\lambda_{\max}} \int y(\lambda/\lambda_m) v_\lambda(\lambda) d\lambda,$$

где  $v_\lambda(\lambda)$  – относительная видность монохроматического излучения на длине волны  $\lambda$ .

## 1.2.1. ИСТОЧНИКИ ИСКУССТВЕННОГО ОСВЕЩЕНИЯ

Для создания искусственного освещения, как правило, используются электрические источники света, излучение которых возникает в результате прямого или опосредованного преобразования электрической энергии.

К наиболее распространенным электрическим источникам света относятся

лампы накаливания, люминесцентные и газоразрядные. В лампах накаливания излучающим элементом является вольфрамовая нить, помещенная в стеклянный баллон с инертным газом и разогреваемая электрическим током до высокой температуры (2500...3000 К). Спектр излучения ламп накаливания непрерывный. Максимум спектральной плотности излучения приходится на ближнюю инфракрасную область (1,0...1,2 мкм). Видимое излучение составляет не более 10...12% лучистого потока, причем основная часть приходится на оранжево-красную часть спектра. А в ультрафиолетовой и инфракрасной областях спектр излучения ламп накаливания вследствие поглощения в стеклянном баллоне заметно отличается от спектра излучения черного тела с соответствующей цветовой температурой.

В источниках излучения с лампами накаливания обычно используются отражающие и светорассеивающие элементы. Расположение и форма этих элементов в значительной мере определяют индикаторы излучения источников освещения. Задать индикаторы излучения, как правило, можно лишь приближенно.

Люминесцентные лампы в настоящее время очень широко используются в источниках освещения общественных, выставочных, торговых и других помещений. Они выполняются в виде цилиндрической трубы, заполненной аргоном с парами ртути.

В люминесцентных лампах используется электрический разряд в парах ртути низкого давления, из-за чего возникает мощное излучение на нескольких длинах волн в ультрафиолетовой и видимой частях спектра. Внутренняя поверхность трубы люминесцентной лампы покрыта тонким слоем люминофора, который, поглощая коротковолновое излучение, излучает сплошной спектр. Подбором люминофора можно в широких пределах менять форму спектральной плотности потока излучения, создавая имитацию той или иной цветовой температуры. Промышленностью выпускаются люминесцентные лампы нескольких типов: дневного света марки ЛД с  $T_c = 6000$  К; белого света ЛБ с  $T_c = 3450$  К; холодного белого света ЛХБ с  $T_c = 2800$  К; используются также лампы марок ЛЕК с  $T_c = 4000$  К и ЛХЕ с  $T_c = 5200$  К.

Источники освещения с люминесцентными лампами обычно представляют собой сборку из нескольких ламп с общим отражателем и светорассеивателем. При моделировании такой источник освещения можно представлять в виде плоской светящейся площадки.

К газоразрядным относятся лампы, в которых используется непосредственное излучение электрического разряда в газе. В ртутных лампах высокого давления (до 1 МПа) основная энергия при электрическом разряде сосредоточена на длинах волн  $\lambda = 404,7; 435,8; 546$  и  $579$  нм, т.е. в сине-зеленой части спектра. Отсутствие в излучении ртутных ламп спектральных составляющих в красной области спектра приводит к заметным искажениям цветопередачи. Поэтому в ртутных лампах применяются специальные меры по улучшению спектрального состава излучения. В ртутно-люминесцентных лампах используются стеклянные колбы, покрытые изнутри люминофором с достаточным излучением в длинноволновой части видимого спектра. В металлогалоидных лампах к парам ртути добавляются галогениды (обычно йодиды) натрия, индия, таллия и других металлов, дающие излучения в желто-оранжевой области спектра.

В качестве источников очень большой яркости используются также ксеноновые лампы высокого и сверхвысокого давлений. Излучение этих ламп определяется дуговым разрядом в ксеноне. Спектральный состав излучения близок к дневному излучению.

В ряде систем в качестве источников освещения используются лазеры. Отличительными особенностями этих источников являются высокая монохроматичность и направленность излучения. В большинстве практических задач можно считать, что лазеры излучают на фиксированной длине волны, а диаграмма направленности (индикатриса излучения) может быть аппроксимирована гауссоидой или близкой к ней функцией.

### 1.2.2. ХАРАКТЕРИСТИКИ ИЗЛУЧЕНИЯ ИСТОЧНИКОВ ПРОСТОЙ ГЕОМЕТРИЧЕСКОЙ ФОРМЫ

В инженерной практике обычно используют определенную схематизацию источников излучения. Реальный источник заменяют совокупностью излучающих поверхностей или их участков, в пределах которых можно полагать их яркость постоянной. В процессе моделирования исключают из рассмотрения те поверхности, вклад которых в суммарное излучение незначителен, и не учитывают взаимное влияние излучающих поверхностей.

Процедура расчета освещенности от источника произвольной формы включает определение освещенности от элементарной площадки с последующим интегрированием по излучающей поверхности. При наличии нескольких источников производится суммирование освещенностей, создаваемых каждым из них. Таким образом, для расчетов в принципе достаточно задать форму излучающих поверхностей и их яркость.

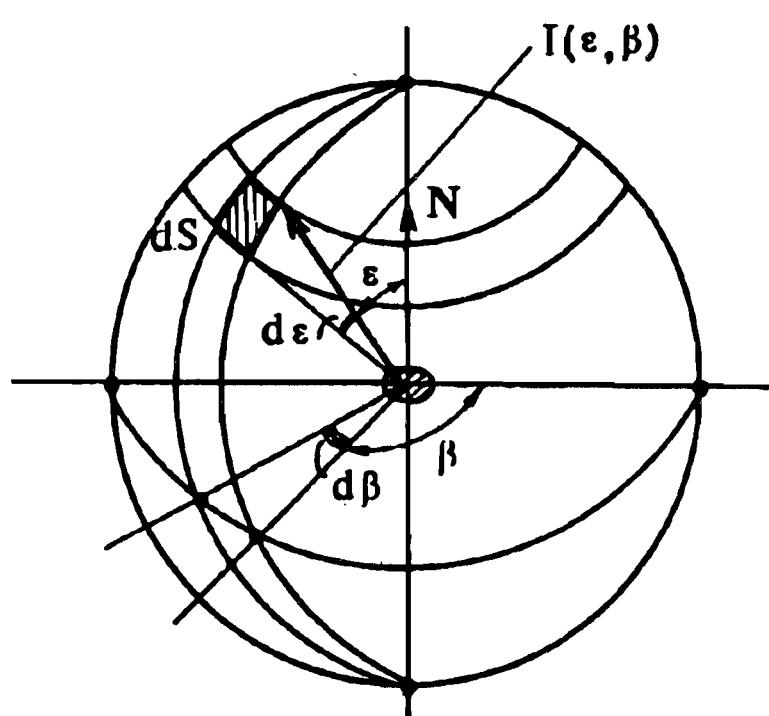


Рис. 1.2.1. Схема для определения связи энергетических характеристик элементарной площадки  $S$

Если источник расположен относительно освещаемой поверхности на расстоянии, которое значительно превышает его геометрические размеры, то его можно считать точечным. В частности, если расстояние от источника в 5 раз превышает его размеры, то погрешность в расчете освещенности, обусловленная заменой его точечным, не будет превышать 1 % [15]. Если источник излучения имеет площадь  $dS$  (рис. 1.2.1), то яркость в заданном направлении равна  $L(\epsilon, \beta)$ , и в соответствии с определением яркости  $L(\epsilon, \beta) = dI(\epsilon, \beta)/(dS \cos\epsilon)$  получаем  $dI(\epsilon, \beta) = L(\epsilon, \beta) \cos\epsilon dS$ , где  $\epsilon$  – угол между нормалью  $N_0$  к излучающей поверхности и направлением, для которого определяется сила света  $I(\epsilon, \beta)$ .

Интегрируя по поверхности излучателя  $S_{изл}$ , определяем силу излучения:

$$I(\epsilon, \beta) = \int L(\epsilon, \beta) \cos \epsilon \, ds. \quad (1.2.5)$$

В моделях источника излучения, как правило, используется предположение, что излучение их отдельных поверхностей носит диффузный (рассеянный) характер, причем яркость каждой отдельной поверхности имеет одинаковую величину по всем направлениям. Для диффузной излучающей поверхности можно записать  $L(\epsilon, \beta) = L_0 = \text{const}$ .

Для плоского диффузного излучателя в соответствии с (1.2.5) получим

$$I(\epsilon) = L_0 S_{\text{изл}} \cos \epsilon. \quad (1.2.6)$$

Излучающую плоскую площадку  $S_{\text{изл}}$  при расчетах можно заменить точечным источником с координатами центра площадки  $x_i, y_i, z_i$  с индикаторисой излучения, определяемой (1.2.6). Угловое распределение излучения площадки в пространстве определяется положением единичного вектора нормали  $N_0$ . Для направления, заданного единичным вектором  $N_q$ , значение  $\cos \epsilon$  равно скалярному произведению  $(N_0, N_q)$ . Так как при синтезе изображений приходится определять силу излучения для большей совокупности направлений с соответствующими значениями  $N_q$  и вычислять для каждого из них  $(N_0, N_q) = \cos \epsilon$ , то целесообразно индикаторису излучения представить не в виде функции угла  $\epsilon$ , а в виде функции от параметра  $q = (N_0, N_q) = \cos \epsilon$ . Это заметно снижает объем вычислений при расчете освещенности от источников излучения, поскольку исключается операция вычисления обратных тригонометрических функций  $\epsilon = \arccos(N_0, N_q)$ . Если использовать переменную  $q$ , то для для плоской излучающей площадки

$$I(q = \cos \epsilon) = I_0 q, \quad (1.2.7)$$

где  $I_0 = L_0 S_{\text{изл}}$ .

Индикаториса излучения плоской площадки является симметричной, ось симметрии определяется вектором  $N_0$ .

Лучистый поток  $\Phi$ , создаваемый источником излучения, определяется суммированием по всем направлениям. Чтобы получить в математической форме взаимосвязь потока и силы излучения, поместим источник излучения в центре воображаемой сферической поверхности (рис.1.2.1). Поток излучения в направлении элемента  $dS$  сферы составляет  $d\Phi = I(\epsilon, \beta) d\Omega$ , где  $d\Omega$  – телесный угол, в котором распространяется поток, падающий на элемент  $ds$ . Если радиус сферы  $R$ , то  $d\Omega = R d\epsilon R \sin \epsilon d\beta / R^2 = \sin \epsilon d\epsilon d\beta$ . Для элементарного потока

$$d\Phi = I(\epsilon, \beta) \sin \epsilon d\epsilon d\beta. \quad (1.2.8)$$

Полный поток, испускаемый источником по всем направлениям, составляет

$$\Phi = \int_{\beta=0}^{2\pi} \int_{\epsilon=-\pi/2}^{\pi/2} I(\epsilon, \beta) \sin \epsilon d\epsilon d\beta.$$

Если распределение излучения источника симметрично относительно вертикальной оси, то

$$\Phi = 2\pi \int_{\epsilon=-\pi/2}^{\pi/2} I(\epsilon) \sin \epsilon d\epsilon. \quad (1.2.9)$$

Отметим, что интегрирование должно проводиться только по той области, где  $I(\epsilon)$  больше или равно нулю. В частности, для плоской площадки область

интегрирования ограничивается диапазоном от  $\epsilon = 0^\circ$  до  $\epsilon = 90^\circ$ . Если использовать переменную  $q = \cos \epsilon$ , то (1.2.9) принимает вид

$$\Phi = 2\pi \left[ \int_{q=1}^0 I(q) dq + \int_{q=-1}^0 I(q) dq \right].$$

Для искусственного освещения довольно часто используются источники с повышенной направленностью излучения. Для моделирования таких источников в машинной графике весьма удобной является аппроксимация в виде степенной функции от  $\cos \epsilon$ :

$$I(\epsilon) = I_0 \cos^m \epsilon; \cos \epsilon \in [0, 1].$$

При использовании переменной индикатриса такого источника представляется в виде

$$I(q) = I_0 q^m; q \in [0, 1], \quad (1.2.10)$$

а его лучистый поток в соответствии с формулой (1.2.10) равен

$$\Phi = 2\pi I_0 / (m+1). \quad (1.2.11)$$

Величину  $m$  для аппроксимирующей функции можно приблизенно получить из индикатрисы излучения  $m = 1,18/e_{0,5}^2$ , где  $e_{0,5}^2$  – половинное значение угла, в пределах которого сила излучения превышает 50 % максимального ее значения. Значение  $m$  для снижения объема вычислений целесообразно выбирать целым.

Достоинством рассмотренной аппроксимации является не только удобство вычислений, но и возможность получения при целых  $m$  аппроксимирующей функции, описывающей с небольшой погрешностью реальные индикатрисы.

### 1.2.3. ОСВЕЩЕННОСТЬ, СОЗДАВАЕМАЯ ТОЧЕЧНЫМИ И ПРОТЯЖЕННЫМИ ИСТОЧНИКАМИ

Пусть точечный источник с индикатрисой  $I(q)$  размещен в точке  $x_i, y_i, z_i$  и освещает элементарную площадку  $dS$ , координаты которой  $x_0, y_0, z_0$  (рис. 1.2.2). Угловую ориентацию индикатрисы излучения и площадки определяют единичные векторы  $n_0$  и  $n_{dS}$ . Освещенность элементарной площадки  $dS$  по определению равна  $E = d\Phi_q / dS$ , где  $d\Phi_q$  – элементарный лучистый поток в направлении  $q$ . Величина  $d\Phi_q$  определяется через силу света в данном направлении  $d\Phi_q = I(q) d\Omega$ .

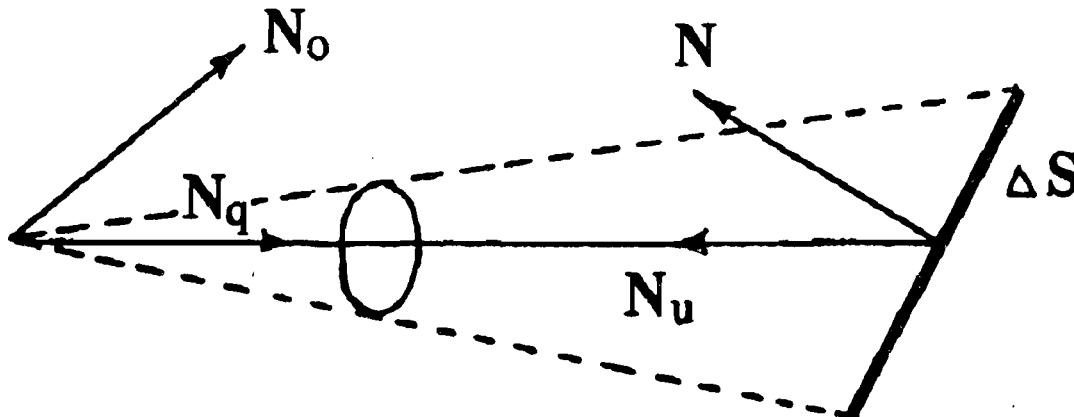


Рис. 1.2.2. Схема определения освещенности от точечного источника

Телесный угол  $d\Omega$ , в пределах которого распространяется поток излучения  $d\Phi_q$ , равен  $d\Omega = dS \cos \alpha / r^2$ , где  $\alpha$  – угол между нормалью к площадке  $dS$  и направлением на источник излучения;  $r$  – расстояние от источника излучения до облучаемой площадки.

После подстановок и преобразований для освещенности  $E$  получим формулу

$$E = I(q) \cos \alpha / r^2, \quad (1.2.12)$$

которая носит название *закона обратных квадратов расстояний*. Полученная формула (1.2.12) по форме совпадает с известной формулой для освещенности от точечных источников. Отличие ее заключается в учете индикатрисы излучения.

Для практических расчетов (1.2.12) требует конкретизации. Вектор  $r$ , определяющий направление на источник излучения

$$r = i(x_0 - x_i) + j(y_0 - y_i) + k(z_0 - z_i), \quad r^2 = (x_0 - x_i)^2 + (y_0 - y_i)^2 + (z_0 - z_i)^2,$$

где  $i, j, k$  – орты соответствующих координатных осей  $x, y, z$ .

Единичный вектор  $N_i$  и  $\cos \alpha$  вычисляются по формулам  $N_i = r/r$ ;  $\cos \alpha = (N_i \cdot N_q)$ , а вычисление параметра  $q$  производится следующим образом:  $N_q = -N_i$ ;  $q = (N_q \cdot N_0)$ , где  $\cdot$  – знак скалярного произведения векторов. В программе расчета должна предусматриваться проверка неотрицательности  $\cos \alpha = (N_i \cdot N)$ .

Рассмотрим освещенность, созданную протяженным источником, площадь которого равна  $S_i$ , а поверхностная яркость –  $L$ . Для анализа разместим освещаемый элемент в центре координатной системы так, чтобы нормаль к элементу совпадала с вертикальной осью координат, и сформируем условную полусферу единичного радиуса с центром в начале координат (рис.1.2.3).

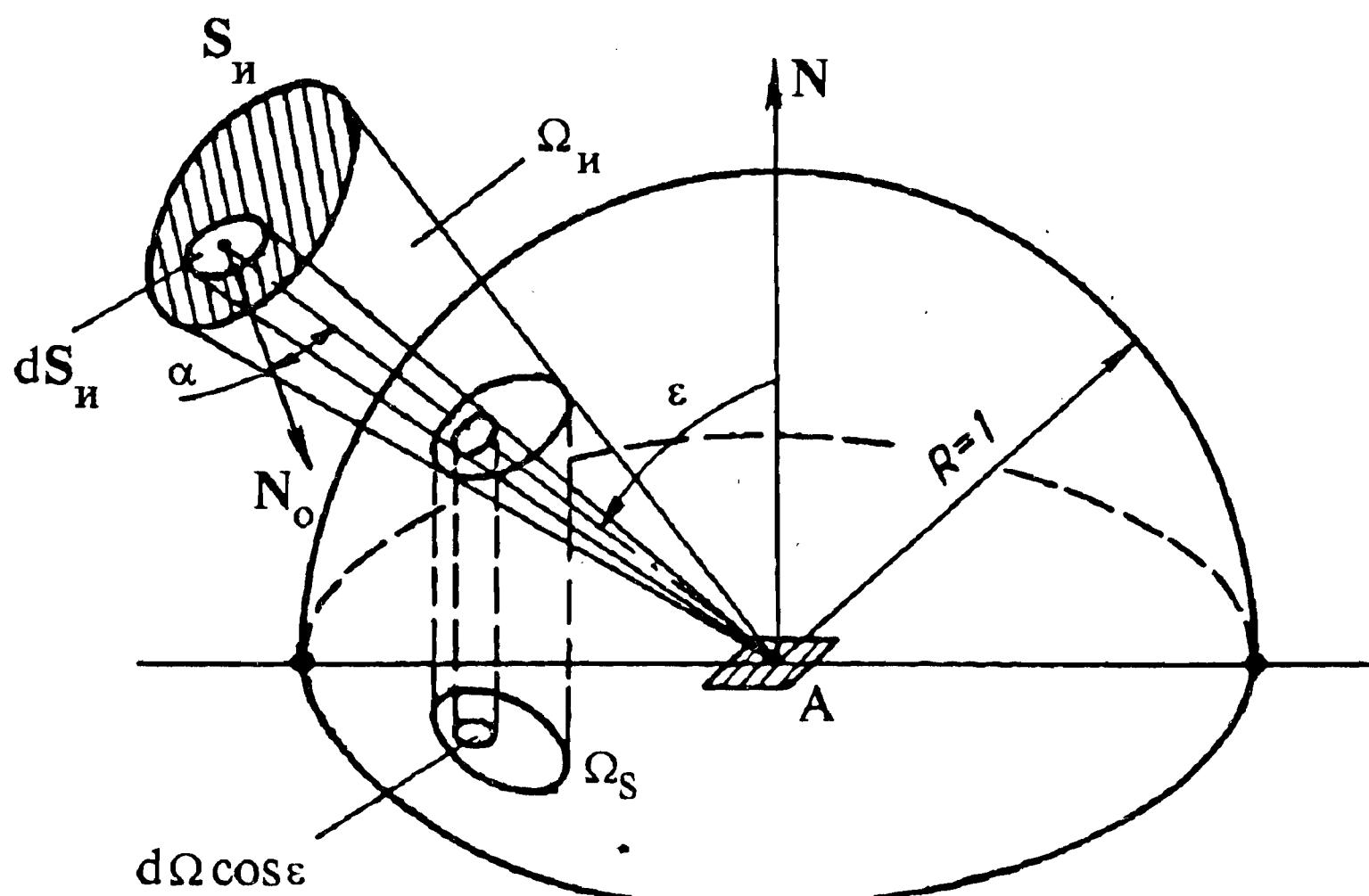


Рис. 1.2.3. Схема определения освещенности от протяженного источника

Освещенность в точке А, созданная элементом  $dS_i$ , будет определяться по формуле

$$dE_A = L dS_i \cos \alpha \cos \epsilon / r^2 = L d\Omega \cos \alpha, \quad (1.2.13)$$

где  $\cos \alpha = (N_i \cdot N)$  и  $\cos \epsilon = (N_q \cdot N_0)$  – косинусы углов для элемента  $dS_i$ ;  $r$  – расстояние между элементом  $dS_i$  и точкой А.

Полная освещенность в точке А от всей поверхности  $S_i$  в соответствии с (1.2.13) может быть вычислена двумя путями:

$$E_A = \int_{S_i} dE_A = L \cos \alpha \cos \epsilon / r^2 dS_i; \quad (1.2.14)$$

$$E_A = \int_{\Omega_i} dE_A = \int_{\Omega_i} L \cos \alpha d\Omega. \quad (1.2.15)$$

В большинстве случаев получить аналитические формулы для  $E_A$  не удается [15], и вычисление должно производиться численными методами даже при постоянной яркости  $L$  по всей светящейся поверхности.

В частности, можно разделить всю светящуюся поверхность на отдельные участки, которые можно заменить точечными источниками, затем найти освещенность суммированием:

$$E_A = \sum_{K=1}^{K_0} L_K \cos \alpha_K \cos \epsilon_K S_K / r_K^2,$$

где  $K_0$  – число участков протяженного источника излучения.

В ряде случаев весьма удобным является использование формулы (1.2.15). Если яркость  $L$  всей поверхности источника одинакова, то

$$E_A = \int_{\Omega_i} \cos \alpha d\Omega = L \Omega_S, \quad (1.2.16)$$

где  $\Omega_S$  – проекция на освещаемую плоскость участка сферической поверхности единичного радиуса, вырезаемого телесным углом  $\Omega_i$ , который характеризует размеры источника.

Формула (1.2.1.6) позволяет получать в ряде ситуаций простые аналитические формулы. Например, требуется рассчитать освещенность горизонтальной площадки, созданную свечением купола с постоянной яркостью  $L_0$ . В этом случае  $\Omega_S = \pi$ , а освещенность  $E_A = \pi L_0$ .

### 1.3. ЕСТЕСТВЕННОЕ ОСВЕЩЕНИЕ

Основным источником, определяющим естественную освещенность, является Солнце. Спектральный состав солнечного излучения на границе атмосферы принято аппроксимировать излучением черного тела с температурой  $T = (5770+15)$  К. Истинное распределение энергии в спектре солнечного излучения несколько отличается от распределения для черного тела с  $T = 5770$  К: в области 0,4...0,75 мкм Солнце излучает больше энергии, чем черный излучатель при  $T = 5770$  К, в ультрафиолетовой области – меньше, а в инфракрасной области отличия несущественны. Солнце как излучатель представляет собой шар и теоретически излучает расходящийся поток лучей, однако из-за большого удаления Солнца его излучение на земной поверхности практически представляет поток параллельных лучей. Энергетическая освещенность, которую создают солнечные лучи на перпендикулярной к ним плоскости вне земной атмосферы на среднем расстоянии от Земли до Солнца, характеризуется солнечной постоянной  $E_{sun} = (1365 \pm 14)$  Вт/м<sup>2</sup>.

Освещенность естественных ландшафтов определяется высотой Солнца над горизонтом  $h_0$  и влиянием атмосферы. Высота Солнца  $h_0$  для района с геодезической широтой  $\phi$  и долготой  $\lambda$  определяется по следующей расчетной формуле:

$$\sin h_0 = \sin \delta_{sun} \sin \phi + \cos \delta_{sun} \cos \phi \cos \Delta\lambda_0, \quad (1.3.1)$$

где  $\delta_{sun}$  – склонение Солнца на дату наблюдения;  $\Delta\lambda_0 = \lambda_0 - \lambda$  – разность долгот Солнца  $\lambda_0$  и наблюдателя (часовой угол).

Разность долгот  $\Delta\lambda_0$  (градус) связана с местным временем  $\tau$  соотношением  $\Delta\lambda_0 = 180 - 15\tau$ , где  $\tau$  – время в часах и его долях.

На заданный момент московского времени  $t_{\text{моск}}$  величина  $\Delta\lambda_0$  определяется следующими равенствами для зимнего и летнего времени соответственно:

$$\Delta\lambda_0 = 225 - 15(t_{\text{моск}} - \eta_0) - \lambda; \quad \Delta\lambda_0 = 240 - 15(t_{\text{моск}} - \eta_0) - \lambda, \quad (1.3.2)$$

где  $\eta_0$  – уравнение времени (поправка по времени) в долях часа.

Склонение Солнца  $\delta_0$  задается таблично, но с достаточной для моделирования точностью может быть определено аналитически:  $\delta_0 = -23,5 \sin(2\pi d_0/365)$ , где  $d_0$  – время в сутках от дневного равноденствия (22 марта) до даты съемки. Значения определяются  $\eta_0$  по номограмме или по таблицам.

Для моделирования реалистичных изображений при естественном освещении необходимо определить также азимут Солнца  $A_0$ , для вычисления которого используются  $\Delta\lambda_0$ ,  $\delta_0$  и  $h_0$ :

$$\sin A_0 = (\sin \Delta\lambda_0 \cos \delta_0) / \cos h_0; \quad \cos A_0 = (\sin \delta_0 - \sin h_0 \sin \phi) / (\cos \phi \cosh_0). \quad (1.3.3)$$

В процедурах синтеза изображений целесообразно использовать единичный вектор  $n_0$ , указывающий направление на Солнце. Если использовать правую топоцентрическую систему координат, в которой ось  $x$  направлена на север, а ось  $z$  перпендикулярна поверхности Земли и направлена в зенит, то составляющие вектора  $n_0$  по осям  $x, y, z$  будут определяться следующими соотношениями:

$$\begin{aligned} n_{0x} &= \cosh_0 \cos A_0 = (\sin \delta_0 - \sin h_0 \sin \phi) / \cos \phi; \\ n_{0y} &= \cosh_0 \sin A_0 = \sin \Delta\lambda_0 \cos \delta_0; \\ n_{0z} &= \cosh_0. \end{aligned} \quad (1.3.4)$$

Отметим, что для характеристик положения Солнца наряду с высотой  $h_0$  используется зенитное расстояние  $z_0 = 90^\circ - h_0$ .

Воздействие атмосферы проявляется в ослаблении прямого солнечного излучения и его рассеивании. В соответствии с этим освещенность земной поверхности определяется двумя световыми потоками: ослабленной прямой радиацией  $e_{\lambda_{\text{пр}}}$  и рассеянной радиацией солнечного излучения  $e_{\lambda_{\text{р}}}$ , идущей к Земле.

Существенная нестабильность свойств атмосферы, значительное число факторов, обуславливающих ее изменчивость, не позволяют давать точный прогноз освещенности. Обычно используются приближенные модели с ограниченным числом параметров, характеризующих оптические свойства атмосферы. Для расчетов широко используется модель средней стандартной атмосферы. Спектральная освещенность, создаваемая Солнцем у поверхности Земли на площадке, перпендикулярной солнечным лучам, при безоблачном небе и стандартной атмосфере определяется формулой

$$e_{\lambda_{\text{пр}}} = e_{0\lambda} e^{-\tau_{0\lambda}} \operatorname{cosech} h_0, \quad (1.3.5)$$

где  $e_{0\lambda}$  – спектральная освещенность, создаваемая солнечным излучением на границе атмосферы;  $\tau_{0\lambda}$  – оптическая толщина атмосферы.

Обобщенным параметром  $\tau_{0\lambda}$  практически можно пользоваться в диапазоне  $\lambda = 0,4 \dots 0,8 \text{ мкм}$ , в пределах которого ослабление прямой солнечной радиации обусловлено в основном молекулярным и аэрозольным рассеиванием (рис. 1.3.1).

Для этого диапазона зависимость  $\tau_{0\lambda}$  от длины волны для стандартной атмосферы описывается эмпирической формулой

$$\tau_{0\lambda} = 0,1(550/\lambda)^4 + (\tau_0 - 0,1)(550/\lambda), \quad (1.3.6)$$

где  $\tau_0$  – оптическая толщина атмосферы при  $\lambda=550$  нм. При вычислениях по (1.3.6) значения  $\lambda$  подставляются в нанометрах.

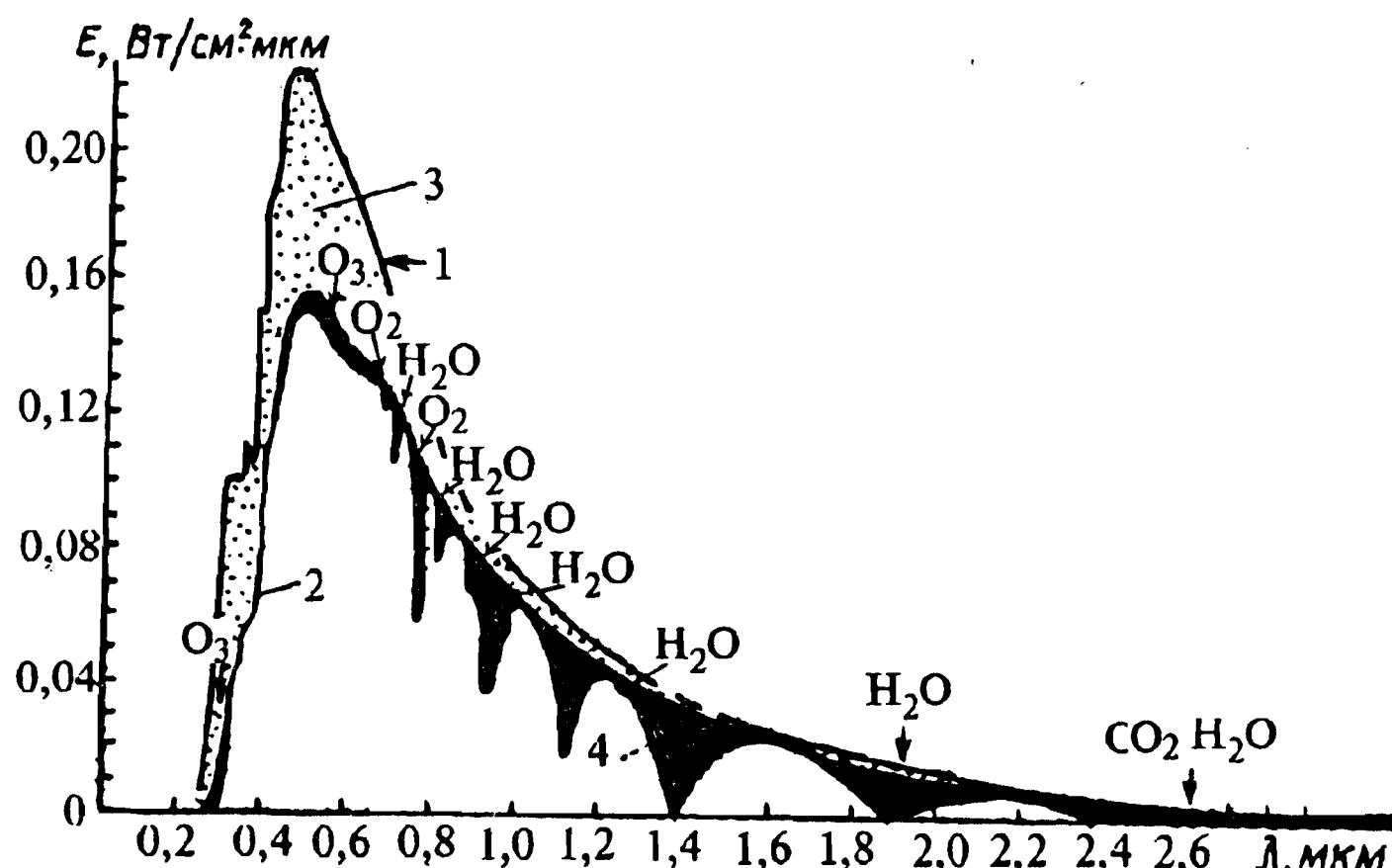


Рис. 1.3.1. Ослабление прямой солнечной радиации в атмосфере:

1 – солнечное излучение на границе атмосферы; 2 – солнечное излучение у земной поверхности; 3 – аэрозольное рассеивание; 4 – поглощение в атмосфере

При расчетах обычно используется несколько типовых значений  $t_0$ . Для среднезамутненной атмосферы  $t_0$  составляет 0,3. Слабой замутненности атмосферы соответствует  $t_0=0,2$ , повышенной замутненности  $t_0=0,4$ , высокой  $t_0=0,5$ .

Освещенность, создаваемая прямым излучением Солнца, на произвольно ориентированной площадке  $\Delta S$  определяется углом между единичным вектором направления на солнце  $N_s$  и единичным вектором нормали  $N$  к площадке  $\Delta S$ :

$$e_{\lambda_{\text{пр}} \Delta S} = e_{\lambda_{\text{пр}}} (N_s \cdot N), \quad (1.3.7)$$

где  $\cdot$  – скалярное произведение векторов  $N_s$  и  $N$ .

В программе синтеза изображений обязательно должно учитываться условие неотрицательности освещенности

$$(N_s \cdot N) \geq 0. \quad (1.3.8)$$

При невыполнении условий (1.3.8) данная сторона площадки  $\Delta S$  не освещена:

$e_{\lambda_{\text{пр}} \Delta S} = 0$ . Единичный вектор нормали к площадке  $\Delta S$  должен быть направлен от поверхности, освещенность которой вычисляется. Это означает, что принципиально площадка  $\Delta S$  характеризуется двумя единичными векторами нормали  $N_1$  и  $N_2$ , определяющими две ее стороны. Очевидно, что  $N_1 = -N_2$ .

Отметим, что из общей формулы для определения освещенности (1.2.23) непосредственно следует приводимая в литературе формула для освещенности земной поверхности. Для горизонтальной земной поверхности  $N_g = [0 \ 0 \ 1]^T$  и, следовательно,  $(N \cdot N_g) = \cos(-z_s) = \cos(90^\circ - h_s) = \sin h_s$ .

Освещенность, создаваемая рассеянной радиацией, определяется яркостью неба. Важность учета рассеянной радиации обусловлена тем, что она определяет освещенность участков сцены, находящихся в тени.

Яркость произвольной точки небосвода  $L_H$  представляет собой функцию четырех основных параметров [13]: высоты Солнца  $h_0$ , пропускания атмосферы  $T$ , зенитного расстояния  $\xi$  точки небосвода и угла  $\psi$  между направлением на Солнце и в заданную точку небосвода.

Расчет освещенности произвольно ориентированной площадки с учетом истинного распределения яркости небосвода требует выполнения численного интегрирования с использованием таблично заданных функций. Это весьма серьезно усложняет процедуру вычисления освещенности точек картинной плоскости. Процедуру вычислений можно существенно упростить, если яркость всех точек небосвода принять одинаковой и равной некоторой усредненной величине. Среднюю яркость небосвода можно аппроксимировать зависимостью вида [6]

$$L_{H\lambda} = (1/\pi)C e_{0\lambda} \sin h_0 (1 - e^{-\tau_{0\lambda}} \operatorname{cosech} \psi). \quad (1.3.9)$$

Величина  $C$  сравнительно слабо зависит от  $h_0$  и  $\tau_{0\lambda}$ . В ряде случаев ее полагают постоянной. Более точное приближение можно получить, если полагать  $C = 0,5(1 + \tau_{0\lambda})$ . При этом различия в результатах, полученных на основе более точных моделей и изложенной выше, невелики. Максимальные различия достигают 20 % лишь при значительной высоте Солнца ( $h_0 > 60^\circ$ ).

Для определения освещенности от небосвода произвольно ориентированной площадки  $\Delta S$  рассмотрим общую схему определения освещенности, создаваемой протяженным источником (рис. 1.3.2).

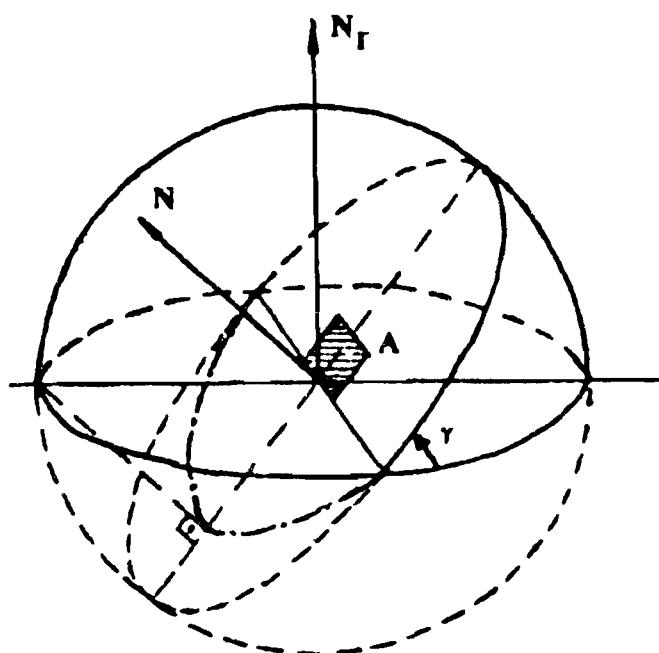


Рис. 1.3.2. Определение освещенности произвольно ориентированной площадки небосводом

В соответствии с (1.2.16) освещенность от небосвода площадки  $\Delta S$  определяется следующим образом:  $E_{\Delta S} = L_H \Omega_S$ , где  $\Omega_S$  – проекция на освещаемую плоскость  $A$ , в которой лежит площадка  $\Delta S$ , видимой части небесной сферы.

Если поверхность  $A$  наклонена на угол  $\gamma$  относительно горизонта, то проекция  $\Omega_S$  телесного угла видимой части небосвода  $\Omega_H$  на плоскость  $A$  определяется простым соотношением  $\Omega_S = \pi/2 + (\pi/2)\cos\gamma = (\pi/2)[1 + (N \cdot N_r)]$ .

В результате освещенность  $e_{\lambda_{\text{PDS}}}$  произвольно ориентированной площадки, обусловленная рассеянной радиацией, составляет

$$e_{\lambda_{\text{PDS}}} = 0,25[1+(N \cdot N_r)] \sinh \left( 1 + \tau_{0\lambda} \right) e_{0\lambda} (1 - e^{-\tau_{0\lambda}} \operatorname{cosech}^2 \theta). \quad (1.3.10)$$

Полная освещенность в спектральном диапазоне от  $\lambda_1$  до  $\lambda_2$  для произвольных поверхностей, не закрытых тенью, в энергетической системе единиц может быть определена интегрированием:

$$E_{\text{c}_{\Delta\lambda}} \int_{\lambda_1}^{\lambda_2} (e_{\lambda_{\text{PDS}}} + e_{\lambda_{\text{PDS}}}) d\lambda. \quad (1.3.11)$$

Для систем, работающих в видимом диапазоне, освещенность наблюдаемого пространства принято указывать в светотехнической системе единиц. Переход к светотехнической системе может быть осуществлен путем учета относительной видности монохроматических излучений  $v_{\lambda}(\lambda)$ :

$$E_C = 683 \int_{\lambda_1}^{\lambda_2} (e_{\lambda_{\text{PDS}}} + e_{\lambda_{\text{PDS}}}) v_{\lambda}(\lambda) d\lambda. \quad (1.3.12)$$

Интегрирование в соотношении (1.3.12) выполняется в пределах от  $\lambda_1 = 0,38$  мкм до  $\lambda_2 = 0,76$  мкм. За пределами этого диапазона значения  $v_{\lambda}(\lambda)$  практически являются нулевыми.

Хотя переход от энергетической системы к светотехнической не вызывает принципиальных затруднений, однако для систем видимого диапазона удобнее пользоваться расчетными формулами, выражающими освещенность непосредственно в светотехнической системе. Для таких расчетов может быть использовано соотношение, базирующееся на известном в [6], но дополненное учетом наклона освещаемой площадки:

$$E_{C_{\Delta S}} = E_{C} [(N \cdot N_r) P \operatorname{cosech}^2 \theta + 0,25 C (1 + \cos \gamma) (1 - P \operatorname{cosech}^2 \theta)], \quad (1.3.13)$$

где  $E_C = 13300$  лк – освещенность плоскости, перпендикулярной к лучам Солнца на границе атмосферы в светотехнической системе единиц;  $P, C$  – коэффициенты, характеризующие прозрачность и рассеивание в атмосфере.

Для средних параметров стандартной атмосферы  $P = 0,734$ ;  $C = 0,48$ . В соответствии с (1.2.29) максимальная освещенность горизонтальной площадки на земной поверхности для стандартных условий составляет 106000 лк (при  $\theta = 90^\circ$ ).

На величину естественной освещенности большое влияние оказывает характер облачности. Наличие облачности вызывает значительное увеличение рассеянного излучения. При разорванной облачности освещенность "на Солнце" оказывается на 10...30 % выше, чем при безоблачной погоде, а освещенность в тени может возрастать до двукратной величины. Это обстоятельство является причиной значительного разброса в экспериментальных данных по освещенности в тени и оправдывает применение в машинной графике относительно простых моделей расчета освещенности, использование поправочных коэффициентов, увеличивающих значение освещенности в тени по сравнению с расчетными при углах Солнца  $\theta > 10...15^\circ$ .

## 1.4. МОДЕЛИРОВАНИЕ ОТРАЖЕННОГО ИЗЛУЧЕНИЯ ОБЪЕКТОВ

Отражающие свойства объектов описываются коэффициентами отражения, коэффициентами яркости и индикатрисой отражения.

В основу классификации объектов по характеру отражения падающего на них лучистого потока положено пространственное распределение отраженного ими излучения, т.е. индикатриса отражения. Определяющее влияние на индикатрису излучения оказывает структура поверхности объекта. По виду индикатрисы отражения принято выделять четыре типа поверхностей (рис. 1.4.1).

*Ортотропные поверхности* отражают падающий на них лучистый поток диффузно или равномерно по всем направлениям. Ортотропные поверхности в литературе часто называют *диффузными* или *ламбертовскими*. Ламбертовские отражатели являются доминирующими среди естественных и искусственных объектов. В видимой области ортотропными поверхностями являются пески, рыхлый снег, неоформленный бетон, сухой асфальт, большинство грунтов. Отличительная особенность диффузных поверхностей – независимость их яркости от положения наблюдателя (рис. 1.4.1, а).

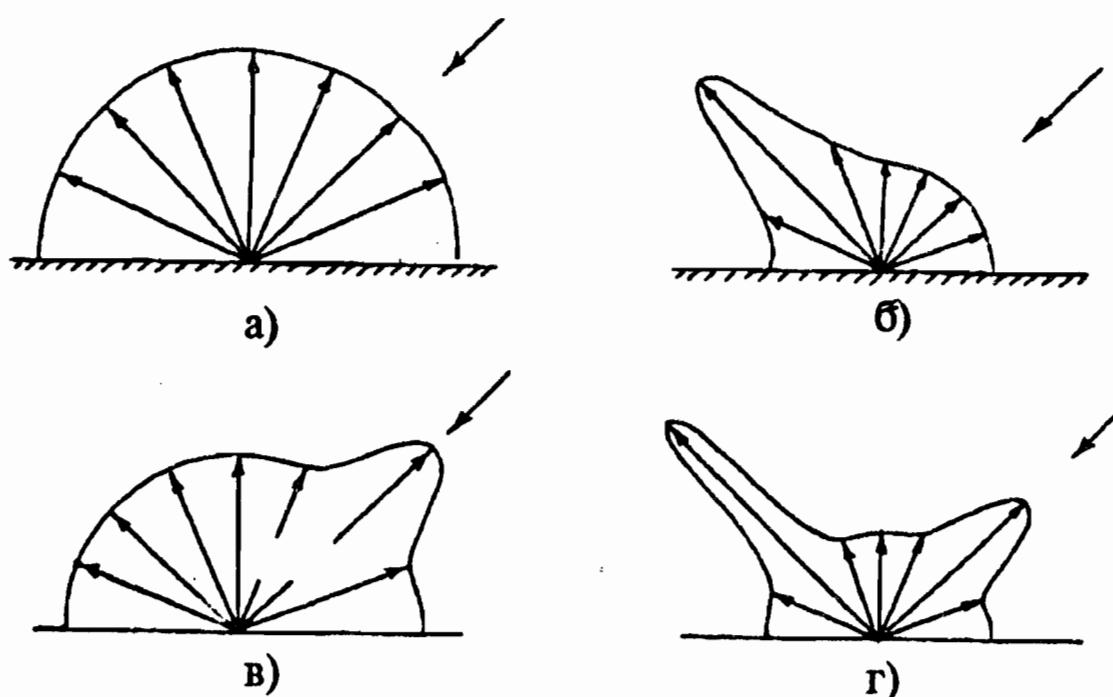


Рис. 1.4.1. Вид индикатрис различных типов поверхностей:

а – ортотропной; б – с зеркальным отражением; в – с обратным отражением; г – смешанной

*Зеркальные поверхности* отражают падающий на них поток пре-имущественно в плоскостях падающих лучей и под углом, равным углу падения. В видимой области к зеркально отражающим относятся чистые стеклянные поверхности, пластики, обработанные металлические поверхности, чистый лед, обнаженный скальный грунт, сухие каменные поверхности, поверхности водных бассейнов в спокойном состоянии. Следует отметить, что применительно к реальным объектам термин "зеркальная поверхность" указывает на направленный характер отражения падающего лучистого потока, но не означает, что отражение происходит в полном соответствии с законами геометрической оптики. Для реальных зеркальных отражений положение максимума отраженного излучения определяется в соответствии с законами геометрической оптики: угол отражения  $\epsilon_1$  равен углу падения  $\epsilon_2$ . При этом падающее излучение рассеивается в некотором телесном угле относительно направления максимума (рис. 1.4.1, б). При моделировании для

упрощения принято полагать, что индикатриса зеркального отражения симметрична относительно максимума.

Поверхности, для которых закон отражения выполняется достаточно строго, будем называть *идеально отражающими* в отличие от реальных зеркальных поверхностей.

К третьему типу относят поверхности, отражающие излучение преимущественно в направлении к источнику (рис.1.4.1,в). Для обозначения этого типа поверхностей используются различные термины, единого установившегося термина пока нет. Их называют *изрытыми, антизеркальными, световозвращающими*. Отражение от объекта в направлении источника называют *обратным блеском* или просто *обратным отражением*. Такое отражение типично для сельскохозяйственных культур, лугов и другой растительности.

Четвертый тип составляют поверхности со смешанным отражением. Для таких поверхностей характерно наличие двух или трех типов отражений. В общем случае для поверхностей со смешанным отражением можно выделить диффузную, зеркальную и обратную составляющие, а индикатриса имеет два максимума (рис.1.4.1,г). Смешанное отражение общего вида наблюдается у рисовых полей, покрытых росой лугов и других аналогичных объектов.

Различие в типах индикатрис обусловлено структурой отражающей поверхности и степенью их шероховатости. Для гладких поверхностей характерно наличие зеркального отражения. Идеально отражающими являются поверхности, удовлетворяющие критерию Рэлея:

$$h < \lambda / (16 \cos\theta), \quad (1.4.1)$$

где  $h$  – высота шероховатости;  $\theta$  – угол падения излучения.

С увеличением высоты шероховатостей зеркальная компонента снижается, и отражение приобретает диффузный характер. В ряде случаев диффузное отражение является преобладающим и для объектов с гладкими поверхностями. Примерами таких объектов являются молочное стекло и вода со значительной концентрацией взвешенных частиц. Диффузное излучение таких объектов обусловлено тем, что значительная часть падающего лучистого потока проникает в приповерхностный слой и рассеивается массой мелких неоднородностей. В результате наряду с зеркальным отражением от поверхности раздела объекта и внешней среды наблюдается диффузное излучение из внутренних областей объекта.

При значительной высоте шероховатостей существенную роль играет микроструктура отражающей поверхности. Для исследования общих закономерностей отражения поверхности сложной структуры используют специальные модели.

Одной из наиболее разработанных аналитических моделей является фасцентная модель отражения Торрэнса–Спэрроу. В этой модели поверхность представляется в виде совокупности случайно ориентированных микроскопических зеркальных граней. Отражение от каждой микрограммы определяется по формуле Френеля, затем методами геометрической оптики учитывается затенение микрограмм соседними и маскирование (экранирование) части зеркально отраженного излучения соседними микрограммами. Модель Торрэнса–Спэрроу позволяет в аналитической форме учесть длину волны и угол падения освещивающих лучей. Для многих природных образований поверхность представляют аналогичным образом, но отражение каждой из микроплощадок принимается ортотропным. Такая модель качественно подтверждает возможность появления зеркальной и обратной составляющих отраженного излучения у объектов, все элементы которых отражают диффузно.

Для анализа особенностей отражения от растительности, покрытой

листвой, используют слоистую модель, каждый отражающий слой которой образован отдельными, в общем случае не перекрывающимися площадками определенных форм и размеров и обладающими ортотропным отражением. Специфика отражения в таких слоистых структурах обусловливается затенением отражающих площадок нижних слоев вышележащими. Получить аналитическое решение для такой модели чрезвычайно сложно, и обычно моделирование выполняется методом Монте-Карло. Результаты моделирования показывают, что поверхности такой структуры обладают обратным отражением. Количественные результаты близки к экспериментальным.

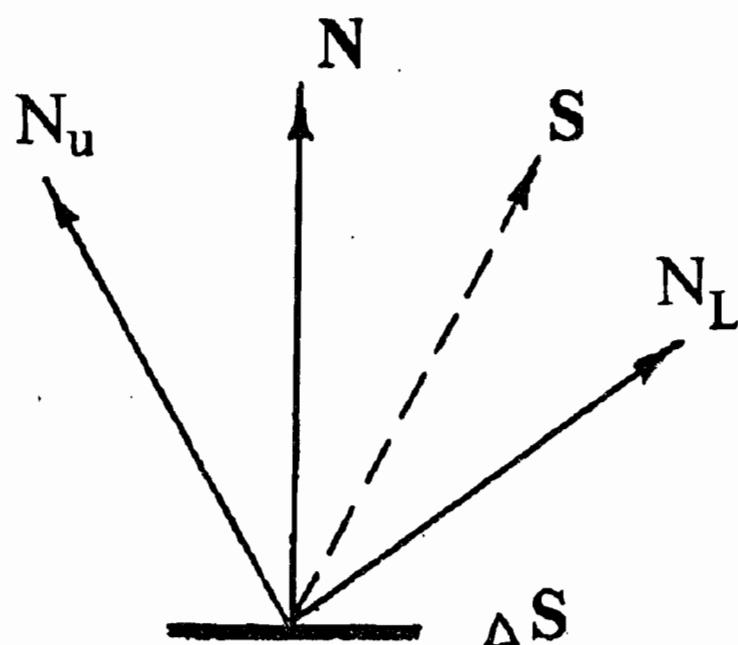


Рис. 1.4.2. Диаграмма направлений

Модели, основанные на статистическом описании структуры отражающих поверхностей, сложны, расчеты на их основе связаны со значительным объемом вычислений. Это существенно ограничивает их непосредственное использование в машинной графике. При моделировании отраженного излучения обычно ограничиваются приближенными моделями, в которых учитываются лишь главные особенности рассеянного светового потока поверхностью объекта. Как правило, полагают, что форма индикаторы отражения не зависит от длины волны и для каждой составляющей отраженного излучения ее можно описать осесимметричной функцией. Для составляющих зеркального и обратного отражений наиболее удобной является степенная функция от косинуса угла между направлениями наблюдения и максимума отражения.

Рассмотрим модель определения яркости площадки  $\Delta S$ , на которую под углом  $\theta$  падает лучистый поток от источника освещения (рис. 1.4.2). Пусть наблюдение данной площадки осуществляется в направлении, определяемом единичным вектором  $N_L$ . Яркость наблюдаемой площадки в общем случае определяется прямым излучением источника и рассеянным излучением окружающей предмет полусферы:

$$L = \gamma_a E_p / \pi + \gamma_{\Sigma} E_{\text{пр}} / \pi, \quad (1.4.2)$$

где  $E_p$  и  $E_{\text{пр}}$  – освещенности, обусловленные рассеянным и прямым излучениями;  $\gamma_a$  – коэффициент яркости площадки для рассеянного излучения;  $\gamma_{\Sigma}$  – суммарный коэффициент яркости площадки в заданном направлении.

Каждая из компонент яркости в (1.4.2) в свою очередь включает диффузную, зеркальную и обратную составляющие. Лучистый поток, обусловленный отражением рассеянного излучения, составляет

$$\Phi_{\text{отр}} = \gamma_a E_p \Delta S. \quad (1.4.3)$$

С другой стороны, используя коэффициенты отражения, получаем

$$\Phi_{\text{отр}} = (\rho_d + \rho_{3\cap} + \rho_{\text{обр}\cap}) E_p \Delta S, \quad (1.4.4)$$

где  $\rho_d$  – коэффициент диффузного отражения;  $\rho_{3\cap}$ ,  $\rho_{\text{обр}\cap}$  – коэффициенты зеркального и обратного отражений при рассеянном освещении.

Из (1.4.3) и (1.4.4) получаем

$$r_a = \rho_d + \rho_{3\cap} + \rho_{\text{обр}\cap}. \quad (1.4.5)$$

Коэффициенты отражения  $\rho_{3\cap}$  и  $\rho_{\text{обр}\cap}$  принципиально могут быть определены путем интегрирования по всем углам падения лучей в пределах полусферы [15], если известны зависимости коэффициентов отражения от угла падения лучей. Аналитическое определение  $\rho_{3\cap}$  возможно только для оптически гладкой поверхности диэлектриков. В частности, для спокойной поверхности воды  $\rho_{3\cap} = 0,067$ , для поверхности стекла  $\rho_{3\cap} = 0,095...0,13$ . Величины  $\rho_{3\cap}$  и  $\rho_{\text{обр}\cap}$  для растительности обычно невелики и составляют 0,001...0,01, для таких поверхностей основной вклад дает диффузное рассеивание.

С учетом ориентации площадки  $\Delta S$  для яркости, обусловленной отражением рассеянного излучения, получим

$$r_a E_p / \pi = 0,5 r_a [1 + (N \cdot N_r)] L_{\text{сф}}, \quad (1.4.6)$$

где  $N_r$  – нормаль к горизонтальной поверхности сцены;  $L_{\text{сф}}$  – средняя яркость полусферы, отражающей моделируемую сцену.

Яркость поверхности, обусловленную отражением прямого излучения освещаемого источника, можно определить следующим образом:

$$r_{\Sigma} E_{\text{пр}} / \pi = [r_d + r_{3m} \cos^n \beta_3 + r_{\text{обр}m} \cos^p \beta_{\text{обр}}] E_0 \cos \theta / \pi, \quad (1.4.7)$$

где  $r_{3m}$ ,  $n$ ,  $\beta_3$ ,  $r_{\text{обр}m}$ ,  $p$ ,  $\beta_{\text{обр}}$  – соответственно максимальные значения коэффициента яркости, показатели сжатия индикатора отражения, углы между направлением наблюдения и максимумами для зеркальной и обратной составляющих отраженного излучения;  $E_0$  – освещенность, создаваемая источником в точке на площадке, перпендикулярной падающим лучом.

Поскольку при синтезе изображений направления задаются векторами, то все косинусы углов в (1.4.7) целесообразно выразить в виде скалярных произведений соответствующих векторов. В результате (1.4.7) приводится к виду

$$r_{\Sigma} E_{\text{пр}} / \pi = (E_0 / \pi) [r_d + r_{3m} (N_L S)^n + r_{\text{обр}m} (N_u \cdot N_L)^p] (N_u \cdot N). \quad (1.4.8)$$

Вектор  $S$  определяется через векторы  $N_u$  и  $N$ :  $S = M_S N_u$ , где  $M_S$  – матрица, определяющая направление зеркального отражения.

Матрица  $M_S$  представляет собой матрицу поворота вокруг  $N$  на угол  $\delta = -180^\circ$ . Если составляющие вектора нормали  $N$  к выбранной системе координат равны  $N_x$ ,  $N_y$  и  $N_z$ , то

$$M_S = \begin{bmatrix} 2N_x^2 - 1 & 2N_x N_y & 2N_x N_z \\ 2N_y N_x & 2N_y^2 - 1 & 2N_y N_z \\ 2N_z N_x & 2N_z N_y & 2N_z^2 - 1 \end{bmatrix}.$$

Таким образом, яркость наблюдаемой площадки определяется формулой

$$L = 0,5r_a [1 + (N \cdot N_r)] L_{\text{сф}} + (E_0/\pi) [r_d + r_{3m} (N_L S)^n + r_{\text{обр}_m} (N_u \cdot N_L)^p] (N_u \cdot N). \quad (1.4.9)$$

Для расчетов по (1.4.9) должны быть заданы четыре коэффициента яркости  $r_a$ ,

$r_d$ ,  $r_{3m}$ ,  $r_{\text{обр}_m}$ . Для большинства объектов они могут быть получены лишь экспериментально. Укажем количественные данные  $r_d$  для нескольких поверхностей: лес хвойный – 0,04...0,08; кирпич красный – 0,20; стена беленая – 0,70; мел очищенный – 0,90.

Данных по коэффициентам яркости для зеркальной и обратной составляющих в литературе приводится очень мало. Коэффициент яркости обратного отражения для растительности близок по величине к  $r_d$ , в отдельных случаях может превышать  $r_d$  в 2...3 раза. При углах падения лучей  $\theta > 60...70^\circ$  величина  $r_{\text{обр}_m}$  уменьшается. Значения  $p$  для растительности лежат в пределах 10...100. Повышение густоты растительности сопровождается увеличением  $p$ .

Зеркальное отражение от растительности при углах падения лучей  $\theta \leq 45^\circ$  выражено очень слабо, а при  $\theta > 45^\circ$  быстро возрастает с увеличением  $\theta$ . Рост  $r_{3m}$  приблизительно происходит пропорционально третьей–пятой степеням  $\theta$ . При углах падения лучей  $\theta = 70...80^\circ$  величина  $r_{3m}$  для посевов зерновых культур в 4...5 раз превышает  $r_d$ . Весьма значительный рост  $r_{3m}$  с увеличением  $\theta$  характерен для спокойных водных поверхностей. При углах падения лучей  $75...80^\circ$   $r_{3m}$  может в 20...30 раз превышать  $r_d$  и быть в 2...3 раза ярче, чем идеальный диффузный отражатель.

Специфику зеркального отражения реальных объектов в значительной мере объясняет теория отражения излучения на гладкой границе изотропных диэлектриков, развитая Френелем. Поэтому формулы Френеля для естественно поляризованного света [15] могут быть использованы при моделировании зеркального отражения как определенное приближение. Чтобы использовать френелевский коэффициент отражения  $r_3$ , выразим зеркальную составляющую отраженного излучения через этот коэффициент. Зеркально отраженный от площадки  $\Delta S$  лучистый поток составляет

$$\Phi_{\text{отр}_3} = r_3 E_0 \cos\theta \Delta S. \quad (1.4.10)$$

С другой стороны, величину  $\Phi_{\text{отр}_3}$  в соответствии с (1.2.9) можно вычислить следующим образом:

$$\Phi_{\text{отр}_3} = 2\pi \Delta S \int_{-\pi/2}^{\pi/2} E_0 \pi^{-1} \cos\theta r_{3m} \cos^n \beta \sin\beta \cos(\theta + \beta) d\beta. \quad (1.4.11)$$

Так как значения подынтегральной функции имеют существенные значения в сравнительно узком диапазоне углов  $\beta$ , можно полагать  $\cos(\theta + \beta) \approx \cos\theta$ .

Тогда

$$\Phi_{\text{отр}_3} = 2 \Delta S r_{3m} E_0 \cos^2 \theta / (n+1). \quad (1.4.12)$$

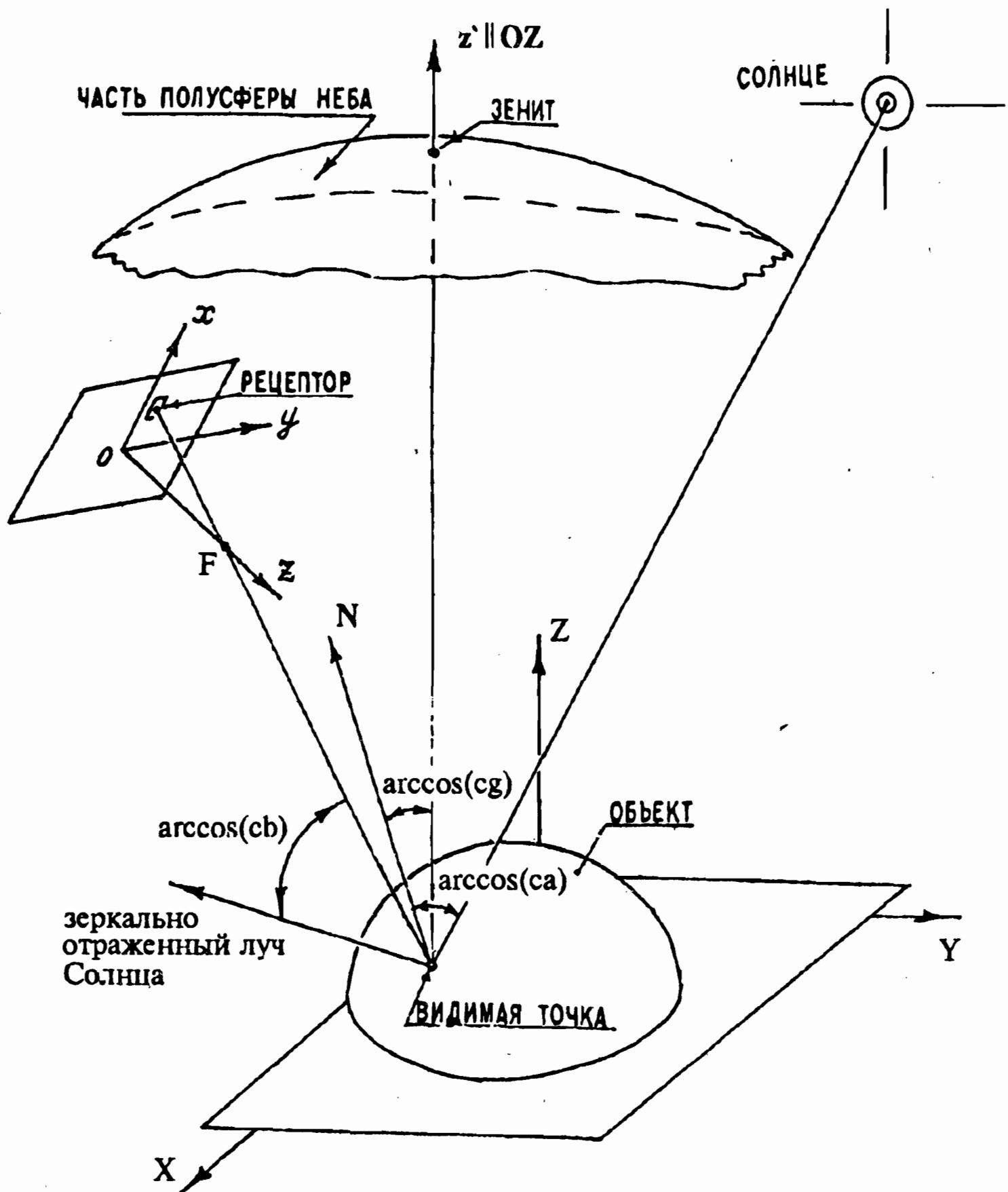


Рис. 1.4.3. Пояснение расположения векторов и углов к программе E2 (см. приложение)

Из (1.42) и (1.44) определяем связь  $\rho_3$  и  $r_{3m}$ :

$$r_{3m} = 0.5(n+1)\rho_3 / \cos\theta. \quad (1.4.13)$$

С учетом (1.4.13) формулу (1.4.9) можно привести к виду

$$L = 0.5r_a [1 + (N \cdot N_r)] L_{c\phi} + (E_0 / \pi) \{ [r_d + r_{obr.m} (N_u \cdot N_L)^p] (N_u \cdot N_L) (N_L S)^n +$$

$$+r_{\text{обр}_m}(N_u \cdot N_L)^p](N_u \cdot N) + 0,5(n+1)\rho_3(N_L S)^n \} . \quad (1.4.14)$$

Формула (1.4.14) может быть легко обобщена на случай освещения несколькими источниками.

В приложении приведена программная реализация на Турбо Паскале упрощенной модели расчета освещенности (модули E2, ALFF, ВЕГТА). Пояснения к использованию E2 приведены в комментариях к тексту программы, на рис. 1.4.3 показаны направления векторов и правила отсчета углов.

## 1.5. ПРОЕКЦИИ В ТРЕХМЕРНОЙ МАШИННОЙ ГРАФИКЕ И ИХ МАТЕМАТИЧЕСКИЕ МОДЕЛИ

Все виды проекций, используемые в инженерной графике, а также реализуемые в системах формирования изображений, представляют собой комбинации двух главных типов проекций: параллельной и перспективной (центральной).

### 1.5.1. ПАРАЛЛЕЛЬНАЯ ПРОЕКЦИЯ

В инженерной графике и начертательной геометрии преимущественно используется параллельная проекция. Точки предмета проецируются на поверхность пучком лучей, параллельных заданному направлению  $V$  (рис. 1.5.1).

Координаты точек изображения при параллельной проекции определяются путем совместного решения уравнения прямой, проходящей через предметную точку A параллельно единичному вектору, и уравнения поверхности проекции:

$$\begin{aligned} (X_A - X_{A'})/V_X &= (Y_A - Y_{A'})/V_Y = (Z_A - Z_{A'})/V_Z; \\ F(X_{A'}, Y_{A'}, Z_{A'}) &= 0, \end{aligned} \quad (1.5.1)$$

где  $X_A, Y_A, Z_A$  – координаты точки предмета;  $X_{A'}, Y_{A'}, Z_{A'}$  – координаты точки изображения;  $F(X_{A'}, Y_{A'}, Z_{A'})$  – уравнение поверхности проекции.

Если проецирование осуществляется на плоскость и проектирующие лучи перпендикулярны к ней, то проекция называется ортогональной или перпендикулярной. Этот вид проекции широко используется в техническом черчении. Если оси X, Y лежат в плоскости проекции, а Z перпендикулярна ей, то при представлении предмета в координатном базисе плоскости проекции, координаты точек изображения можно определить по координатам точек предмета с помощью простого соотношения

$$[X_{A'} \ Y_{A'} \ Z_{A'}] = [X_A \ Y_A \ Z_0], \quad (1.5.2)$$

где  $Z_0 = \text{const}$  – координата плоскости проекции по оси Z.

Как следует из (1.5.2), для получения ортогональной проекции принципиально достаточно определить координаты XA и YA предмета. Однако этим не ограничивается процедура построения трехмерной модели на экране дисплея. Предмет обычно задается в своей объектной системе координат XYZ, оси которой в исходном состоянии параллельны экранной системе дисплея xyz. Изображение предмета отображается на экране дисплея с некоторым масштабным коэффициентом m, а начало объектной системы координат располагается в точке  $x_0, y_0, z_0$  экрана. Для этого более общего случая можно записать

$$\begin{bmatrix} x_{A'} \\ y_{A'} \\ z_{A'} \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + m \begin{bmatrix} 1 & X_A \\ 1 & Y_A \\ 0 & Z_A \end{bmatrix}, \quad (1.5.3)$$

Выбором значения  $m$  можно осуществлять масштабирование изображения объекта.

Чтобы обеспечить наиболее наглядное отображение объемности предмета, обычно выбирается положение предмета относительно плоскости проекции. В техническом черчении положение предмета определяется типом аксонометрической проекции. Чтобы получить аксонометрическую проекцию, предмет вместе со связанный системой координат разворачивается таким образом, что проекции координатных осей получают определенную ориентацию друг относительно друга, а отрезки, взятые по координатным осям, отображаются на проекции с определенным соотношением масштабных коэффициентов.

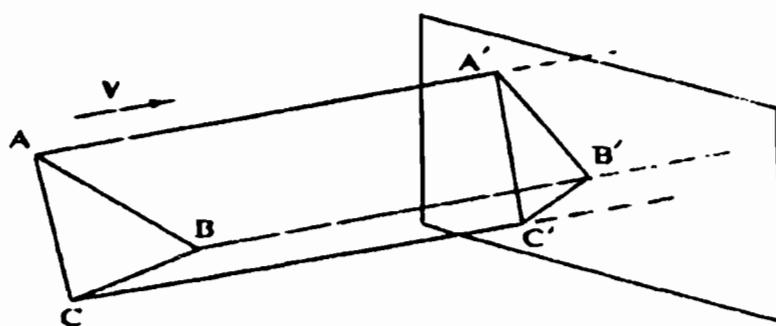


Рис. 1.5.1. Параллельная проекция

Чтобы в машинной графике воспроизвести предмет в заданной проекции, необходимо определить матрицу преобразования координат. Основой для этого являются принципы построения изображения в данной проекции. В частности, в изометрической проекции координатные оси предмета  $X, Y, Z$  отображаются на плоскости экрана дисплея под углом  $120^\circ$ , а масштабные коэффициенты по всем трем осям одинаковы (рис.1.5.2). С учетом этих свойств изометрической проекции получим

$$\begin{bmatrix} x_{A'} \\ y_{A'} \\ z_{A'} \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + m \begin{bmatrix} 1 & 3^{0.5}/2 & 3^{0.5}/2 \\ 1 & -0.5 & 1 & -0.5 \\ 0 & 3^{-0.5} & 3^{-0.5} & 3^{-0.5} \end{bmatrix} \begin{bmatrix} X_A \\ Y_A \\ Z_A \end{bmatrix}. \quad (1.5.4)$$

Аналогично можно получить формулу преобразования для фронтальной проекции, при которой оси  $X, Y$  проецируются параллельно осям  $x, y$  дисплея, а проекция оси  $Z$  совпадает с биссектрисой угла между осями  $x$  и  $y$  (рис.1.5.2). В этом случае преобразование координат выполняется следующим образом:

$$\begin{bmatrix} x_{A'} \\ y_{A'} \\ z_{A'} \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + m \begin{bmatrix} 1 & 1 & -2^{0.5}/4 \\ 1 & 0 & 2^{0.5}/4 \\ 0 & 2^{0.5}/4 & (2/3)^{0.5} \end{bmatrix} \begin{bmatrix} X_A \\ Y_A \\ Z_A \end{bmatrix}. \quad (1.5.5)$$

Можно отметить, что фронтальная проекция не является линейным преобразованием пространства предметов.

Формулы преобразования (1.5.4) и (1.5.5) принципиально позволяют обеспечить вычерчивание на дисплее изображений в изометрической или фронтальной проекции, если задано математическое описание предмета. Однако такой узкопрактический подход не позволяет в полной мере использовать возможности машинной графики. В системах трехмерной машинной графики нет необходимости фиксировать какие-либо конкретные виды проекций. В программах формирования

трехмерных изображений обычно предусматриваются операции вращения предмета вокруг двух или трех осей. Это позволяет пользователю получать такой вид объекта, который наиболее выразительно отображает его геометрическую форму.

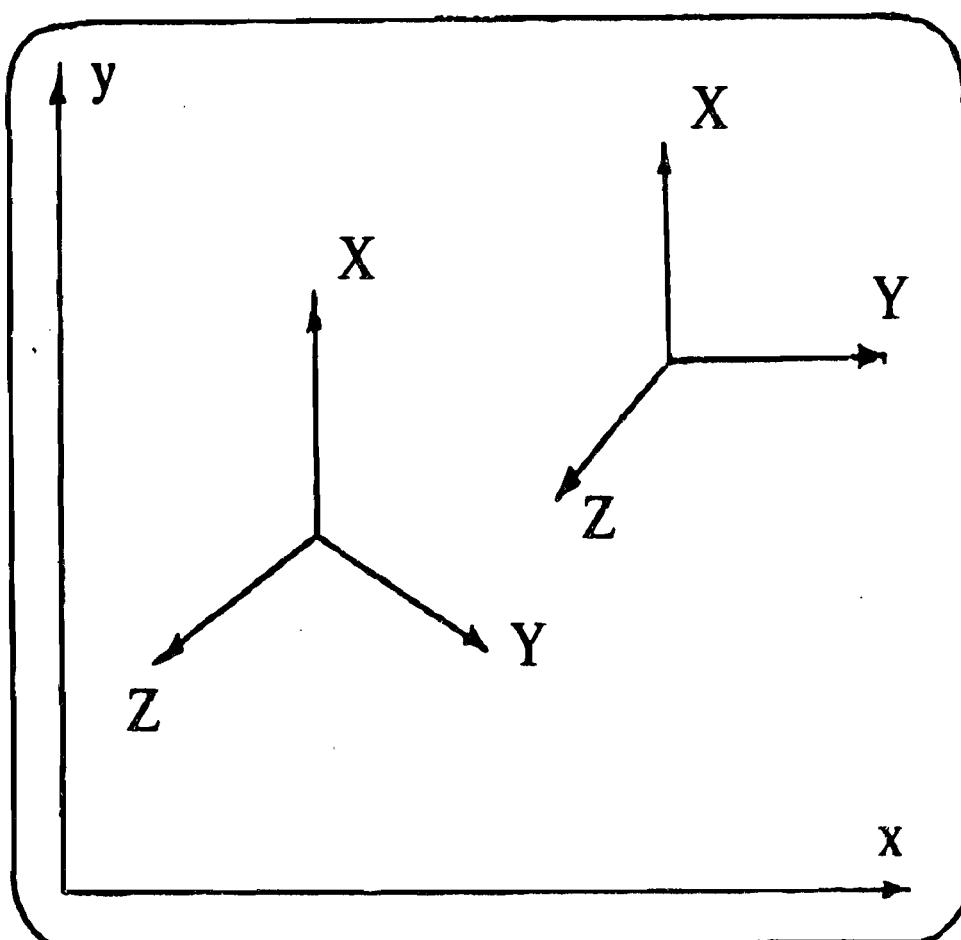


Рис. 1.5.2. Отображение координатных осей предмета на экране дисплея в изометрической и ортогональной проекциях

### 1.5.2. ЦЕНТРАЛЬНАЯ (ПЕРСПЕКТИВНАЯ) ПРОЕКЦИЯ

При центральной проекции все проектирующие лучи проходят через определенную точку пространства  $S$  – центр проекции. Физическим устройством, реализующим центральную проекцию, является объектив. При визуальном наблюдении роль объектива выполняет глаз. В объективе лучи, соединяющие сопряженные точки в пространстве предметов и изображений, проходят через заднюю главную точку, являющуюся центром проекции (рис.1.5.3). Из этого основного свойства центральной проекции вытекает математический метод построения изображения: координаты каждой точки изображения могут быть вычислены путем определения точки пересечения прямой, проходящей через предметную точку  $A$  и центр проекции  $S$ , с поверхностью проекции (изображения). Если в выбранной объектной системе координат  $OXYZ$  известны координаты точек  $A$  и  $S$ , а также уравнение поверхности изображения  $\phi(X, Y, Z) = 0$ , то координаты точки изображения  $A'$  определяются в результате решения системы уравнений

$$\begin{cases} (X_A - X_{A'}) / (X_A - X_S) = (Y_A - Y_{A'}) / (Y_A - Y_S) = (Z_A - Z_{A'}) / (Z_A - Z_S) : \\ \phi(X_{A'}, Y_{A'}, Z_{A'}) = 0. \end{cases} \quad (1.5.6)$$

Поверхность проекции в большинстве случаев можно считать плоской. Это приближение достаточно точно выполняется и для глаза. Хотя светочувствительная поверхность глаза – сетчатка имеет почти сферическую форму, для области ясного зрения, ограниченной угловым размером в несколько градусов, ее вполне можно считать плоской.

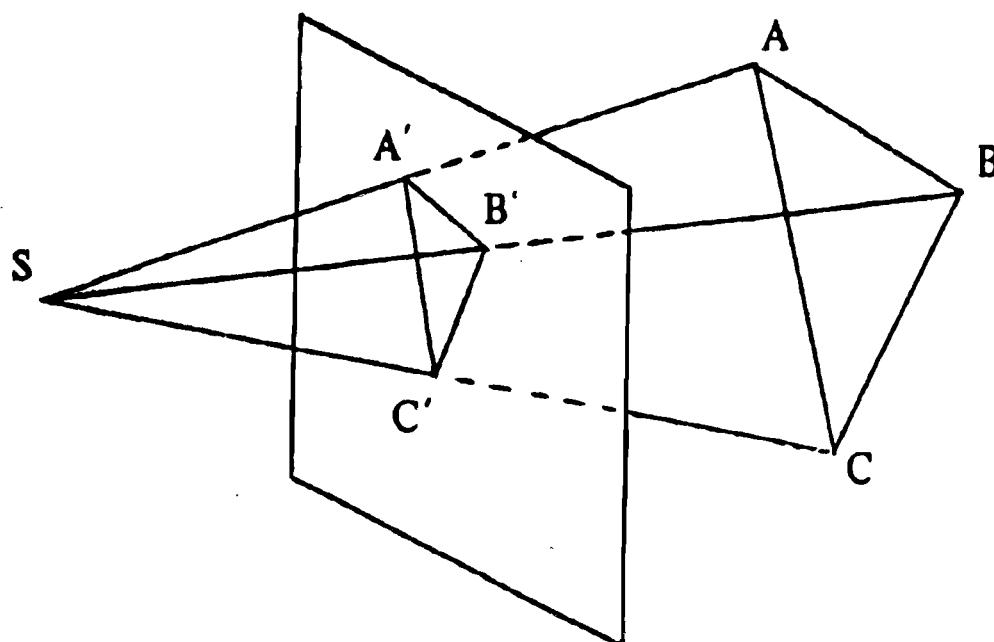


Рис. 1.5.3. Общая схема центральной проекции

В соответствии с законами оптики для получения резкого изображения необходимо, чтобы светочувствительная поверхность была перпендикулярна оптической оси объектива и располагалась на определенном расстоянии от центра проектирования, которое, как правило, принимают равным фокусному расстоянию  $f$ . Фактически изображение располагается на картинном расстоянии от центра проектирования, которое всегда больше фокусного. Однако если предмет удален от объектива на расстояние  $L >> f$ , то разница между картинным и фокусным расстоянием незначительна. Таким образом, положение плоскости изображения легко фиксировано относительно центра проекции  $S$  и оптической оси объектива. Если объектив разворачивается с тем, чтобы в его поле зрения попадали определенные объекты, то вместе с ним должна разворачиваться и плоскость изображения.

Если учесть отмеченные особенности центральной проекции в реальных устройствах формирования изображения, то связь координат точек пространства предметов и пространства изображений может быть выражена в иной форме, чем в системе уравнений (1.5.7). Введем систему координат  $xuz$  для плоскости изображения, связанную систему координат объектива  $x'y'z'$  и систему координат пространства предметов  $XYZ$  (рис. 1.5.4). Особенность центральной проекции можно выразить следующим образом: векторы  $r_{SA}$  и  $R_{AS}$ , соединяющие центр проекции  $S$  с сопряженными точками  $A$  и  $A'$ , являются коллинеарными. Отсюда следует

$$R_{AS} = \lambda r_{SA'}, \quad (1.5.7)$$

где  $\lambda$  – константа для данной пары точек  $A$  и  $A'$ .

Учитывая, что съемочная камера может быть развернута на углы  $\phi, \theta$  и  $\psi$  относительно осей  $x', y', z'$ , из (1.5.8) получаем

$$\begin{bmatrix} X - X_S \\ Y - Y_S \\ Z - Z_S \end{bmatrix} = \lambda [C] \begin{bmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{bmatrix}, \quad (1.5.8)$$

где  $X_S, Y_S, Z_S$  – координаты центра проектирования  $S$  в системе  $XYZ$ ;  $x_0, y_0, z_0$  – координаты центра проектирования  $S$  в системе  $xuz$ .

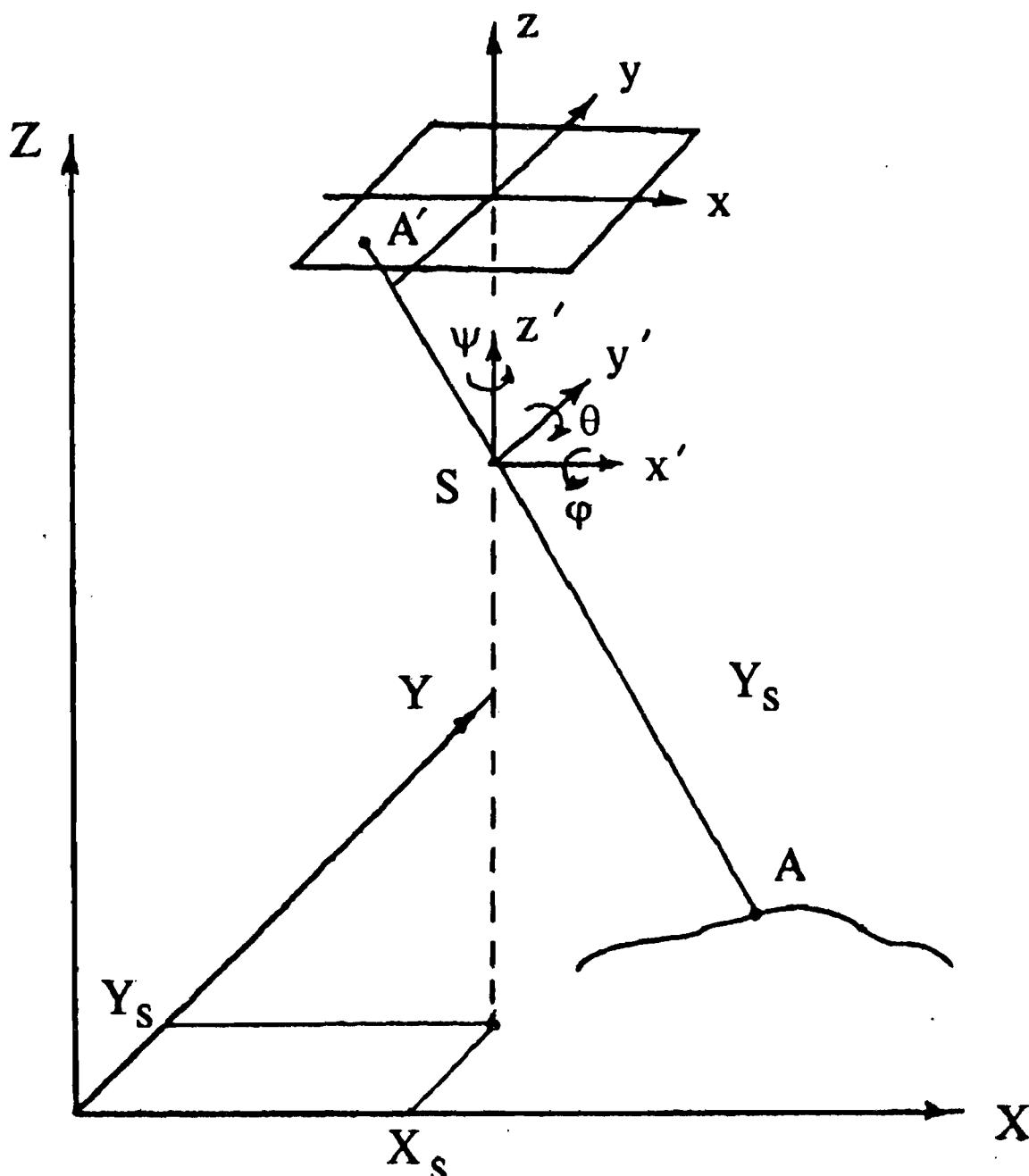


Рис. 1.5.4. Схема разворотов плоскости изображения

Если учесть, что поверхность проекции является плоской ( $z = 0$ ), начало координат системы  $xyz$ , как правило, совпадает с главной точкой картинной плоскости ( $x_0 = y_0 = 0$ ), расположенной на расстоянии  $f$  от  $S$  ( $z_0 = -f$ ), то

$$\begin{bmatrix} X - X_S \\ Y - Y_S \\ Z - Z_S \end{bmatrix} = \lambda [C] \begin{bmatrix} x \\ y \\ f \end{bmatrix} \quad (1.5.9)$$

Исключая в (1.5.9) константу путем деления первой и второй строк на третью, получаем уравнения, связывающие координаты сопряженных точек в системах  $XYZ$  и  $xyz$ :

$$\begin{aligned} (X - X_S) &= (Z - Z_S) (c_{11}x + c_{12}y + c_{13}f) / (c_{31}x + c_{32}y + c_{33}f); \\ (Y - Y_S) &= (Z - Z_S) (c_{21}x + c_{22}y + c_{23}f) / (c_{31}x + c_{32}y + c_{33}f). \end{aligned} \quad (1.5.10)$$

Из системы (1.5.10) по координатам точек изображения  $x, y$  можно определить координаты сопряженных точек  $X, Y, Z$  в пространстве предметов, если задано уравнение наблюдаемой поверхности  $F(X, Y, Z) = 0$ . Затем по координатам  $X, Y, Z$  точки  $A$ , уравнению поверхности  $F(X, Y, Z) = 0$  и известным условиям освещения могут

быть определены атрибуты точки А (яркость, цвет) и рассчитаны соответствующие атрибуты точки изображения А'. Описанная здесь кратко процедура синтеза изображений основана на отслеживании луча, исходящего из пространства изображений в пространство предметов, т.е. в направлении, противоположном ходу лучей в реальной системе. Этот подход в машинной графике назовем *методом обратного трассирования лучей*.

Из уравнения (1.5.9) можно получить формулы прямого преобразования

$$\begin{aligned} x &= f[c_{11}(X-X_S) + c_{12}(Y-Y_S) + c_{13}(Z-Z_S)]/[c_{13}(X-X_S) + c_{23}(Y-Y_S) + c_{33}(Z-Z_S)]; \\ y &= f[c_{21}(X-X_S) + c_{22}(Y-Y_S) + c_{32}(Z-Z_S)]/[c_{13}(X-X_S) + c_{23}(Y-Y_S) + c_{33}(Z-Z_S)]. \end{aligned} \quad (1.5.11)$$

В этом случае по координатам X, Y, Z точки А предмета и ее атрибутам определяются координаты x, y и атрибуты точки изображения А'.

Формулы (1.5.10) и (1.5.11) широко используются в фотограмметрии [30]. Они удобны при моделировании изображений, получаемых съемочными системами, анализе изменений масштаба по полю изображения. При моделировании крупных планов фотограмметрические формулы практически не используются, предпочтение отдается описанию трассирующих лучей уравнением прямой.

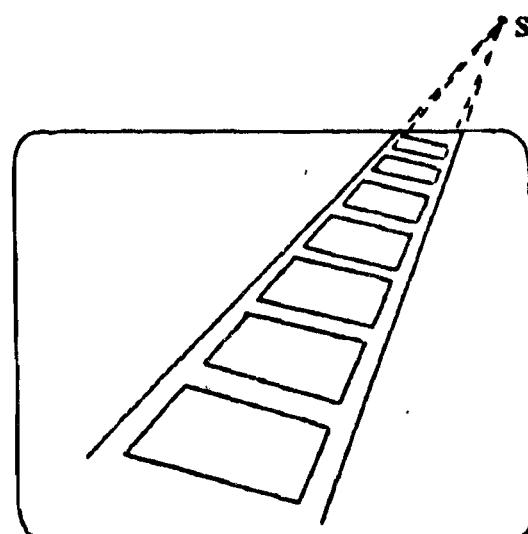


Рис. 1.5.5. Перспективное изображение, построенное с использованием точки схода лучей

Характерной особенностью центральной проекции является существенное различие в масштабах изображения предметов, находящихся на различных расстояниях от центра проектирования. Это связано с уменьшением угловых размеров предмета (и соответственно с уменьшением линейных размеров в плоскости изображения) при удалении от съемочной сцены. На рис. 1.5.5 приведен результат съемки предмета в виде полосы с нанесением на нее рисунка из периодически повторяющихся прямоугольников. Изменение ширины полосы и размеров прямоугольников создает ощущение глубины пространства. Принципиально изображение может быть рассчитано, например, по формулам (1.5.12), но его можно построить с достаточной степенью точности, если задать точку схода лучей. Расчеты с использованием координаты точки схода лучей значительно проще. Поэтому такой подход широко используется при имитации визуально наблюдаемой обстановки в видеотренажерах.

### 1.5.3. ОПИСАНИЕ ПРЕОБРАЗОВАНИЙ С ИСПОЛЬЗОВАНИЕМ ОДНОРОДНЫХ КООРДИНАТ

Классические геометрические преобразования (сдвиги, повороты) описываются с помощью математических моделей, основанных на использовании матриц размера 3x3. При классическом подходе каждое из преобразований

представляется отдельной матрицей. Для машинной графики это приводит во многих случаях к увеличению объема вычислений. Возникает необходимость в использовании математического аппарата, обеспечивающего более компактное описание геометрических преобразований. Наибольшее распространение для задач машинной графики получил *метод однородных координат*. В основе этого метода лежит представление о том, что каждая точка в N-мерном пространстве может рассматриваться как проекция точки из (N+1)-мерного пространства. В частности, точка в трехмерном пространстве представляется четырьмя составляющими  $w_x, w_y, w_z, w$ , где  $w$  может принимать любое значение. На практике в основном используется  $w = 1$ , что соответствует нормализованным координатам  $(x, y, z, 1)$ .

Свойства однородных координат позволяют выражать с помощью единой матрицы все преобразования: сдвиги, повороты и даже проекции (аксонометрические или центральные), а также любые сочетания преобразований в виде произведения матриц. Использование однородных координат позволяет применять единый математический аппарат для пространственных преобразований (поворотов, масштабирования, переноса) точек, прямых, квадратичных и бикубических поверхностей и линий.

Для трехмерной машинной графики все преобразования могут быть описаны матрицей  $4 \times 4$  следующего вида:

Классическая матрица поворотов	$\begin{array}{ccc c} a_{11} & a_{12} & a_{13} & d \\ a_{21} & a_{22} & a_{23} & e \\ a_{31} & a_{32} & a_{33} & f \\ \hline x & y & z & S \end{array}$	Векторы проекций
Вектор сдвига (переноса)		Общий масштаб

Основные преобразования выражаются с помощью матрицы  $4 \times 4$  следующим образом:

поворот на угол  $\phi$  вокруг оси x

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & \sin\phi & 0 \\ 0 & -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

поворот на угол  $\theta$  вокруг оси y

$$R_y = \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

поворот на угол  $\psi$  вокруг оси z

$$R_z = \begin{bmatrix} \cos\psi & \sin\psi & 0 & 0 \\ -\sin\psi & \cos\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

сдвиг на вектор  $(x, y, z)$

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x & y & z & 1 \end{bmatrix};$$

геометрическое преобразование

$$M = \begin{bmatrix} p & 0 & 0 & 0 \\ 0 & q & 0 & 0 \\ 0 & 0 & r & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Совокупность операций по преобразованию координат предмета описывается произведением матриц, которое затем приводится к единой матрице для всех элементов и точек предмета.

Предположим, что задано осуществить повороты предмета на угол  $\phi_0$  вокруг оси, параллельной оси  $x$ ; и на угол  $\psi_0$  вокруг оси  $z$ , проходящих через точку с координатами  $(x_0, y_0, z_0)$ . Эта операция будет описываться произведением четырех матриц: матрицы  $T_1$ , описывающей сдвиг для совмещения точки  $(x_0, y_0, z_0)$  с началом координат; двух матриц  $R_x$  и  $R_z$ , описывающих повороты вокруг соответствующих осей; матрицы  $T_2$ , описывающей сдвиг для возвращения точки  $(x_0, y_0, z_0)$  в первоначальное положение.

Результирующая матрица  $M$  имеет вид  $M = T_1 R_x R_z T_2$ .

#### 1.5.4. ПРОЕКЦИИ ПРИ РАЗЛИЧНЫХ ВИДАХ СЪЕМКИ

Одной из областей трехмерной машинной графики является моделирование изображений, получаемых съемочными камерами. В зависимости от технической реализации съемочной камеры можно выделить три основных вида съемки: *кадровую, щелевую и панорамную*.

Геометрическая схема кадровой съемки полностью совпадает со схемой, изображенной на рис.1.5.4. При кадровой съемке можно пренебрегать поступательным и угловым перемещениями камеры. Кадровые снимки формируются по законам центральной проекции. Связка проектирующих лучей сохраняет неизменное положение в пространстве. Основными параметрами, необходимыми для моделирования кадрового снимка, являются пространственные координаты центра проектирования, углы разворота и фокусное расстояние объектива. Принципиально можно промоделировать и смаз изображения, возникающий из-за движения камеры во время экспонирования кадра. Для учета движения съемочной камеры необходимо получить уравнения движения оптического изображения в ней. Эта задача требует значительного увеличения объема вычислений и решается лишь при специальных исследованиях.

При щелевой съемке изображение получают последовательно строка за строкой (рис.1.5.6). В плоскости изображения устанавливается непрозрачный экран с узкой щелью, через которую изображение регистрируется на фотопленке. Роль щели может также играть линейка фотоприемников, преобразующая в электрический сигнал одну строку в пространстве предметов. Сканирование пространства предметов проекцией экспозиционной щели (или проекцией линейки фотоприемников) осуществляется перемещением съемочной камеры вдоль координаты  $X$ . При перемещении съемочной камеры изображение в фокальной плоскости перемещается со скоростью  $v_x$ . Для получения резкого изображения фотопленка должна протягиваться относительно экспозиционной щели синхронно с изображением  $v_{\text{фп}} = v_x$ . Если изображение регистрируется линейкой приемников, то период считывания сигналов линейки также должен быть синхронизирован с движением изображения. Период считывания  $T_c$  должен удовлетворять соотношению

$$v_x T_c = v_{\text{фп}}, \quad (1.5.12)$$

где  $x_{\text{ФП}}$  – ширина светочувствительной области (размер фотоприемника) в направлении движения изображения.

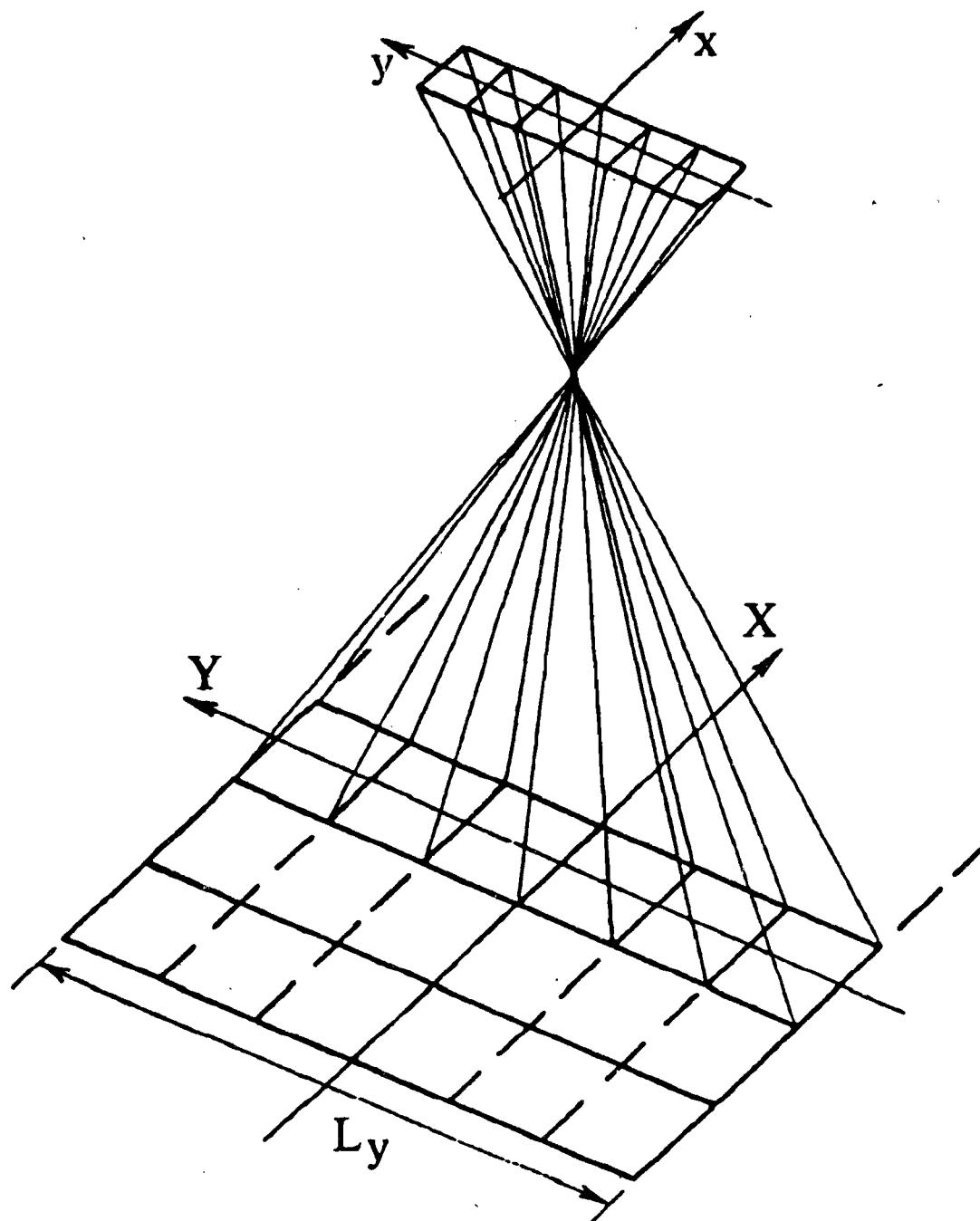


Рис. 1.5.6. Схема щелевой съемки

Условие слитности изображения (1.5.12) можно сформулировать в иной форме: период считывания строк равен временному интервалу, в течение которого съемочная камера перемещается на расстояние, равное ширине строки в пространстве предметов. При этом строки в пространстве предметов будут примыкать друг к другу без пропусков и наложений. Исходя из принципов получения щелевого снимка, несложно определить методы его моделирования. Для этого необходимо определить размер проекции фотоприемника (пикселя) в пространстве предметов в направлении движения съемочной камеры, затем задать ряд последовательных положений съемочной камеры с шагом, равным размеру пикселя по оси X, и для каждого положения камеры построить строки изображения по законам центральной проекции. Получаемое изображение представляет собой сочетание двух проекций: все сечения, параллельные оси Y, определяются законами центральной проекции, а сечения, параллельные оси X, формируются в результате параллельной проекции с различными углами наклона лучей к плоскости проекции.

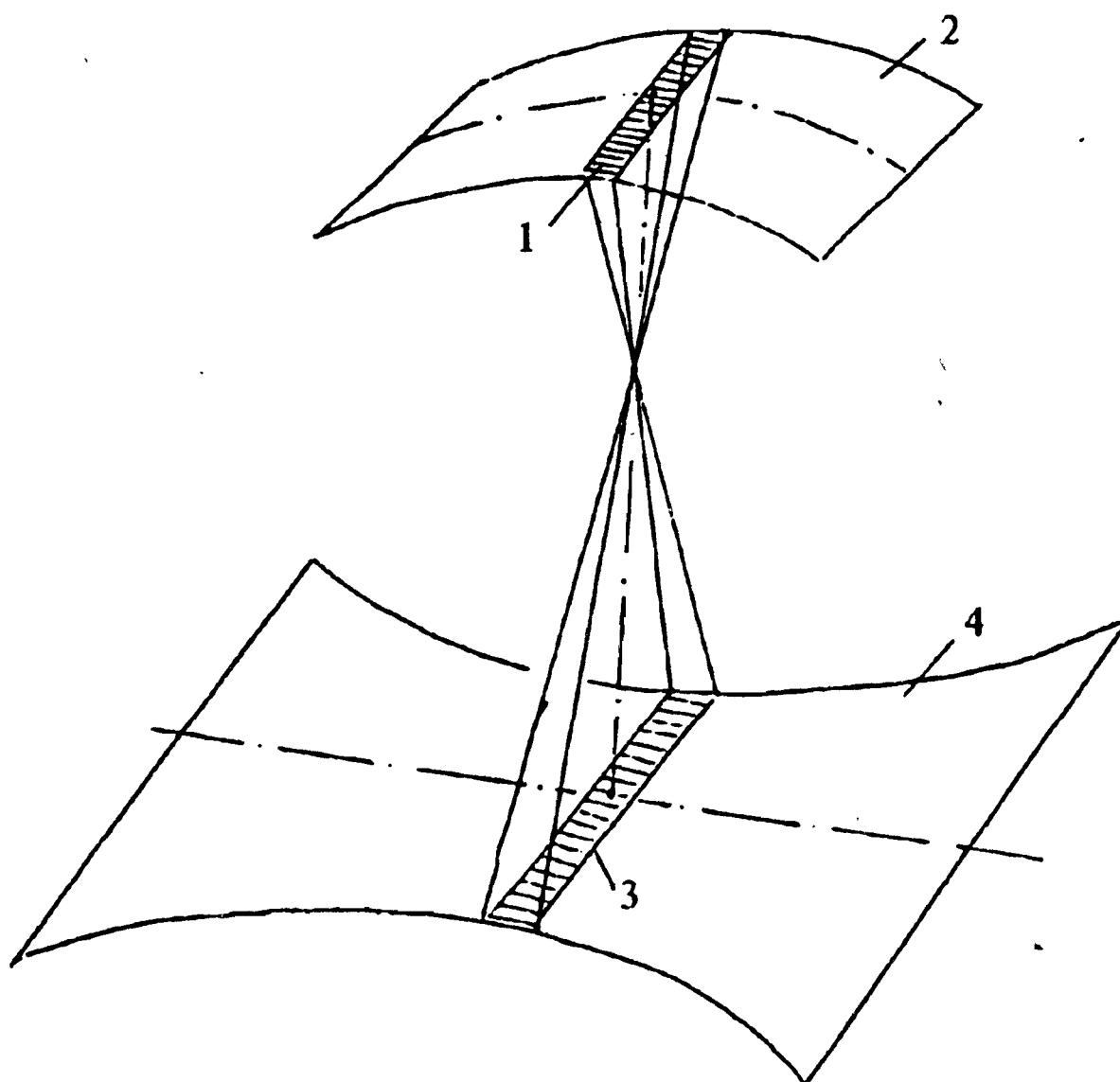


Рис. 1.5.7. Схема панорамной съемки:

1 – панорамирующая щель; 2 – поверхность проекции; 3 – отображение панорамирующей щели в пространстве предметов; 4 – участок пространства предметов, отображаемый на панорамном снимке

При панорамной съемке мгновенное поле изображения также имеет вид узкой щели (рис.1.5.7). Объектив съемочной камеры вместе с экспозиционной щелью поворачивается вокруг оси панорамирования, проходящей через центр проекции. Эта ось параллельна длинной стороне щели. В результате происходит поворот плоского пучка проектирующих лучей, формирующего строки панорамного изображения, параллельные оси X. В общем случае при поступательном перемещении съемочной камеры для моделирования каждой строки панорамного изображения необходимо задавать не только угловое положение связки проектирующих лучей, но и координаты центра проекции, соответствующие данному моменту времени. Однако особенности панорамных снимков в первом приближении можно оценить, не учитывая поступательное перемещение съемочной камеры. Как следует из рис.1.5.7, панорамное изображение при неподвижной съемочной камере представляет собой центральную проекцию пространства предметов на цилиндрическую поверхность.

Близкой к панорамной является съемка с линейно-строчным сканированием. Связка проектирующих лучей разворачивается сканирующим устройством параллельно оси Y, а регистрация осуществляется одним или несколькими фотоприемниками в плоскости изображения. В результате формируется одна строка или субкадр из нескольких строк. Следующая строка или субкадр формируются при перемещении

съемочной камеры на расстояние, равное ширине их проекции в пространстве предметов по оси X.

Краткий обзор различных видов съемки показывает, что для реальных съемочных систем проецирование в плоскость изображения, как правило, носит более сложный характер, чем параллельная или центральная проекция. Однако у всех видов съемки есть и общее: в любой фиксированный момент времени локальный участок изображения строится по законам центральной проекции. Если учесть это, то для моделирования любого вида съемки достаточно определить последовательный ряд параметров центральной проекции и построить соответствующие этому ряду участки изображения.

## 1.6. ТЕКСТУРА ИЗОБРАЖЕНИЯ

Под текстурой понимают специфические изменения тона (цвета) в изображении объекта или его некоторой части. К текстуре относят также наличие на изображении характерных линий. Наряду с термином "текстура" широко используются такие, как "узор", "рисунок", "фактура".

Текстура может представлять упорядоченное изменение тона в виде геометрически правильных или почти правильных рисунков. Подобную текстуру имеют, например, кирпичная кладка, кафельная облицовка, специально раскрашенные поверхности: шахматная доска, пешеходный переход и т.д. Другой тип – стохастическая текстура. Она присуща естественным объектам и, как правило, является следствием шероховатости наблюдаемых объектов.

Одним из основных принципов формирования текстуры изображения является перенос регулярного или стохастического рисунка на поверхность объекта. Этот подход и его различные варианты излагаются в ряде работ, краткий обзор которых приведен в [49]. Для нанесения рисунка необходимо решить задачу преобразования систем координат. Если рисунок задан в пространстве текстуры в системе координат ( $u, w$ ), а поверхность – в системе координат ( $\theta, \varphi$ ), то для переноса рисунка нужно найти или задать функцию отображения одного пространства на другое, т.е. определить соотношения  $\theta = f(u, w)$ ,  $\varphi = g(u, w)$  или  $u = r(\theta, \varphi)$ ,  $w = s(\theta, \varphi)$ .

Во многих практических случаях функции отображения линейны:  $\theta = Au + Bw + C$ ,  $\varphi = Du + Ew + F$ . Коэффициенты A, B, C, D, E, F вычисляются из соотношений между известными точками в системе координат.

Характерные точки узора из пространства текстуры переносятся в объективное пространство, затем в пространство изображений и определенным образом соединяются отрезком прямых. В процедурах нанесения узоров часто используется разбиение как поверхностей, так и узоров на фрагменты.

При переносе на поверхность предмета текстуры, представляющей непрерывное регулярное или случайное поле яркости, наиболее подходящим является метод обратного трассирования лучей. Центр каждого пикселя изображения проецируется на поверхность предмета, и по координатам точки на поверхности определяется соответствующая ей точка в пространстве текстуры. Чтобы устранить эффекты, связанные с пространственной дискретизацией, используют процедуры сглаживания. Достаточно простой процедурой сглаживания является трассирование четырех точек, соответствующих углам пикселя, и использование среднего значения яркости текстуры для этих четырех точек. Использование четырех точек пикселя позволяет оценивать достаточность процедуры сглаживания. Если разность яркостей для указанных четырех точек больше некоторой пороговой величины, можно произвести разбиение пикселя на четыре части и более точно вычислить яркость для данного пикселя.

Формирование стохастических текстурных полей, близких к естественным, представляет сложную в вычислительном отношении процедуру [46]. Для синтеза текстур предпочтительными считаются авторегрессионная модель и модели с использованием методов линейного программирования. Синтез осуществляется с использованием статистических характеристик текстур-прототипов: плотности вероятности и автокорреляционной функции. Для определения плотности вероятности в пределах некоторого окна измеряют гистограмму и для ее отображения используют первые четыре момента – среднее значение, стандартное отклонение, асимметрию и эксцесс. Форму автокорреляционной функции выражают через ее четыре центральных пространственных момента. В результате формируется восьмимерный вектор признаков текстуры. Наряду с описанным основным вариантом определения параметров, характеризующих стохастическую текстуру, используют ряд упрощенных процедур, обеспечивающих снижение вычислительных затрат.

Синтезированные на основе прототипов стохастические текстуры хорошо моделируют естественные шероховатые поверхности, если они обладают сравнительно небольшой кривизной. Для поверхностей со значительной кривизной (шар, эллипсоид и близкие к ним) перенос на них даже стохастической текстуры не обеспечивает их реалистичного вида. Возникает ощущение, что текстура нарисована на гладкой поверхности. Причина этого заключается в том, что текстура, сформированная без учета формы объекта, не передает изменений освещенности, обусловленных рельефом поверхности. Для реальных шероховатых поверхностей вектор нормали содержит небольшую случайную составляющую, которая определяет характер изменения освещенности в изображении. Чтобы получить изображение, близкое к реалистичному, необходимо смоделировать именно этот фактор. В методе моделирования рельефа поверхности  $Q(u,w)$  с помощью вспомогательной функции  $F(u,w)$  в направлении нормали к поверхности вносится возмущение, что приводит к изменению отражения от элементов вновь образованной поверхности.

Нормаль  $N$  к исходной поверхности определяется векторным произведением  $N = Q'_u Q'_w$ , где  $Q'_u$  и  $Q'_w$  – частные производные по направлениям  $u$  и  $w$ .

Нормаль  $N'$  к возмущенной поверхности при малом возмущении  $F(u,w)$  вычисляется по формуле  $N' = N + F_u(N Q'_w)/|N| + F_w(Q'_u N)/|N|$ , где  $F_u$  и  $F_w$  – сечения функции  $F(u,w)$  в данной точке по направлениям  $u$  и  $w$ .

В качестве функции  $F(u,w)$  могут быть использованы как аналитические функции, так и заданные в виде цифрового массива. В частности, в виде цифрового массива можно задать стохастическую текстуру. Изображение при использовании данного метода отображения шероховатости поверхности приобретает вид, чрезвычайно близкий к реалистичному. Необходимый объем вычислений для алгоритма возмущения нормали весьма значителен. Он приблизительно в 4...5 раз больше, чем при отображении текстуры без учета рельефа.

Аналогичные методы возмущения используют для придания естественного вида цветным поверхностям. К исходному цвету добавляется случайная или псевдослучайная составляющая другого цвета. В результате точки поверхности приобретают цвет, являющийся интерполяцией между двумя крайними цветами. Этот метод подходит для моделирования поверхностей из гравия, цемента и т.п.

Наиболее сложным в вычислительном отношении является формирование текстуры изображений, содержащих существенные нерегулярности. К этому типу относятся изображения значительного числа природных объектов: каменные осыпи, деревья, облака, всхолмленные ландшафты и т.п. Для формирования изображений подобных объектов в настоящее время стали широко использоваться фрактальные

поверхности, состоящие из случайно заданных полигональных или биполигональных поверхностей. Фрактальные поверхности получают путем рекурсивного разбиения поверхности на фрагменты. Число разбиений обычно весьма велико – от нескольких тысяч до миллионов.

## 1.7. МОДЕЛИРОВАНИЕ ЭНЕРГЕТИЧЕСКИХ ПРЕОБРАЗОВАНИЙ В СИСТЕМАХ ФОРМИРОВАНИЯ ИЗОБРАЖЕНИЙ

Первым звеном, осуществляющим преобразования лучистого потока от элементов пространства предметов, является оптическая система, с помощью которой формируется первичное изображение на фоточувствительной поверхности фотопреобразователя. Геометрические и энергетические характеристики первичного изображения обычно определяются на основе геометрической оптики, в соответствии с которой первичное изображение можно рассматривать как центральную проекцию наблюдаемого пространства на картинную плоскость. Центральная проекция обеспечивает точное моделирование процессов геометрического и энергетического преобразования, но не позволяет охарактеризовать передачу мелкой структуры изображения, поскольку сам принцип центральной проекции не предусматривает учета дифракции, aberrаций и других явлений, приводящих к размытию изображения. Для моделирования возможностей системы по воспроизведению мелких деталей регистрируемого сюжета необходима отдельная процедура.

При анализе энергетических преобразований и определении освещенности изображений в картинной плоскости будем полагать, что расстояние от плоскости предметов до входного зрачка оптической системы значительно больше фокусного расстояния ( $H \gg f$ ), а угловая ширина индикатрисы излучения любого участка наблюдаемых поверхностей существенно превышает угловой размер входного зрачка оптики из любой точки пространства предметов. Для определения освещенности первичного изображения воспользуемся методами центральной проекции (рис.1.7.1). Элементу  $dS$  на поверхности наблюдения соответствует элемент  $dS'$  в картинной плоскости. Так как проективное преобразование оптической системы является центральным, то угол  $d\omega$ , опирающийся на площадку  $dS$ , равен углу  $d\omega'$  в пространстве изображений. Отсюда следует

$$dS \cos \epsilon' / (H^2 / \cos^2 \epsilon') = dS' \cos w / (f^2 / \cos^2 w). \quad (1.7.1)$$

Лучистый поток, собираемый системой от элемента  $dS$ , составляет

$$d\Phi = L dS \cos \epsilon' d\Omega \tau_{oc} \tau_{cp}, \quad (1.7.2)$$

где  $d\Omega$  – телесный угол, образуемый косинусом лучей, поступающих от точки объекта в оптическую систему;  $\tau_{oc} = k(w) \tau_{opt}$  – коэффициент ослабления излучения оптической системой, равный произведению коэффициента виньетирования  $k(w)$  и коэффициента пропускания оптики  $\tau_{opt}$ ;  $\tau_{cp}$  – пропускание слоя среды между объектом и оптической системой.

Учитывая, что  $d\Omega = S_{opt} \cos w / (H^2 / \cos^2 \epsilon') = \pi D_{opt}^2 \cos w \cos^2 \epsilon' / (4H^2)$ , получаем

$$d\Phi = \pi D_{opt}^2 L \cos w \cos^3 \epsilon' k(w) \tau_{oc} \tau_{cp} dS / (4H^2). \quad (1.7.3)$$

Величина освещенности в плоскости первичного изображения определяется следующим образом:  $E_{iz} = d\Phi / dS' = \pi D_{opt}^2 L \cos w \cos^3 \epsilon' k(w) \tau_{oc} \tau_{cp} dS / (4H^2 dS')$ .

Так как согласно (1.7.1)  $dS/dS' = \cos^3 w \cos^{-3} \epsilon' (H/f)^2$ , то окончательно получаем

$$E_{iz} = \frac{\pi}{4} (D_{opt}/f)^2 L k(w) \tau_{opt} \tau_{cp} \cos^4 w. \quad (1.7.4)$$

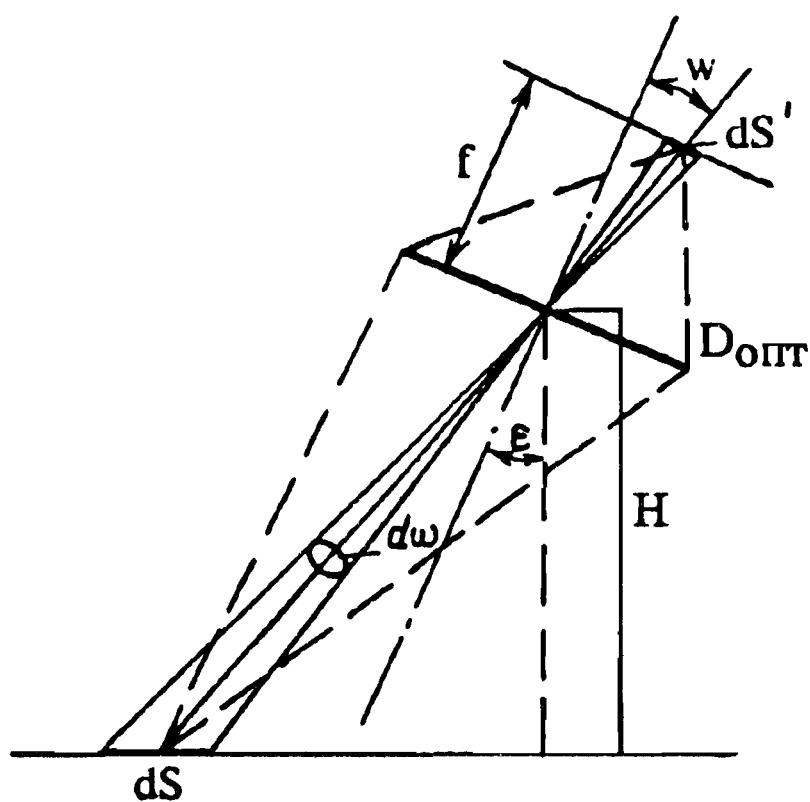


Рис. 1.7.1. Геометрическая схема для определения освещенности первичного изображения

Формула (1.7.4) справедлива для интегральных величин. Если необходимо определить спектральную освещенность, то формула будет иметь аналогичный вид, но вместо интегральных в нее будут входить соответствующие спектральные величины.

Рассмотрим зависимость освещенности первичного изображения от различных факторов.

Неравномерность освещенности по полю изображения характеризуется коэффициентом  $G(w) = k(w) \cos^4 w$ . Этот коэффициент можно выразить через координаты первичного изображения  $x$  и  $y$ , отсчитываемые от главной точки картинной плоскости:  $G(w) = k(w) / [1 + (x^2 + y^2)/f^2]^2$ .

Коэффициент  $G(w)$  имеет максимальное значение при центральной части изображения  $G(w) = 1$  при  $w = 0$  и  $k(w) = 1$  и уменьшается для периферийных областей. При размере первичного изображения  $(x^2 + y^2)^{0.5}/f < 0.2$ , что приблизительно соответствует углам поля зрения  $20\ldots 25^\circ$ , падение освещенности на краях изображения не превышает 10 %. Целесообразность моделирования неравномерности освещенности по полю зрения возникает, как правило, для систем со сравнительно большим угловым полем зрения  $2w > 30\ldots 40^\circ$  или в тех случаях, когда в оптической системе имеет место значительное виньетирование. Аргументом функции виньетирования  $k(w)$  является  $w$ , которую можно вычислить по координатам точки изображения  $w = \arctg[(x^2 + y^2)^{0.5}/f]$ .

Другим важным вопросом для процедуры синтеза является моделирование влияния дальности до объекта на величину освещенности получаемого первичного изображения. В ряде работ по машинной графике [48, 60] в формулах для расчета освещенности вводится сомножитель  $I/(d+R)$ , где  $d$  – эмпирически подбираемая константа по субъективной оценке изобразительных свойств синтезируемого изображения. Качественным обоснованием этой модели является то, что интенсивность света обратно пропорциональна квадрату расстояния от источника. Поскольку введение сомножителя  $I/(d+R)$  на практике позволяет в ряде случаев повышать

реалистичность синтезируемых изображений, целесообразно рассмотреть, в каких случаях оно совпадает с реальным изменением освещенности, определяемым (1.7.4).

Если среда распространения является непоглощающей ( $\tau_{\text{ср}} = 1$ ), то в соответствии с (1.7.4) при постоянной яркости  $L$  наблюдаемого сюжета освещенность первичного изображения  $E_{\text{из}}$  не зависит от дальности наблюдения и угла визирования  $\varepsilon$ . При освещении сюжета точечным источником яркость объектов будет действительно обратно пропорциональна квадрату расстояния до источника, но расстояние между объективом (глазом) и наблюдаемой поверхностью влияния на освещенность первичного изображения не оказывает. При естественном освещении яркость объектов при несущественном поглощении в среде (состояние чистой атмосферы) не зависит от дальности до них.

Зависимость освещенности изображения объектов в съемочной системе от дальности до них связана лишь с затуханием (светорассеиванием) в среде распространения излучения (атмосфере, воде и т.п.). В соответствии с законом Бугера–Ламберта затухание в среде описывается экспоненциальной зависимостью  $\tau_{\text{ср}} = \exp(-\beta R)$ , где  $\beta$  – коэффициент затухания;  $R$  – расстояние между данной точкой сюжета и центром проекции.

Для атмосферы  $\beta$  лежит обычно в диапазоне от  $20 \text{ км}^{-1}$  (сильный туман) до  $0,08 \text{ км}^{-1}$  (хорошая видимость).

Использование закона Бугера–Ламберта для учета затухания в среде дает точную модель для расчета освещенности. Сравним ее с приведенным ранее вариантом учета дальности до объектов. При небольшом общем затухании  $\beta R < 0,5$  экспоненциальную зависимость можно представить дробной рациональной функцией:

$$\exp(-\beta R) \approx 1/(1+\beta R) = \beta^{-1}/(\beta^{-1}+R).$$

Таким образом, при небольшом общем затухании характер зависимости освещенности от дальности до объекта практически совпадает при использовании экспоненциальной функции от  $R$  и функции  $1/(d + R)$ , если принять  $d = 1/\beta$ . С ростом величины  $\beta R$  расхождение в результатах расчета освещенности быстро возрастает, причем модель вида  $1/(d + R)$  дает завышенные значения освещенности первичного изображения для удаленных объектов. Например, при  $\beta R = 1$  модель  $1/(d + R)$  дает величину освещенности, завышенную примерно в 1,3 раза, при  $\beta R = 2$  – в 2,5 раза, при  $\beta R = 4$  – в 10 раз. При моделировании реальных съемочных систем это может привести к существенному завышению возможностей системы по регистрации удаленных объектов.

Среди фактов, которые целесообразно учитывать при моделировании реальных съемочных систем, следует отметить наличие в реальных сюжетах дымки. Результирующая освещенность содержит составляющие от дымки и наблюдаемых объектов. Дымку характеризуют либо яркость  $L_{\text{дм}}$ , либо коэффициент яркости  $r_{\text{дм}}$ . При расчете освещенности яркость равна сумме яркостей объекта и дымки:  $L = L_{\text{об}} + L_{\text{дм}}$ . Дымка в полной мере должна быть отражена на синтезируемом изображении при моделировании снимков фотографических систем или систем визуального наблюдения. В оптико-электронных системах постоянная составляющая дымки на изображении может отсутствовать, поскольку ее можно устраниć соответствующим выбором нуля шкалы сигналов в электронном тракте. В машинной графике во многих случаях абсолютные величины освещенности не рассчитываются. Моделируются лишь относительные изменения освещенности, затем

выбором некоторого множителя полученный диапазон освещенностей первичного изображения укладывается в определенный диапазон яркостей экрана дисплея. Такой подход обусловлен тем, что на практике параметры систем выбираются так, чтобы выходное изображение было близким к оптимальному для визуального наблюдения. Аналогичная картина имеет место и при непосредственном визуальном наблюдении: глаз адаптируется к освещенности сюжета [14].

Необходимость в расчете абсолютных величин сигналов возникает лишь в тех случаях, когда при синтезе изображений ставится задача отобразить влияние действующих на выходе шумов. Значение освещенности в плоскости первичного изображения определяет уровень фотонного шума. Определение уровня шумов требует учета характеристик фотопреобразователя электронного тракта усиления и преобразования сигналов регистратора. Подобная задача выходит за рамки машинной графики и детально здесь рассматриваться не будет.

## 1.8. СИНТЕЗ ИЗОБРАЖЕНИЙ С УЧЕТОМ СПЕЦИФИКИ ИХ ФОРМИРОВАНИЯ В РЕАЛЬНЫХ УСЛОВИЯХ

Синтез изображений на основе центральной поверхности соответствует приближению геометрической оптики. Каждой точке в пространстве предметов соответствует точка в пространстве изображений. Поэтому процедура такого синтеза ориентирована на генерирование идеального изображения. Для многих задач такой подход к синтезу изображений является вполне приемлемым, например когда синтез изображений осуществляется для оценки внешнего вида изделий, распределения объектов в пространстве и т.п. Однако существует целая группа прикладных задач, требующих от системы синтеза не идеальных по качеству изображений, а близких к тем, которые реально формируются в реальных устройствах и системах. Наиболее распространенной задачей этого типа являются моделирование работы систем формирования изображений и оценка влияния технических решений на выходное изображение.

С точки зрения создания программного обеспечения синтеза изображений наиболее рациональным является формирование процедур программных модулей, которые путем математических преобразований приводили бы синтезированное на первом этапе идеальное изображение к изображению, формируемому реальными устройствами. Такой подход обеспечивает универсальность программного обеспечения системы синтеза.

Изображения, получаемые в реальных оптических, фотографических и оптико-электронных устройствах, отличаются от изображений, построенных по законам геометрической оптики, размытием, наличием шумов, строчной или растровой структуры. Все эти факторы снижают качество изображения и приводят к некоторой потере информации. Размытие возникает вследствие неидеальности передаточных функций всех звеньев тракта преобразования оптического сигнала в изображение. Каждая точка в пространстве предметов отображается в пространстве изображений в виде пятна с распределением освещенности, пропорциональным  $h(x,y)$ . Функция  $h(x,y)$  называется *функцией рассеивания точки* или *импульсной реакцией системы*. Поскольку каждую функцию можно представить в виде комбинации точек, то выходное изображение в реальной системе можно представить как сумму реакций системы на каждую точку предмета:

$$E'(x,y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} E(\xi,\eta) h(x,y,\xi,\eta) d\xi d\eta, \quad (1.8.1)$$

где  $E'(x,y)$  – освещенность неразмытого, геометрически точного изображения (в приближении геометрической оптики).

Соотношение (1.8.1) называется *интегралом суперпозиции*, так как функция  $E'(x,y)$  выражает результат наложения реакций системы. В интеграле суперпозиции полагается, что реакция системы зависит от координат точки изображения  $x,y$ .

Для многих систем функция рассеяния  $h(x,y)$  незначительно изменяется по полю изображения. Такие системы принято называть *пространственно инвариантными*. Выходное распределение освещенности для этих систем выражается интегралом свертки:

$$E'(x,y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} E(\xi,\eta) h(x-\xi,y-\eta) d\xi d\eta. \quad (1.8.2)$$

В символической форме свертку записывают в следующем виде:  $E'(x,y) = E(x,y) \otimes h(x,y)$ , где  $\otimes$  – символ свертки.

Для реальных систем формирования изображений функция рассеяния  $h(x,y)$  определяется совместным действием нескольких факторов: дифракции на входном зрачке объектива; aberrации и дефокусировки объектива; скоростного сдвига изображения (смаза); атмосферной турбулентности и др. Каждый из этих факторов характеризуется своей функцией рассеяния  $h_i(x,y)$ . Результирующая функция рассеяния определяется как результат свертки функций рассеяния всех  $n$  звеньев:

$$h(x,y) = h_1(x,y) \otimes h_2(x,y) \otimes \dots \otimes h_n(x,y). \quad (1.8.3)$$

Операция свертки может быть упрощена несколькими путями. Очень часто двухмерная функция рассеяния может быть представлена в виде произведения двух одномерных функций:  $h(x,y) = h_1(x) h_2(y)$ . Это позволяет перейти к операциям одномерной свертки. Для одномерной свертки известен ряд алгоритмов, в которых сокращено число умножений из-за некоторого увеличения количества операций сложения. Простейший из них – алгоритм Винограда ускоренной свертки.

Непосредственно для двухмерных сверток сокращение объема вычислений обеспечивает алгоритм Нуссбаумера, в котором используются полиномиальные преобразования, позволяющие заменять значительную часть умножений циклическими сдвигами.

Известен класс алгоритмов вычисления свертки сигналов, вообще не требующий операций умножения. В них применяются теоретико-числовые преобразования.

Операция свертки широко используется в цифровых трактах современных телевизионных и оптико-электронных систем. Если свертка используется в цифровом тракте, ее, как правило, называют *фильтрацией*, а ядро свертки  $h_{kl}$  – *функцией фильтра*. Моделирование работы цифровых фильтров принципиально не отличается от описанной процедуры свертки.

В машинной графике свертка используется как средство улучшения синтезированных изображений для устранения искажений, вызванных аппроксимацией формы предметов или связанных с дискретизацией изображения. В качестве фильтров используются простое прямоугольное, пирамидальное, усеченное гауссовское и некоторые другие ядра свертки.

Специфической особенностью ряда оптико-электронных систем является *растровая структура* формируемого изображения. Она обусловлена структурой фотоприемных устройств, представляющих собой двухмерную матрицу (мозаику) светочувствительных элементов (рис.1.8.1). Такие фотоприемные

устройства усредняют освещенность первичного изображения в пределах каждой светочувствительной площадки. Величина сигнала с выхода каждого приемника  $I_{mn}$  определяется как интеграл по его площади:

$$I_{mn} = \int_{mA-a/2}^{mA+a/2} \int_{bN-b/2}^{bN+b/2} E'(x,y) dx dy, \quad (1.8.4)$$

где  $m, n$  – соответственно номер строки и столбца фотоприемной матрицы.

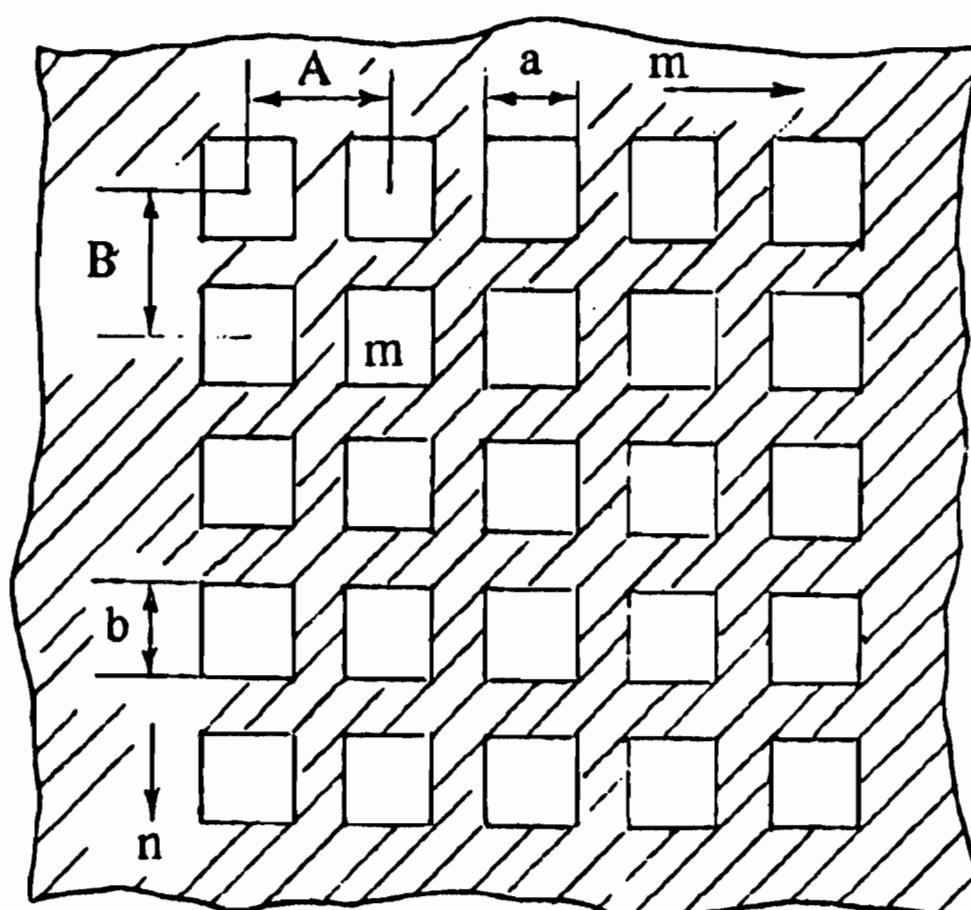


Рис. 1.8.1. Фотоприемная структура

Полученное значение  $I_{mn}$  характеризует сигнальную компоненту. Кроме сигнала в системе присутствуют шумы. Наличие шумов обусловлено случайной природой формирования отклика видеодатчиков (или фотопленки) на падающий лучистый поток. Шум в изображении представляет собой аддитивную компоненту. Следовательно, распределение яркости в зашумленном изображении  $I'_{mn}$  может быть промоделировано следующим образом:

$$I'_{mn} = I_{mn} + \hat{N}_{mn}, \quad (1.8.5)$$

где  $\hat{N}_{mn}$  – случайная величина.

Простейшие модели шума – это модели со статистически независимыми отсчетами  $\hat{N}_{mn}$ . Шумовые отсчеты  $\hat{N}_{mn}$  обычно генерируются в виде реализаций псевдослучайных последовательностей с заданными статистическими свойствами. Для получения требуемых последовательностей базовыми являются алгоритмы генерирования независимых псевдослучайных чисел с равномерным распределением. Основными требованиями к базовому алгоритму являются близость плотности распределения генерируемых чисел к равномерной и независимость отсчетов. Проверка закона распределения выполняется обычно по гистограмме, а для проверки независимости отсчетов можно использовать непосредственную систему машинной графики и свойство зрения обнаруживать на изображении регулярные структуры.

Последовательность псевдослучайных чисел преобразуют в формат экрана дисплея и выводят в виде изображения. Если при рассмотрении такого изображения на нем не обнаруживаются заметные структуры, то псевдослучайные числа вполне можно считать независимыми.

Если в моделируемом изображении отношение сигнала  $I_{mn}$  к среднеквадратической величине последовательности  $N_{mn}$  сравнительно большое (свыше 10), то генератор псевдослучайных чисел с равномерным законом распределения удовлетворительно воспроизводит шумы. Отклонения синтезированного изображения от реального не очень существенны. Если же отношение сигнал-шум невелико, необходимо более строгое моделирование. Очень часто для моделирования шумов генерируются в этих случаях псевдослучайные числа с гауссовским распределением вероятностей.

Для систем со строчной структурой (в частности, телевизионных) для отображения шумов используют одномерные модели с коррелированными отсчетами. Наличие корреляции шумов в случае телевизионной развертки обусловлено непрерывным движением считающего луча по строке, в результате чего шумовое воздействие с некоторой области распространяется на несколько соседних отсчетов. Аналогичная картина имеет место и в сканирующих оптико-электронных приборах.

Таким образом, все основные особенности реальных изображений могут быть воспроизведены системой машинной графики, хотя в отдельных случаях для учета специфики реальных систем требуется определенная изобретательность.

## Глава 2. ОПИСАНИЕ ГЕОМЕТРИЧЕСКИХ ФОРМ

### 2.1. ОПИСАНИЕ ПОВЕРХНОСТЕЙ

**Параметрическое описание поверхностей.** Поверхности, заданные в форме  $X = X(u,t)$ ,  $Y = Y(u,t)$ ,  $Z = Z(u,t)$ , где  $u,t$  – параметры, изменяющиеся в заданных пределах, относятся к классу параметрических. Для одной фиксированной пары значений  $u,t$  можно вычислить только положение одной точки поверхности. Для полного представления о всей поверхности необходимо с определенным шагом перебрать множество пар  $u,t$  из диапазона их изменений, вычисляя для каждой пары значение  $X,Y,Z$  в трехмерном пространстве. Любую поверхность, описанную неявно, можно представить и параметрически, но обратное правило в большинстве случаев не выполняется. Приведем несколько примеров параметрического описания.

Плоскость, проходящая через точку  $(X_0, Y_0, Z_0)$  и векторы  $n_1$  и  $n_2$ , исходящие из этой точки, определяется уравнением [23]  $\mathbf{r} = \mathbf{r}_0 + u \mathbf{n}_1 + t \mathbf{n}_2$  или в выражении для компонент  $X = X_0 + u n_{1X} + t n_{2X}$ ,  $Y = Y_0 + u n_{1Y} + t n_{2Y}$ ,  $Z = Z_0 + u n_{1Z} + t n_{2Z}$ , где  $\mathbf{r} = iX + jY + kZ$ ;  $\mathbf{r}_0 = iX_0 + jY_0 + kZ_0$ ;  $n_{mX}$ ,  $n_{mY}$ ,  $n_{mZ}$  – проекции вектора  $n_m$  ( $m = 1,2$ ) на оси  $OX$ ,  $OY$ ,  $OZ$  соответственно.

Приведенное уравнение опишет прямоугольник со сторонами, длины которых равны соответственно  $|n_1|$  и  $|n_2|$ , если единичные векторы  $n_1$  и  $n_2$  будут расположены перпендикулярно друг другу, а диапазон изменения  $u,t$  будет выбран от нуля до единицы.

Нормаль  $\mathbf{N}$  к плоскости, заданной в параметрическом виде, может быть определена как векторное произведение:  $\mathbf{N} = \mathbf{n}_1 \times \mathbf{n}_2$ . Вектор  $\mathbf{N}$  ориентирован так, чтобы тройка  $\mathbf{n}_1$ ,  $\mathbf{n}_2$ ,  $\mathbf{N}$  составляла правую систему осей.

Эллипсоид вида  $(x/a)^2 + (y/b)^2 + (z/c)^2 = 1$  параметрически представляется в форме  $X = a \cos \theta \cos \varphi$ ,  $Y = b \cos \theta \cos \varphi$ ,  $Z = c \sin \theta$ , где  $\theta$  – долгота,  $\varphi$  – широта.

Нормаль к поверхности эллипса определяется следующим образом:  $\mathbf{N} = i b c \cos \theta \cos \varphi + j c a \sin \theta \cos \varphi + k a b \sin \varphi$ .

В общем случае нормаль  $\mathbf{N} = i N_X + j N_Y + k N_Z$  к параметрически заданной поверхности в точке  $M(X_0, Y_0, Z_0)$  имеет вид [25,28,40]:

$$N_x = \begin{vmatrix} \frac{\delta Y}{\delta u}|_{(.)M} & \frac{\delta Z}{\delta u}|_{(.)M} \\ \frac{\delta Y}{\delta t}|_{(.)M} & \frac{\delta X}{\delta t}|_{(.)M} \end{vmatrix};$$

$$N_y = \begin{vmatrix} \frac{\delta Z}{\delta u}|_{(.)M} & \frac{\delta X}{\delta u}|_{(.)M} \\ \frac{\delta Z}{\delta t}|_{(.)M} & \frac{\delta Y}{\delta t}|_{(.)M} \end{vmatrix};$$

$$N_z = \begin{vmatrix} \frac{\delta X}{\delta u}|_{(.)M} & \frac{\delta Y}{\delta u}|_{(.)M} \\ \frac{\delta X}{\delta t}|_{(.)M} & \frac{\delta Y}{\delta t}|_{(.)M} \end{vmatrix}.$$

Особенно важными для практики являются параметрические бикубические поверхности, математические аспекты описания которых рассмотрены в § 3.4.4. Бикубические поверхности являются простейшими среди форм поверхностей, с помощью которых достигается непрерывность составной функции и ее первых производных. Другими словами, функция, составленная из нескольких смежных бикубических участков, будет обладать непрерывностью и гладкостью в местах стыка. Обычно бикубические участки — это гладкие изогнутые четырехугольники, представление о которых могут дать листы металла, бумаги и других материалов, обладающих упругостью. Описанию и изображению бикубических поверхностей посвящена обширная литература [54, 62, 63, 79, 99, 104, 146], что обусловлено свойствами этих поверхностей описывать любые геометрические формы. К недостаткам такой формы задания поверхностей следует отнести трудоемкость описания и большие вычислительные затраты. Последние определяются необходимостью численных, а не аналитических методов математических решений.

Основным преимуществом параметрического описания является возможность передачи геометрической формы очень сложных поверхностей, которые другими методами описать очень сложно. Например, даже такое сложное тело, как винтообразная улитка, может быть описано [88] посредством параметрического

представления суммы трех векторов: первого, вокруг которого завивается улитка; второго, конец которого очерчивает спираль, а начало скользит по первому; и третьего, начало которого скользит по спирали, а конец вращается вокруг спирали. Другим примером является описание тора [54], симметричного относительно оси OZ и плоскости XOY:

$$\begin{cases} X = (R + a \cos u) \cos \theta; \\ Y = (R + a \cos u) \sin \theta; \\ Z = a \sin u, \end{cases}$$

где  $a$  – радиус кольцевого “баллона” тора;  $R$  – расстояние от центра тора до оси “баллона”;  $u = [0, 2\pi]$ ;  $\theta = [0, 2\pi]$ .

Неявное описание типа  $f(X, Y, Z) = 0$  этих и многих других поверхностей невозможно.

Другое преимущество параметрического описания заключается в приспособленности к физическим процессам управления резцом в станках с числовым программным управлением. Резец в этом случае должен вытачивать деталь, двигаясь в пространстве по законам, заданным в параметрической модели [35]. Этот подход является единственным приемлемым для моделирования очень сложных гладких участков посредством аппроксимации сплайнами [49].

Однако параметрическое описание подразумевает, что исходной позицией светового луча, строящего изображение, является точка на объекте. Это затрудняет применение алгоритмов компьютерного синтеза изображений, предполагающих иную начальную позицию луча, например метода трассирования лучей. Это, в свою очередь, ведет к ограничениям по изобразительным свойствам: иллюстрации параметрических поверхностей, как правило, не имеют теней, не передают прозрачности и зеркального отражения соседних объектов [31,32].

Параметрические поверхности очень легко ограничиваются в пространстве путем задания пределов изменения параметров. Например, наружная поверхность дольки апельсина в виде 1/8 шара радиуса  $r$  задается в виде  $X = r \sin u \cos \theta$ ,  $Y = r \sin u \sin \theta$ ,  $Z = r \cos u$ , где  $\theta = [0, \pi/4]$ ;  $u = [-\pi/2, \pi/2]$ .

Описание ограничений в неявно заданных поверхностях достигается более сложным путем.

**Описание поверхности неявными функциями** заключается в моделировании поверхностей следующей математической формой:  $f(X, Y, Z) = 0$ , где  $X, Y, Z$  – координаты объектного пространства.

В качестве функции  $f$  могут быть функции различных порядков, однако из-за сложности математической обработки на практике обычно ограничиваются первой и второй степенью. Известны аналитические методы [16,23] решения уравнений третьей и четвертой степени, однако поверхности, описываемые функциями такой степени, незначительно расширяют возможности геометрической имитации форм, а вычислительные затраты резко возрастают. Поверхности первого порядка типа  $AX + BY + CZ + D = 0$ , где  $A, B, C, D$  – коэффициенты, представляют собой плоскости. Поверхности второго порядка типа  $AX^2 + BY^2 + CZ^2 + 2DXY + 2EYZ + 2FZX + 2GX + 2HY + 2JZ + K = 0$  в зависимости от значения коэффициентов  $A - K$  могут описывать две плоскости, конусы, гиперболоиды параболоиды и эллипсоиды. Неявная форма задания поверхности органично приспособлена для использования в методе твердотельного описания объектов и при трассировании лучей, так как существуют простые приемы определения взаимного положения точки и поверхности такого типа, определения точки пересечения прямой и поверхности. Математические свойства таких поверхностей рассмотрены в § 2.2.

**Поточечное описание** поверхностей заключается в представлении поверхности множеством отдельных точек, принадлежащих этой поверхности. Теоретически при бесконечном увеличении числа точек такая модель обеспечивает непрерывную форму описания. Точки, используемые для описания, должны располагаться достаточно часто, чтобы можно было воспринять поверхность без грубых потерь и искажения информации. Основной особенностью такого описания в отличие от других подходов является отсутствие информации о поверхности между точками. Например, при задании полигональных поверхностей [60] вершины каждого плоского многоугольника, а следовательно, и вся модель хотя и могут быть описаны точками, но предполагается, что между точками располагаются участки плоскостей. Поточечное описание поверхностей применяют в тех случаях, когда поверхность очень сложна, не обладает гладкостью, а детальное представление многочисленных геометрических особенностей важно для практики. К поверхностям такого типа можно отнести участки грунта на других планетах, формы малых небесных тел, информация о которых доставлена с искусственного спутника в виде нескольких стереопар, микрообъекты, снятые с помощью электронных микроскопов, и другие образования со сложной причудливой формой.

Практическая цель обработки таких поверхностей методами машинной графики заключается, как правило, в построении разноракурсных изображений с целью наилучшего восприятия формы, в том числе в создании мультиликационных фильмов на основе синтезированных изображений.

Исходная информации о поточечно описанных объектах представляется в виде матрицы трехмерных координат точек (см. § 4.1.5). Определение этих координат производят автоматизированными методами обработки стереопар. В простейшем случае координаты точки определяют на стереоприборе путем визуального отождествления положений соответствующих точек [30], однако трудозатраты на обработку будут исключительно велики. Известны реально работающие системы, которые автоматически распознают соответствующие точки на кадрах стереопары и известными расчетными методами устанавливают трехмерные координаты каждой точки в относительной или абсолютной объектной системе координат. Процесс “снятия” координат должен учитывать требования достаточной частоты расположения точек в пространстве. Основная идея, определяющая требования к выбору пространственной частоты точек, заключается в том, чтобы достичь состояния, когда две соседние точки на поверхности объекта спроектировались максимум в два соседних рецептора на экране. Алгоритм изображения поточечно описанных объектов подробно изложен в § 4.1.5. Описываемый подход применяется на практике достаточно редко, что связано с трудоемкостью снятия данных об объекте, большими вычислительными затратами и значительным объемом исходных данных на описание объекта.

## 2.2. ОБЩИЕ ХАРАКТЕРИСТИКИ ОСНОВНЫХ ТИПОВ ПОВЕРХНОСТЕЙ

**Поверхности первого порядка.** В системе координат XYZ объекта любой его отдельный плоский участок лежит в плоскости, описываемой уравнением  $f_1(X, Y, Z) = AX + BY + CX + D = 0$  или в матричном виде  $[X \ Y \ Z \ 1] P = 0$ , где  $P = [A \ B \ C \ D]^T$ .

Изменив компоненты матрицы  $P$ , можно описать плоскость любой ориентации и положения, однако каждый раз это будет описание бесконечно простирающейся плоскости. Реальный плоский участок ограничен плоской кривой или ломаной линией. Наиболее удобен для рассмотрения случай ограничения

выпуклым многоугольником. Все другие случаи, как криволинейного ограничения, так и невыпуклой фигуры, могут быть сведены к указанному выше путем аппроксимации или разбиения на выпуклые подфигуры. Примером плоского примитива является крыло самолета. Границные точки многоугольника однозначно (по или против часовой стрелки) оцифровывают и записывают их координаты в матрицу  $\Lambda$ :

$$\Lambda = \begin{bmatrix} X_1 & Y_1 & Z_1 \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ X_N & Y_N & Z_N \end{bmatrix},$$

где  $N$  — число граничных точек,  $N \geq 3$ .

Уравнение плоскости определяют на основании координат трех точек:  $T_1 (X_1, Y_1, Z_1)$ ,  $T_2 (X_2, Y_2, Z_2)$ ,  $T_3 (X_3, Y_3, Z_3)$ , заведомо не лежащих на одной прямой [11]:

$$\begin{vmatrix} X - X_1 & Y - Y_1 & Z - Z_1 \\ X_2 - X_1 & Y_2 - Y_1 & Z_2 - Z_1 \\ X_3 - X_1 & Y_3 - Y_1 & Z_3 - Z_1 \end{vmatrix} = 0.$$

Программа PL3T (см. приложение) представляет реализацию этого решения на языке Турбо Паскаль.

Нормальный вектор  $\mathbf{N}$  плоскости определяется следующим образом:  $\mathbf{N} = i\mathbf{A} + j\mathbf{B} + k\mathbf{C}$ .

Направлен этот вектор в сторону полупространства, где значения скалярного поля  $f_1(X, Y, Z)$  положительны. Из поверхностей первого порядка можно составить описание поверхности объекта типа *полигонального поля* [41,60]. Таким полем называют серию смежных многоугольников, не имеющих разрывов между собой. Каждое ребро поля является общим для смежных многоугольников. Таким образом, составная функция, описывающая поверхность, обладает непрерывностью, а производная имеет разрывы в местах стыка участков плоскостей. Простота обработки плоскостей заслуженно привлекла внимание многих исследователей и разработчиков коммерческих профессиональных программных

продуктов. В [34,49,67,90] описаны алгоритмы построения и изображения полигональных полей.

**Поверхности второго порядка.** Из аналитической геометрии [11,39] известно, что функция вида

$$f_2(X, Y, Z) = AX^2 + BY^2 + CZ^2 + 2DXY + 2EYZ + 2FZX + 2GX + 2HY + 2JZ + K = 0 \quad (2.2.1)$$

в зависимости от выбора коэффициентов A, B, ..., K может описывать поверхности эллипсоида, гиперболоида, конуса, параболоида, цилиндра или двух плоскостей. Все поверхности второго порядка, за исключением эллипсоида, не локализованы в пространстве и простираются в бесконечность. Поэтому только эллипсоид может самостоятельно образовывать объемный примитив, все другие квадратичные формы требуют пространственного ограничения линией или другими поверхностями.

Наиболее удобно с вычислительной точки зрения представлять квадратичную функцию (2.2.1) в матричном виде:

$$f_2(X, Y, Z) = [X \ Y \ Z \ 1] P [X \ Y \ Z \ 1]^T, \quad (2.2.2)$$

где

$$P = \begin{bmatrix} A & D & F & G \\ D & B & E & H \\ F & E & C & J \\ G & H & J & K \end{bmatrix}. \quad (2.2.3)$$

Нормальный вектор к квадратичной поверхности в точке (X, Y, Z) определяется по правилу

$$\mathbf{N} = \Omega \Psi, \quad (2.2.4)$$

где

$$\Psi = \begin{bmatrix} AX + DY + EZ + G \\ BY + DX + EZ + H \\ CZ + FX + EY + J \end{bmatrix};$$

$$\Omega = [i \ j \ k],$$

а i, j, k – орты осей OX, OY, OZ. Направлен такой вектор по градиенту скалярного поля  $f_2(X, Y, Z)$ , т. е. в ту сторону, где наблюдается возрастание значений  $f_2(X, Y, Z)$ . Так как функция  $f_2(X, Y, Z)$  является монотонной и однократно знакопеременной, то нормаль  $\mathbf{N}$  направлена в ту часть подпространства, где значения

функции положительны. Например, нормаль к поверхности шара, заданного уравнением  $X^2 + Y^2 + Z^2 - 1 = 0$ , нацелена внутрь шара, а нормаль к поверхности того же шара, но заданного в виде  $-X^2 - Y^2 - Z^2 + 1 = 0$ , уже направлена наружу шара.

Явное задание квадратичной поверхности применяют в машинной графике в методе обратного трассирования лучей, при прямом трассировании используют параметрическую форму типа  $X = X(u,v); Y = Y(u,v); Z = Z(u,v)$ . Каждая квадратичная поверхность описывается своей системой уравнений. Приведем примеры для эллипсоида:  $X = a \sin u \cos v; Y = b \sin u \sin v; Z = c \cos u$  и эллиптического параболоида:  $X = \sqrt{p} u \cos v; Y = \sqrt{q} u \sin v; Z = 0,5u^2$ , где  $a, b, c$  – размеры главных полуосей эллипсоида;  $p, q$  – коэффициенты сжатия параболоида вдоль осей  $OX, OY$  соответственно.

В машинной графике часто используется операция пространственного переноса предварительно заданной квадратичной поверхности. Известны [19] способы определения коэффициентов уравнения квадратичной поверхности (2.2.2) при изменении системы координат. Так, если в системе координат  $X' Y' Z'$  квадратичная поверхность задается матрицей  $P'$ , а для перевода некоторой точки  $(X', Y', Z')$  из этой системы в другую –  $X Y Z$  необходимо применить преобразование  $F$ :

$$[X \ Y \ Z \ 1] = [X' \ Y' \ Z' \ 1] F,$$

где  $F$  – матрица размера  $4 \times 4$  элемента (см. §3.4.1 и [19]), то новая матрица  $P$  квадратичной поверхности в новой системе координат  $X Y Z$  будет определяться по правилу

$$P = F^{-1} P' (F^{-1})^T. \quad (2.2.5.)$$

Структура матриц  $P$  и  $P'$  приведена в (2.2.3.). Подобным образом осуществляется переход в новую систему координат параметрически заданной поверхности. Если в системе координат  $X' Y' Z'$  квадратичная поверхность задается в виде  $X' = X'(u, v); Y' = Y'(u, v), Z' = Z'(u, v)$ , а новая система  $X Y Z$  связана со старой зависимостью  $[X \ Y \ Z \ 1] = [X' \ Y' \ Z' \ 1] F$ , то новые параметрические описания  $X(u, v), Y(u, v), Z(u, v)$  могут быть получены следующей операцией:

$$[X(u, v) \ Y(u, v) \ Z(u, v) \ 1] = [X'(u, v) \ Y'(u, v) \ Z'(u, v) \ 1] F.$$

Приведем пример преобразования функции шара  $f(X, Y, Z) = X'^2 + Y'^2 + Z'^2 - 1 = 0$  из системы координат  $X'Y'Z'$  в систему  $XYZ$ ,

которая образована путем сдвига  $X'Y'Z'$  на пять единиц по каждой из координатных осей.

В матричном виде шар описывается формой

$$[X' \ Y' \ Z' \ 1] P' [X' \ Y' \ Z' \ 1]^T = 0,$$

где

$$P' = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

В соответствии с правилами построения матрицы преобразования координат  $F$  (см. § 3.4.1) последняя имеет вид

$$F = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -5 & -5 & -5 & -1 \end{bmatrix}.$$

Новая матрица  $P$ , описывающая шар в системе  $XYZ$ , определяется правилом

$$\begin{aligned} P &= F^{-1} P' (F^{-1})^T = \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 5 & 5 & 5 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 5 \\ 5 & 5 & 5 & 74 \end{bmatrix}. \end{aligned}$$

Проведя операции умножения матриц, можно получить новую форму описания шара в системе  $XYZ$  в каноническом виде:  $f(X, Y, Z) = [X \ Y \ Z \ 1] P [X \ Y \ Z \ 1]^T = (x+5)^2 + (y+5)^2 + (z+5)^2 - 1 = 0$ .

Заметим, что при преобразовании координат по правилу (2.2.4) нормальный вектор сохраняет свою исходную ориентацию, что важно при вычислении освещенности на поверхности квадратичной функции. Например, если квадратичная функция задана в системе  $X'Y'Z'$  и нормальный вектор из точки поверхности направлен в сторону выпуклости поверхности, то при переносе и вращении поверхности по правилу (2.2.4) новый нормальный вектор из этой точки также направлен в сторону выпуклости. Докажем этот факт.

Действительно, известно, что если  $(X_A', Y_A', Z_A')$  – координаты точки  $A$  в системе  $X'Y'Z'$ , а системы  $XYZ$  и  $X'Y'Z'$  связаны между собой матрицей преобразования  $F'$ :

$$[X \ Y \ Z \ 1] = [X' \ Y' \ Z' \ 1] F',$$

$$\text{то } [X_A \ Y_A \ Z_A \ 1] = [X'_A \ Y'_A \ Z'_A \ 1] F'.$$

Кроме того, если квадратичная поверхность в системе  $X Y Z$  описывается матрицей  $P$ , а в системе  $X' Y' Z'$  – матрицей  $\rho$ , то  $P = (F')^{-1} \rho ((F')^{-1})^T$ . Тогда справедливы следующие выражения:

$$\begin{aligned} f_2(X_A, Y_A, Z_A) &= [X_A \ Y_A \ Z_A \ 1] P [X_A \ Y_A \ Z_A \ 1]^T = \\ &= [X'_A \ Y'_A \ Z'_A \ 1] F' (F')^{-1} \rho ((F')^{-1})^T ([X'_A \ Y'_A \ Z'_A \ 1])^T = \\ &= [X'_A \ Y'_A \ Z'_A \ 1] E \rho (F'(F')^{-1})^T [X'_A \ Y'_A \ Z'_A \ 1]^T = \\ &= [X_A \ Y_A \ Z_A \ 1] E \rho E^T [X'_A \ Y' \ Z'_A \ 1]^T = \\ &= [X' \ Y' \ Z' \ 1] \rho [X' \ Y' \ Z' \ 1]^T = f_2(X'_A, Y'_A, Z'_A), \end{aligned}$$

где  $E$  – единичная матрица  $4 \times 4$  элемента.

Значение функции в одной и той же точке пространства инвариантно к смене системы координат, в которой эта функция задана. Пользуясь принципом относительности, содержание этой фразы можно интерпретировать так: значение функции  $f(X, Y, Z)$ , описывающей поверхность уравнением  $f(X, Y, Z) = 0$  в точке, перемещающейся по единому закону с поверхностью, есть константа. Эти положения объясняют и инвариантность ориентации нормали к поверхности при перемещении самой поверхности.

**Поверхности типа экструзий.** Название этих поверхностей происходит от английского слова *extrusion*, означающего выталкивание, выдавливание. С такими поверхностями встречаются довольно часто: это и металлические профили, выдавленные из расплава, и керамические пустотелые кирпичи, выдавленные из глины. К экструзиям относят и поверхности вращения, которые вырезаны резцом из заготовки.

Достаточно широкий класс машиностроительных деталей, предметов быта, архитектурных форм может быть представлен как результат вращения кривой или ломаной линии относительно некоторой оси. Обычно поверхности такого типа описывают в виде усеченных конусов, примыкающих друг к другу торцами (рис. 2.2.1, а, б).

Кривую линию, являющуюся линией вращения фигуры, аппроксимируют ломаной линией. Каждый отдельный участок последней становится образующей отдельного конуса. Таким образом, если кривая аппроксимируется ломаной из ста отрезков, то вся фигура представляется из ста попарно смежных конусов. Описание конуса может быть как неявным, так и параметрическим в зависимости от алгоритма синтеза изображения.

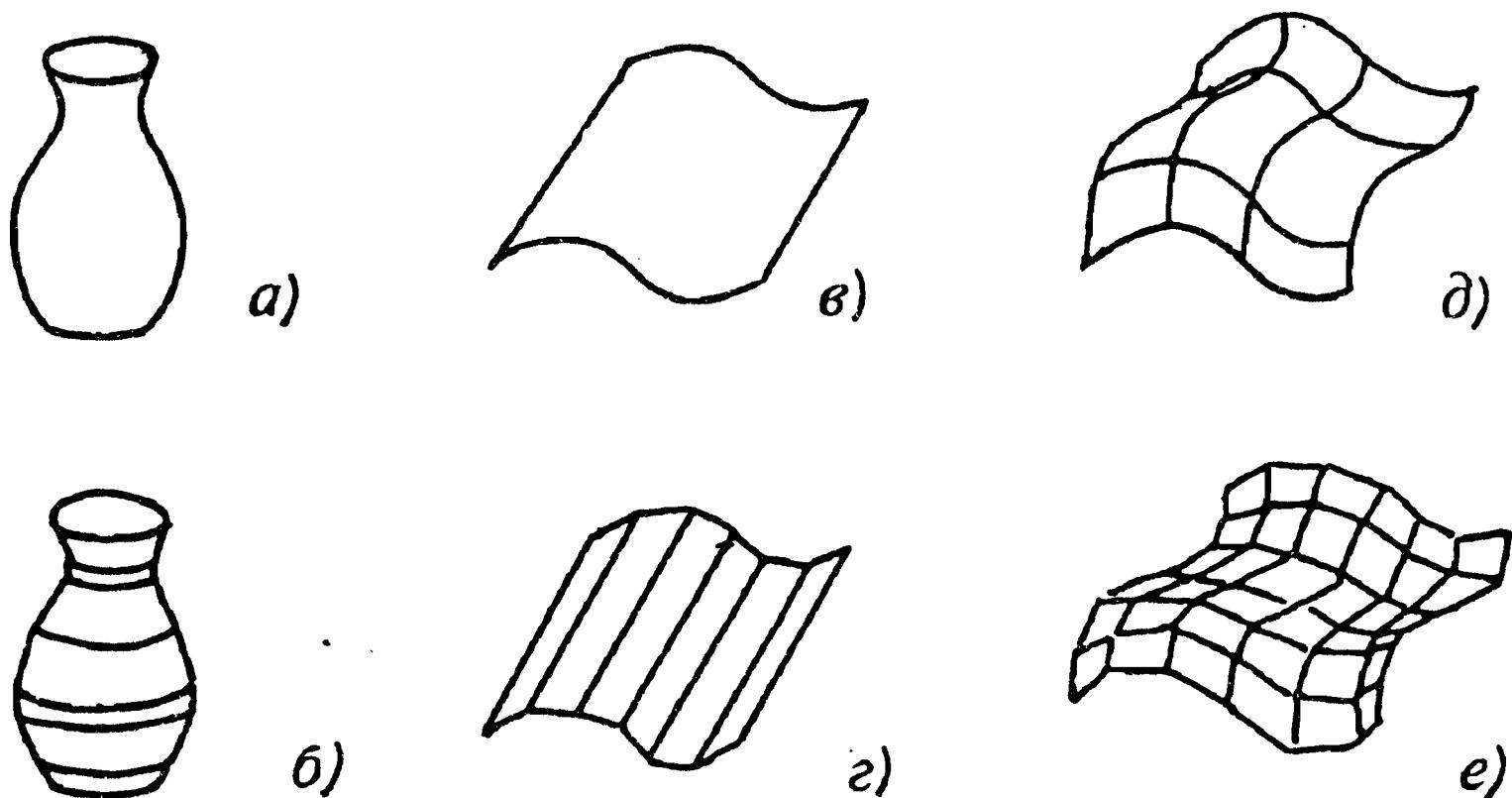


Рис. 2.2.1. Поверхности типа экструзий

Другими представителями поверхностей-экструзий являются поверхности, образованные путем параллельного переноса кривой линии вдоль некоторой прямой. Как и в других случаях, кривую аппроксимируют ломаной линией, а всю поверхность фигуры представляют множеством смежных четырехугольников. Две стороны каждого четырехугольника параллельны направляющей прямой, а две остальные параллельны соответствующему отрезку ломаной (рис.2.2.1,в,г). Если в качестве направляющей используется кривая линия, то она, в свою очередь, также аппроксимируется ломаной. В пределах каждого отрезка направляющей ломаной линии поверхность представляется лентой смежных четырехугольников, которая сопрягается с соседней лентой, построенной в направлении следующего отрезка направляющей ломаной (рис.2.2.1,д,е).

**Фрактальные поверхности.** Естественные сцены часто не могут быть описаны на основе традиционных приемов, базирующихся на использовании непрерывных функций. Однако можно заметить, что большинство природных сцен статистически родственны. В [89] приводится анализ нерегулярности изображений береговой линии, полученных при съемке из космоса с высот 10; 3 км и, наконец, при наблюдении с уровня поверхности Земли. При этом установлено, что при любом уровне разрешения береговая линия может быть смоделирована

и нарисована посредством объединения участков небольших прямолинейных сегментов. Причем при переходе на каждый следующий, более высокий уровень разрешения, который был предварительно аппроксимирован одним прямолинейным сегментом, этот сегмент теперь должен быть вероятностным образом разбит на последовательность линейных сегментов, и так далее и до бесконечности. На основании этого свойства — постоянства статистического закона порождения деталей природных образований при переходе от низких к более высоким уровням разрешения — построен метод использования фрактальных поверхностей.

В переводе с английского "фрактальный" означает состоящий из частиц, частей. Такими поверхностями называют класс нерегулярных геометрических форм, задаваемых вероятностным образом на основе исходного описания низкого разрешения. Случайный закон, с помощью которого исходная линия или поверхность дробится на несколько более мелких линий или поверхностей, подбирается опытным путем по критерию визуального согласования синтезированного изображения с реальной моделируемой сценой [43,80]. Наиболее часто фрактальные поверхности используются для моделирования горного ландшафта. Предварительно горный массив описывают очень приближенно полигональным полем из четырехугольников. Каждый четырехугольник разбивают с помощью случайной функции на четыре фигуры меньших размеров, причем эти фигуры вероятностным образом сдвигают относительно плоскости исходного четырехугольника, сохраняя для каждой фигуры по одной общей вершине с исходным четырехугольником. Каждую фигуру вновь делят, и так далее до достижения желаемого уровня изрезанности поверхности. Построение изображения осуществляется путем удаления скрытых поверхностей и закраски множества сгенерированных четырехугольников. Изображения, созданные на основе фрактальных поверхностей, только статистически идентичны реальным объектам, поэтому от них нельзя требовать идеальной точности.

### 2.3. МОДЕЛИ ОБЪЕКТОВ И ИХ КЛАССИФИКАЦИЯ

В практической деятельности получили распространение три основных типа моделей объектов [53,76,102,105,121,143] : описание объекта поверхностями, сплошными телами и описание

типа проволочной сетки. Первый подход представляет объект в виде тонких поверхностей, под которыми находится пустое пространство, не заполненное материалом объекта. Эллипсоид, рассматриваемый в рамках поверхностного описания, следует ассоциировать с неразбитой скорлупой совершенно пустого внутри яйца. Описания подобного типа встречаются в машинной графике достаточно часто, именно в рамках этого метода конструируют полигонные поля и бикубические участки [83—86]. Описание сплошными телами подразумевает, что объекту илициальному примитиву принадлежат все точки объекта или примитива — как наружные, так и внутренние. В данном случае эллипсоид следует понимать как однородно заполненное яйцо. Описание типа проволочной сетки заключается в представлении поверхности серией пересекающихся линий, принадлежащих поверхности объекта [27,137].

Существующие системы описания сплошными телами [83—86,123,125] могут быть разделены на три класса. Во-первых, существуют системы, которые основываются на пространственной декомпозиции объема, содержащего моделируемый объект, на массив участков или трехмерных ячеек, которые могут быть заполнены телом объекта. Такие методы представления объектов называют *ячеичными* или системами *пространственного заполнения*. Во-вторых, существуют системы, в которых объект представляют как комбинацию простых форм примитивов, в качестве которых могут быть использованы, например, кубы, шары, цилиндры. Сам объект с точки зрения геометрической конструкции может быть представлен древовидной структурой, терминальными вершинами которой являются примитивы, а нетерминальными — правила, указывающие, каким образом должны комбинироваться примитивы. Сама модель эквивалентна инструкции [77,87] следующего типа: "Взять шар", "Вычесть из него цилиндрический примитив", "Перенести результат в новую точку пространства", ... . Такие модели называют моделями *геометрических сплошных конструктивов*. Наконец, существуют подходы, в которых объект задается только границами. В этой системе объект следует описывать состоящим из поверхностей такого-то конкретного вида, которые ограничены краями, часть краев является общей для нескольких поверхностей, края пересекаются в таких-то точках.

**Ячеичные методы.** Идеи этого метода очень просты.

Ограниченный участок пространства, охватывающий весь моделируемый объект, считается разбитым на большое число дискретных кубических ячеек. В простейшем случае размеры ребра куба равны единице измерения длины. Моделирующая система должна просто записать информацию о принадлежности или непринадлежности каждого куба телу объекта. Структура данных представляется трехмерной матрицей, в которой каждый элемент соответствует пространственной ячейке. С одной стороны, ячеичный метод имеет преимущества, определенные простотой, с другой – недостатки, связанные с большим объемом памяти, требуемой для записи объекта с высоким разрешением.

Для преодоления этого недостатка разработаны системы, которые используют идею разбиения ячеек на подъячейки меньшего размера. Последние применяются в тех случаях, когда ячейка захватывает границу объекта, и здесь в целях повышения разрешения действуются подъячейки, регулярно заполняющие ячейку-границу.

**Моделирование сплошными геометрическими конструктивами.** Такой метод называют еще твердотельным моделированием (*constructive solid geometry*). Он представляет сложные объекты составленными из простых объемных примитивов. Обычно эти примитивы включают кубы, цилиндры, конусы, эллипсоиды и другие подобные фигуры (см. § 3.2.1). Булевы операции над примитивами позволяют достигать объединения, вычитания и выделения общей части примитивов [33]. Структура данных модели этого вида идентична бинарному дереву (рис. 3.2.9). Заметим, что построение и обработка дерева осуществляются снизу вверх. Узлы (нетерминальные вершины) дерева являются операторами над примитивами, а листья – примитивами. Структура описанного вида может быть усложнена внесением операторов переноса, поворота, масштабного преобразования.

В большинстве систем поверхностного моделирования с примитивами тесно связано понятие полупространства. Любая поверхность, которая подразделяет трехмерное пространство на два несовпадающих региона, определяет два полупространства [95]. Например, плоскость делит пространство на два полупространства с каждой стороны от плоскости. Полупространство может иметь конечный объем. Например, такой объем локализован внутри поверхности эллипсоида. Заметим, что не все функции образуют полупространства, выделяемые

формальными методами. Например, функция  $x$  посредством уравнения  $x=0$  создает два полупространства, в одном из которых  $x > 0$ , в другом  $x < 0$ , а функция  $|x|$  этим свойством не обладает, т.е. в последнем случае формально различить между собой два полупространства невозможно:  $x = 0$  и  $|x| = 0$  описывают одну и ту же плоскость.

Подробно вычислительные аспекты твердотельного представления объектов описаны в § 3.2. Можно выделить [125] следующие преимущества этого подхода: концептуальная простота, малый объем требуемой памяти, принципиальная застрахованность от создания противоречивых конструкций, приспособленность к усложнению модели, относительная простота представления частей и сечений общей сцены. К недостаткам обычно относят следующие: метод построения объекта ограничен рамками булевых операций, метод требует вычислительноемких алгоритмов обработки, невозможность использования параметрически описанных поверхностей, которые не позволяют устанавливать положение полупространства, сложности создания и обработки объектов, поверхность которых описана функциями более чем второй степени.

**Представление объекта границами.** При этом подходе в памяти машины сохраняют все те элементы, которые создают границы объекта. Такими элементами являются поверхности, края поверхностей и указатели пересечений поверхностей. Одновременно хранят так называемую топологическую информацию, которая показывает, как элементы связаны друг с другом. Про поверхности края и пересечения можно сказать, что они являются топологической основой конструкции объекта.

При моделировании объекта границами структура данных строится одновременно с процессом создания модели. Это означает, что "проводочное" изображение может быть построено достаточно просто и быстро на любой стадии обработки объекта, так как задание краев присутствует в структуре данных явным образом, а для изображения уже имеющихся краев необходимо использовать соответствующий алгоритм их проецирования на экран. Правила определения краев для тел, составленных из поверхностей первого и второго порядка, приведены в [75]. Модели описания краями допускают использование булевых операций и операций над множествами, но они становятся вычислительно сложными, так как любая булева операция

приводит к появлению новых краев и новых фактов пересечений, которые, в свою очередь, должны быть определены.

Как было показано ранее, для твердотельной модели булевы операции над примитивами гарантируют успешное построение модели. В случае представления границами операторы обеспечивают гарантию генерации правильной модели. Закон Euler [125] говорит о связи между числом поверхностей  $\Pi$ , краев  $K$  и числом фактов пересечений  $\Phi$ :  $\Pi - K + \Phi = 2$ . Например, если конус и плоскость ( $\Pi=2$ ) пересекаются дважды ( $\Phi=2$ ), то число краев (линий пересечения) равно двум. Если же эти поверхности пересекаются один раз, то число линий пересечения равно единице.

Преимущества описания границами заключаются в следующем: большие возможности геометрического моделирования форм, чем при описании сплошными телами; быстрый и эффективный доступ к геометрической информации, которая требуется для выполнения прорисовки или в других прикладных целях; относительно простое создание геометрических поверхностей свободных форм.

К недостаткам этого подхода можно отнести: больший объем исходных данных, чем при твердотельном описании; модель логически менее устойчива, чем твердотельная, т. е. возможно создание противоречивых конструкций; сложности построения вариаций форм.

В заключение этого раздела приведем структуру моделей, описывающих геометрию объектов (рис. 2.3.1).

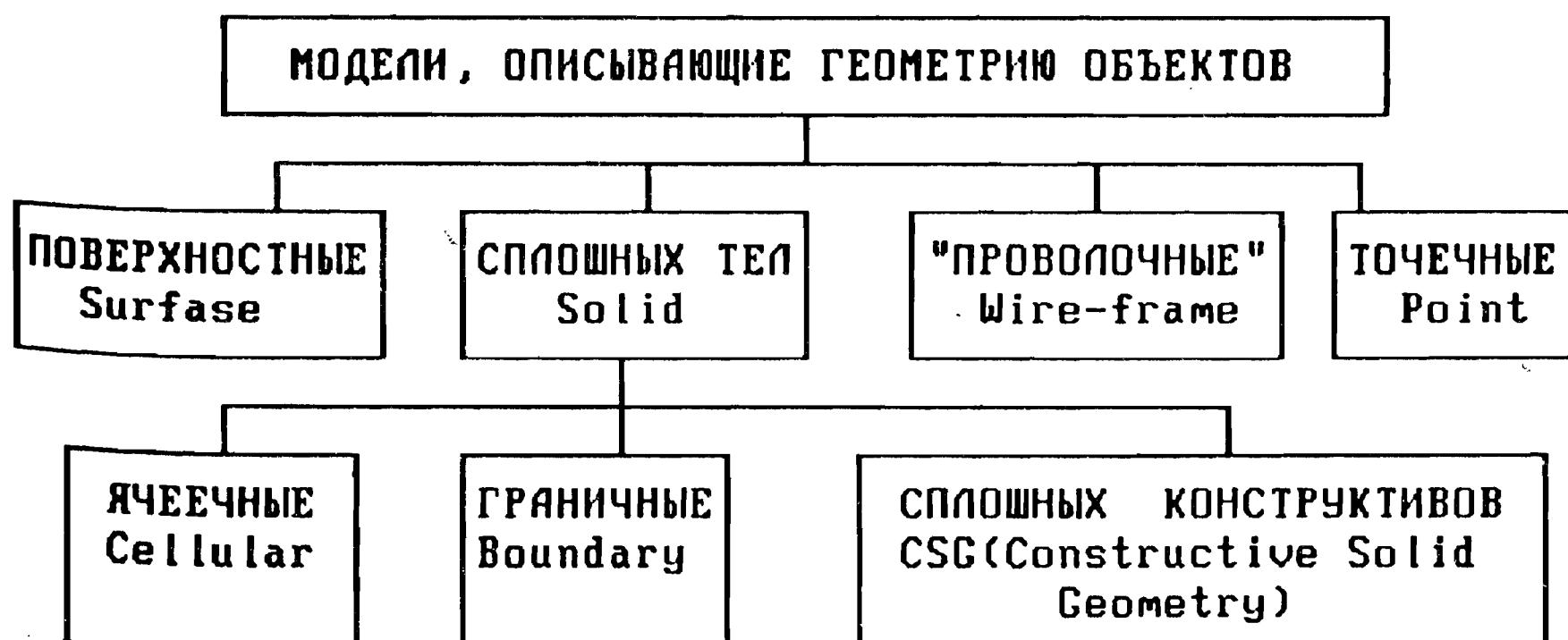


Рис. 2.3.1. Классификация моделей, описывающих геометрию объектов

## Глава 3. СИНТЕЗ ИЗОБРАЖЕНИЙ МЕТОДОМ ОБРАТНОГО ТРАССИРОВАНИЯ ЛУЧЕЙ

**Общие положения.** Название “трассирование лучей” происходит от слова “путь” (трасса), что связано с моделированием геометрического пути каждого светового луча, участвующего в построении изображения. Процесс моделирования основывается на законах геометрической оптики, таких как законы преломления, отражения, прямолинейности, обратимости хода световых лучей и др.

Заметим, что световой луч характеризует направление распространения энергии в пространстве в предположении бесконечно малой длины оптического излучения. Это предположение позволяет абстрагироваться от волновой природы оптического излучения и формулировать оптические законы на языке геометрии [47].

Потому можно сказать, что лучевая оптика — это модель распространения света в оптических приборах и средах, а, в свою очередь, трассирование лучей — модель лучевой оптики применительно к задачам компьютерной графики.

Основная идея метода сводится к повторению на ЭВМ всех геометрических преобразований, которые бы совершил каждый световой луч на пути источник—объект—приемник. Хотя действительных лучей бесконечно много, для построения изображения достаточно ограничиться рассмотрением хода лучей, попадающих в центры рецепторов или исходящих из ограниченного числа точек на изображаемой поверхности.

При компьютерном моделировании, как и в ряде разделов геометрической оптики, реальный ход лучей в объективах не анализируют, а используют для построения изображения так называемые кардинальные элементы оптической системы (главные и фокальные точки, а также соответствующие плоскости). В соответствии с принципами геометрической оптики сопряженные точки в пространстве предметов и изображений лежат на прямой, проходящей через заднюю главную точку оптической системы. На основании закона обратимости [7,8] хода световых лучей можно синтезировать путь луча как в направлении объект—изображение, так и в обратном. Поэтому можно различать два способа трассирования лучей: прямое и обратное. При прямом трассировании за исходную позицию берут вычисляемую на изображаемой поверхности точку 1 (рис.3.1.1,а) и моделируют путь

луча из нее как на источник света 2, так и на приемник изображения 3. При обратном трассировании за исходную позицию берут центр рецептора 1 (рис.3.1.1,б) на приемнике изображения и моделируют путь луча из него на объект 2 и далее от объекта на источник света 3 [65,91,145].

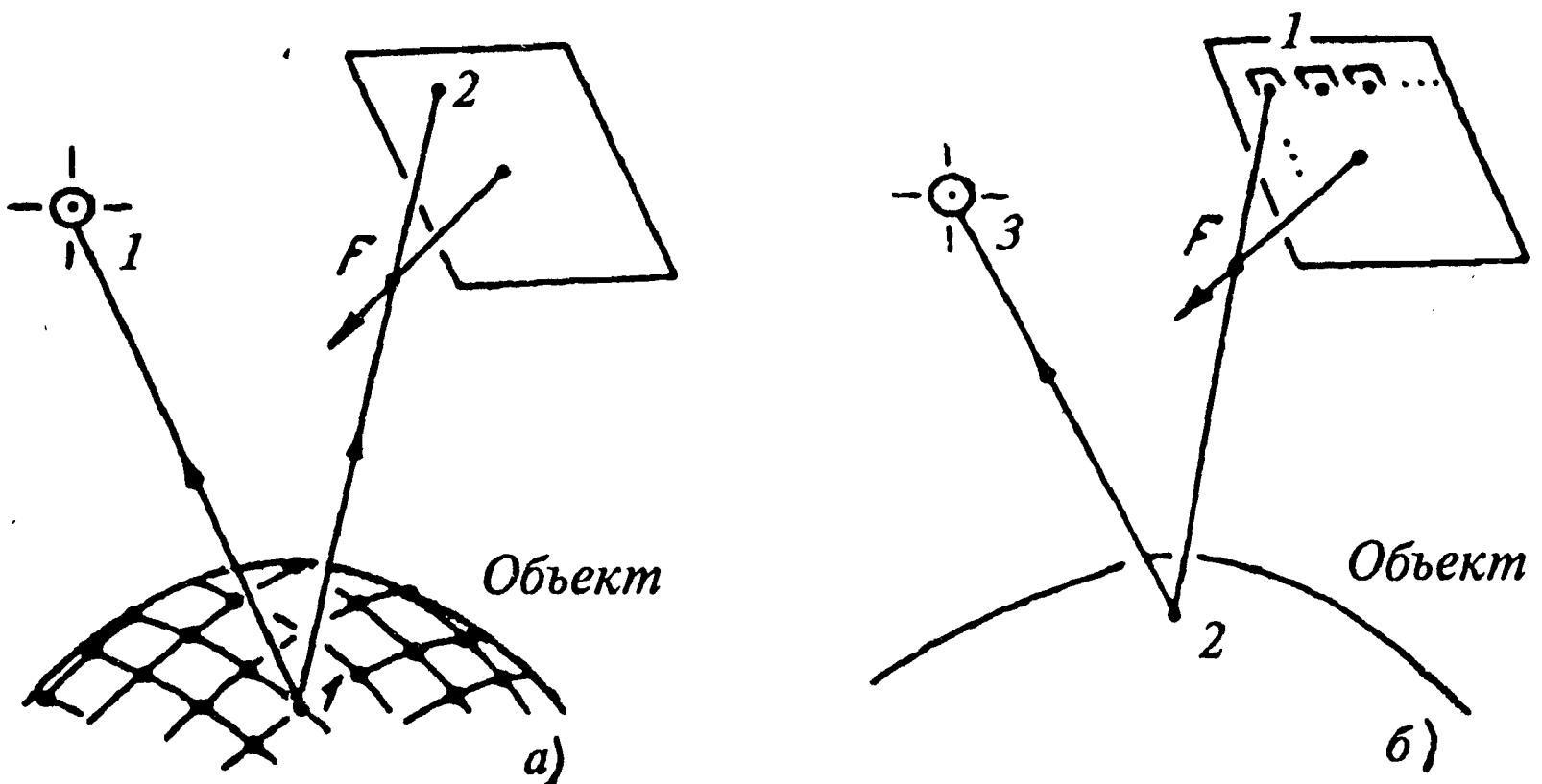


Рис. 3.1.1. Ход лучей при обратном (а) и прямом (б) трассировании лучей

Опишем системы координат, в которых решается задача синтезирования изображения при трассировании лучей. Будем называть *сценой* совокупность изображаемых объектов, включая при необходимости поверхность основания. Правая прямоугольная система координат, общая для всей сцены, соответственно будет называться системой координат сцены  $X_C Y_C Z_C$  (рис.3.1.2). *Объектом* будем называть совокупность точек пространства, объединенных функциональной общностью с точки зрения конкретной целевой задачи. Соответственно для каждого объекта введем правую прямоугольную систему координат  $XYZ$ . Систему координат  $xyz$  наблюдательной системы будем называть *экранной*. В отличие от других ее обычно выбирают левой прямоугольной, что объясняется исторически сложившейся традицией [41]: оси  $x$  и  $y$  располагают соответственно вправо и вверх при рассматривании изображения, а ось  $z$  – из центра экранной системы в сторону объекта (рис.3.1.2,3.1.3). Проведя аналогию с физическими устройствами, можно сравнить ось  $z$  с

главным лучом объектива, плоскость  $xy$  — с задней фокальной плоскостью, а центр проекции  $F$ , который располагают на оси  $oz$  в точке  $(0,0,f)$ , с задней главной точкой объектива.

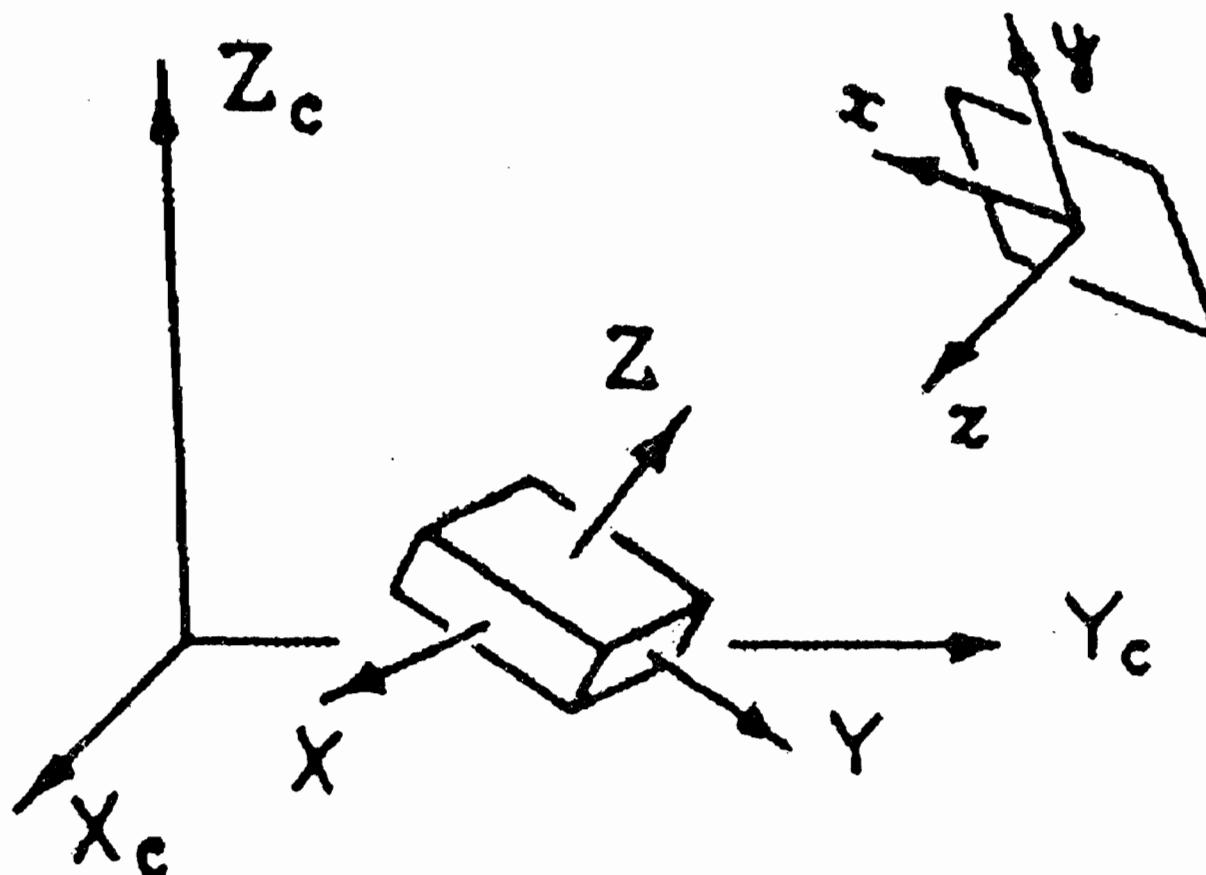


Рис. 3.1.2. Системы координат сцены  $X_C Y_C Z_C$ , объекта  $XYZ$  и экрана  $xuz$

Метод трассирования лучей получил достаточно широкое распространение на практике, его вычислительным и алгоритмическим особенностям посвящены многие исследования [62, 115, 131, 145], что связано с относительной простотой и универсальностью метода и возможностью его реализации на многопроцессорных вычислительных устройствах.

### 3.1. МОДЕЛЬ ПРИЕМНИКА СВЕТА И ЕГО РАЗМЕЩЕНИЕ В ПРОСТРАНСТВЕ

Поскольку исходной позицией трассирования лучей является точка центра рецептора, то обрабатывающий алгоритм должен начинать расчеты с определения пространственного положения любого рецептора. В плоскости  $xy$  экранной системы симметрично относительно осей расположим матрицу  $I \times J$  точечных приемников (рис. 3.1.3, а). Расстояния между центрами соседних рецепторов

обозначим  $c'$  и  $d'$  вдоль осей  $x$  и  $y$  соответственно. Каждый  $j$ -й рецептор ( $i=1(1)I$ ,  $j=1(1)J$ ) имеет координаты  $(x_{ij}, y_{ij}, 0)$  в экранной системе. Эти координаты вычисляют по номеру рецептора:  $x_{ij} = -c'(j-J/2-1/2)$ ;  $y_{ij} = d'(I/2-i+1/2)$ . При симметричном расположении рецепторов относительно осей  $ox$  и  $oy$  последние выражения справедливы как для четного (рис.3.1.3,б), так и для нечетного (рис.3.1.3,в) числа строк и столбцов.

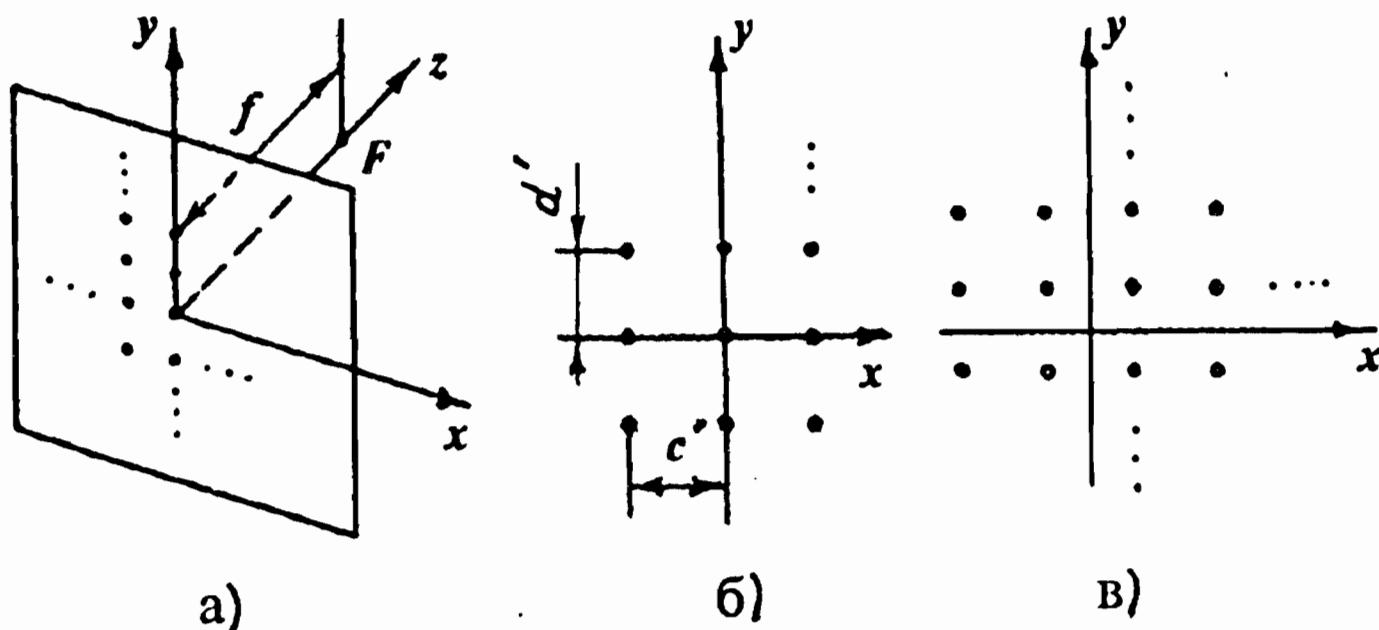


Рис. 3.1.3. Расположение рецепторов и центра проекции –  $F$  в экранной системе координат (а) при нечетном (б) и четном (в) числе рецепторов

Экранная система должна управляемо перемещаться и переориентироваться в объектной системе координат для того, чтобы моделировать масштаб и ракурс. Покажем, как связаны эти системы. Из [41, 119, 120] известно, что координаты  $(x, y, z)$  некоторой точки в экранной системе могут быть пересчитаны в объектную –  $(X, Y, Z)$  по правилу

$$[ X \ Y \ Z \ 1 ] = [ x \ y \ z \ 1 ] M^{-1}, \quad (3.1.1)$$

где  $M^{-1}$  – матрица  $4 \times 4$  элемента, являющаяся обратной к матрице преобразования  $M$ , связывающей объектную правую и экранную левую системы.

Матрица преобразования  $M$  имеет следующий вид [20]:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -X_0 & -Y_0 & -Z_0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\alpha & 0 & \sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\psi & \sin\psi & 0 \\ 0 & \sin\psi & -\cos\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1.2)$$

где  $X_0, Y_0, Z_0$  – координаты центра экранной системы в объектной

(рис.3.1.4);  $\alpha$  – угол, определяющий положение главного луча  $z$  по азимуту и отсчитывающийся между отрицательным направлением оси  $Y$  и проекцией главного луча на плоскость  $XY$ ;  $\psi$  – угол, определяющий положение главного луча  $z$  относительно плоскости  $XY$  и отсчитывающейся в направлении, показанном на рис.3.1.4.

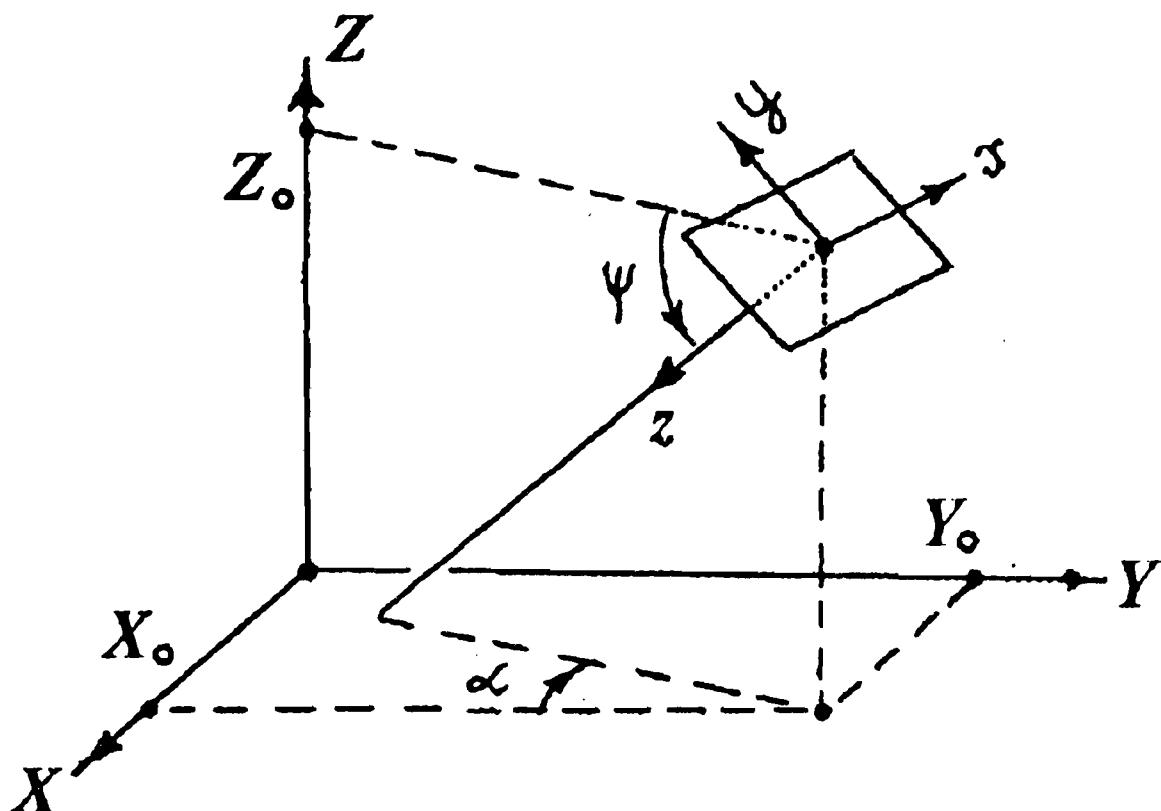


Рис. 3.1.4. Взаимная ориентация объектной и экранной систем координат

Для того чтобы избежать численного обращения матриц, целесообразно использовать полученную аналитически обратную матрицу  $M^{-1}$ :

$$M^{-1} = \begin{bmatrix} -\cos\alpha & \sin\alpha & 0 & 0 \\ -\sin\alpha \sin\psi & -\sin\psi \cos\alpha & \cos\psi & 0 \\ -\sin\alpha \cos\psi & -\cos\alpha \cos\psi & -\sin\psi & 0 \\ X_0 & Y_0 & Z_0 & 1 \end{bmatrix} \quad (3.1.3)$$

Пользуясь правилами преобразования, можно определить координаты  $(X_{ij}, Y_{ij}, Z_{ij})$   $ij$ -го рецептора в объектной системе:

$$[X_{ij} \ Y_{ij} \ Z_{ij} \ 1] = [x_{ij} \ y_{ij} \ z_{ij} \ 1] M^{-1} \quad (3.1.4)$$

и, аналогично, положение точки проекции  $F$ :

$$[X_F \ Y_F \ Z_F \ 1] = [0 \ 0 \ f \ 1] M^{-1}. \quad (3.1.5)$$

Здесь и далее часто используется представление координат точки в виде строки с четырьмя элементами. В данном случае четвертый элемент равен единице и не несет информационно полезной нагрузки. Такое представление обеспечивает универсальность метода для описания преобразований не только

точек, но и прямых линий, отрезков прямых, кривых линий, плоскостей, участков плоскостей квадратичных и бикубических поверхностей.

## 3.2. МОДЕЛЬ ОБЪЕКТА

### 3.2.1. ПРИМИТИВЫ – БАЗОВЫЕ СТРОИТЕЛЬНЫЕ ЭЛЕМЕНТЫ ОБЪЕКТА

Наиболее удобно представлять трехмерные объекты для метода обратного трассирования лучей в виде отдельных строительных блоков, поверхности которых обычно описываются функциями первого и второго порядка. Выбор таких функций обусловлен необходимостью аналитического, а не численного решения уравнений пересечения светового луча с поверхностями. Описание бикубических поверхностей представлено в § 3.4.4.

Будем называть *функциональным объемом* некоторую часть пространства (необязательно конечную), которая охватывается поверхностью одной функции. Для того чтобы однозначно определить, какой участок полупространства относится к телу объекта, а какой вне его, установим следующее правило: принадлежащим телу объекта считается подпространство, выделяемое поверхностью  $f(X, Y, Z) = 0$ , в любой точке которого значение скалярного поля  $f(X, Y, Z) > 0$ . Назовем такое подпространство *положительным*, а смежное с ним и лежащее по другую сторону от поверхности функции – *отрицательным*. При соблюдении этой договоренности, автоматически выполняется условие направленности вектора нормали внутрь тела, т. е. в сторону положительного подпространства, так как градиент скалярного поля описываемых классов направлен нормально от поверхности в сторону возрастания значения  $f(X, Y, Z)$ .

*Объемными примитивами* будем называть конечные участки пространства, ограниченные одной или несколькими функционально описанными поверхностями. Очень часто в качестве примитива используют функциональный объем, ограниченный плоскостями – многогранник. Примитивы, естественно, должны обеспечивать удобство конструирования из них производных тел и обладать относительной математической простотой.

*Плоским примитивом* будем называть часть плоскости, ограниченную замкнутой линией, состоящей из конечного числа прямолинейных или криволинейных участков.

Для одного и того же примитива характерны неизменное

количество ограничивающих его тело поверхностей и стандартный вид функций, описывающих эти поверхности. Параметры функций являются варьируемыми, этим достигается изменение формы примитива (например из эллипсода в шар), их пространственного положения и ориентации. Наиболее употребительные типы примитивов показаны на рис.3.2.1: а – тетраэдр, б – параллелепипед, в – цилиндр, г – эллипсоид, д – конус, е – часть плоскости.

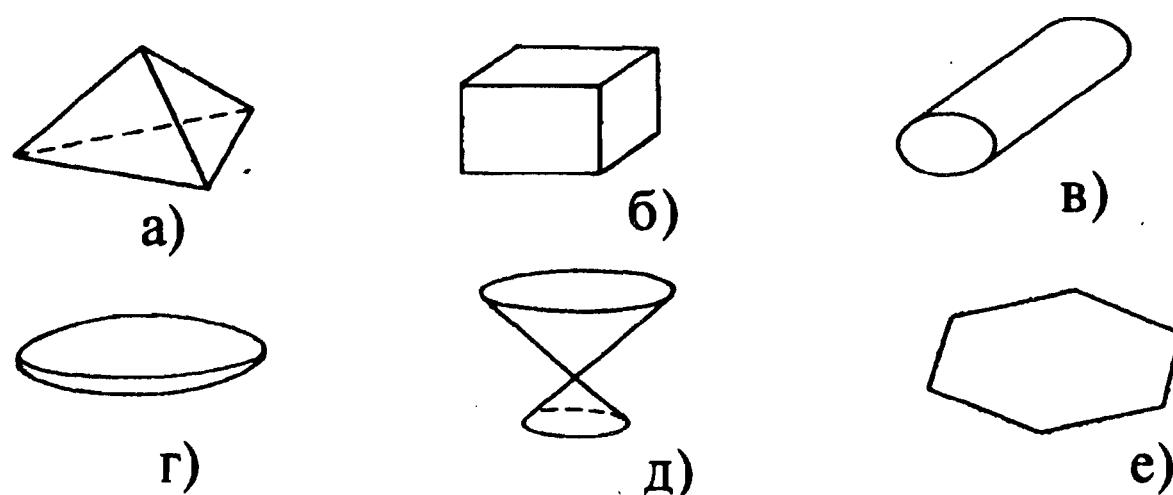


Рис. 3.2.1. Типичные примитивы

Изображения некоторых примитивов, полученные методом машинной графики, показаны на рис.3.2.2 - 3.2.6.

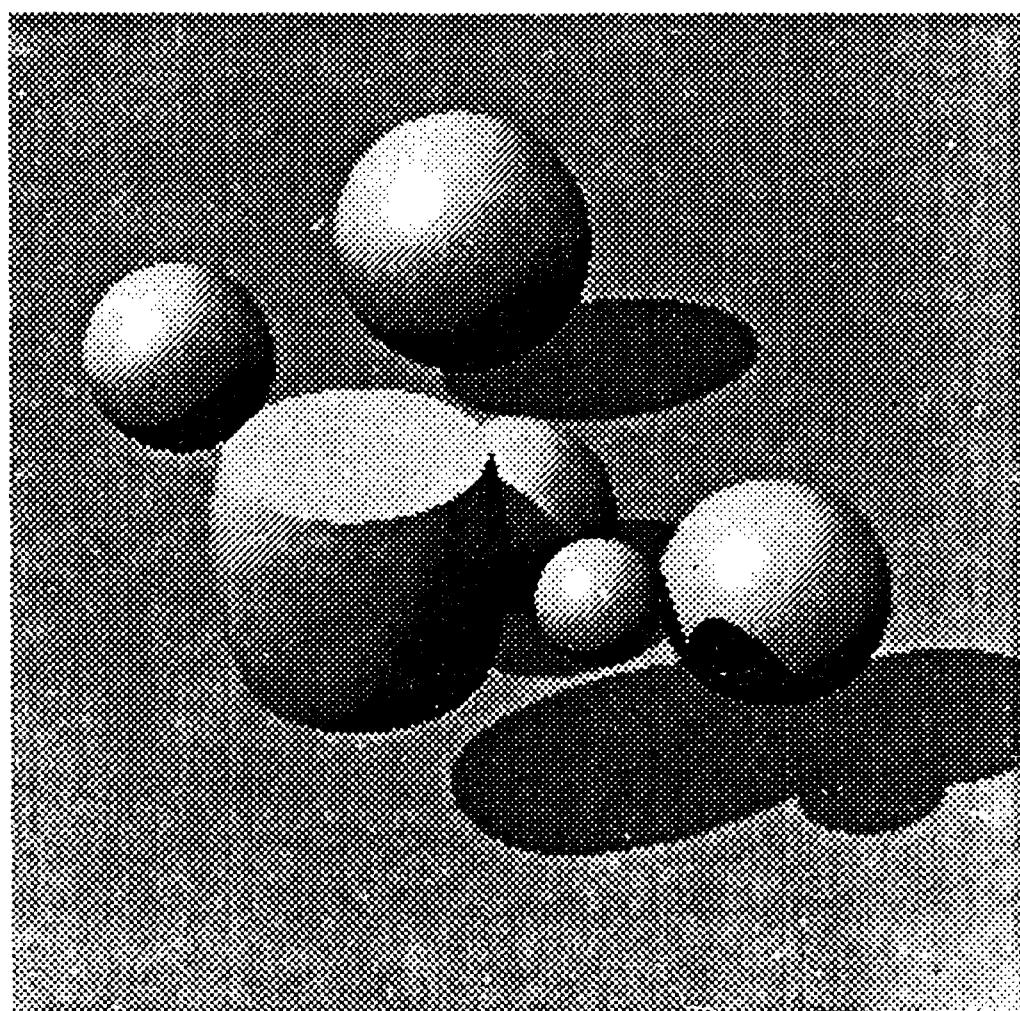


Рис. 3.2.2. Сцена, составленная из эллипсоидов

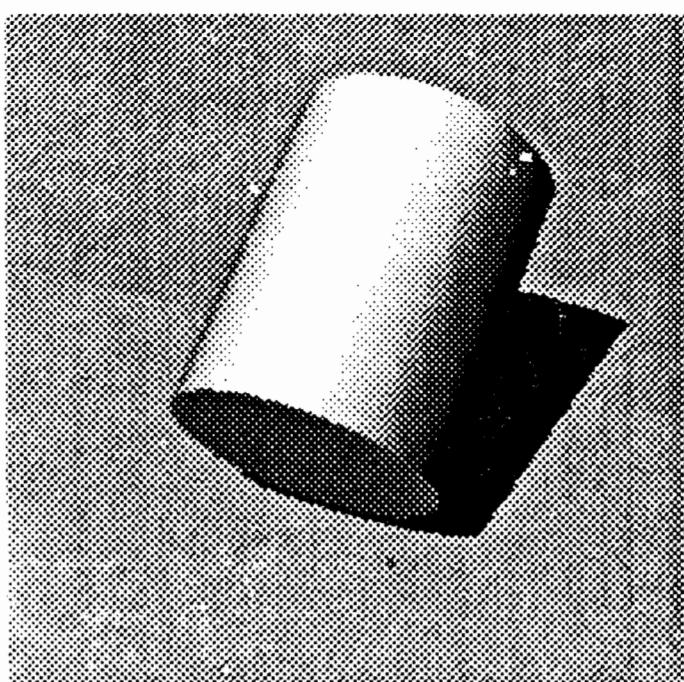


Рис. 3.2.3. Цилиндр

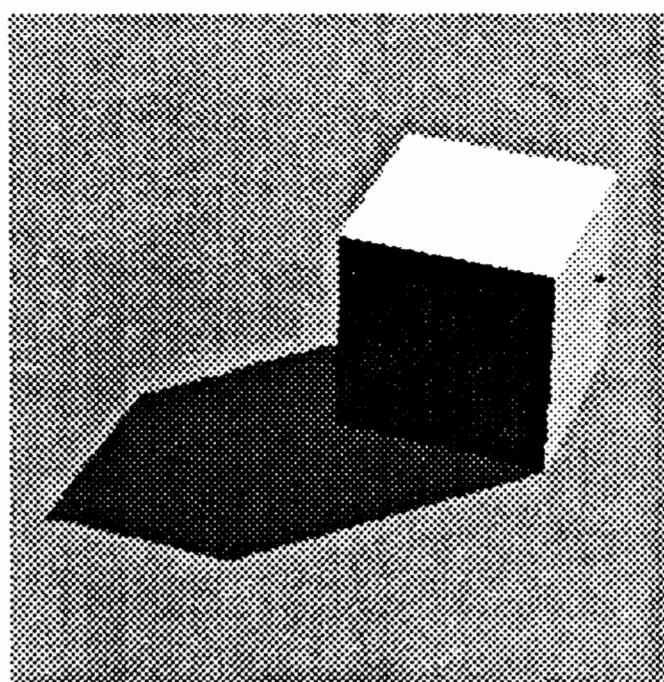
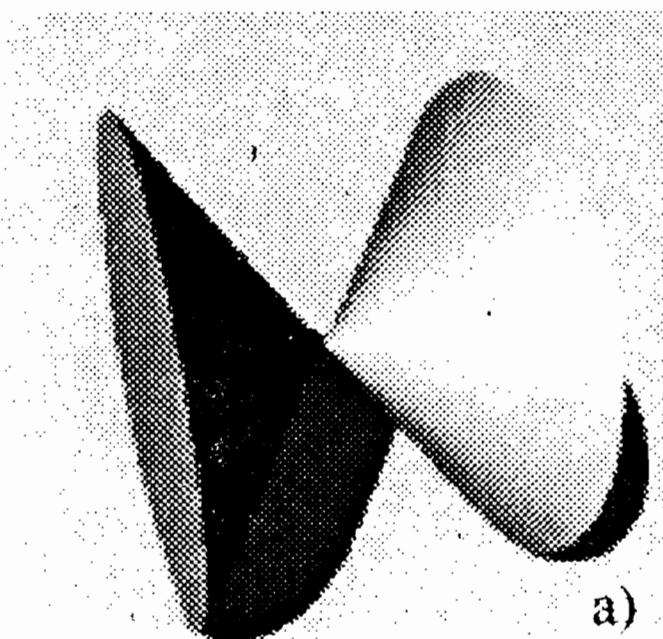
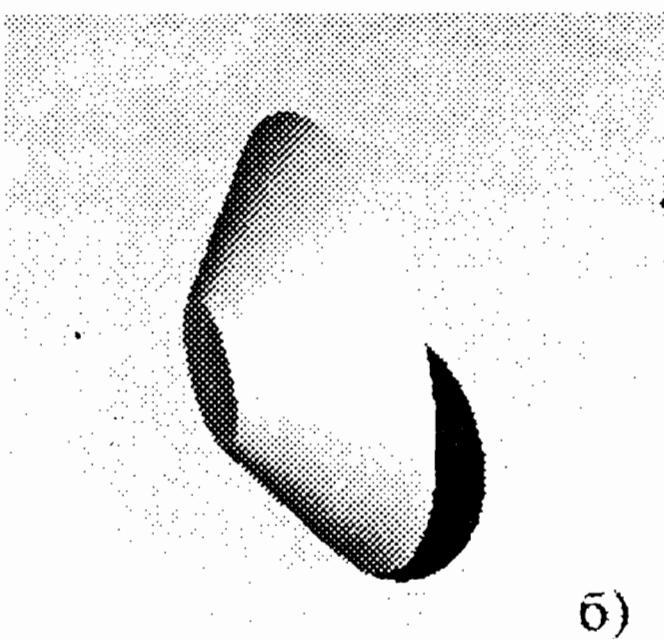


Рис. 3.2.4. Параллелепипед



а)



б)

Рис. 3.2.5. Конус двухсторонний(а) и односторонний(б)

Приведем пример математического описания цилиндрического примитива в виде кругового цилиндра с плоскими торцами, перпендикулярными оси. Математическая модель примитива состоит из уравнения цилиндра

$$-(X-X_0)^2 + (Y-Y_0)^2 + (Z-Z_0)^2 - [A(X-X_0)+B(Y-Y_0)+C(Z-Z_0)]^2 - R^2 = 0, \quad (3.2.1)$$

где  $X_0, Y_0, Z_0$  – координаты любой точки на оси цилиндра;  $A, B, C$  – компоненты направляющего вектора оси цилиндра, и уравнений торцевых поверхностей  $AX+BY+CZ+(-AX_1-BY_1-CZ_1)=0$ ,  $-AX-BY-CZ+(AX_2+BY_2+CZ_2)=0$ , где  $(X_1, Y_1, Z_1), (X_2, Y_2, Z_2)$  – координаты осевых точек первом и втором торце соответственно.

Для всех поверхностей примитива сохраним правило размещения положительных подпространств внутри тела

примитива. Так, если в состав примитива входят К уравнений вида  $f_k(X, Y, Z) = 0$ , где  $k = 1(1)K$ , то на этапе конструирования примитива устанавливают состояние:

$$\forall k : f_k(X_{\text{вн}}, Y_{\text{вн}}, Z_{\text{вн}}) > 0, \quad (3.2.2)$$

где  $(X_{\text{вн}}, Y_{\text{вн}}, Z_{\text{вн}})$  – любая заведомо внутренняя точка тела примитива. Достижение указанного состояния осуществляется подбором знака в уравнении  $\pm f_k(X, Y, Z) = 0$ .

Формальное правило преобразования исходно заданной функции  $f_k(X, Y, Z)$  в форму  $\pm f_k^+(X, Y, Z)$ , обеспечивающую положительность подпространства внутри примитива, представляется в виде:  $f_k^+(X, Y, Z) = \text{sign}[f_k(X_{\text{вн}}, Y_{\text{вн}}, Z_{\text{вн}})] f_k(X, Y, Z)$ .

Напомним, что положительность значения всех функций внутри тела примитива необходима для локализации примитива в пространстве и однозначного определения положения нормали – внутрь примитива.

На рис. 3.2.6 показан пример изображения истребителя, сконструированного из геометрических примитивов.

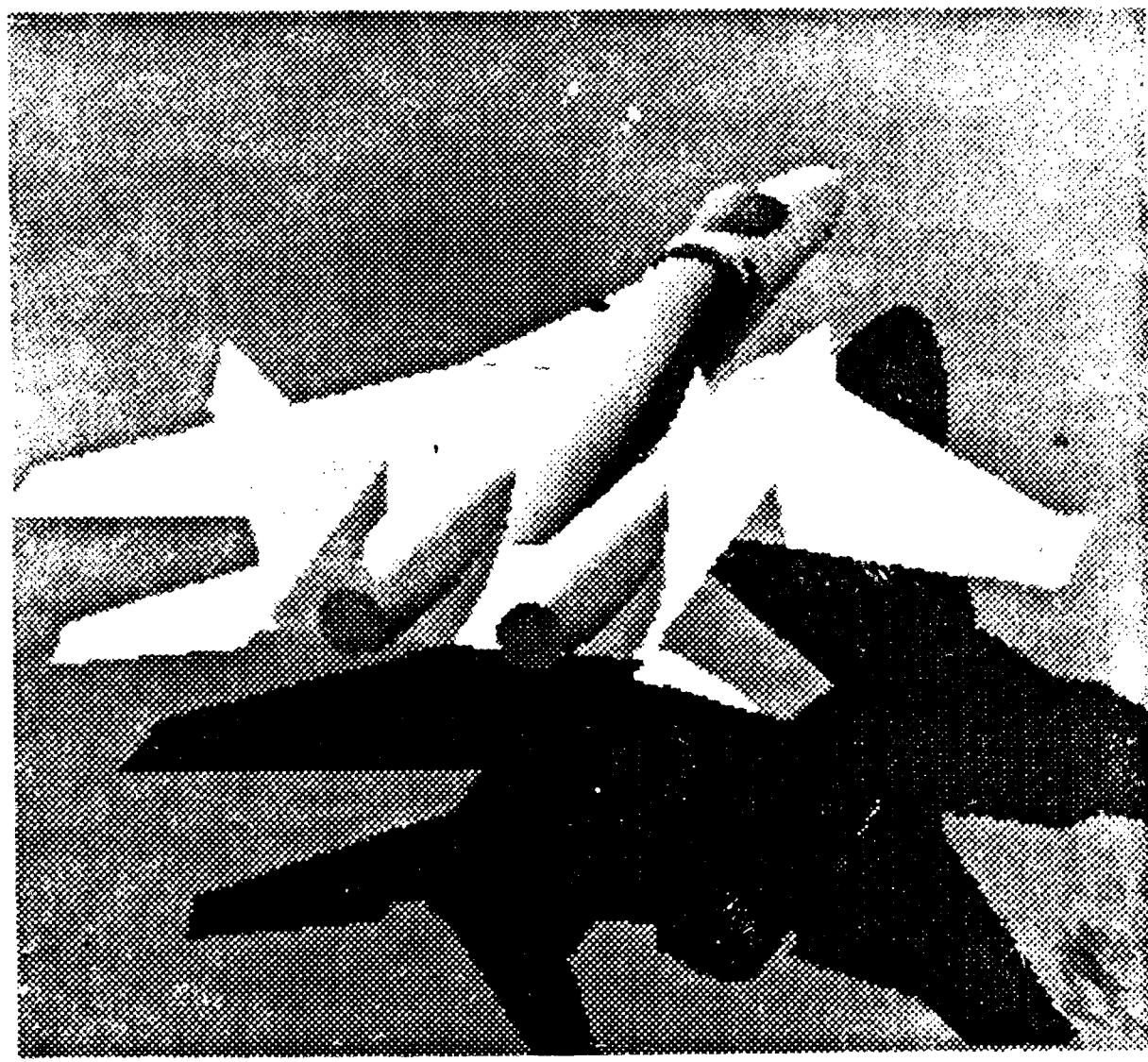


Рис. 3.2.6. Истребитель

### 3.2.2. ПРОСТРАНСТВЕННЫЕ КОМБИНАЦИИ ПРИМИТИВОВ

Примитивы могут пространственно комбинироваться друг с другом, образуя более сложные формы, которые будем называть *строительными блоками* или просто *блоками*. Блоки обеспечивают создание функциональной детали или целого узла. На рис.3.2.7 показаны строительный блок и примитивы, на которых он построен. Над примитивами допустимы следующие пространственные операции взаимодействия друг с другом: “+” – объединение; “&” – пересечение; “-” – вычитание.

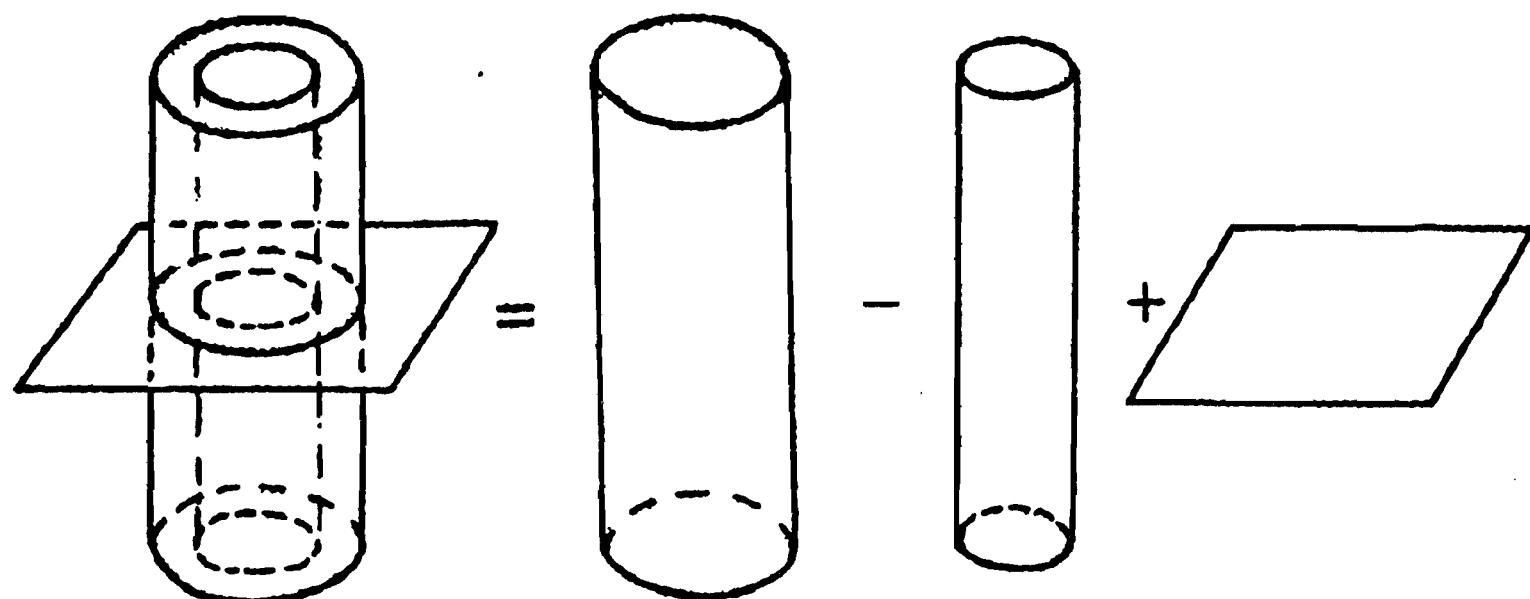


Рис. 3.2.7. Декомпозиция строительного блока на примитивы

*Объединением* или *суммой* примитивов  $\Pi_1$  и  $\Pi_2$  называется тело, каждая точка которого принадлежит хотя бы одному из объединяющихся примитивов:  $(\cdot) A \in (\Pi_1 + \Pi_2)$  если  $(\cdot) A \in \Pi_1$  или  $(\cdot) A \in \Pi_2$ .

*Пересечением* или *общей частью* примитивов  $\Pi_1$  и  $\Pi_2$  называется тело, каждая точка которого одновременно принадлежит обоим примитивам:  $(\cdot) A \in (\Pi_1 \& \Pi_2)$  если  $(\cdot) A \in \Pi_1$  и  $(\cdot) A \in \Pi_2$ . *Вычитанием* примитива  $\Pi_1$  из  $\Pi_2$  называется тело, каждая точка которого принадлежит  $\Pi_1$ , но не принадлежит  $\Pi_2$ :  $(\cdot) A \in (\Pi_1 - \Pi_2)$  если  $(\cdot) A \in \Pi_1$  и  $(\cdot) A \notin \Pi_2$ .

Заметим, что  $\Pi_1 + \Pi_2 = \Pi_2 + \Pi_1$ ,  $\Pi_1 \& \Pi_2 = \Pi_2 \& \Pi_1$ , но  $\Pi_2 - \Pi_1 \neq \neg \Pi_1 - \Pi_2$ .

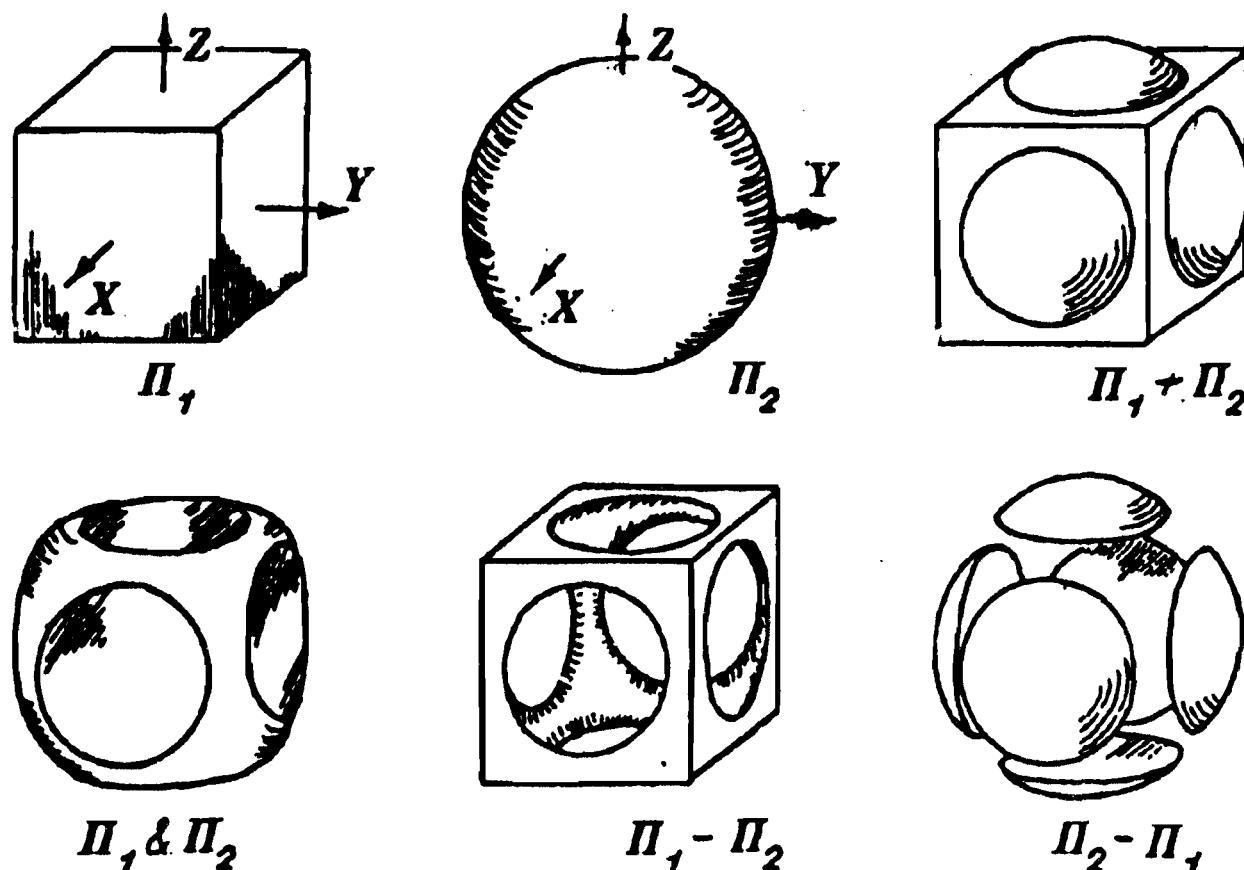


Рис. 3.2.8. Иллюстрация возможностей создания разных форм путем пространственной комбинации куба и шара

На рис.3.2.8 показаны примитивы  $P_1$  и  $P_2$  и их пространственные комбинации.

### 3.2.3. ФОРМАЛИЗОВАННАЯ МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ОБЪЕКТА

Любой пространственный объект, образованный путем комбинации примитивов можно описать структурой [9, 125], корнем которой является сам объект, вершинами – примитивы, а в узлах ветвей определены операции пространственных комбинаций. Например, на рис.3.2.9 показаны объекты  $O_1$  и  $O_2$ , описание структуры которых представляются в виде

$$O_1 = -(P_1 + (-P_2)) + P_3 = P_3 - (P_1 - P_2); O_2 = P_1 + (P_2 \& P_1), \quad (3.2.2)$$

где операция  $-P$  – взятия дополнения означает, что подразумевают объект, занимающий все трехмерное пространство за исключением точек, принадлежащих поверхности и внутренней области примитива  $P$ . Взятые дополнения еще обозначают  $\bar{P}$  и результат операции называют *отрицанием примитива*. Ранее введенное правило вычитания примитивов  $P_1 - P_2$  сводится к следующему:  $P_1 - P_2 = P_1 \& (-P_2)$ .

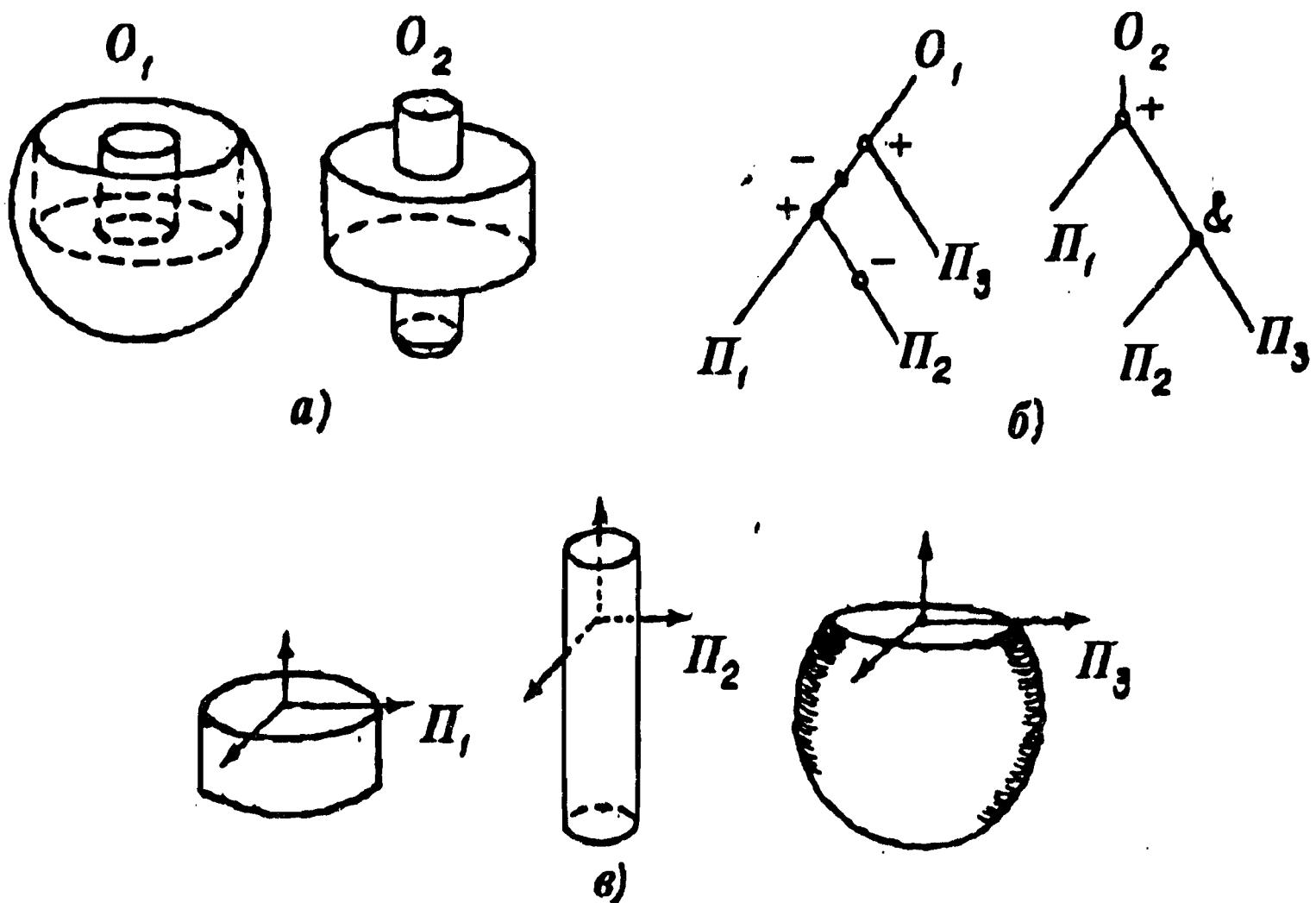


Рис. 3.2.9. Объекты (а), графическое изображение их математических моделей(б) и слагающие примитивы(в)

Множество примитивов  $P_1, P_2, \dots$ , все трехмерное пространство I и пространство нулевого объема E (пустое пространство) образуют булеву алгебру [23]. Путем пространственного сложения (+), умножения (&), взятия дополнения (-) может быть сконструирован любой комбинационный объект из исходного состава примитивов. При этих операциях справедливы следующие свойства булевой алгебры [23]:

1.  $P_1 + P_2 = P_2 + P_1$ ;
2.  $P_1 \& P_2 = P_2 \& P_1$ ;
3.  $P_1 + (P_2 + P_3) = (P_1 + P_2) + P_3$ ;
4.  $P_1 \& (P_2 \& P_3) = (P_1 \& P_2) \& P_3$ ;
5.  $P_1 \& (P_1 + P_2) = (P_1 \& P_2) \& P_1 + P_2$ ;
6.  $P_1 + P_2 \& P_3 = (P_1 + P_2) \& (P_1 + P_3)$ ;
7.  $P_1 + P_1 = P_1 \& P_1 = P_1$ ;
8.  $P_1 + P_1 = P_2$  в том и только в том случае, если  $P_1 \& P_2 = P_1$ ;
9.  $P_1 + E = P_1$ ,  $P_1 \& E = E$ ;
10.  $P_1 \& I = P_1$ ,  $P_1 + I = I$ ;

11.  $\Pi_1 + (-\Pi_1) = I$ ,  $\Pi_1 \& (-\Pi_1) = E$ ;
12.  $\Pi_1 \& (\Pi_1 + \Pi_2) = \Pi_1 + \Pi_1 \& \Pi_2 = \Pi_1$ ;
13.  $- (\Pi_1 + \Pi_2) = (-\Pi_1) \& (-\Pi_2)$ ;
14.  $- (\Pi_1 \& \Pi_2) = (-\Pi_1) + (-\Pi_2)$ ;
15.  $-(-\Pi_1) = \Pi_1$ ,  $-I = E$ ,  $-E = I$ ;
16.  $\Pi_1 + (-\Pi_1) \& \Pi_2 = \Pi_1 + \Pi_1$ ;
17.  $\Pi_1 \& \Pi_2 + \Pi_1 \& \Pi_3 + \Pi_2 \& (-\Pi_3) = \Pi_1 \& \Pi_3 + \Pi_2 \& (-\Pi_3)$ .

В соответствии с законом де Моргана (свойства 13 и 14) булево выражение, описывающее объект, можно представить в виде суммы (объединения) произведений (пересечений) примитивов или их отрицаний:

$$O = \bigcup_{q=1}^{2^D} \left( \bigcap_{d=1}^D z_i \right), \quad (3.2.3)$$

где  $D$  – число примитивов, входящих в состав объекта,  $d=1(1)D$ ;  $z_i$  это  $+\Pi_i$  или  $-\Pi_i$ ;  $q$  – номер текущего произведения,  $q = 1(1)2^D$ .

Форма (3.2.3) называется *каноническим видом булевого описания объекта*. Формализованное описание объекта в виде правила комбинирования примитивов совместно с информацией о типе каждого примитива, коэффициентов функций поверхностей каждого примитива и оптических характеристик поверхностей составляет полное представление объекта.

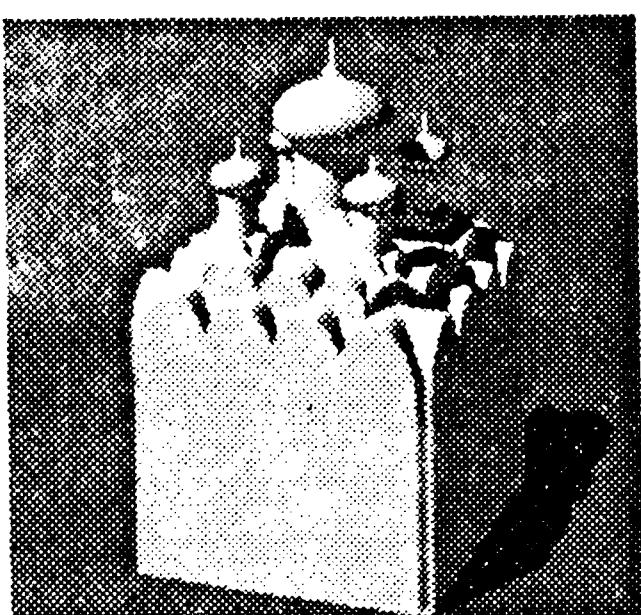


Рис. 3.2.10. Церковь

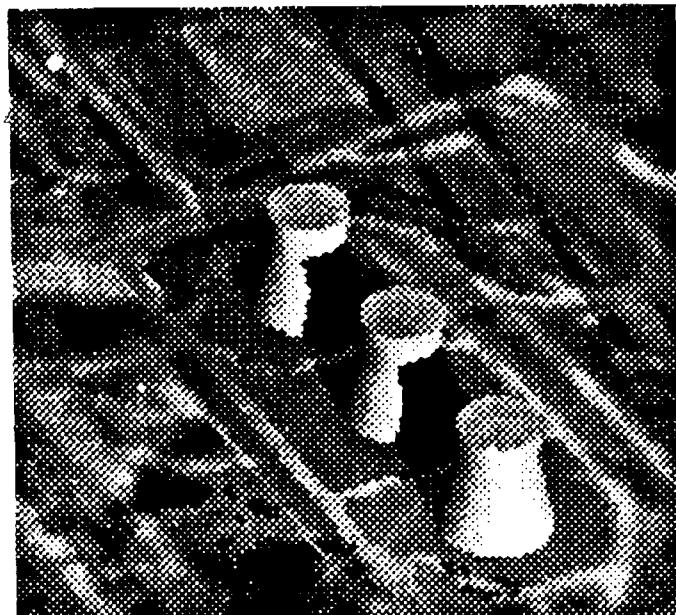


Рис. 3.2.11. Градирни ТЭС

В качестве иллюстрации изложенных принципов моделирования объекта приведем изображения церкви (рис.3.2.10) и технологических установок ТЭС (рис.3.2.11). Первый объект

составлен из эллипсоидов, цилиндров, параллелепипеда и примитивов с поверхностью бикубического описания. Второй объект составлен из усеченных конусов. В обоих случаях применялись операции пространственного сложения и вычитания примитивов.

### 3.2.4. ВЗАИМНОЕ ПОЛОЖЕНИЕ ТОЧКИ И ОБЪЕКТА

Введенные правила задания примитивов позволяют формальными методами определять взаимное расположение любой точки и конкретного примитива. Зная правила объединения примитивов в объект, возможно определить взаимное положение точки и объекта. Взаимное положение охарактеризуем через признак или функцию принадлежности  $\varphi(X, Y, Z; \Phi)$  (в более краткой записи  $\varphi(\Phi)$ ), где  $(X, Y, Z)$  – координаты точки;  $\Phi$  – обозначение примитива, объекта или другой фигуры. Функция  $\varphi$  принимает значение -1, если точка  $(X, Y, Z)$  находится вне объекта  $\Phi$ ,  $\varphi = 0$ , если точка лежит на поверхности  $\Phi$ ;  $\varphi = +1$ , если точка лежит внутри  $\Phi$ . В формальных терминах положение точки и примитива определяется следующим образом. Пусть примитив  $\Pi$  состоит из  $K$  ( $k = 1(1)K$ ) уравнений вида  $\pm f_k(X, Y, Z) = 0$ , где знаки функций выбраны по правилу (3.2.2). Тогда

$$\varphi(\Phi) = 1, \text{ если } \forall k: f_k(X, Y, Z) > 0;$$

$$\varphi(\Phi) = 0, \text{ если } \{(\exists \rho: f_\rho(X, Y, Z) = 0) \cup (\forall k \neq \rho: f_k(X, Y, Z) \geq 0)\}, \text{ где } \rho \in [1, K];$$

$$\varphi(\Phi) = -1, \text{ если } \exists j: f_j(X, Y, Z) < 0, \text{ где } j \in [1, K].$$

Точка лежит внутри примитива, если в этой точке значения всех функций, слагающих поверхность примитива, положительны. Точка лежит на поверхности примитива, если существует хотя бы одна функциональная поверхность, значение которой в точке равно нулю, а значения остальных функций неотрицательны. Точка лежит вне примитива, если существует хотя бы одна функция поверхности, значение которой в этой точке отрицательно.

Рассмотрим теперь правила определения взаимного положения точки и объекта, которые очень важны для реализации процесса построения изображения. При трассировании лучей световая прямая пересекается с множеством поверхностей различных примитивов. Прежде всего следует установить факт принадлежности очередной точки пересечения поверхности объекта. Этот факт зависит от взаимного расположения точки и примитивов и правила пространственного комбинирования примитивов.

Таблица 3.2.1 – это правила определения положения точки по отношению к комбинации пары примитивов А и В.

**Таблица 3.2.1.**

$\phi(A)$	$\phi(B)$	$\phi(-A)$	$\phi(A-B)$	$\phi(B-A)$	$\phi(A+B)$	$\phi(A \& B)$
-1	-1	1	-1	-1	-1	-1
-1	0	1	-1	0	0	-1
-1	1	1	-1	1	1	-1
0	-1	0	0	-1	0	-1
0	0	0	-1	-1	0	0
0	1	0	-1	0	1	0
1	-1	-1	1	-1	1	-1
1	0	-1	0	-1	1	0
1	1	-1	-1	-1	1	1

Заметим, что если  $\phi(-A) = 0$ , то  $\phi(A) = 0$ , т. е. поверхность фигуры и ее дополнения являются общими.

В табл. 3.2.1 под примитивами А и В понимаются пространственно ограниченные выпуклые трехмерные тела. На основании составленной таблицы формальными методами может быть определено положение точки по отношению к сложному объекту, состоящему из многих примитивов. Например, пусть точка лежит на поверхности С, т.е.  $\phi(C) = 0$ ; вне А, т.е.  $\phi(A) = -1$ ; внутри В, т.е.  $\phi(B) = -1$ , а объект задается описанием  $O = (A-B) \& C$ . Необходимо определить  $\phi(O)$ . Сначала определим  $\phi(A-B) = (-1) - (-1) = -1$ , затем  $\phi(O) = (-1) \& (\phi(C)) = \phi[(-1) \& (0)] = -1$ . В соответствии с принятой интерпретацией булевого выражения результат означает, что точка лежит вне объекта.

Приведем другой пример (рис. 3.2.12). Объект О сконструирован по правилу  $O = (\Pi_1 - \Pi_2) + (\Pi_3 - \Pi_4)$ , точка А принадлежит косому торцу примитива  $\Pi_1$  и лежит на его оси, т. е.  $\phi(\Pi_1; A) = 0$ . Соответственно  $\phi(\Pi_2; A) = 1$ ,  $\phi(\Pi_3; A) = 0$ ,  $\phi(\Pi_4; A) = 1$ . Определим положение точки по отношению к объекту:  $\phi(\Pi_1 - \Pi_2; A) = -1$ ,  $\phi(\Pi_3 - \Pi_4; A) = -1$ , тогда  $\phi[(\Pi_1 - \Pi_2) + (\Pi_3 - \Pi_4)] = [(-1) + (-1)] = -1$ , т.е. точка А не принадлежит объекту, как и показано на рисунке.

На рис.3.2.13 показаны два объекта  $O_1$  и  $O_2$ , составленные из примитивов  $\Pi_1, \Pi_2, \Pi_3$  по правилам  $O_1 = \Pi_3 - (\Pi_1 \& \Pi_2)$ ,  $O_2 = (\Pi_3 - \Pi_1) + \Pi_2$ . Положение точки В показано на изображениях примитивов; тогда  $\phi(\Pi_1; B) = -1, \phi(\Pi_2; B) = 1, \phi(\Pi_3; B) = -1$ . Пользуясь

табл. 3.2.1, можно получить:  $\phi(\Pi_1 \& \Pi_2; B) = \phi[(-1) \& (1)] = -1$ ,  $\phi(O_1) = \phi[(-1) - (-1)] = -1$ , т. е. точка В лежит вне объекта  $O_1$ . Для второго объекта справедливо  $\phi(\Pi_3 - \Pi_1) = \phi[(-1) - (-1)] = -1$ ; тогда  $\phi(O_2) = \phi[(-1) + (1)] = 1$ , т. е. точка В лежит внутри  $O_2$ , как и показано на рис.3.2.13.

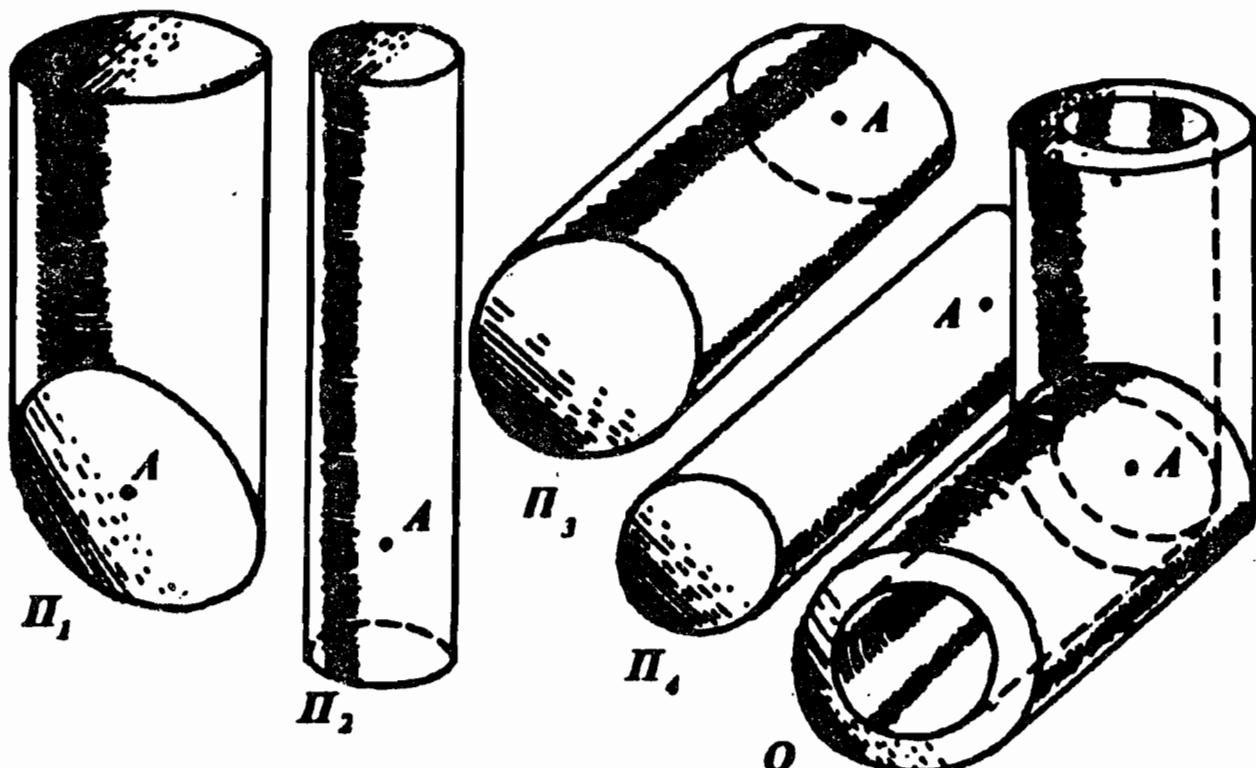


Рис. 3.2.12. Положение точки А относительно примитивов и объекта

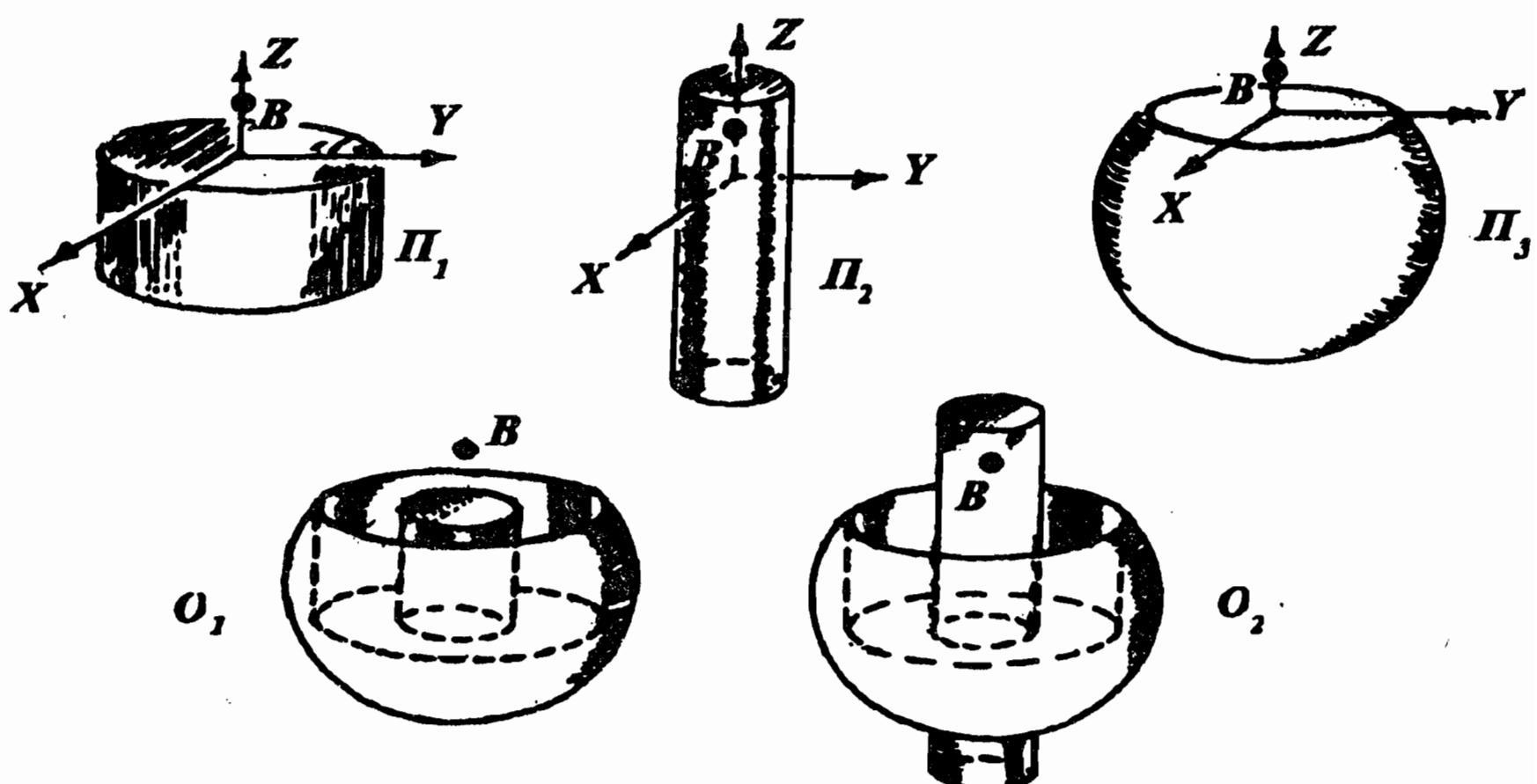


Рис. 3.2.13. Положение точки В относительно примитивов и объекта

### 3.3. ОПРЕДЕЛЕНИЕ ВИДИМЫХ И ЗАТЕНЕННЫХ ТОЧЕК

Для определения освещенности изображения необходимо установить для каждого рецептора, какую точку объекта он “видит”, каковы ориентация нормали в этой точке, отражательную способность и другие необходимые данные. Пусть объект  $O$  содержит  $D$  примитивов  $\Pi_1, \Pi_2, \Pi_3, \dots, \Pi_d, \dots, \Pi_D$ , которые пространственно комбинируются друг с другом в форме (3.2.2), (3.2.3). Обозначим  $K_d$  число поверхностей, слагающих  $d$ -й примитив; соответственно  $k_d$  – текущая  $k$ -я поверхность в примитиве под номером  $d$ ;  $k_d = 1(1)K_d$ . Степень функции  $k_d$ -й поверхности обозначим  $S_{k_d}$ . В этих обозначениях определим пересечения светового луча: сначала с каждым примитивом, а затем со всем объектом.

#### 3.3.1. ПЕРЕСЕЧЕНИЯ СВЕТОВОГО ЛУЧА С ПРИМИТИВОМ

Пусть  $d$ -й примитив содержит  $K_d$  поверхностей, организация которых осуществлена по правилу положительности внутренней области примитива (3.2.2). Тогда для определения всех точек пересечения прямой, исходящей из  $ij$ -го рецептора через центр проекции  $F$ , и  $d$ -го примитива необходимо решить  $K_d$  систем вида

$$\begin{cases} (X - X_{ij}) / (X_F - X_{ij}) = (Y - Y_{ij}) / (Y_F - Y_{ij}) = (Z - Z_{ij}) / (Z_F - Z_{ij}); \\ f_{k_d}(X, Y, Z) = 0. \end{cases} \quad (3.3.1)$$

Каждая такая система может: вообще не иметь решений (луч минует поверхность); иметь единственное решение (луч пересекает плоскость или касается выпуклой поверхности), иметь несколько решений (луч пересекает кривую поверхность несколько раз) и, наконец, бесконечное множество решений (луч совпадает с поверхностью). Алгоритмы и программы пересечения поверхностей первого и второго порядка со световым лучом приведены в § 3.4. и приложении.

Исключим последнюю группу решений из-за неопределенности положения точки пересечения. В общем же случае каждая  $k_d$ -я система может давать  $S_{k_d}$  решений. Но все точки-решения принадлежат поверхности примитива. Точка решения  $(X, Y, Z)S_{k_d}$ , принадлежащая поверхности  $k_d$ , принадлежит

поверхности  $d$ -го примитива при выполнении условия  $\forall \rho \neq k_d : f_\rho(X, Y, Z) s_{k_d} \geq 0$ , где  $s_{k_d}$  – номер текущего решения для  $k$ -й поверхности,  $s_{k_d} = 1(1)S_{k_d}$ ;  $\rho = 1(1)K_d$  – текущий номер поверхности в  $d$ -м примитиве.

Другими словами, точка, принадлежащая некоторой поверхности, в свою очередь входящей в описание примитива, принадлежит поверхности примитива, если для всех остальных поверхностей эта точка находится в неотрицательной части пространства.

Исключив из дальнейшего анализа те решения, которые не удовлетворяют последнему условию, получим набор  $B_d$  точек – действительных пересечений, точно лежащих на поверхности примитива. Ряд из них может дублироваться, что возникает в случае прохождения луча через границу смежных поверхностей. Так как априори известно, что примитив выпуклый, то прямая может пересечь его поверхность максимально дважды. Остальные решения могут относиться к различным поверхностям, но физически соответствуют одним и тем же точкам. Например, при пересечении куба через две диаметрально противоположные вершины по большой диагонали образуют шесть решений, принадлежащих соответствующим плоскостям и одновременно самой поверхности куба. Фактически эти решения представляют собой две одинаковые тройки. Однако так как решения получены в различных частях программы, то из-за ошибок представления чисел с плавающей запятой физически одинаковые координаты могут несколько отличаться. Отберем две точки фактического пересечения луча и выпуклого примитива. Из всех возможных  $B_d$  претендентов выберем ближайшую  $(X_1, Y_1, Z_1)$  и самую удаленную точку  $(X_2, Y_2, Z_2)$ . Здесь возможны различные подходы. В самом общем случае должны выполняться соотношения  $(X_1 - X_F)^2 + (Y_1 - Y_F)^2 + (Z_1 - Z_F)^2 \leq (X_b - X_F)^2 + (Y_b - Y_F)^2 + (Z_b - Z_F)^2$ , а также  $(X_2 - X_F)^2 + (Y_2 - Y_F)^2 + (Z_2 - Z_F)^2 \geq (X_b - X_F)^2 + (Y_b - Y_F)^2 + (Z_b - Z_F)^2$  для любого  $b$ , где  $b = 1(1)B_d$  – текущий номер точки действительного пересечения луча с примитивом.

Для дальнейшего анализа необходимо сопоставлять каждую точку пересечения с поверхностью, которой она принадлежит. Поэтому информация о пересечении луча с  $d$ -м примитивом представим в виде матрицы координат точек пересечения:

$$T_d = \begin{bmatrix} X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \end{bmatrix}$$

и матрицы номеров поверхностей, которым принадлежат точки  $(X_1, Y_1, Z_1)$  и  $(X_2, Y_2, Z_2)$ :

$$WHO_d = \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}.$$

Для случая одинаковых решений (одного решения) каждая матрица имеет одну строку.

Ключевые элементы алгоритма определения точек пересечения прямой и примитива выглядят следующим образом.

1. Устанавливают  $k_d = 1$ , где  $k$  - номер обрабатываемой поверхности в составе примитива  $d$ . Устанавливают FLAG = 0, где FLAG – индикатор отсутствия (0), наличия (1) решений.

2. Решают систему (3.3.1).

3. Если решений нет, то  $k_d = k_d + 1$  (пока  $k_d \leq K_d$ ) и возврат на шаг 2.

4. Если решения есть, а их в общем случае может быть  $S_{k_d}$ :  $(X, Y, Z)_1, (X, Y, Z)_2, \dots (X, Y, Z)_{S_{k_d}}$ , то устанавливают  $s_{k_d} = 1$ , где  $s_{k_d}$  – номер текущего решения  $k_d$ -й поверхности.

5. Для всех  $K_d$  поверхностей, кроме  $k_d$ -й проверяют условие  $f_p(X, Y, Z)_{s_{k_d}} \geq 0 \quad \forall p \neq k, p=1(1)K_d$ .

6. Если условие не выполняется, то  $s_{k_d} = s_{k_d} + 1$  (пока  $s_{k_d} \leq S_{k_d}$ ) и возврат на шаг 5.

7. Если условие 5 выполняется и FLAG = 0, то точка  $(X, Y, Z)_{s_{k_d}}$  размещается в две первые строки матрицы  $T$ , а WHO заполняется:  $WHO(1,1)=WHO(2,1)=k_d$ .

8. Если условие 5 выполняется и FLAG = 1, то алгоритм ветвится: если  $(X, Y, Z)_{s_{k_d}}$  ближе к точке  $(X_F, Y_F, Z_F)$ , чем точка из первой строки матрицы  $T$ , то  $T(1,*) = (X, Y, Z)_{s_{k_d}}$ ,  $WHO(1,1)=k_d$ ; если же  $(X, Y, Z)_{s_{k_d}}$  дальше от  $(X_F, Y_F, Z_F)$ , чем точка во второй строке матрицы  $T$ , то  $E(2,*) = (X, Y, Z)_{s_{k_d}}$ ,  $WHO(2,1)=k_d$ .

9.  $\text{FLAG} = 1$ ,  $k_d = k_d + 1$  (пока  $k_d \leq K_d$ ), возврат на шаг 2.

В результате определения пересечений светового луча с примитивом устанавливается сам факт наличия пересечения, и в том случае вычисляются координаты двух точек пересечения. Одновременно запоминаются номера поверхностей внутри описания  $d$ -го примитива, которым эти точки принадлежат.

### 3.3.2. ПЕРЕСЕЧЕНИЯ СВЕТОВОГО ЛУЧА С КОМБИНАЦИЕЙ ПРИМИТИВОВ

После того как задача определения точек пересечений решена для всех примитивов, входящих в состав объекта, необходимо выделить среди всей этой совокупности точку, ближайшую к наблюдателю, т.е. видимую. Однако формально ближайшая точка может не принадлежать поверхности объекта, как это бывает, например, для близких к наблюдателю вычитающихся примитивов.

Для каждого  $d$ -го примитива ранее были составлены матрицы:  $T_d$  – координат пересечения луча с примитивом и матрица  $\text{WHO}_d$  – указателей номеров поверхностей внутри примитива, которым принадлежат соответствующие точки пересечения из  $T_d$ . Сведем все  $D$  матриц  $T_d$  и  $D$  матриц  $\text{WHO}_d$  соответственно в общие для всего объекта блочные матрицы  $T$  и  $\text{WHO}$ . Матрица  $T$  представляет собой матрицу-столбец, первый элемент которой есть матрица  $T_1$ ,  $d$ -й –  $T_d$ , а последний –  $T_D$ . Матрица  $\text{WHO}$  имеет следующую структуру:

$$\text{WHO} = \begin{bmatrix} \text{WHO}_1 & | & 1 \\ & | & 1 \\ \hline : & | & : \\ \hline \text{WHO}_d & | & d \\ & | & d \\ \hline : & | & : \\ \hline \text{WHO}_D & | & D \\ & | & D \end{bmatrix}$$

В общем случае размерность матрицы  $T$  составляет  $2D \times 3$ , матрицы  $\text{WHO}$  –  $2D \times 2$ . В каждой строке  $\text{WHO}$  первое число указывает номер поверхности, второе – номер примитива для точки, координаты которой помещены в одноименной строке  $T$ .

Переформируем  $T$  так, чтобы в первой ее строке помещалась ближайшая к источнику луча точка, остальные же точки могут

произвольно располагаться на более низких позициях. Соответственно и в том же порядке переставим строки матрицы WHO, получим новую матрицу WHO'. Это необходимо для того, чтобы не потерять связь между точкой и конкретной поверхностью.

Теперь после выделения ближайшей точки  $(X_B, Y_B, Z_B)$  следует убедиться в том, что она принадлежит поверхности объекта. Для этого необходимо оценить положение точки по отношению к каждому примитиву, входящему в состав объекта, а затем, пользуясь формальными правилами (табл. 3.2.1), установить относительное положение точки и объекта. Положение точки и каждого примитива  $\Pi_1, \dots, \Pi_d, \dots, \Pi_D$  определяется функцией принадлежности  $\phi(X_B, Y_B, Z_B; \Pi_d)$ , вычисляемой в соответствии с правилами § 3.2.4. Положение точки  $(X_B, Y_B, Z_B)$  и объекта  $O = f(\Pi_1, \dots, \Pi_d, \dots, \Pi_D)$  устанавливают путем попарной подстановки значений  $\phi(X_B, Y_B, Z_B; \Pi_d)$ ,  $d = 1(1)D$ , в формализованное описание (3.2.2) и (3.2.3) для объекта. Преобразования значений функции принадлежности над парами примитивов приведены в табл. 3.2.1.

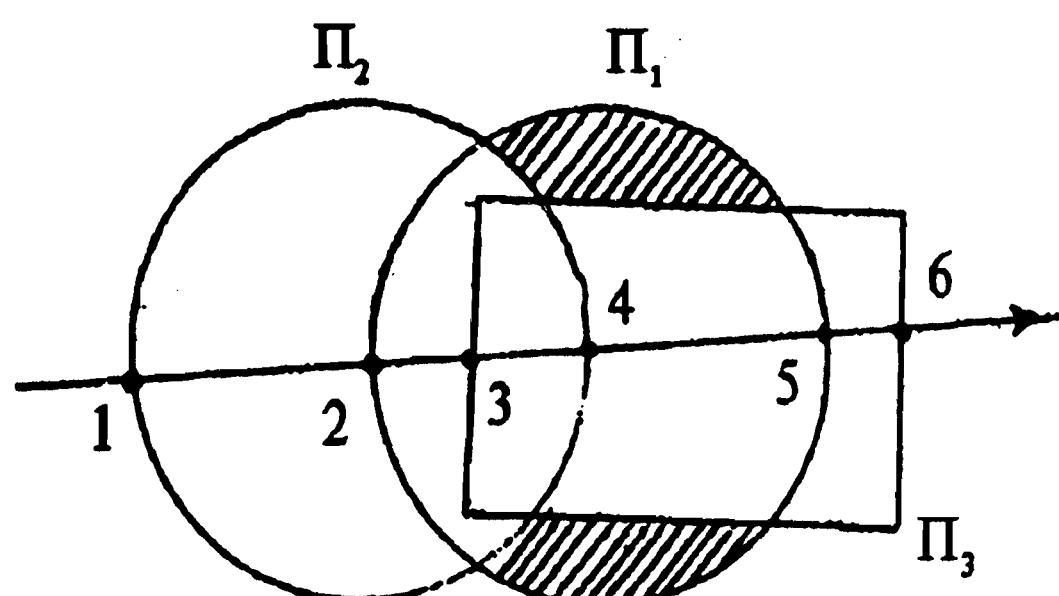


Рис. 3.3.1. Пересечение светового луча со сложной комбинацией примитивов

Например, пусть объект  $O$  составлен из примитивов  $O = (\Pi_1 - \Pi_2) - \Pi_3$  (рис. 3.3.1). Световой луч пересекает эту комбинацию в точках 1–6. Ближайшей к источнику луча будет точка 1. Функции принадлежности этой точки и примитивов принимают значения  $\phi(X_1, Y_1, Z_1; \Pi_1) = -1$ ;  $\phi(X_1, Y_1, Z_1; \Pi_2) = 0$ ;  $\phi(X_1, Y_1, Z_1; \Pi_3) = -1$ ; тогда  $\phi(X_1, Y_1, Z_1; \Pi_1 - \Pi_2) = -1$ ;  $\phi(X_1, Y_1, Z_1; (\Pi_1 - \Pi_2) - \Pi_3) = -1$ . Таким образом, точка 1, будучи ближайшей к

источнику луча, не является видимой, так как находится вне объекта.

В случае, если ближайшая точка не лежит на поверхности объекта, как это случилось в приведенном примере, ее следует устраниить из дальнейшего анализа, а среди оставшихся точек-пересечений вновь выделить ближайшую и вновь проверить ее на принадлежность объекту. Этот процесс следует продолжать либо до исчерпания всех точек, либо до нахождения как очередной ближайшей, так и одновременно принадлежащей объекту точки. В первом случае констатируется факт непересечения луча и объекта, во втором – осуществляется определение координат видимой точки ( $X_B, Y_B, Z_B$ ).

Так как одновременно с каждым переформированием Т однотипно переформировывалась WHO, то номера поверхности и примитива для видимой точки прочитываются соответственно из первого и второго столбца первой строки WHO.

В описанной процедуре используется алгоритм поиска ближайшей ( $X_B, Y_B, Z_B$ ) точки к некоторой другой точке ( $X_F, Y_F, Z_F$ ) среди множества точек ( $X_n, Y_n, Z_n$ ),  $n=1(1)N$ , которые все принадлежат одной прямой – световому лучу.

В том случае, если объект и прямая заданы в объектной системе координат, то наиболее общим критерием ближайшей точки является следующий:  $(X_B, Y_B, Z_B) : (X_F - X_B)^2 + (Y_F - Y_B)^2 + (Z_F - Z_B)^2 \leq (X_F - X_n)^2 + (Y_F - Y_n)^2 + (Z_F - Z_n)^2 \forall n$ . В частных прикладных задачах критерий может быть упрощен до следующего вида:  $(X_B, Y_B, Z_B) : \text{abs}(X_F - X_B) \leq \text{abs}(X_F - X_n) \forall n$ , или  $(X_B, Y_B, Z_B) : \text{abs}(Y_F - Y_B) \leq \text{abs}(Y_F - Y_n) \forall n$ , или  $(X_B, Y_B, Z_B) : \text{abs}(Z_F - Z_B) \leq \text{abs}(Z_F - Z_n) \forall n$ .

Применение последних трех правил оправдано только в том случае, когда априорно известна непараллельность любого светового луча плоскостям  $YZ, XZ, XY$ . Так, например, целесообразно использование последнего критерия при моделировании аэросъемки. В этом случае ни один трассирующий луч не параллелен плоскости Земли ( $XY$ ) на пути камера – объект и Солнце – объект.

В случае, когда объект и приемник света заданы в экранной системе координат, наиболее пригодным является следующий хорошо известный подход:  $(x_B, y_B, z_B) : z_B \leq z_n \forall n$ .

Таким образом, формальными методами возможно определение точек пересечения светового луча с любой пространственной комбинацией примитивов. Определение ближайшей (видимой) точки может достигаться с использованием различных критериев, выбор которых зависит от смысла прикладной задачи.

### 3.3.3. ОПРЕДЕЛЕНИЕ ЗАТЕНЕННЫХ ТОЧЕК

Множество точек объекта, видимых множеством рецепторов, качественно различается между собой. Одни точки освещаются прямыми лучами света, другие – находятся в тени. Алгоритм вычисления освещенности для этих групп точек существенно отличается, поэтому предварительно следует установить признак освещенности – затененности для каждой видимой точки. Идея решения определяется очевидным фактом: видимая точка находится "на свету", если она не загораживается от источника другими поверхностями, или, другими словами, эта точка является ближайшей к источнику света среди всех остальных, лежащих на световом луче [73].

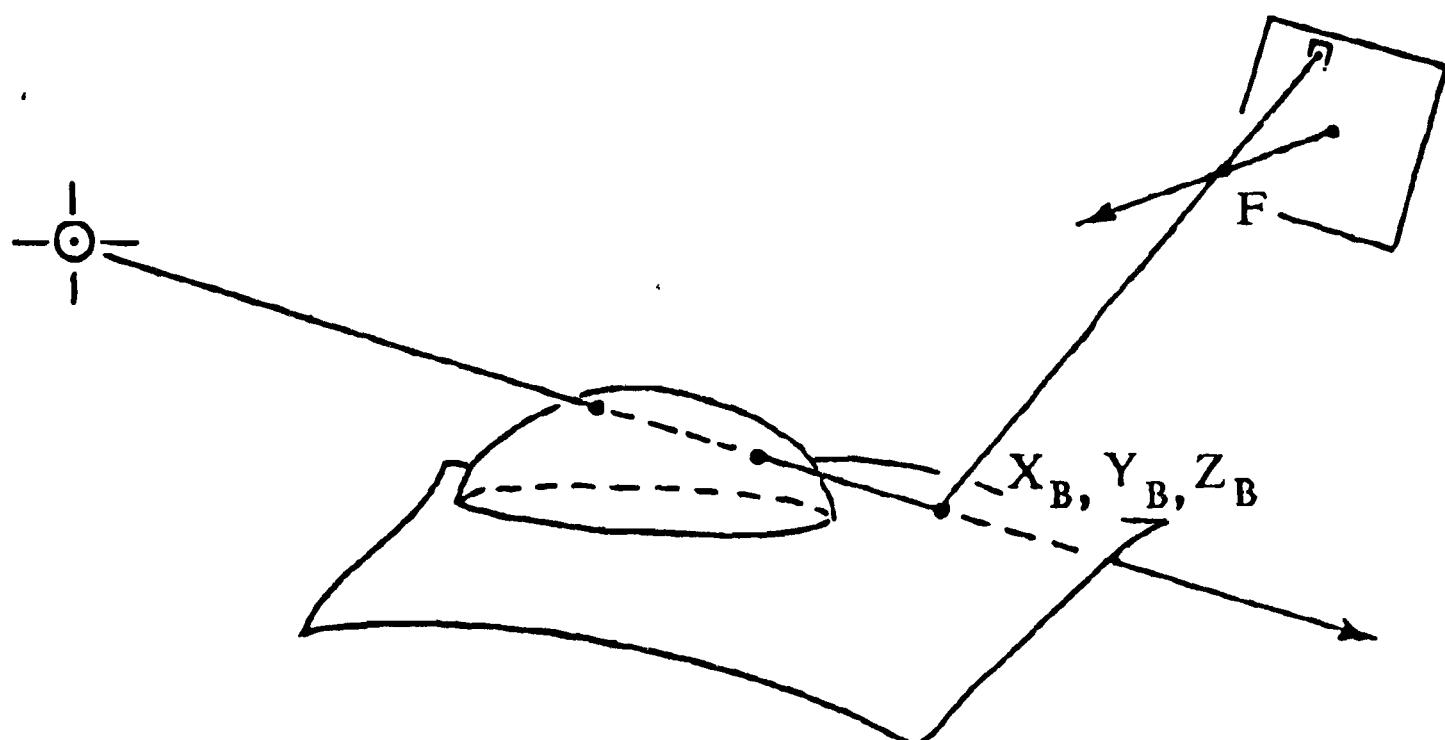


Рис. 3.3.2. Определение факта затененности точки

На рис.3.3.2 показана видимая точка  $(X_B, Y_B, Z_B)$  из  $i$ -го рецептора. Для определения ее затененности необходимо провести световой луч из источника света  $(X, Y, Z)$  на видимую точку  $(X - X_B)/(X_{\text{sun}} - X_B) = (Y - Y_B)/(Y_{\text{sun}} - Y_B) = (Z - Z_B)/(Z_{\text{sun}} - Z_B)$  и найти точки

пересечения этой прямой со всеми поверхностями, входящими в состав объекта. Так как объект состоит из примитивов, сложно комбинирующихся между собой, то на первом этапе следует определить пересечения луча с поверхностью каждого примитива. Если примитив содержит К поверхностей вида  $f_k(X, Y, Z) = 0$ , где  $k=1(1)K$ , то необходимо К раз решить систему вида:

$$\begin{cases} (X - X_B)/(X_{\text{sun}} - X_B) = (Y - Y_B)/(Y_{\text{sun}} - Y_B) = (Z - Z_B)/(Z_{\text{sun}} - Z_B); \\ f_k(X, Y, Z) = 0. \end{cases}$$

Среди множества точек-решений отбирают действительные, а затем среди последних – ближайшую точку к точечному источнику и одновременно лежащую на поверхности объекта.

Заметим, что алгоритм определения факта затененности совершенно идентичен алгоритму определения видимости, но в последнем необходимо поменять точку центра проектирования на точку источника излучения, а точку центра рецептора на видимую точку. Если результатом работы алгоритма являются координаты видимой точки (т.е. ближайшей к центру проекции), то результатом работы алгоритма затененности являются координаты точки, ближайшей к источнику света.

Очевидно, что если видимая из рецептора точка  $(X_B, Y_B, Z_B)$  является ближайшей  $(X_{\text{Bsun}}, Y_{\text{Bsun}}, Z_{\text{Bsun}})$  к источнику света, то она освещена прямыми лучами, т.е. точка  $(X_B, Y_B, Z_B)$  освещена, если  $[X_B \ Y_B \ Z_B] = [X_{\text{Bsun}} \ Y_{\text{Bsun}} \ Z_{\text{Bsun}}]$ .

Когда же видимая рецептором точка не является ближайшей к источнику света, то она затенена, т.е.  $(X_B, Y_B, Z_B)$  затенена, если  $[X_B \ Y_B \ Z_B] \neq [X_{\text{Bsun}} \ Y_{\text{Bsun}} \ Z_{\text{Bsun}}]$ .

Проведя подобный комплекс вычислений для всех  $I \times J$  рецепторов, удается получить информацию о координатах  $(X_B, Y_B, Z_B)_{ij}$  видимой из каждого рецептора точки, о номере примитива, номере функции поверхности, которой видимая точка принадлежит, и признак затененности-освещенности.

Заметим, что при неизменном ракурсе и вариации положения источника света нет необходимости каждый раз вновь определять видимые точки. Координаты видимых точек в этом случае остаются неизменными, изменяется только факт их затенения-освещения.

### 3.4. ИЗОБРАЖЕНИЕ ОСНОВНЫХ ПРИМИТИВОВ МЕТОДОМ ТРАССИРОВАНИЯ ЛУЧЕЙ

Алгоритм синтеза изображений требует конкретного описания правил изображения некоторых стандартных примитивов. К последним можно отнести многоугольники, многогранники, шары и эллипсоиды, параболоиды, цилиндры, конусы; поверхности типа экструзий; бикубические участки. В § 3.4.1–3.4.4 подробно рассмотрены правила обработки примитивов на основе плоских многоугольников, многогранников, поверхностей второго порядка, бикубических поверхностей. Напомним, что поверхности типа экструзий образуются вращением некоторой ломаной линии вокруг заданной оси или с помощью параллельного переноса ломаной вдоль некоторой прямой. В первом случае поверхность представляют в виде усеченных конусов, соприкасающихся торцами, а во втором – в виде смежно расположенных многоугольников. В этом изложении примитивы такого вида специально не рассматриваются, так как могут быть составлены из независимых комбинационных частей. В [95, 74] также могут быть найдены конкретные правила изображения основных примитивов методом трассирования лучей.

#### 3.4.1. ПЛОСКИЕ МНОГОУГОЛЬНИКИ

Объект, описываемый только многоугольниками, может быть преобразован в изображение несколькими методами: Робертса, сканирующей строки, прямым, обратным трассированием и др. Самым быстродействующим для таких объектов является метод сканирующей строки. Однако если объект наряду с плоскостями содержит поверхности более высоких порядков и требуется высокое качество изображения, то лучше применять метод трассирования лучей. Этот подход имеет специфические особенности.

Пусть некоторый пространственный объект, составленный из сплошных выпуклых объемных примитивов, включает и несколько плоских участков. Рассмотрим такой участок отдельно. В объектной системе плоскость, которой он принадлежит, может быть описана уравнением

$$AX+BY+CZ+D = 0 \quad (3.4.1)$$

или в матричном виде  $[X \ Y \ Z \ 1] P = 0$ , где  $X, Y, Z$  – координаты любой точки на плоскости;  $P = [A \ B \ C \ D]^T$ .

Иногда плоский участок задают тремя точками при условии, что они не принадлежат одной прямой. По этой информации можно восстановить уравнение плоскости (3.4.1) как это показано в программе PL3T(см. приложение).

Уравнение (3.4.1) описывает бесконечно простирающуюся плоскость. Реальные плоские детали ограничены замкнутой линией. Рассмотрим наиболее простой с вычислительной позиции случай, когда форма линии – многоугольник. Примерами таких плоских примитивов являются крыло самолета, поверхность стола, обложка книги и т.д. (рис. 3.4.1).

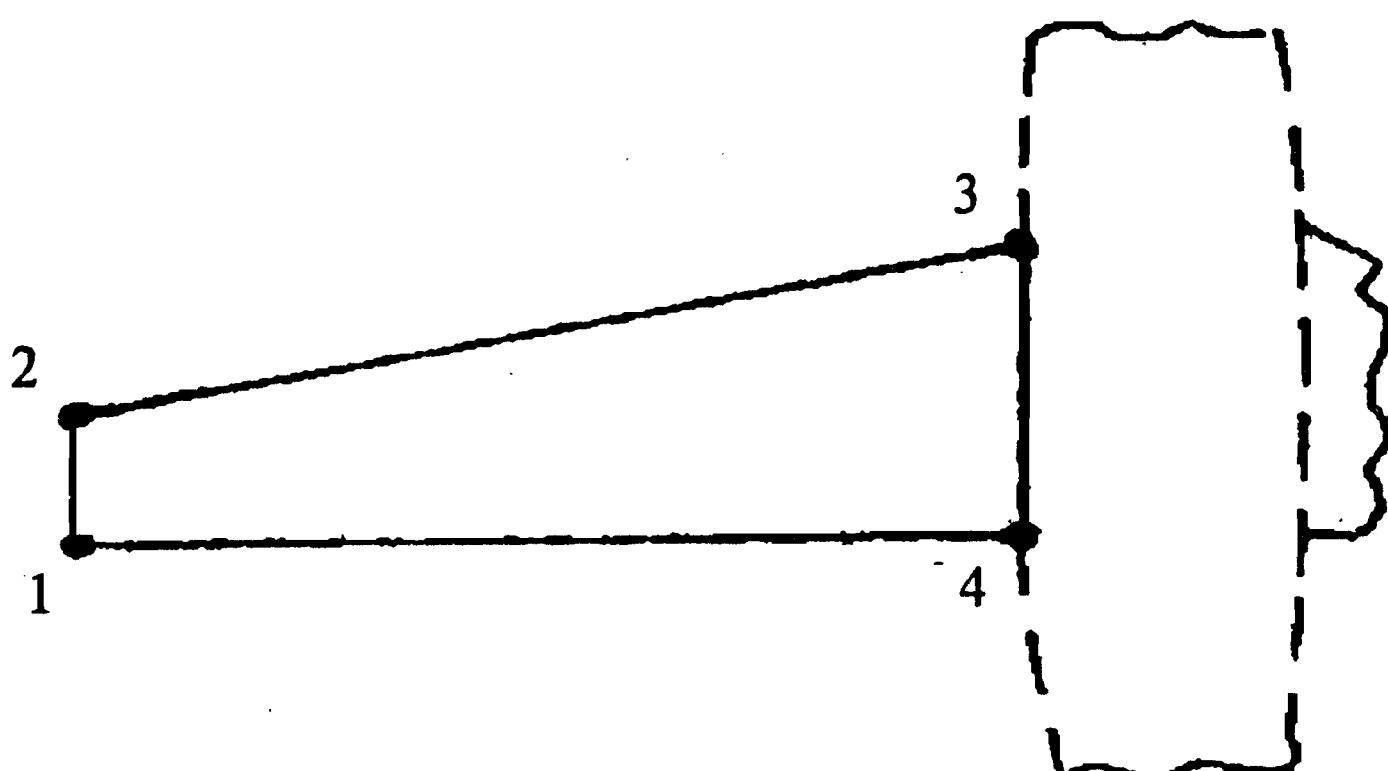


Рис. 3.4.1. Крыло самолета – пример плоского примитива

Границные точки такого многоугольника оцифруем однообразно по (или против) часовой стрелке и запишем их координаты в матрицу

$$\Lambda = \begin{bmatrix} X_1 & Y_1 & Z_1 \\ \vdots & \vdots & \vdots \\ X_N & Y_N & Z_N \end{bmatrix},$$

где  $N$  – количество вершин многоугольника,  $N \geq 3$ .

Матрицы параметров уравнения плоскости  $P$  и матрица граничных точек  $L$  однозначно задают плоский участок объекта.

В соответствии с методом трассирования лучей необходимо найти пересечение прямой с данным примитивом. Координаты такого пересечения находят решением системы

$$\begin{cases} (X - X_F) / (X_F - X_{ij}) = (Y - Y_F) / (Y_F - Y_{ij}) = (Z - Z_F) / (Z_F - Z_{ij}); \\ AX + BY + CZ + D = 0, \end{cases}$$

где  $X_F, Y_F, Z_F$  – координаты точки фокуса F в системе XYZ;  $X_{ij}, Y_{ij}, Z_{ij}$  – координаты ij-того рецептора в системе XYZ.

Отдельный световой луч и плоскость могут быть ориентированы друг относительно друга произвольно, поэтому возможны три исхода совместного решения: отсутствие решений, бесконечное множество решений и единственное решение. Решения отсутствуют при параллельности, но не совпадении прямой и плоскости, другими словами, при перпендикулярности вектора нормали  $N = iA + jB + kC$  к плоскости направляющего вектора  $L = i(X_F - X_{ij}) + j(Y_F - Y_{ij}) + k(Z_F - Z_{ij})$  светового луча, т.е., при выполнении условия

$$[N L] \cap [AX_F + BY_F + CZ_F + D \neq 0], \quad (3.4.2)$$

где  $i, j, k$  – орты осей OX, OY, OZ соответственно;  $\cap$  – знак логического умножения событий.

Физически выполнение этого условия означает невидимость поверхности из данного рецептора  $(X_{ij}, Y_{ij}, Z_{ij})$ , так как луч проходит мимо плоскости.

Бесконечное множество решений возникает при условии совпадения светового луча и плоскости, то есть, при одновременном выполнении условий параллельности и наличия общей точки:

$$[N L] \cap [AX_F + BY_F + CZ_F + D = 0], \quad (3.4.3)$$

Так как информация о поверхности, которой луч касается, но не пересекает, остается неопределенной для присмника света, то справедливо интерпретировать последнее условие как невидимость тонкой поверхности. Объединяя условия (3.4.2), (3.4.3) как условия невидимости, можно установить, что прямая не пересекается с плоскостью, если выполняется условие  $NL = 0$ .

В противном случае решение единственное:

$$Z_\Pi = Z_F + (Z_F - Z_{ij})(-AX_F - BY_F - CZ_F - D) / [A(X_F - X_{ij}) + B(Y_F - Y_{ij}) + C(Z_F - Z_{ij})];$$

$$X_\Pi = (Z_\Pi - Z_F)(X_F - X_{ij}) / (Z_F - Z_{ij}) + X_F;$$

$$Y_\Pi = (Z_\Pi - Z_F)(Y_F - Y_{ij}) / (Z_F - Z_{ij}) + Y_F,$$

где  $(X_\Pi, Y_\Pi, Z_\Pi)$  – координаты пересечения луча и плоскости.

В программах PLOS, DIRECT (см. приложение) приведен пример реализации этого алгоритма на Турбо Паскале.

Приведенное решение получено для бесконечной плоскости. Так как плоский участок объекта ограничен выпуклым многоугольником, то полученное ранее единственное решение необходимо оценить на принадлежность внутренней области этого многоугольника. В объектной системе координат уже заданы координаты крайних точек в виде матрицы  $L$  и получены координаты точки пересечения  $T = [X_{\Pi} \ Y_{\Pi} \ Z_{\Pi}]$ . Так как все эти точки принадлежат одной и той же плоскости, то для упрощения вычислений можно отказаться от третьей координаты и пользоваться информацией о двух измерениях. Спроектируем точки, описанные в строках матрицы  $L$  и матрице  $T$ , например, на плоскость XOY. Тогда размерность этих матриц сократится на один столбец (рис. 3.4.2):

$$L = \begin{bmatrix} X_1 & Y_1 \\ \vdots & \vdots \\ X_N & Y_N \end{bmatrix}, \quad T_2 = [X_{\Pi} \ Y_{\Pi}].$$

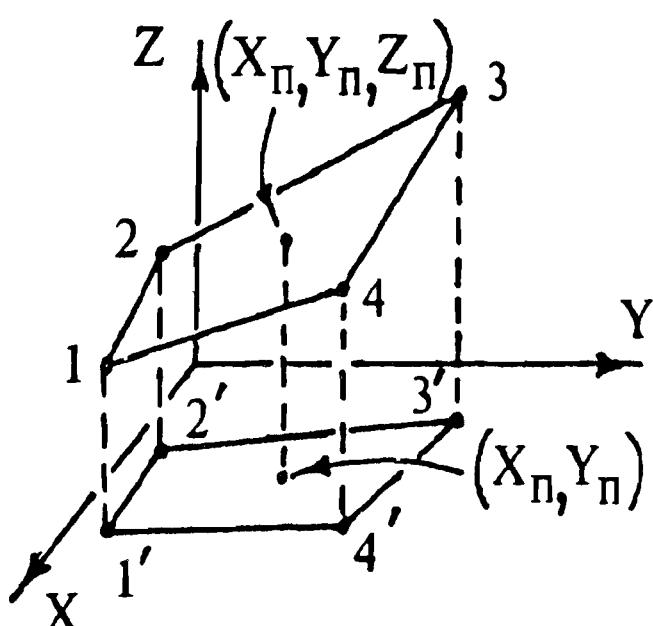


Рис. 3.4.2. Проецирование многоугольника на координатную плоскость

Из проективной геометрии [51] известно, что если точка  $(X_{\Pi}, Y_{\Pi}, Z_{\Pi})$  принадлежит внутренней части многоугольника (1-4), то проекция  $(X_{\Pi}, Y_{\Pi})$  этой точки также принадлежит внутренней части проективного многоугольника (1'-4') (рис. 3.4.2), т.е. если точка и многоугольник лежат в одной плоскости, а проекция многоугольника не вырождается в отрезок прямой, то оценку принадлежности точки этому многоугольнику можно проводить в двухмерном пространстве (в данном случае на XYZ).

Известно [39], что прямая  $f(X, Y)$  на плоскости XOY делит

последнюю на две полуплоскости, причем в одной из них  $f(X, Y) > 0$ , в другой  $f(X, Y) < 0$  и, естественно, на самой прямой  $f(X, Y) = 0$ . Воспользуемся этим правилом для нахождения такой формы задания функции  $f_i(X, Y)$  для прямой, соединяющей две соседние вершины многоугольника, при которой любая точка  $(X_{\text{вн}}, Y_{\text{вн}})$ , принадлежащая внутренней части многоугольника, давало бы положительное значение:  $f_i(X_{\text{вн}}, Y_{\text{вн}}) > 0$  – для любой  $i$ -й стороны многоугольника  $i = 1(1)N$ . Для этого сначала построим уравнения прямых, проходящих через соседние крайние точки многоугольника в однозначной последовательности (все по часовой или все против часовой), т. е. первая прямая проходит через первую и вторую точки; вторая – через вторую и третью, ..., последняя – через последнюю  $N$ -ю и первую точки (рис. 3.4.3). Уравнение  $i$ -й прямой для  $i < N$  будет иметь вид

$$f_i(X, Y) = (X - X_i)(Y_{i+1} - Y_i) - (X_{i+1} - X_i)(Y - Y_i) = 0. \quad (3.4.4)$$

Для  $N$ -ной прямой в последней записи следует заменить индекс  $i$  на  $N$ , а  $i+1$  на первый.

Напомним, что для определения факта принадлежности точки пересечения  $(X_{\Pi}, Y_{\Pi})$  внутренней области многоугольника необходимо так установить знак функции  $f_i(X, Y)$  для всех  $i$ , чтобы значение этой функции во внутренней области многогранника было положительным. Априорно в этом нет уверенности, поэтому следует проверить экспериментально факт положительности внутренней области для всех  $N$  прямых, а в случае отрицательности – изменить знак  $f_i(X, Y)$  на противоположный. Заметим, что для всех  $i < N$  точка под номером  $i+2$  должна находиться в положительной полуплоскости относительно уравнения  $i$ -й прямой, а для  $N$ -й прямой такой точкой является точка под номером 1. Тогда вычислим значение функции  $f_1(X_3, Y_3)$  и изменим знак функции  $f_1(X, Y)$  на противоположный, если  $f_1(X_3, Y_3) < 0$ , т. е. проведем операцию  $f_1(X, Y) = f_1(X, Y)\text{sign}(f_1(X_3, Y_3))$ . Таким образом, удалось добиться положительности внутренней области для первой прямой.

Для остальных  $N-1$  прямых нет необходимости опытным путем оценивать их знак во внутренней области, так как они относятся к внутренней области точно так, как и первая прямая. Т. е., не только первая, но и все  $N$  ( $i=1(1)N$ ) прямые должны быть преобразованы по правилу  $f_i(X, Y) = f_1(X, Y)\text{sign}(f_1(X_3, Y_3))$  или,

другими словами, при  $f_1(X_3, Y_3) < 0$  уравнение (3.4.4) изменяется:  
 $f_i(X, Y) = (X - X_i)(Y_i - Y_{i+1}) - (X_i - X_{i+1})(Y - Y_i) = 0.$

Распространение "опыта" первой прямой на все остальные оправдано, так как может быть достаточно просто доказано следующее положение. Если задан замкнутый плоский выпуклый контур из  $N$  отрезков прямых, направляющие векторы которых однозначно расположены по часовой (или однозначно против часовой) стрелке при обходе контура, то для любых двух  $i$ -го и  $j$ -го ребер, где  $i \in [1, N]$ ,  $j \in [1, N]$  справедливо  $\text{sign}(f_i(X, Y)) = \text{sign}(f_j(X, Y))$ .

Введем знак  $\text{ind}$ , означающий  $i$  или  $j$ . Тогда справедливо  
 $f_{\text{ind}}(X, Y) = (X - X_{\text{ind}})(Y_{\text{ind}+1} - Y_{\text{ind}}) - (Y - Y_{\text{ind}})(X_{\text{ind}+1} - X_{\text{ind}})$ , или  
 $f_{\text{ind}}(X, Y) = (X - X_{\text{ind}+1})(Y_{\text{ind}+1} - Y_{\text{ind}}) - (Y - Y_{\text{ind}+1})(X_{\text{ind}+1} - X_{\text{ind}})$ .

Приведем доказательство. Рассмотрим участок выпуклого замкнутого многоугольника. Этот участок состоит из трех последовательных отрезков (ребер) под номерами  $I, I+1, I+2$  и имеет соответственно угловые точки (вершины)  $i, i+1, i+2, i+3$  (рис. 3.4.4). Так как многоугольник выпуклый, то  $(\cdot)i+2$  и  $(\cdot)i+3$  лежат по одну сторону плоскости от прямой  $I$ , следовательно

$$\text{sign } f_I(i+2) = \text{sign } f_I(i+3). \quad (3.4.5)$$

Аналогично  $(\cdot)i+3$  и  $(\cdot)i$  лежат по одну сторону плоскости от отрезка  $I+1$ , и тогда

$$\text{sign } f_{I+1}(i) = \text{sign } f_{I+1}(i+3). \quad (3.4.6)$$

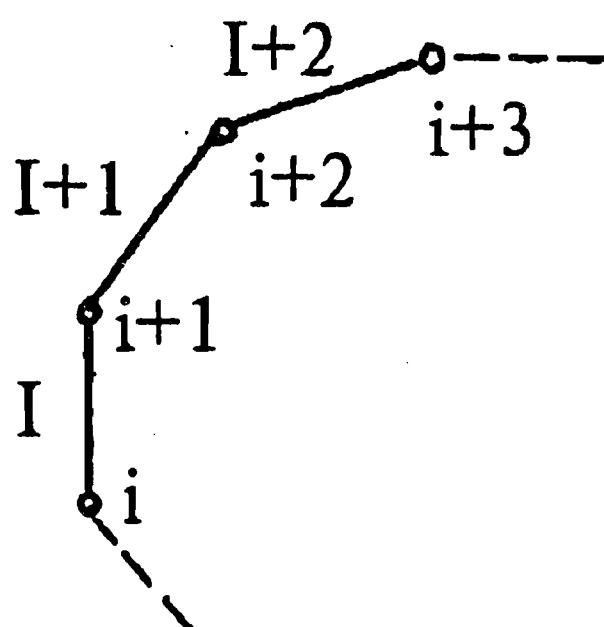


Рис. 3.4.3. Участок выпуклого многоугольника

Докажем сначала, что точка  $i+3$  лежит в полуплоскости одного знака относительно отрезков  $I$  и  $I+1$ , т. е. что справедливо

$$\text{sign } f_I(i+3) = \text{sign } f_{I+1}(i+3).$$

В соответствии со значением  $f_{\text{ind}}(X, Y) \text{ sign } f_I(i+2) = X_{i+2}Y_{i+1} - X_{i+2}Y_i - X_iY_{i+1} + X_iY_i - Y_{i+2}X_{i+1} + Y_{i+2}X_i + Y_iX_{i+1} - X_iY_i$ ;  $\text{sign } f_{I+1}(i) = X_iY_{i+2} - X_iY_{i+1} - X_{i+1}Y_{i+2} + X_{i+1}Y_{i+1} - Y_iX_{i+2} + Y_iX_{i+1} + Y_{i+1}X_{i+2} - Y_{i+1}X_{i+1}$ . Сравнивая правые части последних выражений, получаем  $\text{sign } f_{I+1}(i) = \text{sign } f_I(i+2)$ , тогда из (3.4.5), (3.4.6) следует, что  $\text{sign } f_I(i+3) = \text{sign } f_{I+1}(i+3)$ .

Распространяя по аналогии полученные выводы для любой другой тройки последовательных отрезков, убеждаемся в справедливости приведенного выше положения. Оно может быть сформулировано в другой форме: знак внутренней области замкнутой зоны, заданной в однозначной последовательности нумерации вершин (все по или против часовой стрелки), определяется знаком функции, описывающей любое ребро зоны.

Процедура AUTOZON (см. приложение) реализует достижение положительности значения внутренней области сразу для нескольких выпуклых многоугольников путем оценки знака функции, описывающей первое по порядку ребро внутри многоугольника. Обработка многих многоугольников в одной процедуре связана с тем, что обычно в сцене присутствуют несколько плоских участков с одинаковым числом ребер.

После такого преобразования факт принадлежности точки  $(X_\Pi, Y_\Pi)$  внутренней области многоугольника имеет место при выполнении условия

$$\sum_{i=1}^N \text{sign } [(f_i(X_\Pi, Y_\Pi))] = N.$$

Соответственно точка лежит вне фигуры, если найдется хотя бы одна такая  $k$ -я прямая ( $k \in [1, N]$ ), что  $f_k(X_\Pi, Y_\Pi) \leq 0$ .

Приведенные рассуждения справедливы для случая проецирования плоского примитива на плоскость XOY, хотя они работоспособны при проецировании на любые другие координатные плоскости, лишь бы проекция многоугольника не вырождалась в отрезок прямой. Исходные данные о плоском примитиве—многоугольнике в трехмерном пространстве задаются в виде коэффициентов уравнения плоскости  $P = [A \ B \ C \ D]^T$  и одной из трех матриц:

$$\Lambda_1 = \begin{bmatrix} X_1 & Y_1 \\ \vdots & \vdots \\ X_N & Y_N \end{bmatrix}, \text{ или } \Lambda_2 = \begin{bmatrix} Y_1 & Z_1 \\ \vdots & \vdots \\ Y_N & Z_N \end{bmatrix}, \text{ или } \Lambda_3 = \begin{bmatrix} Z_1 & X_1 \\ \vdots & \vdots \\ Z_N & X_N \end{bmatrix}.$$

В более компактном виде указывают двумерную матрицу  $\Lambda$  размера Nx2 и индикатор IND, который принимает значения 1, 2, 3. Если IND = 1, то в матрице  $\Lambda$  задана информация о координатах XY; если IND=2 – YZ; если IND=3 – ZX.

Пользоваться двухмерным представлением информации о координатах крайних точек многоугольника достаточно удобно, так как обычно ввод исходных данных производится с чертежа в трех видах, совпадающих с плоскостями XY, YZ, ZX. При вырождении проекции многоугольника в отрезок прямой информация о координатах крайних точек в этой проекции не используется.

Таким образом, при любой ориентации светового луча и многоугольника удается определить факт их пересечения. В программах KRILO, YNZONA (см. приложение) показаны примеры реализации изложенных принципов пересечения прямой и многоугольника на Турбо Паскале. Подпрограмма YNZONA использует в качестве входных параметров матрицу проекций направляющих векторов-ребер, которая формируется путем предварительного обращения к процедуре AVTOZON (см. приложение).

Для вычисления освещенности в видимой точке необходимо знать направление и ориентацию нормального вектора к плоскости. Известно, что нормальный вектор к плоскости записывается в виде  $\mathbf{N}=i\mathbf{A}+j\mathbf{B}+k\mathbf{C}$ . В соответствии с [44] этот вектор направлен в сторону положительного полупространства относительно плоскости, т.е., если X,Y,Z – координаты конца нормального вектора  $\mathbf{N}$ , то  $AX+BY+CZ>0$ . Так как участок плоскости является бесконечно тонким примитивом, т. е. не имеет внутренней области пространства, то при пересечении луча с плоскостью возникает неопределенность: какой же стороне плоскости принадлежит точка пересечения? Определение стороны весьма важно, так как при одном источнике освещения всегда выполняется следующий факт: если одна сторона плоскости освещена, то другая – в тени. Приведем один из возможных алгоритмов разрешения неопределенности, который может быть назван *методом адаптивной нормали*.

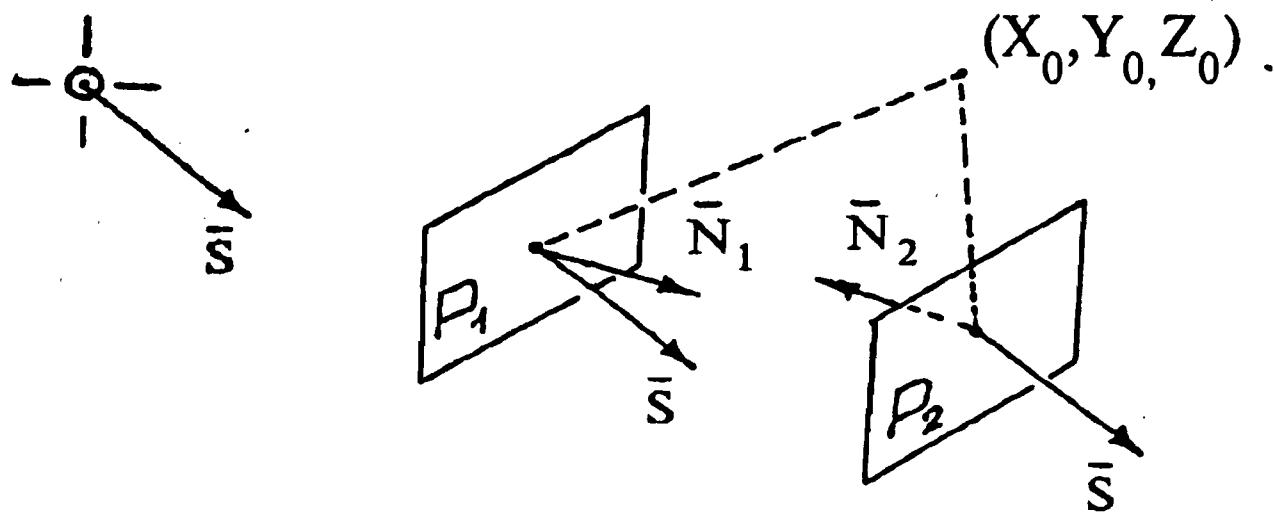


Рис. 3.4.4. Определение факта освещенности или затененности плоскостей

Плоскость  $P_1$  находится в тени, если источник света  $S(X_{\text{sun}}, Y_{\text{sun}}, Z_{\text{sun}})$  и приемник  $(X_0, Y_0, Z_0)$  расположены по разные стороны от плоскости (рис. 3.4.4), т.е. при выполнении условия  $\text{sign}(AX_0 + BY_0 + CZ_0 + D) \neq \text{sign} (AX_{\text{sun}} + BY_{\text{sun}} + CZ_{\text{sun}} + D)$ . При его невыполнении плоскость  $P_2$  освещена прямыми лучами. Направим нормаль для этого случая в сторону от плоскости, где находится приемник и источник:  $N' = N \text{ sign} (AX_0 + BY_0 + CZ_0 + D)$  или  $N' = -N \text{ sign} (AX_{\text{sun}} + BY_{\text{sun}} + CZ_{\text{sun}} + D)$ . При расчете освещенности в области тени от слабых или распределенных источников ориентируем нормаль в ту сторону по отношению к плоскости, где находится приемник:  $N' = N \text{ sign} (AX_0 + BY_0 + CZ_0 + D)$ ; это позволяет определять факт затененности-освещенности плоскости, а также, пользуясь правилами расчета освещенности, производить яркостное заполнение участков изображения, на которые проецируется плоскость.

В процессе конструирования объектов и получения изображений возникает необходимость пространственного перемещения частей объекта, в том числе и плоских многоугольников. Перемещение многоугольника в пространстве означает изменение координат его крайних точек и коэффициентов уравнения плоскости. Пусть в системе  $XYZ$  плоскость задана матрицей  $P = [A \ B \ C \ D]^T$ , крайние точки — матрицей

$$\Lambda_4 = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ X_N & Y_N & Z_N & 1 \end{bmatrix},$$

которая образуется путем присоединения к матрице  $\Lambda$  столбца

единиц. Необходимо определить координаты той же плоскости  $P'$  и координаты крайних точек —  $\Lambda'_4$ , но уже в другой системе координат  $X'Y'Z'$ . Данная задача эквивалентна перемещению многоугольника внутри одной и той же системы координат. Известно [60, 19], что для любой точки, имеющей координаты  $(X, Y, Z)$  и  $(X', Y', Z')$  в системах  $XYZ$  и  $X'Y'Z'$  соответственно, выполняется соотношение

$$[X \ Y \ Z \ 1] W = [X' \ Y' \ Z' \ 1], \quad (3.4.7)$$

а для любой плоскости, издаваемой в указанных координатных системах соответственно матрицами  $P = [A \ B \ C \ D]^T$  и  $P' = [A' \ B' \ C' \ D']$  справедливо

$$P = WP. \quad (3.4.8)$$

Матрица  $W$  в формуле (3.4.7) имеет вид

$$W = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -X_0 & -Y_0 & -Z_0 & 1 \end{bmatrix} \begin{bmatrix} \cos\alpha & 0 & \sin\alpha & 1 \\ 0 & 1 & 0 & 0 \\ -\sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \\ \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \sin\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\psi & -\sin\psi & 0 & 0 \\ \sin\psi & \cos\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3.4.9)$$

где  $X_0, Y_0, Z_0$  — координаты точки  $O'$  в системе  $XYZ$ ;  $\alpha, \theta, \psi$  — углы поворота системы  $XYZ$  последовательно вокруг осей  $OY$ ,  $OX$ ,  $OZ$  соответственно до совпадения с осями системы  $X'Y'Z'$  (правое вращение). Тогда можно получить  $P' = W^{-1}P; \Lambda'_4 = \Lambda_4 W$ .

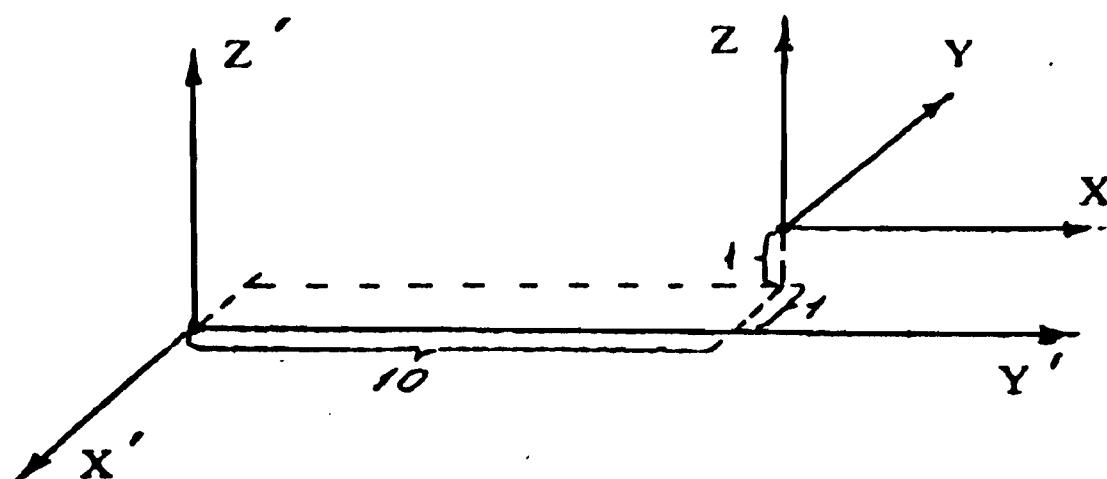


Рис. 3.4.5. Пример взаимного положения систем координат

При перемещении многоугольника происходит изменение коэффициентов уравнения плоскости и принципиально возможно неуправляемое изменение ориентации нормали, что может

привести к грубым ошибкам при вычислении освещенности в тонах. На самом деле этого не происходит, что может быть доказано следующим образом. Покажем, что если в системе координат  $X'Y'Z'$  точка  $A(x',y',z')$  удовлетворяет для плоскости  $A'X'+B'Y'+C'Z'+D'=0$  нестрогому неравенству  $A'x'+B'y'+C'z'+D' \geq 0$ , то знак неравенства сохраняется при перемещении и вращении системы координат по правилу (3.4.7), т. е.  $AX+BY+CZ+D \geq 0$ , где  $X,Y,Z$  – координаты точки  $x',y',z'$  в новой системе координат  $XYZ$ ;  $A,B,C,D$  – координаты уравнения плоскости в системе  $XYZ$ . Другими словами, знак полупространства относительно плоскости инвариантен к перемещению системы координат.

Действительно, ранее было показано, что справедливы выражения:

$$[X \ Y \ Z \ 1] = [x' \ y' \ z' \ 1] W^{-1}; \quad (3.4.10)$$

$$[A \ B \ C \ D]^T = W [A' \ B' \ C' \ D'] \quad (3.4.11)$$

Из условия доказываемого положения следует, что произведение  $[x' \ y' \ z' \ 1][A' \ B' \ C' \ D']^T \geq 0$ . Проверим справедливость сохранения закона неравенства при изменении системы координат, т.е. оценим знак  $[X \ Y \ Z \ 1][A \ B \ C \ D]^T$ . Подставив вместо каждой из матриц-сомножителей последнего выражения их эквиваленты из (3.4.9) и (3.4.10), получим  $[X \ Y \ Z \ 1][A \ B \ C \ D] = = [x' \ y' \ z' \ 1] W^{-1}; W [A' \ B' \ C' \ D'] = [x' \ y' \ z' \ 1][A' \ B' \ C' \ D']$ . То есть значение функции  $f$ , описывающей плоскость уравнением  $f = 0$ , в некоторой точке пространства не изменяется при перемещении и плоскости, и точки по одному закону. Поэтому направление нормали сохраняется неизменным.

### 3.4.2. ВЫПУКЛЫЕ МНОГОГРАННИКИ

Выпуклые многогранники часто используются в качестве примитивов в машинной графике. Невыпуклые фигуры отдельно рассматриваться не будут, поскольку они всегда могут быть разделены на несколько смежных выпуклых. Пусть в системе координат объекта  $XYZ$  многогранник из  $K$  плоских граней задан посредством матрицы

$$R = \begin{bmatrix} A_1 & \dots & A_k & \dots & A_K \\ B_1 & \dots & B_k & \dots & B_K \\ C_1 & \dots & C_k & \dots & C_K \\ D_1 & \dots & D_k & \dots & D_K \end{bmatrix},$$

где  $A_k, B_k, C_k, D_k$  – коэффициент уравнения для  $k$ -й плоскости,  $k=1(1)K$ ,  $K \geq 4$ .

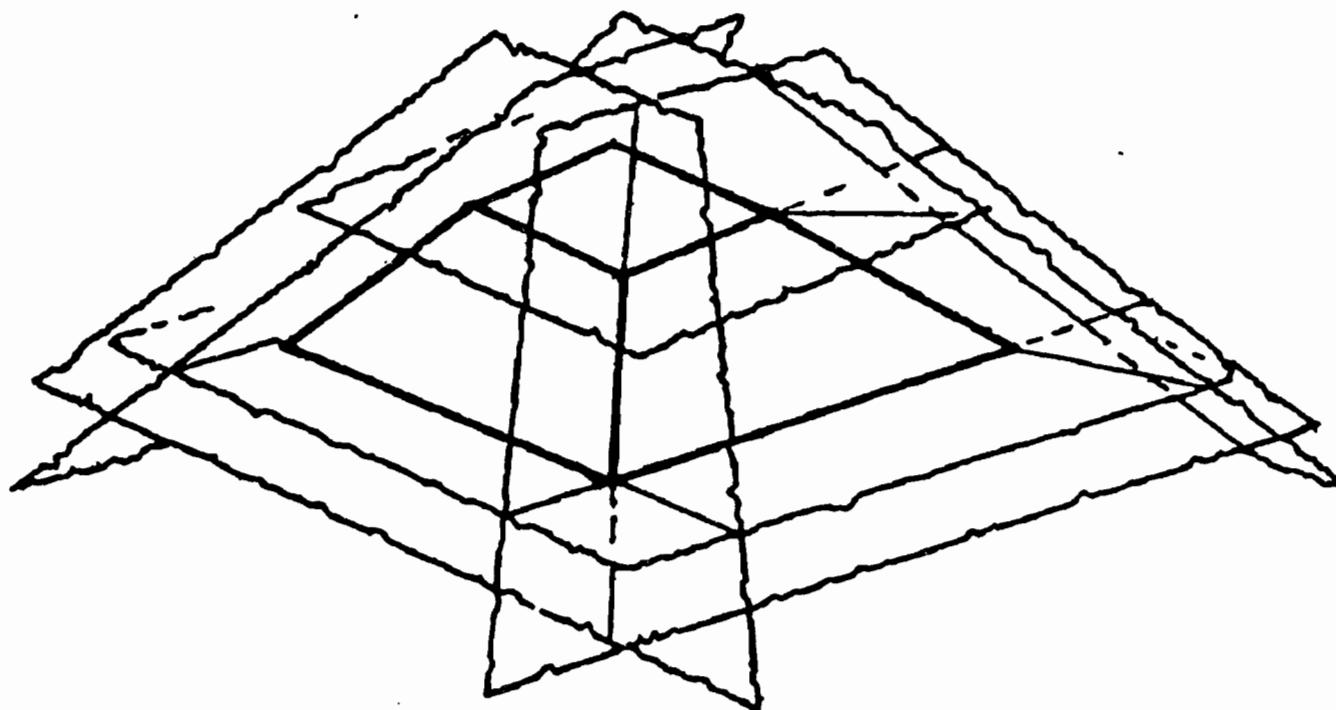


Рис. 3.4.6. Выпуклый многогранник и образующие его плоскости

Плоскости, описывающие грани, простираются в бесконечность (рис. 3.4.6). Для того чтобы выделить из этого набора плоскостей центральное ядро – многогранник, зададим последнему особые свойства по сравнению с наружным относительно него полупространством. В соответствии с договоренностью о положительности внутренней области многогранника выберем знак функции каждой плоскости  $f_k(X, Y, Z)$  таким, чтобы для любой точки  $(X_B, Y_B, Z_B)$ , лежащей на поверхности или внутри многогранника, значение этой функции было неотрицательным:  $f_k(X_B, Y_B, Z_B) \geq 0 \forall k$ , где  $f_k(X, Y, Z) = A_k X + B_k Y + C_k Z + D_k$ . Тогда для каждой плоскости положительное полупространство включает примитив, а отрицательное находится снаружи примитива.

В соответствии с методом трассирования лучей необходимо найти координаты точек пересечения светового луча, проходящего через точки  $(X_F, Y_F, Z_F)$  – центр проекции и  $(X_{ij}, Y_{ij}, Z_{ij})$  – центр рецептора, с многогранником. Так как в общем случае прямая пересекает все плоскости, то сначала найдем решение  $K$  систем вида

$$\begin{cases} (X - X_F)/(X_F - X_{ij}) = (Y - Y_F)/(Y_F - Y_{ij}) = (Z - Z_F)/(Z_F - Z_{ij}); \\ A_k X + B_k Y + C_k Z + D_k = 0. \end{cases}$$

Решение каждой такой системы осуществляется в соответствии с методами, изложенными в § 3.4.1. С учетом того, что отдельные

$K'$  ( $K' < K$ ) плоскостей могут не иметь пересечения с прямой, получаем в результате  $K-K'$  решений, организованных в матрицу

$$S = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ X_g & Y_g & Z_g & 1 \\ \vdots & \vdots & \vdots & \vdots \\ X_{K-K'} & Y_{K-K'} & Z_{K-K'} & 1 \end{bmatrix},$$

где  $X_g, Y_g, Z_g$  – координаты пересечения  $g$ -й плоскости с прямой,  $g=1(1)K-K'$ . Условие отсутствия пересечений прямой и плоскости или факта бесконечного множества решений, что будем также интерпретировать как невидимость плоскости лучом, эквивалентно условию перпендикулярности направляющего вектора прямой и нормали к плоскости:  $A_k(X_F - X_{ij}) + B_k(Y_F - Y_{ij}) + C_k(Z_F - Z_{ij}) = 0$ . Для того чтобы в последующем сохранить информацию о соответствии точек пересечения с плоскостями пересечения, удалим из матрицы  $R$  соответствующие  $K'$  столбцов:

$$R_H = \begin{bmatrix} A_1 & \dots & A_g & \dots & A_{K-K'} \\ B_1 & \dots & B_g & \dots & B_{K-K'} \\ C_1 & \dots & C_g & \dots & C_{K-K'} \\ D_1 & \dots & D_g & \dots & D_{K-K'} \end{bmatrix},$$

где  $g$  – номер плоскости, которой принадлежит точка  $(X_g, Y_g, Z_g)$ . Эта информация необходима для вычисления нормали к видимой точке.

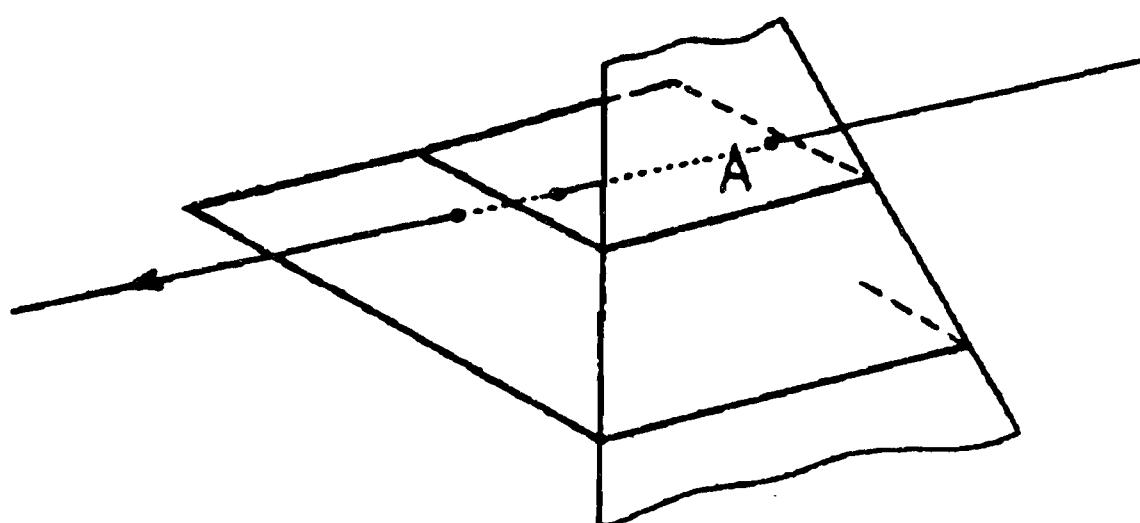


Рис. 3.4.7. Пересечение луча с продолжением грани примитива

Некоторые точки пересечения из приведенных в последней матрице  $S$  хотя и принадлежат каким-то плоскостям, но могут не принадлежать поверхности многогранника. Такие точки лежат за

пределами примитива и следовательно должны быть устранины из последующего анализа, как это необходимо сделать с точкой А на рис.3.4.8. Осуществим операцию оценки знака функций всех плоскостей последовательно для всех точек. Введем матрицу  $D = \text{sign}(S R)$  размера  $(K-K')K$ . Если выполняется условие

$$\bigcap_{k=1}^K (D(g,k) \geq 0),$$

то в матрице  $S$  g-я точка действительно принадлежит поверхности примитива. Если же найдется хотя бы одна g-я плоскость, для которой  $D(g,k) < 0$ , то g-я точка не принадлежит многограннику.

При увеличении числа граней принцип работы алгоритма не меняется. В результате отсеивания всех возможных  $l$  точек, лежащих вне тела примитива, матрица  $S$  сокращается еще на  $l$  строк. В последней матрице сосредоточились все действительные решения, которых в общем случае может быть больше двух из-за возможного пересечения прямой с вершиной многогранника. Устранив всех двойников одной и той же точки, получим в результате только два решения:  $(X_1, Y_1, Z_1)$  и  $(X_2, Y_2, Z_2)$ . Внешняя нормаль к поверхности многогранника в этих точках имеет вид  $N_1 = -iA_{\lambda_1} - jB_{\lambda_1} - kC_{\lambda_1}$ ;  $N_2 = -iA_{\lambda_2} - jB_{\lambda_2} - kC_{\lambda_2}$  где  $\lambda_1, \lambda_2$  – номера плоскостей, которым и принадлежат точки решения.

Эти плоскости определяются путем одновременного удаления столбцов из матрицы  $R_H$  под номерами, равными номерам строк матрицы  $S$ , удаляемыми как лежащие вне примитива или как двойники действительных решений.

В программах KOROB и SOME (см. приложение) приведены примеры решений пересечения прямой и шестигранника. Подпрограмма KOROB определяет возможные пересечения со всеми шестью плоскостями и среди них выбирает ближайшее решение – точку к центру проекции. Оценка расстояния между точками производится в объектной системе координат по одной-единственной координате  $Z$ , что является частным случаем общего алгоритма поиска ближайших точек. Выбор ближайшей точки осуществляется с помощью подпрограммы SOME. Результат, полученный с помощью подпрограммы KOROB, показан на рис. 3.2.4. Для изображения такого простого объекта нет необходимости обязательно применять столь мощное средство как метод трассирования лучей, однако как только сцена становится сложной: генерируются тени, проявляются

криволинейные и прозрачные поверхности и другие эффекты, — применение этого метода становится оправданным. Если же сцена состоит только из выпуклых многогранников, то целесообразно использовать метод сканирующей строки или алгоритм Робертса (см. §4.2, 4.3).

Важной задачей в практике применения методов машинной графики является перемещение многогранника в объектном пространстве. Пусть многогранник задан исходно в правой прямоугольной системе координат  $X'Y'Z'$  матрицей  $R$ , тогда если новая система координат  $XYZ$  связана со старой  $X'Y'Z'$  матрицей преобразования  $W$ :  $[X \ Y \ Z \ 1]W = [X' \ Y' \ Z' \ 1]$ , то в системе  $XYZ$  новые параметры плоскостей, сведенные в матрицу

$$R' = \begin{bmatrix} A'_1 & \dots & A'_K \\ B'_1 & \dots & B'_K \\ C'_1 & \dots & C'_K \\ D'_1 & \dots & D'_K \end{bmatrix},$$

можно вычислить по правилу  $R' = W^{-1}R$ . Этот вывод вытекает из обобщения формулы для случая системы плоскостей.

Как на этапе определения яркости теневой точки, так и для определения принадлежности некоторой точки поверхности многогранника необходимо соблюдение правила ориентации нормалей по всем граням внутри фигуры. При переносе многогранника в объектном пространстве, что эквивалентно выбору новой системы координат, меняются значения коэффициентов плоскостей, а это может нарушить ориентацию нормали. Однако этого не происходит в силу сохранения знаков полупространства для каждой из плоскостей, образующих многогранник. Так, если в некоторой системе координат  $X'Y'Z'$  внутренняя область многогранника, который описан матрицей  $R'$ , положительна относительно каждой грани, т.е. для каждой точки  $(X', Y', Z')$ , заведомо лежащей внутри фигуры, справедливо  $[X' \ Y' \ Z' \ 1] R' = [a_1 \dots a_k \dots a_K]$ , где  $a_k$  — положительное число,  $k=1(1)K$ , то положительность внутренней области сохраняется при любом изменении системы координат.

Действительно, известно, что  $R' = W^{-1}R$ ;  $[X \ Y \ Z \ 1]W = [X' \ Y' \ Z' \ 1]$ ;  $[X \ Y \ Z \ 1] = [[X' \ Y' \ Z' \ 1]W^{-1}]$ , тогда можно получить  $[X \ Y \ Z \ 1]R = [[X' \ Y' \ Z' \ 1]W^{-1}WR'] = [X' \ Y' \ Z' \ 1]R' = [a_1 \dots a_k \dots a_K]$ . Многогранник можно перемещать и вращать относительно его

исходного положения с помощью преобразования координат, не опасаясь нарушения общего алгоритма.

### 3.4.3. КВАДРАТИЧНЫЕ ПОВЕРХНОСТИ

Вопросам обработки квадратичных поверхностей посвящена обширная литература [48, 59, 103, 107, 115, 131], что обусловлено, с одной стороны, относительной простотой их обработки, а с другой – богатством геометрических форм.

Уравнение функции вида  $f(X, Y, Z) = AX^2 + BY^2 + CZ^2 + 2DXY + 2EYZ + 2FZX + 2GX + 2HY + 2JZ + K = 0$  или в матричном виде

$$f(X, Y, Z) = [X \ Y \ Z \ 1] P [X \ Y \ Z \ 1]^T, \quad (3.4.12)$$

где  $P$  – матрица (2.2.3), описывает квадратичную поверхность. В составе примитива квадратичная поверхность может присутствовать самостоятельно (эллипсоид) или в комбинации с плоскостями и другими квадратичными поверхностями. Функции второго порядка являются знакоразделимыми, т.е. их значения в точках, лежащих по разные стороны от поверхности, имеют разные знаки. Это утверждение следует из того, что градиент скалярного поля существует в любой точке поверхности и не равен нулю. Воспользуемся этим свойством для создания примитива с положительным значением скалярного поля  $f(X, Y, Z)$  во внутренней области. Это состояние достигается выбором знака функции  $\pm f(X, Y, Z)$ . Таким образом, конкретизируем понятие примитива как сплошного объемного объекта, поверхность которого состоит из нескольких функций  $f_k(X, Y, Z)$ ,  $k=1(1)K$ , причем примитиву принадлежит то подпространство, отсекаемое поверхностью  $f_k(X, Y, Z)$ , в котором значение функции  $f_k$  положительно.

В соответствии с методом трассирования лучей найдем точки пересечения светового луча с поверхностью второго порядка (3.4.12). Прямую  $(X - X_F)/(X_F - X_{ij}) = (Y - Y_F)/(Y_F - Y_{ij}) = (Z - Z_F)/(Z_F - Z_{ij})$ , проходящую через две точки  $(X_F, Y_F, Z_F)$  и  $(X_{ij}, Y_{ij}, Z_{ij})$ , задаем в параметрическом виде:

$$X = X_F - lt; \quad Y = Y_F - mt; \quad Z = Z_F - nt, \quad (3.4.13)$$

где  $l = X_F - X_{ij}$ ;  $m = Y_F - Y_{ij}$ ;  $n = Z_F - Z_{ij}$ ;  $t$  – переменная.

Подставим выражения из (3.4.13) в (3.4.12):

$$f(t) = [X_F + lt \ Y_F + mt \ Z_F + nt \ 1] P [X_F + lt \ Y_F + mt \ Z_F + nt \ 1]^T.$$

Введем новые матрицы  $\Phi = [X_F \ Y_F \ Z_F \ 1]$  и  $\Pi = [1 \ m \ n \ 1]$ , с помощью которых можно получить [19]

$$f(t) = [\Phi + t\Pi]P[\Phi + t\Pi]^T = t^2 P \Pi^T + t(P \Phi^T + \Phi \Pi^T) + \Phi \Phi^T = 0.$$

Так как  $P \Phi^T$  и  $\Phi \Pi^T$  – скаляры, то нетрудно установить, что  $P \Phi^T = \Phi \Pi^T$ . Действительно,  $P \Phi^T = (\Phi^T)^T = (\Phi^T)^T P^T \Pi^T = \Phi \Pi^T$ . Тогда  $f(t) = t^2 P \Pi^T + 2t P \Phi^T + \Phi \Phi^T = 0$ .

Проверим дискриминант  $D = 4[(P \Phi)^2 - P \Pi^T \Phi \Phi^T]$  полученного квадратичного уравнения. Если  $D \leq 0$ , точки пересечения отсутствуют: при  $D < 0$  луч проходит мимо поверхности без пересечений, при  $D = 0$  луч скользит по ней, касаясь в одной точке, что следует интерпретировать как невидимость поверхности. При  $D > 0$  из решения квадратного уравнения получаем два корня:  $t_{1,2} = \{-P \Phi^T \pm [(P \Phi)^2 - P \Pi^T \Phi \Phi^T]^{0,5}\}/(P \Pi^T)$ , которым в трехмерном пространстве соответствуют координаты  $X_{1,2} = X_F + t_{1,2}$ ;  $Y_{1,2} = Y_F + m t_{1,2}$ ;  $Z_{1,2} = Z_F + n t_{1,2}$ . Проведя подобные вычисления со всеми другими поверхностями, слагающими примитив, получим множество точек, принадлежащих поверхностям. Однако не все они лежат на поверхности примитива, как, например, точка A на рис. 3.4.8. Для устранения таких ложных пересечений воспользуемся принципом положительности внутренней области примитива. Если световой луч пересекает какую-либо поверхность в точке  $(X_{\text{вне}}, Y_{\text{вне}}, Z_{\text{вне}})$  за пределами примитива, то обязательно найдется хотя бы одна  $k$  поверхность ( $k=1(1)K$ ), такая, что  $f_k(X_{\text{вне}}, Y_{\text{вне}}, Z_{\text{вне}}) < 0$ . Так как примитив является выпуклым, то истинных решений может быть только два: одно решение – точка принадлежит передней стенке примитива, второе – задней.

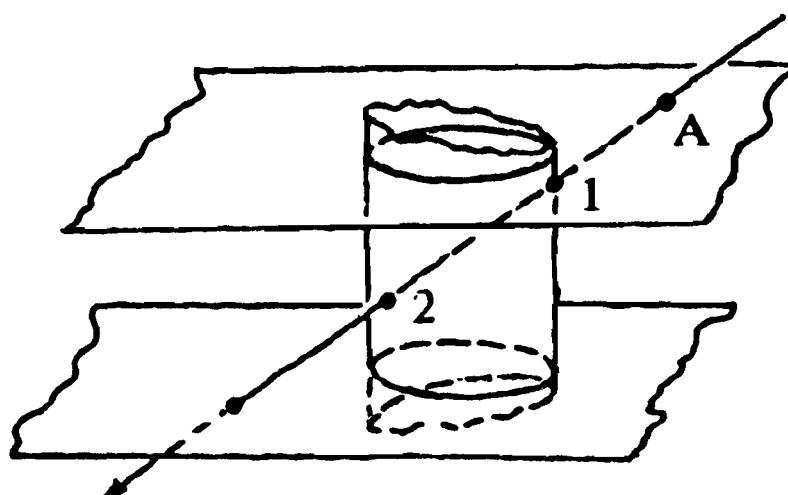


Рис. 3.4.8. Пересечение луча с продолжением поверхностей цилиндра

Имея аппарат определения видимой точки для квадратичного примитива, необходимо быть уверенным в его работоспособности

при перемещении примитива в объектной системе координат. В § 2.2 приведены правила определения коэффициентов уравнения квадратичной поверхности при изменении системы координат. Напомним, что если в системе координат  $X'Y'Z'$  квадратичная поверхность задается матрицей

$$P' = \begin{bmatrix} A' & D' & F' & G' \\ D' & B' & E' & H' \\ F' & E' & C' & J' \\ G' & H' & J' & K' \end{bmatrix},$$

а для перевода некоторой точки  $(X', Y', Z')$  из этой системы в другую –  $XYZ$  необходимо применить преобразование  $F$  вида  $[X \ Y \ Z \ 1] = [X' \ Y' \ Z' \ 1]F$ , где  $F$  – матрица размера  $4 \times 4$  элемента (3.4.9), то новая матрица

$$P = \begin{bmatrix} A & D & F & G \\ D & B & E & H \\ F & E & C & J \\ G & H & J & K \end{bmatrix}$$

квадратичной поверхности в новой системе координат  $XYZ$  будет определяться по правилу  $P = F^{-1}P'(F^{-1})^T$ .

Для метода трассирования лучей и твердотельной конструкции примитива существенно важным является инвариантность свойства положительности внутренней области примитива по отношению к преобразованию координат (перемещениям примитива). В соответствии с доказательством, приведенным в § 2.2, наблюдается не только инвариантность знака скалярного поля второго порядка внутри примитива, но и сохранение ориентации нормали при перемещениях примитива.

Нормальный вектор в точке  $(X, Y, Z)$  квадратичной поверхности, построенной по принципу положительности внутренней области, должен определяться по правилу

$$\mathbf{N} = -\Omega\psi, \quad (3.4.14)$$

где  $\Omega = [i \ j \ k]$ .

Функция  $\psi$  представляет собой вектор-столбец

$$\psi = \begin{bmatrix} AX + DY + EZ + G \\ BY + DX + EZ + H \\ CZ + FX + EY + J \end{bmatrix}.$$

Вектор  $\mathbf{N}$  направлен по градиенту скалярного поля  $f(X, Y, Z)$ , т.е. в сторону возрастания значений  $f(X, Y, Z)$ . В выражение (3.4.14)

введем знак минус, так как для вычисления освещенности следует определять внешнюю нормаль, а вследствие ранее принятого правила выбора положительности поля внутри примитива возрастание значений квадратичной функции происходит по направлению внутрь примитива.

В приложении приведены программы ALLOID, ALLIPS, CILIND, KONYC, PICA, используемые для изображения наиболее часто используемых примитивов второго порядка: ALLOID – эллипсоида, усеченного двумя плоскостями; ALLIPS – эллипсоида; CILIND – кругового цилиндра со свободно-ориентированной осью и закрытыми торцами; KONYC – конуса, ограниченного двумя плоскостями, перпендикулярными оси; PICA – неограниченного конуса. Все эти подпрограммы используют в свою очередь алгоритм решения квадратного уравнения KVADR и служебную подпрограмму NA8FIR, которая из нескольких точек пересечения прямой с примитивом сначала выбирает действительно лежащие на поверхности, а из последних – ближайшую к центру проекции. Процедура обработки эллипсоида и конуса ориентирована на расположение осей этих фигур параллельно координатным осям. Так как все решения пересечения прямой с примитивами ориентированы только на поиск одной ближайшей точки, то подпрограммы можно применять в сценах только с пространственно складывающимися фигурами.

На рис. 3.2.2 показаны несколько шаров и усеченный эллипсоид (бочка), иллюстрирующие изобразительные свойства примитива типа эллипсоида. Изображение построено с использованием подпрограмм ALLIPS и ALLOID. Заметим, что шарообразные поверхности всегда проявляют блик, конечно, при их достаточной гладкости. Плоские же поверхности (торец бочки на рис. 3.2.2) проявляют блик только в очень узком диапазоне условий наблюдения и освещения. Изображение цилиндра, для которого использовано математическое описание в виде (3.2.1) показано на рис. 3.2.3. Рисунки 3.2.5,а,б демонстрируют возможности изменения облика конического примитива при перемещении отсекающих плоскостей вдоль оси конуса. Фигуру на рис. 3.2.5 называют *бантик*, фигуру на рис. 3.2.6 – *усеченный конус*.

### 3.4.4. БИКУБИЧЕСКИЕ ПОВЕРХНОСТИ

С помощью бикубических поверхностей можно описать гладкую поверхность произвольной формы [1, 79, 99, 104]. Обычно этими функциями описывают смежные участки криволинейной поверхности, добиваясь гладкости в местах стыка в результате координатного совпадения смежных угловых точек и совпадения первых производных. Рассматриваемые функции являются функциями наименьшей степени, с помощью которых достигается гладкость составной поверхности. Известно [41, 49, 60], что участок такой поверхности  $X=X(s,t)$ ,  $Y=Y(s,t)$ ,  $Z=Z(s,t)$  может быть представлен параметрически, например, уравнение для  $X(s,t)$ :

$$X(s,t) = a_{11}s^3t^3 + a_{12}s^3t^2 + a_{13}s^3t + a_{14}s^3 + a_{21}s^2t^3 + a_{22}s^2t^2 + a_{23}s^2t + a_{24}s^2 + a_{31}st^3 + a_{32}st^2 + a_{33}st + a_{34}s + a_{41}t^3 + a_{42}t^2 + a_{43}t + a_{44},$$

где  $s, t$  – параметры, изменяющиеся в некотором фиксированном диапазоне, обычно  $t \in [0,1], s \in [0,1]$ ;  $a_{11}, \dots, a_{44}$  – постоянные коэффициенты в пределах данной поверхности, которые могут быть объединены в матрицу  $C_X$  размера  $4 \times 4$ .

Аналогичные выражения существуют для  $Y(s,t)$ ,  $Z(s,t)$ . Обозначим  $C_X, C_Y, C_Z$  матрицы размера  $4 \times 4$  коэффициентов при переменных в выражениях для  $X(s,t)$ ,  $Y(s,t)$ ,  $Z(s,t)$  соответственно и введем векторы-строки  $S = [s^3 \ s^2 \ s \ 1]$ ,  $T = [t^3 \ t^2 \ t \ 1]$ , тогда [60]

$$X(s,t) = S C_X T^T; \quad Y(s,t) = S C_Y T^T; \quad Z(s,t) = S C_Z T^T. \quad (3.4.15)$$

Основной задачей конструирования криволинейной поверхности из бикубических участков является задание коэффициентов бикубического многочлена  $C_X, C_Y, C_Z$  внутри каждого участка через координаты общих угловых точек, касательные вектора в этих точках или систему, так называемых *управляющих точек*. В зависимости от типа имеющихся исходных данных о геометрии моделируемой поверхности применяют представление зависимостей (3.4.15) в виде форм Эрмита, Безье, В-сплайнов [60, 146, 160]. Форма Эрмита предполагает построение бикубического участка в виде изогнутого четырехугольника, к углам которого могут быть пристыкованы другие подобной формы задания:  $X(s,t) = S M_h Q_X M_h^T T^T$ ;  $Y(s,t) = S M_h Q_Y M_h^T T^T$ ;  $Z(s,t) = S M_h Q_Z M_h^T T^T$ , где

$$M_h = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix},$$

$Q_X, Q_Y, Q_Z$  – матрицы  $4 \times 4$  элементов, которые строятся по информации о координатах четырех угловых точек  $(X_i, Y_i, Z_i)$ ,  $i = 1, 2, 3, 4$ , и частных производных  $dX_i/ds$ ,  $dX_i/dt$ ,  $d^2X_i/dsdt$ ,  $dY_i/ds$ ,  $dY_i/dt$ ,  $d^2Y_i/dsdt$  в этих точках. Например,

$$Q_X = \begin{bmatrix} X_1 & X_2 & dX_1/dt & dX_2/dt \\ X_3 & X_4 & dX_3/dt & dX_4/dt \\ dX_1/ds & dX_2/ds & d^2X_1/dsdt & d^2X_2/dsdt \\ dX_3/ds & dX_4/ds & d^2X_3/dsdt & d^2X_4/dsdt \end{bmatrix}.$$

Выражения для  $Q_Y, Q_Z$  определяются аналогично путем замены в последнем выражении  $X$  на  $Y$  и  $Z$  соответственно. Здесь переменная  $t$  меняется от 0 до 1 при переходе от точки  $i = 1$  до  $i = 2$  и  $i = 3$  до  $i = 4$ , а переменная  $s$  – соответственно при переходе от точки  $i = 2$  до  $i = 4$  и от  $i = 1$  до  $i = 3$ .

При переходе от одного участка бикубической поверхности к другому должны обеспечиваться совпадение координат всех точек граничных кривых и совпадение первых производных в этих точках. Это условие достигается идентичностью соответствующих строк (при сшивке кусков вдоль ребра  $t=[0,1]$ ) или столбцов (при сшивке вдоль ребра  $s=[0,1]$ ) [54].

Бикубический многочлен в форме Эрмита позволяет достигать непрерывности производных нулевого и первого порядка. Заметим для сравнения, что в полигональном поле достигается непрерывность только нулевого порядка, что проявляется на изображении в виде резких изломов. Недостатком задания кривых поверхностей в форме Эрмита является сложность интерпретации и получения значений частных производных (тангенсов углов наклона касательных векторов) в угловых точках.

Бикубические поверхности в форме Безье имеют вид [60]  
 $X(s,t)=SM_B P_X M_B^T T^T$ ;  $Y(s,t)=SM_B P_Y M_B^T T^T$ ;  $Z(s,t)=SM_B P_Z M_B^T T^T$ ;  
где

$$M_B = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix},$$

а  $P_X, P_Y, P_Z$  – матрицы  $4 \times 4$  соответственно координат X, Y, Z для 16 управляемых точек.

Так,

$$P_X = \begin{bmatrix} X_{11} & X_{12} & X_{13} & X_{14} \\ X_{21} & X_{22} & X_{23} & X_{24} \\ X_{31} & X_{32} & X_{33} & X_{34} \\ X_{41} & X_{42} & X_{43} & X_{44} \end{bmatrix},$$

где  $(X, Y, Z)_{11}, (X, Y, Z)_{14}, (X, Y, Z)_{41}, (X, Y, Z)_{44}$  – координаты четырех угловых точек, а четыре тройки остальных точек с индексами 21, 22, 12; 13, 23, 24; 43, 33, 34; 42, 32, 31 являются концами касательных векторов, как это показано на рис. 3.4.9.

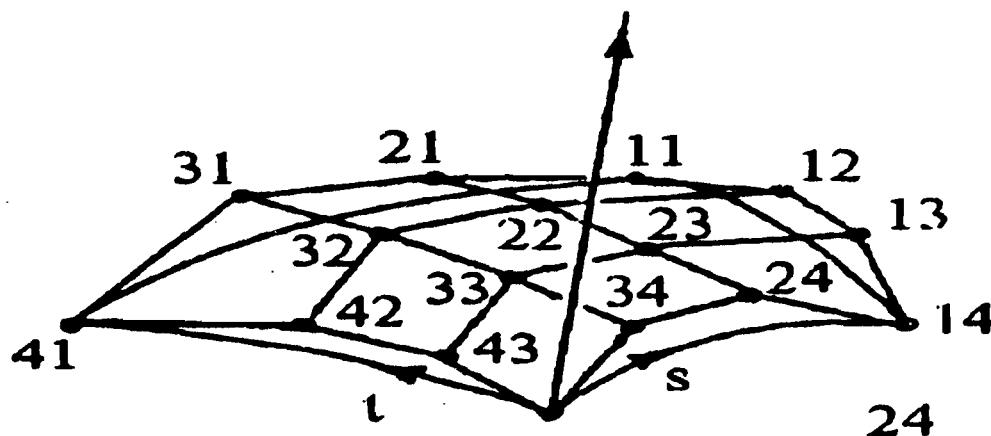


Рис. 3.4.9. Размещение угловых и управляемых точек

Таким образом, геометрия бикубического куска определяется положением 16 управляемых точек. Пространственное положение этих точек значительно легче интерпретируется, чем параметры в форме Эрмита.

Для гладкого перехода от одного бикубического куска в другой необходимо [54] пространственное совпадение четырех смежных управляемых точек и отсутствие изломов в стыке соприкасающихся плоских управляемых полей в направлении поперек стыка.

Бикубические куски в форме B-сплайнов имеют вид  $X(s, t) = S M_S P_X M_S^T T^T$ ;  $Y(s, t) = S M_S P_Y M_S^T T^T$ ;  $Z(s, t) = S M_S P_Z M_S^T T^T$ ,

где

$$M_S = (1/6) \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix},$$

а  $P_X, P_Y, P_Z$  – задают соответственно координаты 16 управляющих точек.

Поверхности, определяемые В-сплайнами, обладают непрерывностью первой и второй производных и значительными возможностями по моделированию поверхностей со сложной топографией.

Метод трассирования лучей требует определения точки пересечения луча с такой поверхностью. Координаты этой точки могут быть установлены при решении следующей системы трех бикубических уравнений с тремя неизвестными  $s, t, \gamma$ :

$$X_0 + l\gamma = SC_X T^T; Y_0 + m\gamma = SC_Y T^T; Z_0 + n\gamma = SC_Z T^T,$$

где  $\gamma$  – переменная в параметрической форме представления светового луча;  $l, m, n$  – проекции направляющего вектора светового луча;  $X_0, Y_0, Z_0$  – координаты точки испускания светового луча (источник света или центр проекции), заметим, что в этой точке  $\gamma=0$ . Геометрический смысл решения последней системы заключается в том, что точка пересечения прямой  $X=X_0 + l\gamma, Y=Y_0 + m\gamma, Z=Z_0 + n\gamma$  с поверхностью  $X=X(s,t), Y=Y(s,t), Z=Z(s,t)$  одновременно принадлежит и прямой, и поверхности.

Система приведенных уравнений сводится к одному уравнению шестой степени, что указывает на невозможность его аналитического решения. Численное решение достигается методом Ньютона [16, 74, 138], посредством которого получают тройку значений  $s, t, \gamma$ , первые два из которых определяют положение точки пересечения на поверхности, а последнее используется для определения видимой точки из серии претендующих для данного светового луча.

Если найденная точка видимая, то необходимо определить ее яркость, для чего, в свою очередь, следует найти уравнение нормального вектора в данной точке поверхности. Известно [23], что уравнение ненормированной нормали  $N$  к поверхности вида  $X=X(s,t), Y=Y(s,t), Z=Z(s,t)$  в точке  $s, t$  имеет вид

$$N=id(Y,Z)/d(s,t)+jd(Z,X)/d(s,t)+kd(X,Y)/d(s,t),$$

где  $i, j, k$  – орты осей  $OX, OY, OZ$  соответственно;  $d(f_1(s,t), f_2(s,t))/d(s,t)$  – оператор Якобиана [23]. Для нахождения компонент нормального вектора введем  $S' = dS/ds = [3s^2 \ 2s \ 1 \ 0]$ ;  $(T^T)' = -dT^T/dt = [3t^2 \ 2t \ 1 \ 0]$ . В этих обозначениях определим выражения для следующих частных производных, входящих в якобианы:

$$dX/ds = S' C_X T'; \quad dX/dt = S C_X (T^T)';$$

$$dY/ds = S' C_Y T^T; \quad dY/dt = S C_Y (T^T)';$$

$$dZ/ds = S' C_Z T^T; \quad dZ/dt = S C_Z (T^T)'. \quad (3.4.15)$$

После их подстановки можно получить окончательное выражение для вектора нормали:

$$\mathbf{N} = i(S' C_Y T^T S C_Z T^T - S C_Y (T^T)' S' C_Z T^T) + j(S' C_Z T^T S C_X T^T - S C_Z (T^T)' S' C_X T^T) + k(S' C_X T^T S C_Y T^T - S C_X (T^T)' S' C_Y T^T). \quad (3.4.16)$$

Для определения освещенности бикубической поверхности от небесного излучателя, а также в случаях преломления и зеркального отражения необходимо знать не только положение, но и направление нормального вектора. Из дифференциальной геометрии [44] известно, что положительные направления линий  $s, t$  и нормаль, найденная по правилу (3.4.16), образуют правую систему осей. Поэтому управлять ориентацией нормального вектора можно посредством выбора направления линий  $s$  и  $t$  на этапе создания математической конструкции объекта. На рис.3.4.9 показан вектор внешней нормали для выбранных направлений.

Преобразование координат, используемое для перемещения бикубической поверхности, приводит к изменению вида матриц  $C_X, C_Y, C_Z$ . Пусть поверхность задана в форме Безье:  $C_X = M B P_X M_B^T$ ;  $C_Y = M B P_Y M_B^T$ ;  $C_Z = M B P_Z M_B^T$ , а система координат  $XYZ$ , в которой определена поверхность, и система  $X'Y'Z'$  связаны соотношением  $[X \ Y \ Z \ 1]W = [X' \ Y' \ Z' \ 1]$ , где  $W$  – матрица  $4 \times 4$  элемента координатного преобразования. Тогда в системе  $X'Y'Z'$   $ij$ -е элементы геометрических матриц  $P_{X'}, P_{Y'}, P_{Z'}$  определяются по формуле

$$[P_{X'ij} \quad P_{Y'ij} \quad P_{Z'ij} \quad 1] = [P_{Xij} \quad P_{Yij} \quad P_{Zij} \quad 1] W,$$

где, в частности,  $P_{Xij}, P_{X'ij}$  –  $ij$ -е элементы матриц  $P_X, P_{X'}$ ;  $i=1(1)4$ ,  $j=1(1)4$ . При пространственном перемещении и вращении бикубического участка ориентация нормали по отношению к

самой поверхности остается неизменной, т. е. если нормаль была направлена в сторону выпуклости, то это свойство сохранится.

### 3.5. ИЗОБРАЖЕНИЕ ЗЕРКАЛЬНЫХ И ПРЕЛОМЛЯЮЩИХ ПОВЕРХНОСТЕЙ

Трассирование лучей является единственным пригодным инструментом изображения зеркальных и преломляющих поверхностей, что связано с высокой степенью адекватности модели реальным оптическим процессам. В оптике луч О (рис.3.5.1), исходящий из источника света и падающий на некоторую поверхность 1, расщепляется на преломленный 1п, отраженный 1з и множество диффузных 1д. Каждый из этих дочерних лучей при встрече с очередной поверхностью вновь дробится на новые вторичные лучи, например, преломленный луч 1п дробится на 2з, 2п и серию лучей 2д. В рецептор наблюдательной системы попадает лишь малая часть циркулирующих в сцене лучей, которые также являются преломленными, отраженными и диффузно рассеянными от участка поверхности, видимого рецептором.

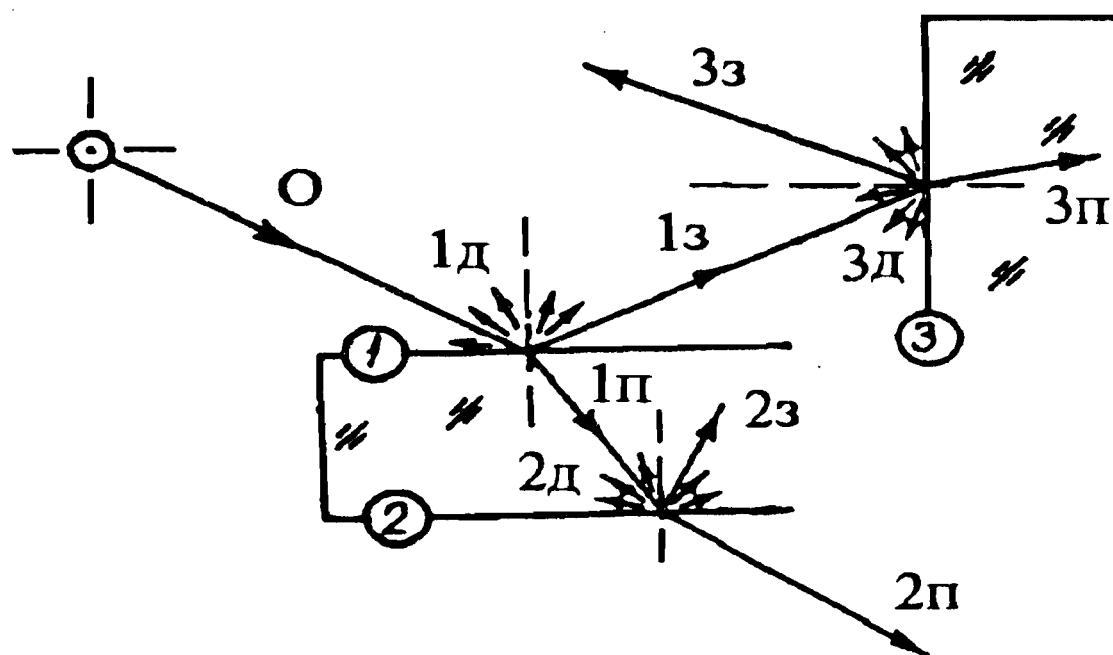


Рис. 3.5.1. Расщепление луча, излучаемого источником света

Метод трассирования лучей повторяет природу, но в обратном направлении распространения света – от рецептора к объекту, что вполне корректно из-за известного принципа обратимости хода светового луча [8]. Для изображения поверхностей, обладающих одновременно диффузными, зеркальными и преломляющими свойствами, в модели отслеживаются пути светового луча по направлению рецептор – центр проекции – ближайшая поверхность, вторичные лучи – очередная ближайшая поверхность – очередные вторичные лучи и т.д. К сожалению,

диффузное отражение приводит к появлению бесконечного числа лучей [8], оттрассировать которые невозможно. Поэтому диффузное отражение не учитывают вовсе или учитывают только для первой к рецептору поверхности. В последнем случае освещенность рассеянного света вычисляют по закону Ламберта. Вторичные лучи, состоящие из зеркальных и преломленных, вновь пересекаются с ближайшими к нам поверхностями, и каждый вновь расщепляется на два (рис.3.5.2). Таким образом, путь луча представляется ветвящимся деревом, “загущенность кроны” которого пропорциональна числу объектов в сцене. Каждый отщепившийся луч ни в природе, ни в модели не отслеживается бесконечно долго: дальнейшее ветвление трассы луча прекращается, когда луч выходит за пределы сцены, или встречается с непрозрачным диффузно рассеивающим телом, или попадает в источник света либо когда интенсивность луча падает ниже порога чувствительности или, наконец, когда число пересечений луча становится слишком большим как для памяти машины, так и для счета.

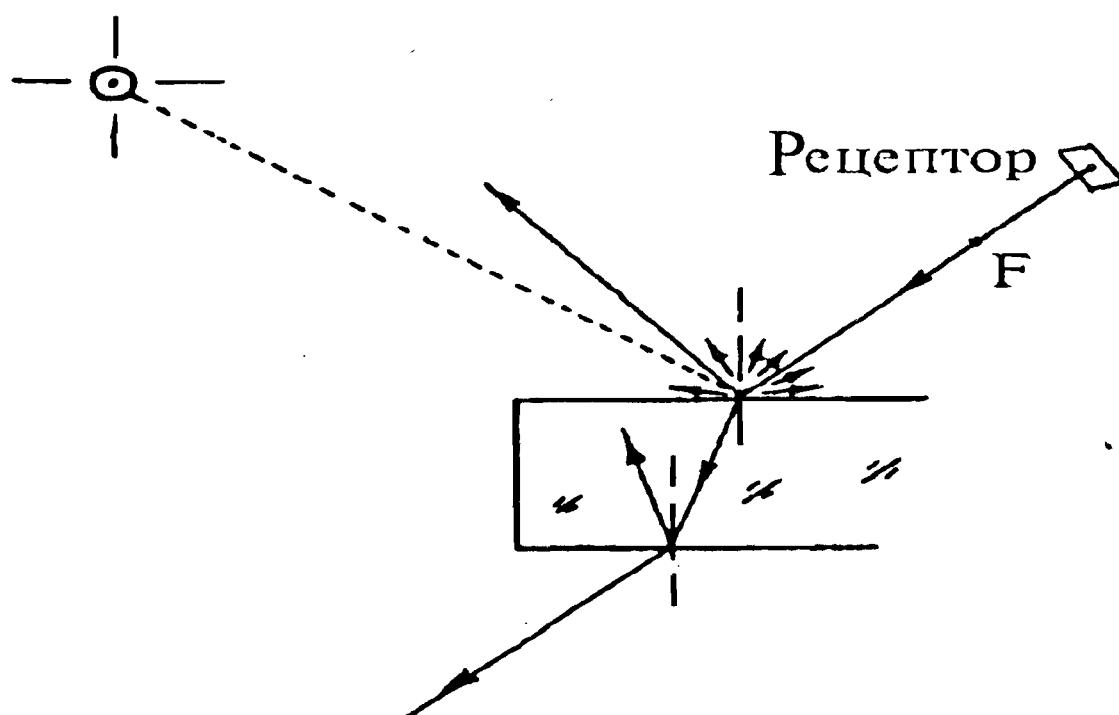


Рис. 3.5.2. Расщепление луча по пути от рецептора к источнику света

Многократно зеркально отражающиеся и преломляющиеся лучи на каждом очередном шаге ветвления процесса требуют определения своего пространственного положения. Ниже приводятся правила определения направления преломленного луча и угла зеркальности.

**Определение вектора преломленного луча.** Для определения трассы луча после прохождения границы двух сред с разными показателями преломления  $n_1$  и  $n_2$  (рис.3.5.3) необходимо определить вектор  $\mathbf{T}$  по известному единичному вектору внешней

нормали  $\mathbf{n}$  и направлению единичного вектора  $\mathbf{v}$  на источник падающего луча. В соответствии с рис.3.5.3 [77] можно показать, что  $\mathbf{T} = \mathbf{a} \cdot \mathbf{n}$ ,  $\mathbf{a} = \operatorname{tg} \alpha_2 \mathbf{b} / |\mathbf{b}| = \mathbf{b} [(\|\mathbf{b}\|/\sin \alpha_2)^2 - \|\mathbf{b}\|]^{-0,5}$ . Так как  $|\mathbf{b}| = |\mathbf{v}'| \sin \alpha_1$ ,  $n_1 \sin \alpha_1 = n_2 \sin \alpha_2$  (закон синусов [8]),  $\mathbf{b} = \mathbf{n} - \mathbf{v}'$ , то можно окончательно получить

$$\mathbf{T} = \{(n_2/n_1)^2 |\mathbf{v}'|^2 - |\mathbf{n} - \mathbf{v}'|^2\}^{-0,5} (\mathbf{n} - \mathbf{v}') - \mathbf{n}, \quad (3.5.1)$$

где  $\mathbf{v}' = \mathbf{v}|\mathbf{n}|/\cos \alpha_1 = \mathbf{v}/(\mathbf{n}\mathbf{v})$ .

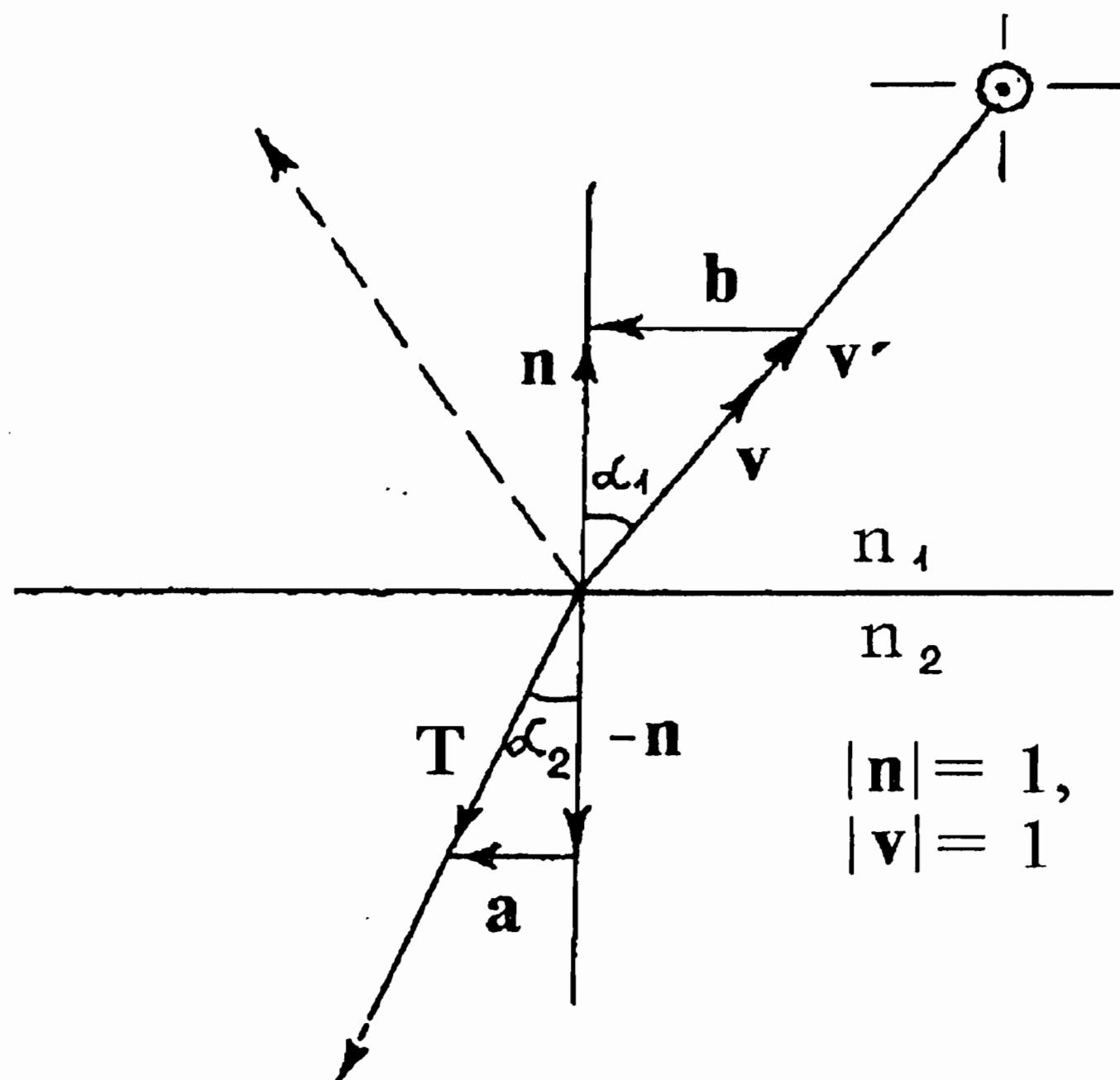


Рис. 3.5.3. Преломление луча на границе двух оптически разнородных сред

В скалярной форме для компонент вектора  $\mathbf{T}$  справедливы следующие соотношения:  $T_x = n_x(B-1)v_x AB$ ;  $T_y = n_y(B-1)v_y AB$ ;  $T_z = n_z(B-1)v_z AB$ , где  $A = (n_x v_x + n_y v_y + n_z v_z)^{-1}$ ;  $B = [(n_2 : n_1)^2(v_x^2 A^2 + v_y^2 A^2 + v_z^2 A^2) - (n_x - v_x A)^2 - (n_y - v_y A)^2 - (n_z - v_z A)^2]^{-0,5}$ . Последние выражения получены путем подстановки в (3.5.1) следующих компонент, слагающих основную формулу:  $v' = v/(nv) =$

$$= i(n_X v_X + n_Y v_Y + n_Z v_Z)^{-1} v_X + j(n_X v_X + n_Y v_Y + n_Z v_Z)^{-1} v_Y + k(n_X v_X + n_Y v_Y + n_Z v_Z)^{-1} v_Z = A(iv_X + jv_Y + kv_Z).$$

$$n - v' = i(n_X - v_X A) + j(n_Y - v_Y A) + k(n_Z - v_Z A), \text{ тогда } |n - v'|^2 = (n_X - v_X A)^2 + (n_Y - v_Y A)^2 + (n_Z - v_Z A)^2, |v'|^2 = A^2 v_X^2 + A^2 v_Y^2 + A^2 v_Z^2.$$

Окончательно получаем, вводя обозначение:  $B = [(n_2/n_1)^2 |v'|^2 - |n - v'|^2]^{-0,5} = [(n_2/n_1)^2 (v_X^2 A^2 + v_Y^2 A^2 + v_Z^2 A^2) - (n_X - v_X A)^2 - (n_Y - v_Y A)^2 - (n_Z - v_Z A)^2]^{-0,5}$ , тогда  $T_X = B(n_X - v_X A) - n_X = n_X(B-1) - v_X AB; T_Y = n_Y(B-1) - v_Y AB; T_Z = n_Z(B-1) - v_Z AB$ .

**Вычисление вектора и угла зеркальности.** Углом зеркальности  $\beta$  будем называть угол между текущим лучом визирования и направлением распространения зеркально отраженного луча. Угол зеркальности показывает, на сколько в угловой мере отклонен светочувствительный приемник от положения зеркального луча наибольшей мощности. Световая энергия убывает с увеличением  $\beta$  по закону, близкому к зависимости  $\cos^n \beta$  ( $n \in [10, 200]$ ).

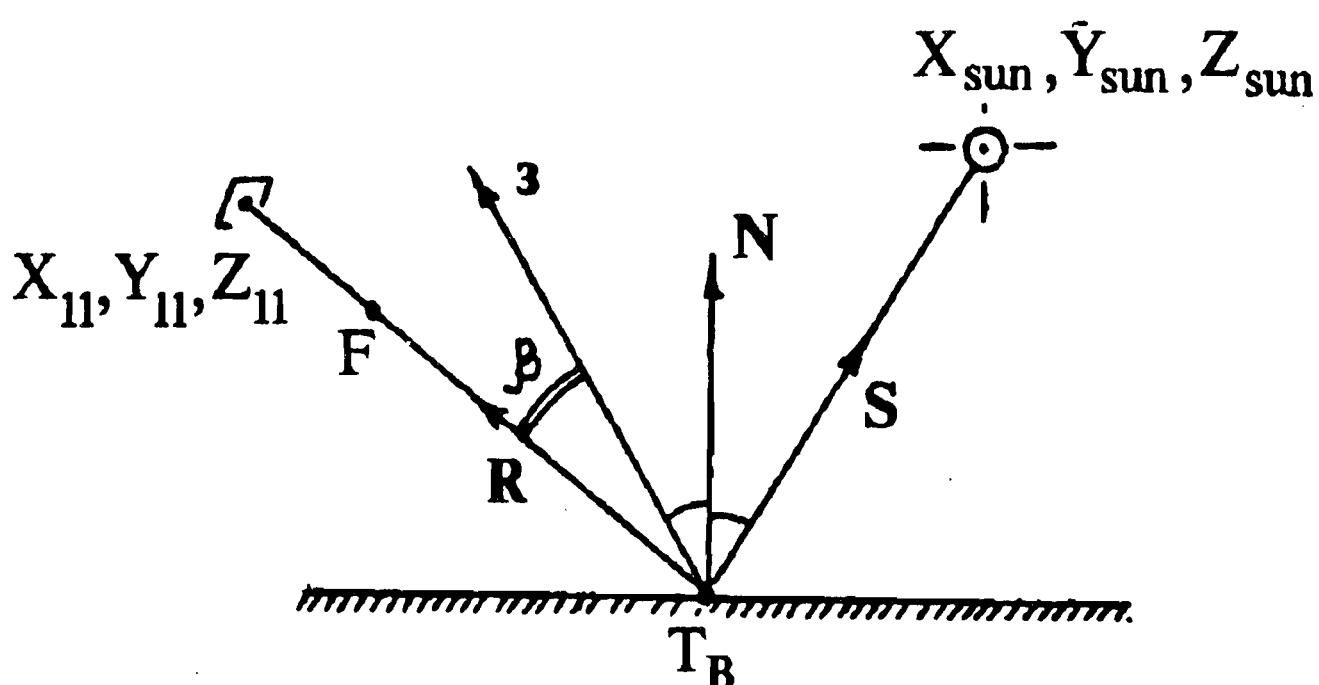


Рис. 3.5.4. Положение векторов при зеркальном отражении

В машинной графике определение освещенности рецепторов от зеркально отражающих поверхностей производится посредством вычисления значения  $\cos \beta$ . Существует несколько способов такого расчета [77, 63]. Приведем один из вычислительно простых. Обозначим:  $T(X_1, Y_1, Z_1)$  – центр рецептора в объектной СК;  $T_B(X_B, Y_B, Z_B)$  – точка, видимая из этого рецептора на поверхности;  $N = iN_X + jN_Y + kN_Z$  – внешняя нормаль к поверхности в видимой

точке;  $S(X_{\text{sun}}, Y_{\text{sun}}, Z_{\text{sun}})$  – положение источника света. Введем векторы:  $S = i(X_{\text{sun}} - X_B) + j(Y_{\text{sun}} - Y_B) + k(Z_{\text{sun}} - Z_B)$ ,  $R = i(X_1 - X_B) + j(Y_1 - Y_B) + k(Z_1 - Z_B)$ . На рис. 3.5.4 показаны данные векторы в плоскости, проходящей через падающий луч  $S$  и нормаль  $N$ . Из законов геометрической оптики известно, что отраженный луч  $z$  также лежит в этой плоскости и может быть вычислен по правилу [61]:  $z = 2N(SN) - S$ . Раскрывая векторную форму, можно получить  $z = i[2N_x(S_x N_x + S_y N_y + S_z N_z) / |N|^2 - S_x] + j[2N_y(S_x N_x + S_y N_y + S_z N_z) / |N|^2 - S_y] + k[2N_z(S_x N_x + S_y N_y + S_z N_z) / |N|^2 - S_z]$ , где  $|N| = \sqrt{(N_x^2 + N_y^2 + N_z^2)^{0.5}}$ .

Угол  $\beta$  между зеркально отраженным лучом  $z$  и направлением на текущий рецептор  $R$  определяется по формуле  $\cos\beta = zR / (|z||R|)$  или в скалярной форме  $\cos\beta = \{R_x(2N_x\rho - S_x) + R_y(2N_y\rho - S_y) + R_z(2N_z\rho - S_z)\} / \{(R_x^2 + R_y^2 + R_z^2)^{0.5} \cdot ((2N_x\rho - S_x)^2 + (2N_y\rho - S_y)^2 + (2N_z\rho - S_z)^2)^{0.5}\}$ , где  $\rho = (S_x N_x + S_y N_y + S_z N_z) / (N_x^2 + N_y^2 + N_z^2)$ .

В программе ВЕТТА (см. приложение) приведены реализация алгоритма вычисления косинуса зеркального угла по известным направлениям: нормали, на источник и на приемник света.

### 3.6. ОСОБЕННОСТИ ОПРЕДЕЛЕНИЯ ОСВЕЩЕННОСТИ

Напомним, что алгоритм трассирования лучей существенно отличается для сцен, содержащих диффузные и бликовые поверхности, и сцен, содержащих прозрачные и зеркальные поверхности. В первом случае объект является единственным преотражателем света от источника, во втором – объект преломляет или отражает свет, который уже отразился от соседних предметов сцены.

Для определения освещенности изображения диффузных и бликовых тел в каждом рецепторе необходимо знать, как это было показано ранее в § 1.1, следующие данные: расстояние от рецептора до видимой точки (если оптическая среда не поглощает света, то этот параметр знать не надо); расстояние от видимой точки до источника света; ориентацию поверхности в районе видимой точки; признак затененности (освещенности от точечного источника света); оптические свойства поверхности; оптические свойства среды распространения света.

Нормальный вектор, определяемый в видимой точке  $V(X_B, Y_B, Z_B)$  поверхности  $f(X, Y, Z)=0$  по правилу

$$N = \text{grad } f(X, Y, Z)|_{(.)V} = i df/dX|_{(.)V} + j df/dY|_{(.)V} + k df/dZ|_{(.)V},$$

направлен внутрь примитива в силу принятого ранее соглашения о выборе положительности внутреннего подпространства. При расчете зеркального отражения и подсветки небом необходимо достижение однозначной ориентации нормали. Установим наружную ориентацию. Тогда для поверхностей, которые образуются путем пересечения или объединения примитивов, наружная нормаль должна вычисляться по правилу

$$N = -\text{grad } f(X, Y, Z)|_{(.)V}.$$

Для поверхностей, которые образуются путем вычитания одного примитива из другого, внешняя нормаль к поверхности образовавшегося тела должна определяться следующим образом:

$$N = \text{grad } f(X, Y, Z)|_{(.)V}.$$

Метод вычисления внешней нормали основывается на априорном определении характера пространственной комбинации примитива, к которому определяется нормаль, с остальными примитивами. Другой подход не требует такого определения и заключается в апостериорной проверке нормали по отношению к направлению на рецептор. Сущность метода заключается в том, что, во-первых, внешняя нормаль должна быть направлена в ту же сторону от поверхности, что и вектор  $R$ , нацеленный из видимой точки на рецептор, а во-вторых, текущее направление нормали неизвестного знака ориентации может быть определено и в случае необходимости изменено на противоположное. Тогда внешняя нормаль вычисляется по правилу

$$N = \text{grad } f(X, Y, Z)|_{(.)V} \text{sign}(\text{grad } f(X, Y, Z)|_{(.)V} R),$$

где  $R = i(X_B - X_F) + j(Y_B - Y_F) + k(Z_B - Z_F)$ .

При наличии в составе объекта только диффузно рассеивающих поверхностей и освещения только точечными источниками ориентация нормали по направлению внутрь-наружу фигуры может не учитываться вовсе при следующей модификации закона Ламберта:  $E = E_0 |\cos \alpha|$ , где  $\alpha$  – угол между направлением на источник и нормалью, ориентированной безразлично: или внутрь или наружу примитива.

Для определения освещенности изображения прозрачных и

зеркальных тел необходимо полностью проследить оптический ход ветвящегося луча по пути от рецептора до объекта сцены. Результирующая освещенность есть сумма освещенностей, приносимых всеми ветвями луча. Освещенность в узлах дерева луча рассчитывается в соответствии с принятой моделью (см. гл.1) источников света и оптическими свойствами поверхностей в этих узлах. Интенсивность света между двумя соседними узлами падает обратно пропорционально квадрату расстояния между точками пересечения для непоглощающих сред. Для поглощающих сред (мутное стекло, вода, ...), кроме того интенсивность падает в зависимости от расстояния по закону Бугера–Ламберта [28, 136]:  $E = E_0 e^{-\alpha r}$ , где  $\alpha$  – коэффициент поглощения среды;  $E_0$  – освещенность при нулевой толщине среды;  $r$  – расстояние, проходимое лучом в среде.

При построении цветных изображений алгоритм вычисления освещенности повторяют трижды: определяют отдельно интенсивности для красного, синего, зеленого лучей с учетом спектральных оптических свойств всех поверхностей. Так как трассы лучей различного света отличаются при прохождении через оптически неоднородные среды, то при моделировании хроматизма необходимо совершенно независимо провести вычисления для всех трех лучей. Если же явление преломления света не учитывается в моделируемой системе, то трассы лучей зеленого, красного и синего цветов полностью совпадают. Изменениям подвержены лишь их интенсивности при диффузном и зеркальном отражении от конкретной поверхности.

# Глава 4. МАТЕМАТИЧЕСКИЕ МЕТОДЫ СИНТЕЗА ИЗОБРАЖЕНИЙ ПУТЕМ ПРОЕКТИРОВАНИЯ ТОЧЕК И ЛИНИЙ НА ЭКРАН

## 4.1. ТРАССИРОВАНИЕ ЛУЧЕЙ ОТ ОБЪЕКТА К ЭКРАНУ

### 4.1.1. ОСНОВНЫЕ ПРИНЦИПЫ

Метод трассирования от объекта к экрану позволяет по вычисленным на поверхности объекта точкам установить координаты их изображения и соответственно именно в эту точку изображения “поместить” информацию об освещенности точки объекта. Если при обратном трассировании лучей априори неизвестно, какую точку “видит” конкретный рецептор, то в данном методе, наоборот, известна точка на объекте, но неизвестен рецептор, который ее видит. Густота точек, покрывающих объект, должна быть достаточной, чтобы соседние точки на объекте проецировались не реже, чем в соседние рецепторы экрана, в противном случае изображение поверхности будет иметь “дыры”.

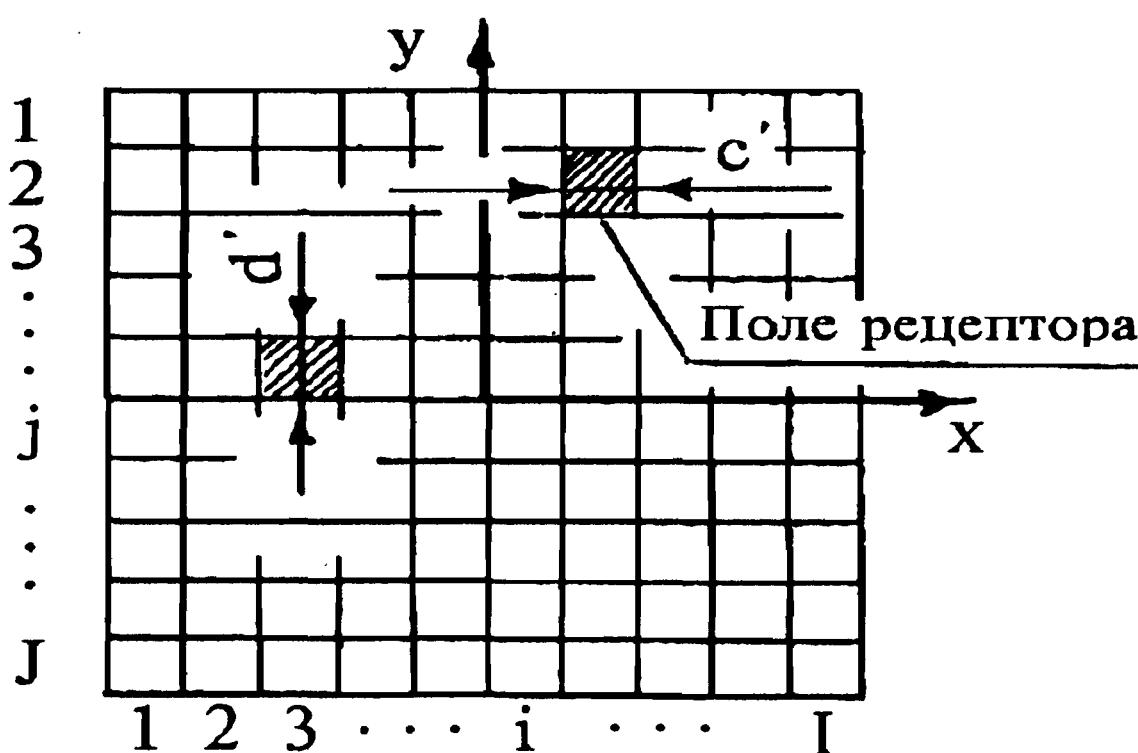


Рис. 4.1.1. Структура рецепторов на экране при прямом трассировании лучей

Модель приемника света представляет собой прямоугольный участок, разбитый без промежутков на  $I \times J$  прямоугольных полей рецепторов со сторонами, параллельными сторонам участка. Последний ориентирован в экранной левой системе координат  $oxyz$  (см. § 3.1) так, что его центр находится в центре системы, ось  $ox$

параллельна строке из J рецепторов, ось oy – столбцу из I рецепторов. Расстояние между центрами соседних рецепторов вдоль оси ox обозначим c', вдоль оси oy – d' (рис.4.1.1). Тогда несложно определить ij-й номер ( $i=(1)I, j=1(1)J$ ) рецептора, полю которого принадлежит точка изображения с координатами x,y:

$$i=\inf(1+I/2-y/d'), \quad j=\inf(1+J/2+x/c'), \quad (4.1.1)$$

где  $\inf$  – функция, определяющая наименьшее целое число от значения аргумента.

При симметричном расположении рецепторов относительно осей координат ox и oy последние выражения справедливы как для четного, так и для нечетного числа строк I и столбцов J.

Экранная система координат может управляемо перемещаться – ориентироваться по отношению к объектной системе координат. Координаты любой точки из объектной системы могут быть пересчитаны в экранную по правилу

$$[x \ y \ z \ 1] = [X \ Y \ Z \ 1] M, \quad (4.1.2)$$

где M – матрица преобразования (см. § 3.1):

$$M = \begin{bmatrix} -\cos\alpha & -\sin\alpha \sin\psi & -\sin\alpha \cos\psi & 0 \\ \sin\alpha & -\cos\alpha \sin\psi & -\cos\alpha \cos\psi & 0 \\ 0 & \cos\psi & -\sin\psi & 0 \\ A & B & C & 1 \end{bmatrix}, \quad (4.1.3)$$

$$A = X_0 \cos\alpha - Y_0 \sin\alpha; \quad B = -Z_0 \cos\psi + \sin\psi(X_0 \sin\alpha + Y_0 \cos\alpha);$$

$$C = Z_0 \sin\psi + \cos\psi(X_0 \sin\alpha + Y_0 \cos\alpha).$$

Правила отсчета углов  $\alpha, \psi$  приведены в § 3.1.

#### 4.1.2. МОДЕЛЬ ОБЪЕКТА

Точно так же, как и при обратном трассировании лучей, модель объекта состоит из сплошных объемных тел и тонких участков поверхностей. Сохраняются структура описания объекта и булево выражение, моделирующее объект. Изменяется лишь форма описания самих поверхностей, слагающих примитивы, ранее она была неявной типа  $f(X,Y,Z)=0$ . Теперь же форма представления поверхности должна быть параметрической:  $X=X(s,t)$ ;  $Y=Y(s,t)$ ;  $Z=Z(s,t)$ , что позволяет перебором параметров  $s,t$  вычислять координаты серии точек на поверхности. При достаточной густоте точек их изображение полностью передает образ самой поверхности.

Возможны многие параметрические задания одной и той же

поверхности: эллипсоид  $x = a \sin s \cos t$ ;  $y = b \sin s \sin t$ ;  $z = c \cos s$ , где  $0 \leq s \leq 2\pi$ ,  $0 \leq t \leq \pi$ ,  $a, b, c$  – коэффициенты. Эллиптический параболоид  $x = p^{-0,5} s \cos t$ ;  $y = q^{-0,5} s \sin t$ ;  $z = 0,5s^2$  и плоскость  $x = x_1 + a_x t$ ;  $y = y_1 + a_y t$ ;  $z = z_1 + a_z t$ , которым в неявном виде соответствуют уравнения

$$x^2/a + y^2/b + z^2/c = 0, \quad x^2/p + y^2/q = 2z;$$

$$(x - x_1)(a \otimes b) + (y - y_1)(a \otimes b) + (z - z_1)(a \otimes b) = 0,$$

где  $a = i a_x + j a_y + k a_z$ ;  $b = i b_x + j b_y + k b_z$ ;  $\otimes$  – знак векторного произведения.

При реализации на ЭВМ обычно организуют цикл расчета координат по одному параметру внутри цикла расчета по другому. Значения приращений  $\Delta t, \Delta t$  выбираются такими, чтобы расстояние между соседними точками  $X(s, t)$ ,  $Y(s, t)$ ,  $Z(s, t)$  и  $X(s + \Delta s, t + \Delta t)$ ,  $Y(s + \Delta s, t + \Delta t)$ ,  $Z(s + \Delta s, t + \Delta t)$  после проецирования на экран было не больше расстояния между центрами соседних рецепторов.

Задавшись границами параметров  $s$  и  $t$ , возможно создание тонких участков поверхностей. Например, задание  $x = a \sin s \cos t$ ;  $y = b \sin s \sin t$ ;  $z = c \cos s$  при  $0 \leq s \leq p$  описывает верхнюю половину эллипсоида.

#### 4.1.3. ОПРЕДЕЛЕНИЕ ВИДИМЫХ ТОЧЕК

Пусть объект состоит из  $D(d=1(1)D)$  примитивов, а каждый примитив – из  $K_d$  поверхностей. Зная правила определения положения точек на поверхности каждого примитива, следует сначала проверить факт принадлежности вычисленной точки объекту, а затем в случае его выполнения определять видимость этой точки.

Принадлежность точки примитива поверхности объекта оценивается по правилам, изложенным в § 3.2.4. Если точка  $(X, Y, Z)$  принадлежит объекту, то она проецируется на экран по следующему правилу:  $[x \quad y \quad z \quad 1] = [X \quad Y \quad Z \quad 1] M$ , где  $M$  – матрица преобразования координат (4.1.3);  $x, y, z$  – координаты точки в экранной системе.

В случае центральной проекции и расположения центра проекции между объектом и экраном проекционные координаты  $x_{\pi}, y_{\pi}$  точки вычисляются по правилу  $x_{\pi} = fx/(f-z)$ ;  $y_{\pi} = fy/(f-z)$ , а для ортогональной проекции  $x_{\pi} = x$ ;  $y_{\pi} = y$ .

Таким образом удается установить однозначное соответствие между вычисленными координатами точки на объекте  $(X, Y, Z)$  и

координатами ее изображения ( $x_{\pi}, y_{\pi}$ ). Установливая подобную связь для всех точек, вычисляемых на объекте, и определяя по правилу (4.1.1) номера соответствующих рецепторов, выявляют видимую точку. На поле одного рецептора может проецироваться достаточно большое число точек объекта. Видимой среди них будет ближайшая к центру проекции или экрану.

Таким образом, определение видимой точки для отдельного рецептора может быть установлено только после проецирования всех точек сцены, что усложняет распараллеливание вычислений между отдельными рецепторами.

Определение факта затененности составляет определенные трудности при прямом трассировании лучей, поэтому данный метод дает быстрые результаты на бесстеневых изображениях, когда центр проекции и источник света совпадают.

#### 4.1.4. ОПРЕДЕЛЕНИЕ ОСВЕЩЕННОСТИ

Для вычисления освещенности необходимо определить нормальный вектор  $\mathbf{N}$  в видимой точке  $X=X(s,t)$ ,  $Y=Y(s,t)$ ,  $Z=Z(s,t)$ . Известно [23], что для явно заданных поверхностей уравнение ненормированной нормали имеет вид

$$\mathbf{N} = i \frac{d(Y,Z)}{d(s,t)} + j \frac{d(Z,X)}{d(s,t)} + k \frac{d(X,Y)}{d(s,t)},$$

где  $i = \frac{d(Y,Z)}{d(s,t)} = dY/ds \frac{dZ}{dt} - dY/dt \frac{dZ}{ds}$ ;

$$j = \frac{d(Z,X)}{d(s,t)} = dZ/ds \frac{dX}{dt} - dZ/dt \frac{dX}{ds};$$

$$k = \frac{d(X,Y)}{d(s,t)} = dX/ds \frac{dY}{dt} - dX/dt \frac{dY}{ds}.$$

Например, пусть задан эллипсоид  $x=a \sin s \cos t$ ;  $y=b \sin s \sin t$ ;  $z=c \cos s$ , где  $s \in [0, \pi]$ ;  $t \in [-\pi/2, \pi/2]$ , тогда в точке  $s=\pi/2$ ;  $t=\pi/2$  ( $X=0$ ,  $Y=b$ ,  $Z=0$ ) значение компонент нормального вектора  $N_x=0$ ;  $N_y=ca$ ;  $N_z=0$ . Во всем остальном алгоритм определения освещенности не имеет особенностей.

Сравнивая описанный метод изображения поверхностей с поточечным алгоритмом сканирующей строки для криволинейных поверхностей [49], нетрудно убедиться, что они во многом имеют общие черты и недостатки: отсутствуют падающие тени и возможности изображения преломляющих и зеркальных поверхностей.

#### 4.1.5. ИЗОБРАЖЕНИЕ ПОТОЧЕЧНО ОПИСАННЫХ ОБЪЕКТОВ

Методом трассирования лучей от объекта к экрану возможно получение изображений объектов, составленных не из поверх-

ностей, а из множества точек. Основная идея метода заключается в изображении каждой точки, описывающей объект, если эта точка видимая. Здесь не происходит никакого достраивания точек по поверхности, что выделяет этот метод среди всех остальных. Примером задачи с данными об объекте в виде точек может служить задача изображения с любых ракурсов геометрически сложной сцены, вся информация о которой представлена одной или несколькими стереопарами-фотоизображениями. Подобные проблемы встречаются при восстановлении формы космических объектов (например, малых планет) по нескольким изображениям со спутника. С использованием известных методов фотограмметрии [30] по такой стереопаре автоматически или интерактивно определяют пространственные координаты множества точек поверхности объекта в некоторой относительной системе координат, и модель объекта может быть описана координатами множества  $K$  точек:

$$T = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ X_k & Y_k & Z_k & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ X_K & Y_K & Z_K & 1 \end{bmatrix}.$$

Очень важно определить требования к плотности размещения точек по поверхности, так как ясно, что при недостаточной их частоте на изображении будут искажения и “провалы”, а при излишней частоте неоправданно возрастут вычислительные затраты. Частота точек на объекте должна быть такой, чтобы любые две соседние точки проецировались в два соседних рецептора на экране или, что менее оптимально, в один рецептор. Следовательно, с учетом масштаба  $1/m$  наблюдения расстояние  $r$  между соседними точками на поверхности объекта должно быть меньше или равно расстоянию  $\lambda$  между центрами соседних рецепторов на экране:  $r/m \leq \lambda$ .

Например, если наблюдение ведется со стократным уменьшением  $m=100$ , а  $\lambda=1$  мм, то расстояние между соседними точками на поверхности не должно превышать 100 мм. При нарушении этого требования соседние точки на объекте спроектируются в два не соседних пикселя, между которыми будет

третий пустой пиксел (или несколько пустых пикселов). На изображении это будет похоже в лучшем случае на отверстие, а в худшем – будет заполнено ложным значением освещенности.

Модель приемника света полностью соответствует изложенной в § 4.1.1. Основная идея определения видимых точек заключается в проекционном переносе всех точек, описывающих объект, в соответствующие рецепторы экрана и в случае попадания нескольких объектных точек в один рецептор в выборе среди последних ближайшей к наблюдателю точки.

Обозначим  $T'$  матрицу размера  $K \times 4$  координат  $K$  точек из  $T$  в экранной системе координат; тогда в соответствии с правилами § 3.1 и § 3.2.4 справедливо  $T' = TM$ , где

$$T = \begin{bmatrix} X_1' & Y_1' & Z_1' & 1 \\ \vdots & \vdots & \vdots & \vdots \\ X_K' & Y_K' & Z_K' & 1 \end{bmatrix}.$$

В случае центральной проекции и расположения центра проекции между объектом и экраном проекционные координаты  $x_k, y_k$  каждой  $k$ -й точки вычисляются по правилу

$$x_k = f X_k' / (f - Z_k'); \quad y_k = f Y_k' / (f - Z_k').$$

Выполнив эту нелинейную операцию для всех  $K$  точек, можно составить матрицу

$$t = \begin{bmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_k & y_k \\ \vdots & \vdots \\ x_K & y_K \end{bmatrix},$$

каждая строка которой есть координаты изображения точки из соответствующей строки матрицы  $T$ . Далее для каждой  $k$ -й точки определим номер экранного рецептора, в поле которого попадают координаты  $x_k, y_k$ . В соответствии с моделью приемника (§ 3.1) номер  $i_k, j_k$  рецептора, в который проецируется  $k$ -я точка, определяется следующим образом:  $i_k = \inf(1 + I/2 - y_k/d');$   $j_k = \inf(1 + J/2 + x_k/c')$ . Вполне понятно, что не все спроектированные точки будут видны. Видимость следует определять для тех групп точек, которые попали в поле одного и того же рецептора. Видимой среди них будет та, которая лежит ближе всех остальных к наблюдателю.

Вычисление освещенности в видимой точке составляет

дополнительную трудность при изображении поточечно описанных объектов, так как для расчета освещенности необходимо знать направление нормали, а определить последнюю в обособленной точке невозможно. Наиболее простой путь решения заключается в нахождении в объектном пространстве нескольких ближайших точек вокруг видимой точки и построении на них как на вершинах полигонального поля. Правила определения направления нормали в вершинах полигонального поля известны [49]. Пусть необходимо определить нормаль в точке  $V_1$ , которая окружена тремя другими ближайшими к ней точками  $V_2, V_3, V_4$ . Тогда вектор  $N_{V_1}$  ненормированной нормали в точке  $V_1$  можно определить следующим образом:  $N_{V_1} = V_1 V_2 \otimes V_1 V_3 + V_1 V_3 \otimes V_1 V_4 + V_1 V_2 \otimes V_1 V_4$ , где  $\otimes$  – знак векторного произведения, а запись  $V_1 V_2$ , например, означает вектор, проведенный из точки  $V_1$  в точку  $V_2$ .

В данном методе получения изображения определенную сложность составляет задача нахождения ближайших точек по отношению к точке поверхности, в которой определяется освещенность. Решение проблемы путем тотального перебора всех точек, описывающих объект, приведет к недопустимым вычислительным затратам. Поэтому более удачной является идея поиска ближайших точек на объекте среди только тех из них общего  $K$  – числа, которые спроектировались на соседние рецепторы по отношению к рецептору, “видящему” анализируемую точку. Так как в результате изгибов поверхности или перекрытия одного объекта другим в смежных рецепторах могут располагаться отнюдь не соседствующие видимые точки и даже не одной и той же поверхности, то проверке на близость к анализируемой видимой точке должны подвергаться не только видимые точки в соседних рецепторах, но и все точки, которые спроектировались в поля смежных рецепторов.

Описанный подход ориентирован на изображение очень сложных поверхностей, информация о которых представлена только координатами множества точек. Одновременно укажем на его недостатки, а именно: трудоемкость описания объекта, игнорирование падающих теней, преломления и зеркального отражения, относительно большое время счета.

## 4.2. МЕТОД СКАНИРУЮЩЕЙ СТРОКИ

Метод сканирующей строки ориентирован прежде всего на изображение сцен, скомпонованных из полигональных полей

или разрозненных многоугольников. Для таких объектов достигается самая высокая вычислительная эффективность по сравнению с другими методами. Теоретически разработаны [69, 144] и практически опробованы методы сканирующей строки для криволинейных поверхностей, однако их вычислительная эффективность сравнима с эффективностью метода трассирования лучей. Описываемый подход целесообразно применять для сцен без преломляющих и зеркально отражающих поверхностей. Изображение теней, как собственных, так и падающих, достижимо.

Алгоритмы построчного сканирования делятся на интервальные и поточечные (использующие z-буфер) для плоских элементов объекта и поточечные для криволинейных объектов. Рассмотрим первый из них, как наиболее характерный и эффективный. Сущность интервального метода заключается в проективном отображении линий-краев из трехмерного пространства в линии на двухмерном изображении, построчном определении видимости и вычислении яркости в интервалах, на которые строка изображения разбивается линиями-краями. Изображение прямой линии также является прямой линией [51], поэтому изменение факта видимости того или иного многоугольника может происходить только при переходе сканирующей строки через прямолинейную границу полей на изображении. Если многоугольники могут протыкать друг друга, то изменение видимости и освещенности может происходить не только при пересечении строки с изображениями ребер многоугольников, но и с изображениями-линиями попарного пересечения многоугольников. Уточним, что *сканирующей строкой* называют текущую строку рецепторов, вдоль которой происходит анализ многоугольников на видимость и соответственно определяется освещенность видимого многоугольника. Освещенность изображения плоских участков объекта есть величина постоянная, поэтому, определив освещенность в одной точке, ее можно распространить на все рецепторы, “видящие” данный плоский участок.

#### 4.2.1. МОДЕЛЬ ОБЪЕКТА ИЗ МНОГОУГОЛЬНИКОВ

Трехмерный объект представляют в виде разрозненных многоугольников, не касающихся и не протыкающих друг друга, и полигональных полей, т. е. множества соприкасающихся без разрывов многоугольников. Случай протыкающихся многоугольников будет рассмотрен отдельно. Поле может быть пространственно замкнутым или открытым. Элементами поля являются

многоугольники в виде треугольников, четырехугольников и т.д. Поле или сцена из отдельных многоугольников описываются через координаты вершин всех  $N$  многоугольников ( $n=1(1)N$ ), входящих в описание объектов:

$$T_n = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ X_k & Y_k & Z_k & 1 \end{bmatrix},$$

где  $n$  – текущий номер многоугольника;  $k$  – количество вершин в многоугольнике;  $T_n$  – матрица вершин  $n$ -го многоугольника.

Для удобства вычислений  $N$  матриц  $T_1, \dots, T_N$  объединяют в одну:

$$T = \begin{bmatrix} T_1 \\ \vdots \\ T_n \\ \vdots \\ T_N \end{bmatrix} = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ X_1 & Y_1 & Z_1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ X_L & Y_L & Z_L & 1 \end{bmatrix},$$

где  $L$  – число ребер многоугольников в сцене,  $L=kN$ ;  $l$  – текущий номер ребра,  $l=1(1)L$ . При описании полигонального поля в целях компактности и устранения лишних вычислений точку, являющуюся общей вершиной нескольких смежных многоугольников, указывают только один раз. В последнем случае предусматривают формальный метод отыскания в матрице всех точек, являющихся вершинами любого  $n$ -го многоугольника.

Рассмотрим в интересах простоты изложения структуру матрицы с повторяющимися точками – общими вершинами треугольников, т.е. первые три точки описывают 1-й треугольник, вторые три – второй и т.д.

#### 4.2.2. МОДЕЛЬ ПРИЕМНИКА

Приемник световой энергии моделируют в виде прямоугольника растровой или векторной структуры. Отличительной особенностью является наличие четко выраженного строчного расположения рецепторов или строк непрерывных линий во всю ширину приемного экрана (рис.4.2.1).

Обозначим расстояние между центрами рецепторов (рис.4.2.1,б)  $a$  и  $b$  вдоль осей  $x$ ,  $y$  экранной системы координат, соответственно расстояние между строками –  $b$ .

Пусть на экране равномерно и симметрично относительно

оси  $x$  размещается  $I$  строк,  $i=1(1)I$ ; тогда уравнение строки под номером  $i$  запишется как  $y = b(I/2-i+0,5)$ , что справедливо как для четного, так и для нечетного числа строк  $I$ .

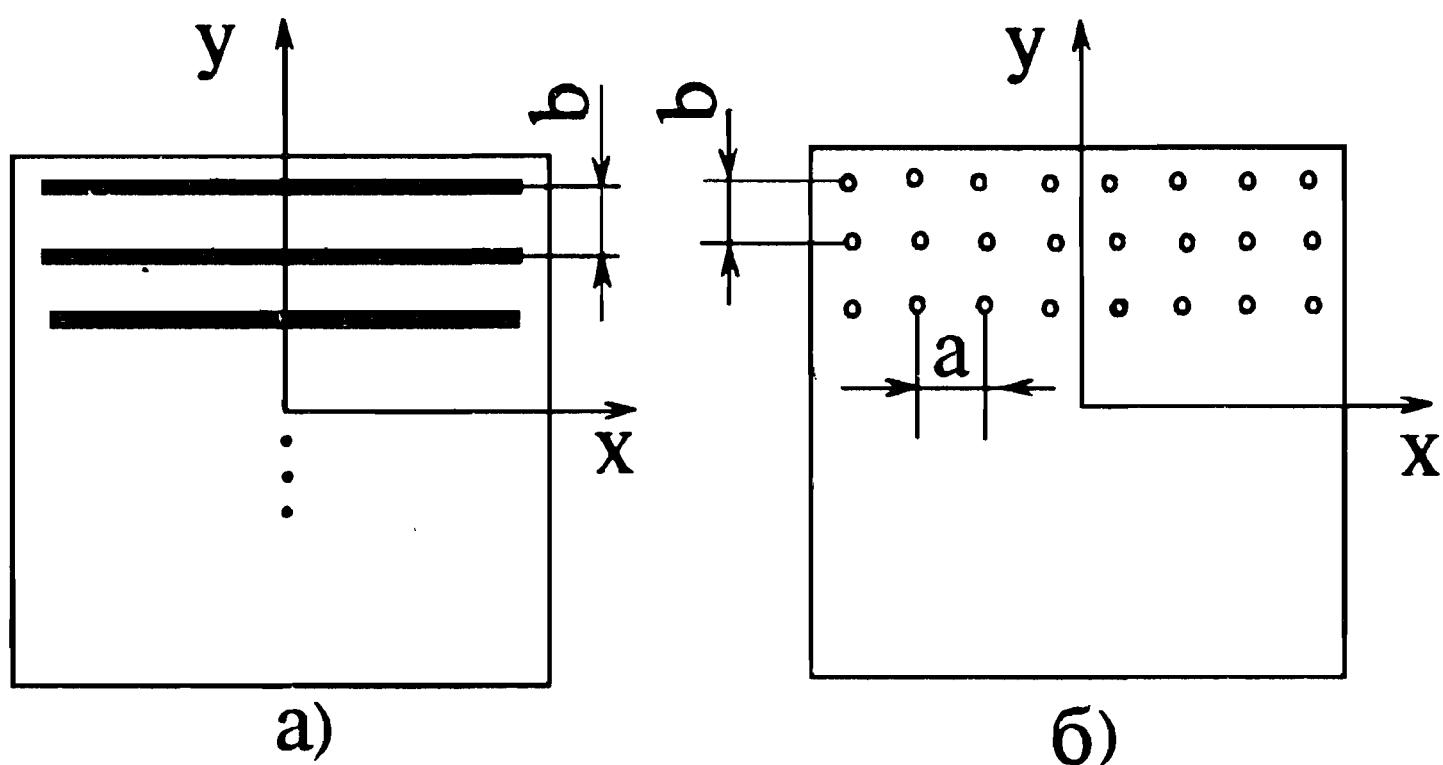


Рис. 4.2.1. Структура приемного экрана в методе сканирующей строки

#### 4.2.3. ИЗОБРАЖЕНИЕ МНОГОУГОЛЬНИКОВ НА ЭКРАНЕ

Пусть экранная и объектная системы координат связаны между собой параметрами сдвига  $X_0, Y_0, Z_0$  и поворота,  $\alpha, \psi$  (рис.3.1.4), где  $X_0, Y_0, Z_0$  – координаты центра экранной системы в объектной системе координат;  $\alpha$  – азимут оси  $oz$ ;  $\psi$  – наклон оси  $oz$ .

Тогда одной операцией легко осуществляется перевод объектных координат вершин многоугольников в экранные:

$$t = \begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_l & y_l & z_l & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_L & y_L & z_L & 1 \end{bmatrix} = TM,$$

где  $M$  – матрица преобразования координат,  $M=f(X_0, Y_0, Z_0, \alpha, \psi)$  (см.(4.1.3));  $L$  – число вершин многоугольников, для случая треугольников  $L=3N$ ;  $l$  – номер текущей вершины;  $l=1(1)L$ .

Проведение этой операции означает, что полигональное поле или отдельные многоугольники теперь определены в

экранной системе. Если применяется ортогональная проекция, то все лучи параллельны оси  $oz$ , координаты изображения  $x_i, y_i$  любой вершины плоскостью соответствуют вычисленным значениям из матрицы  $t$ . Если используется центральная проекция с центром проекции, лежащим между экраном и объектом, то координаты изображения  $x'_1, y'_1$  точки  $(x_1, y_1, z_1)$  определяются по правилу

$$x'_1 = f x_1 / (f - z_1); \quad y'_1 = f y_1 / (f - z_1), \quad (4.2.2)$$

где  $f$  – расстояние от центра проекции до экрана.

Если центр проекции находится за экраном, то

$$x'_1 = f x_1 / z_1; \quad y'_1 = f y_1 / z_1. \quad (4.2.3)$$

Заметим попутно, что операции (4.2.2), (4.2.3) полностью адаптированы к исполнению матричным процессором, так как умножение матриц – обычная стандартная функция матричного процессора, а операция (4.2.3) выполняется путем выделения из  $t$  первого и второго столбцов и поэлементного деления их на третий столбец.

Следовательно, в системе координат экрана можно сформировать матрицу  $t'$  точек-изображений вершин треугольников:

$$t' = \begin{bmatrix} x'_1 & y'_1 \\ \vdots & \vdots \\ x'_L & y'_L \end{bmatrix}.$$

Первые три точки описывают  $N$ -й треугольник поля, вторые три – второй и т.д. Треугольник под номером  $n$  имеет вершины, координаты которых записаны в строках матрицы  $t'$  под номерами  $3n-2; 3n-1; 3n$ . Прямые линии, соединяющие вершины первого треугольника, могут быть описаны следующим образом:  $(x-x'_1)/(x'_2-x'_1) = (y-y'_1)/(y'_2-y'_1)$ ;  $(x-x'_2)/(x'_3-x'_2) = (y-y'_2)/(y'_3-y'_2)$ ;  $(x-x'_3)/(x'_1-x'_3) = (y-y'_3)/(y'_1-y'_3)$ . Уравнения прямых, описывающих ребра  $n$ -го треугольника, образуются из приведенных выражений путем замены индексов 1,2,3 на  $3n-2, 3n-1, 3n$  соответственно.

Таким образом, на приемном экране могут быть построены  $L=3N$  линий-ребер и соответственно  $N$  треугольников. Рассматриваемая  $i$ -я сканирующая строка пересечет не все, а только часть ребер, которые называются *активными* в пределах данной строки. Для  $i$ -й строки, описываемой уравнением  $y_i = b(I/2-i+1/2)$ , активным будет любое ребро, спирающееся на вершины  $(x_{\text{нач}}, y_{\text{нач}})$ ,

$(x_{\text{кон}}, y_{\text{кон}})$ , для которого  $y \in [y_{\text{кон}}, y_{\text{кон}}]$ . После селекции активных ребер определяют координаты  $x$  пересечения их со сканирующей стороной. Например, если ребро первого треугольника активно, то координата  $x$  пересечения определяется решением системы двух уравнений:  $(x - x'_1) / (x'_2 - x'_1) = (y - y'_1) / (y'_2 - y'_1)$  и  $y = b(I/2 - i + 1/2)$ . После решения соответствующих систем для всех активных ребер сканирующая строка оказывается поделенной на множество интервалов, образованных ближайшими друг к другу абсциссами  $x$ . В пределах одного и того же интервала на видимость претендуют несколько отрезков, каждый из которых образован сечением сканирующей плоскости с треугольниками. Сканирующей плоскостью называют плоскость, проходящую через текущую сканирующую строку и центр проекции.

Для определения видимости отрезков и соответствующих им треугольников необходимо вычислить глубину до каждого отрезка (треугольника) в середине интервала [142]. Отрезок, ближайший к наблюдателю (центру проекции), является видимым, и в данном интервале вся сканирующая строка заполняется освещенностью треугольника, которому этот отрезок принадлежит.

Если многоугольники могут пропыкать друг друга, то приведенное правило определения видимости и освещенности сохранится, но список ребер должен быть расширен за счет порождения отрезков – пересечений различных многоугольников попарно между собой. Очевидно, что предварительно необходимо алгоритмически найти все возможные пересечения и занести в память информацию о многоугольниках, которым эти отрезки-пересечения принадлежат. Число анализируемых пар многоугольников равно числу сочетаний из общего числа  $N$  многоугольников по два:  $C_N^2 = N! / \{2(N-2)!\} = N(N-1)/2$ . Например, при  $N=100$  приходится предварительно анализировать пересечения 4950 многоугольников, а при  $N=200$  – уже 19900.

Реализация падающих теней может быть достигнута путем предварительного построения так называемого теневого многоугольника от каждого многоугольника и определения его пересечений со всеми остальными многоугольниками сцены. *Теневым* называют многоугольник, грани которого проходят через ребра многоугольника и источник света. При наличии пересечения в плоскости пересекаемого многоугольника строят вспомогательный теневой многоугольник, который заносят в описание сцены как равноправного участника, но с атрибутами низкой яркости.

Алгоритм построчного сканирования с использованием z-буфера также может использоваться для изображения сцен, составленных из многоугольников. В отличие от интервального алгоритма, где освещенность на сканирующей строке определяется большими прыжками – от интервала к интервалу, здесь анализируется каждый пиксел. Сканирующая строка пересекается с каждым многоугольником (выпуклым) дважды: один раз при входе и один раз при выходе. До этапа определения видимости для каждой сканирующей строки буфер освещенности заполняется некоторым фоновым значением, а текущее значение глубины от рецептора до наблюдаемой точки фона заносится в z-буфер. Оба буфера представляют собой матрицы-строки с числом элементов, равным числу пикселов в строке. Далее анализируется каждый пиксел на сканирующей строке в интервале между входом в треугольник и выходом из него путем сравнения глубины этого пикселя с глубиной, содержащейся в z-буфере. Если рассматриваемый пиксел лежит ближе к наблюдателю, чем буферное значение, то отрезок, на котором лежит пиксел, будет невидимым. По параметрам соответствующего многоугольника вычисляют освещенность и новое значение глубины заносят в z-буфер.

Недостатком приведенного решения является необходимость анализа глубины для каждого пикселя в пределах сканирующей строки. Поточечный метод сканирующей строки в отличие от интервального может быть применен для изображения криволинейных поверхностей.

#### 4.2.4. ИЗОБРАЖЕНИЕ КРИВОЛИНЕЙНЫХ ПОВЕРХНОСТЕЙ

Пусть объект представляет собой участок криволинейной поверхности типа  $X=X(s,t)$ ;  $Y=Y(s,t)$ ;  $Z=Z(s,t)$ . На основании закона связи экранной  $xyz$  и объектной системы  $XYZ$  преобразуем описание поверхности в экранную систему:  $x=x(s,t)$ ;  $y=y(s,t)$ ;  $z=z(s,t)$ . Пересечем эту поверхность со сканирующей плоскостью  $F(x,y,z)$ , которая проходит через  $i$ -ю сканирующую строку и центр проекции  $(0,0,f)$ . Для центральной проекции  $F(x,y,z)$  описывается уравнением

$$F(x,y,z) = \begin{vmatrix} x & y & z - f \\ 0 & y_i & -f \\ 1 & 0 & 0 \end{vmatrix} = 0,$$

а для ортогональной – параллельно оси oz, т.е. уравнение  $F(x,y,z)=0$  вырождается в форму  $y = y_i$ .

Для определения пересечения решим систему из следующих уравнений:

$$\begin{cases} x = x(s,t), y = y(s,t), z = z(s,t); \\ F(x,y,z) = 0 \end{cases},$$

результатом которой является параметрическая кривая, которая, к сожалению, не всегда имеет аналитическую форму и не всегда однозначна. Если же перебирать точки на сканирующей строке от пикселя к пикселу, то численными методами удастся определить параметры s и t в пересечениях световых лучей, исходящих из пикселов и криволинейной поверхности. В случае ортогональной проекции это означает, что для текущего пикселя под номером g ( $g=1(1)G$ ) на i-й сканирующей строке необходимо найти решения  $s = s(x_g, y_i)$ ;  $t = t(x_g, y_i)$ .

По установленным параметрам s, t для g-го пикселя вычисляется удаление до пикселя. Подобные операции выполняются для всех поверхностей сцены. Наименьшее удаление указывает на видимую поверхность, освещенность которой в данном пикселе вычисляется по параметрам s, t. Конкретный алгоритм изображения бикубической поверхности методом сканирующей строки описан в [71].

Таким образом, в общих чертах алгоритм сканирующей строки для криволинейных поверхностей заключается в следующем. Предварительной сортировкой среди множества поверхностей сцены выделяют такие, которые попадают в поле обзора. Затем для каждой сканирующей строки  $y_i$ ,  $i=1(1)I$ , перебирают последовательно пиксели строки  $x_g$ ,  $g=1(1)G$ , и, наконец, последовательно перебирают каждую поверхность. При этом решают уравнения  $s = s(x_g, y_i)$ ;  $t = t(x_g, y_i)$  и вычисляют удаление от каждого g-го пикселя до каждой поверхности. Ближняя поверхность объявляется видимой, ее атрибуты и координаты  $(s, t)_{g,i}$  используются для расчета освещенности в текущем g, i-м рецепторе.

Изображение падающих теней для такого алгоритма затруднено, а собственные тени и блики на освещенной стороне имитируются достаточно просто.

Приведенный алгоритм имеет множество оригинальных модификаций, учитывающих, например, когерентность сканирующих строк [100] или ориентированных только на изображение сфер [124], что приводит к значительному повышению эффективности.

### 4.3. АЛГОРИТМ РОБЕРТСА

Алгоритм получения изображений одиночных выпуклых объектов, составленных из плоских граней, был впервые разработан Робертсом. В переводной литературе метод описан в [41] и наиболее подробно – в [49]. Алгоритм Робертса может быть с успехом применен для изображения множества выпуклых многогранников на одной сцене в виде проволочной модели с удаленными невидимыми линиями. Метод не пригоден непосредственно для передачи падающих теней и других сложных визуальных эффектов. Рассмотрим основные идеи метода на примере одиночного выпуклого многогранника.

#### 4.3.1. МОДЕЛЬ ОБЪЕКТА

Объект-многогранник, состоящий из  $N$  граней, описывается матрицей  $T$  размера  $4 \times N$ , каждый  $n$ -й столбец которой содержит описание  $n$ -й ( $n=1(1)N$ ) плоскости  $A_n X + B_n Y + C_n Z + D_n = 0$  в объектной системе координат XYZ:

$$T = \begin{bmatrix} A_1 & \dots & A_n & \dots & A_N \\ B_1 & \dots & B_n & \dots & B_N \\ C_1 & \dots & C_n & \dots & C_N \\ D_1 & \dots & D_n & \dots & D_N \end{bmatrix}.$$

Минимальное число граней  $N=4$  наблюдается у тетраэдра.

Важное требование к модели объекта в алгоритме Робертса заключается в достижении такой формы (знака) в функции каждой плоскости  $f(X, Y, Z) = A_n X + B_n Y + C_n Z + D_n$ , при которой справедливо  $f_n(X_{\text{ви}}, Y_{\text{ви}}, Z_{\text{ви}}) \geq 0$  для любой точки  $(X_{\text{ви}}, Y_{\text{ви}}, Z_{\text{ви}})$ , заведомо принадлежащей телу многогранника. Достижение этого условия осуществляется путем опытной проверки знака функции относительно внутренней точки, в качестве которой может выступать точка со средним геометрическим положением относительно всех вершин многогранника. Формально правило формирования “положительной” грани может быть представлено

в виде  $f_n^+(X, Y, Z) = f_n(X, Y, Z) \operatorname{sign} f_n(X_{\text{вн}}, Y_{\text{вн}}, Z_{\text{вн}})$ , где  $f_n$  – функция, описывающая  $n$ -ю грань с априори неизвестным знаком относительно внутренней точки;  $f_n^+$  – та же функция, но с положительным значением. При достижении “положительности” всех граней автоматически достигается ориентация нормали  $N_n = iA_n + jB_n + kC_n$  к любой из граней внутрь фигуры.

Кроме параметров уравнений граней необходимо знать координаты вершин каждой грани. Вершины могут быть заданы или вычислены из матрицы  $T$  путем определения общих решений каждой плоскости со всеми остальными. В любом случае для каждой  $n$ -й грани должна быть составлена матрица К-числа вершин:

$$V_n = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ X_K & Y_K & Z_K & 1 \end{bmatrix}.$$

#### 4.3.2. СИНТЕЗ ИЗОБРАЖЕНИЯ

Не все грани, входящие в состав объекта, будут видны наблюдателю. Невидимой будет та, нормаль  $N$  к которой ориентирована в ту же сторону, что и вектор  $R$ , направленный от объекта в сторону наблюдателя, т. е. при выполнении условия неотрицательности скалярного произведения  $NR \geq 0$ .

Эти грани исключаются из дальнейшего анализа. Все оставшиеся являются видимыми. Если грань  $n$  – видимая, а объектная  $XYZ$  и экранная  $xuz$  системы координат связаны преобразованием  $M$  (3.2.1), т.е.  $[x \ y \ z \ 1] = [X \ Y \ Z \ 1]M$ , то для ортогональной проекции координаты вершин  $(x_k, y_k)$  многоугольника–проекции на изображении вычисляются следующим образом:

$$\begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_K & y_K & z_K & 1 \end{bmatrix} = V_n M.$$

Область внутри такой фигуры закрашивается соответствующим значением интенсивности, для вычисления которой и определения факта затененности телом самого многогранника вычисляют скалярное произведение нормального вектора  $N$  и вектора  $S$ , нацеленного из объекта на источник света. Грань освещена при отрицательности скалярного произведения, а величина освещенности выбирается пропорциональной значению  $\cos \alpha = -SN / \|S\| \|N\|$ . Соответственно грань находится в тени при

условии  $SN \geq 0$ . Таким образом, алгоритм Робертса очень простыми действиями достигает успешного построения изображений отдельных выпуклых многогранных объектов.

Основным недостатком метода, определившим ограниченность его распространения, являются неспособность без привлечения других подходов реализовать падающие тени, невозможность передачи зеркальных эффектов и преломления света и, наконец, строгая ориентация метода только на выпуклые многогранники [135].

В [49] приводится модификация алгоритма Робертса, ориентированная на изображение множества выпуклых объектов. Алгоритм выполняется в три этапа. На первом этапе алгоритм Робертса используется в классическом виде для удаления нелицевых (невидимых) граней каждого объекта в отдельности. На втором этапе каждое из множества оставшихся ребер оценивается на экранирование всеми другими телами. Наконец, на третьем этапе определяют отрезки, которые образуют новые ребра при протыкании телами друг друга. Несложно развить этот подход для закрашивания видимых граней, однако проблема падающих теней остается открытой.

# Глава 5. ОРГАНИЗАЦИЯ СИНТЕЗА ИЗОБРАЖЕНИЙ

## 5.1. УПРАВЛЕНИЕ ПРОЦЕССОМ СИНТЕЗА

При практической реализации процесса синтеза изображения на ЭВМ перед пользователем стоит задача оптимального выбора разрешения, обеспечения точного “прицеливания” на объект, выбора фокусного расстояния. Вычислительные мощности расходуются эффективно, когда изображение объекта не выпадает за рамки кадра и занимает весь кадр.

Определим параметры наблюдения, обеспечивающие требуемое качество изображения по дискретности. Будем предполагать, что математическое описание геометрии объекта выполнено с достаточной для данной целевой задачи точностью. Задача будет состоять в определении оптимальной частоты дискретизации в кадре-матрице с фиксированным числом рецепторов. Она может быть решена на основании требований критерия Найквиста [45]: расстояние между центрами дискретизации на изображении должно быть как минимум вдвое меньше самого малого интересующего нас элемента непрерывного изображения. Выполнение критерия Найквиста с потребительской точки зрения означает, что пользователь сумеет увидеть тот малый элемент объекта, который он не хотел потерять в данной целевой задаче. Например, дизайнер автомобиля использует машинную графику для выбора внешнего вида. Дверная ручка имеет наименьший размер в масштабе изображения 3 мм; для данной задачи это размер наименьшей детали, которая существенно влияет на внешний вид автомобиля. В соответствии с критерием Найквиста соседние пиксели на изображении должны размещаться на расстоянии 1,5 мм и менее. Расширим постановку задачи для случая наименьшего требуемого размера на самом объекте.

Расстояние между центрами дискретизации на поверхности объекта и расстояние между рецепторами связаны масштабом съемки. Выберем линейное разрешение  $L$  по поверхности изображаемого объекта при моделировании в масштабе  $1/m$ . Например, наименьший интересуемый размер дверной ручки составляет  $L=3$  см; тогда при съемке в масштабе  $1/10$  расстояние между центрами рецепторов должно быть не более 1,5 мм.

При организации процесса синтеза изображения важно не

только качественное (по дискретности) изображение объекта, но и соблюдение условия попадания интересующего сюжета в поле зрения. Если экран имеет фиксированное количество  $I \times I$  рецепторов, то можно определить расстояние  $d$  между центрами соседних рецепторов, при котором интересующий объект размерами  $E$  весь поместится в кадре. Под  $E$  будем понимать диаметр наименьшего шара, описанного вокруг объекта, и будем считать, что главный луч направлен в центр этого шара; тогда из простых пропорций при центральном проецировании можно получить  $d=fE/[(I-1)R] = E/[m(I-1)]$ , где  $R$  – расстояние от центра проекции до объекта (центра сферы);  $f$  – фокусное расстояние наблюдательной системы;  $m$  – знаменатель масштаба изображения.

Использование этого соотношения позволяет правильно выбирать параметры экрана для уверенного попадания объекта в кадр при условии, что главный луч направлен точно в центр объекта. Например, пусть размер автомобиля  $E=5$  м, кадр имеет  $1000 \times 1000$  рецепторов, масштаб изображения  $1/m=1/10$ ; тогда при прицеливании в центр автомобиля последний не выйдет за пределы кадра, если расстояние между центрами соседних рецепторов будет не менее  $d=5 \cdot 10^{-4}$  м. Заметим, что при выборе максимального размера следует учитывать (приглюсовать) и размер отбрасываемой тени, а при выборе наименьшего размера  $L$  интересующей детали фактического размера  $L_\phi$  из-за наблюдения детали под острым углом  $\phi$  следует уменьшать этот размер до значения  $L = L_\phi \cos\phi$ .

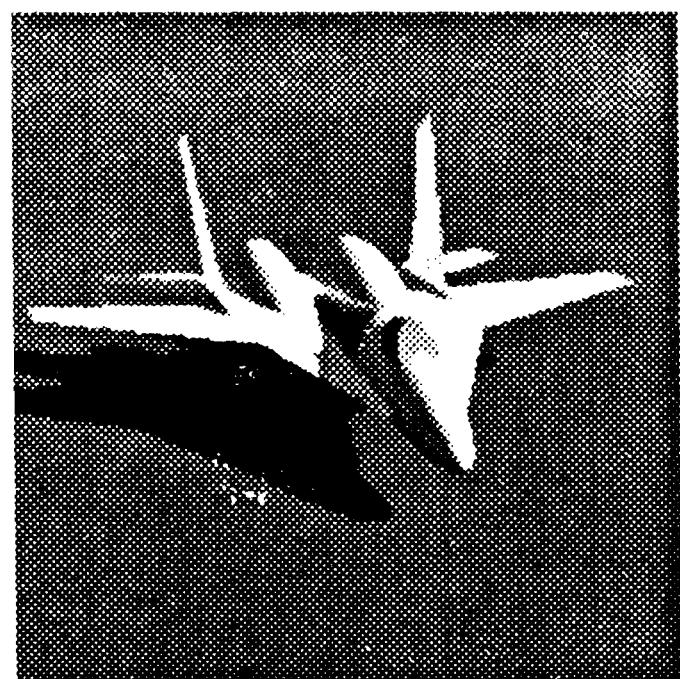
В приведенных соображениях подразумевается прицеливание главной оси в заданную точку на объекте. Покажем метод достижения этого условия. Ориентация главного луча (см. § 3.1) определяется значением углов  $\alpha$  и  $\psi$  в матрице  $M^{-1}$ . Однако в практическом использовании модели неудобно задание ориентации экранной системы через углы, лучше задавать точку прицеливания на объекте, в которую должен попадать главный луч. Выбрав координаты этой точки  $(X_{\text{пр}}, Y_{\text{пр}}, Z_{\text{пр}})$  и центра снимка  $(X_0, Y_0, Z_0)$ , можно получить тригонометрические значения углов, входящих в матрицу  $M^{-1}$ . Из анализа рис. 3.1.4 и правила нахождения угла между вектором и плоскостью можно получить

$$\sin\psi = |Z_{\text{пр}} - Z_0| / \{(X_{\text{пр}} - X_0)^2 + (Y_{\text{пр}} - Y_0)^2 + (Z_{\text{пр}} - Z_0)^2\}^{1/2}.$$

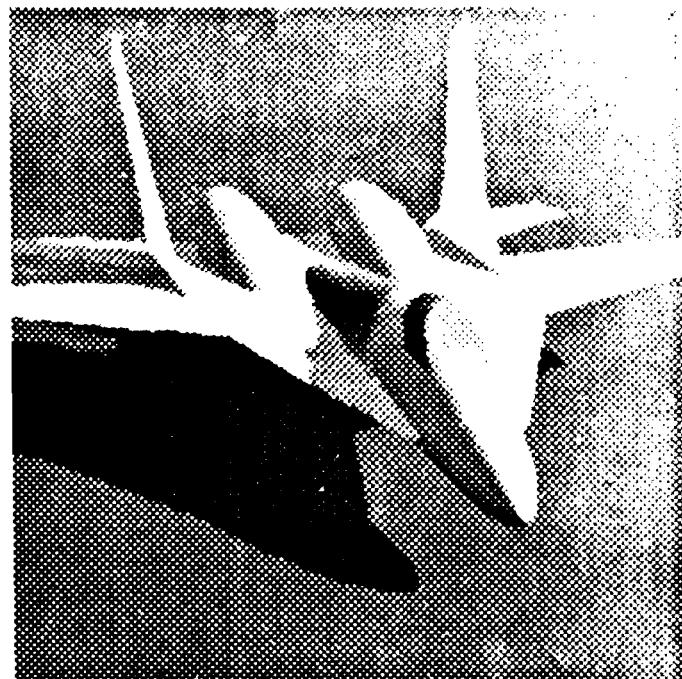
Тригонометрические функции угла  $\alpha$  зависят от четверти в плоскости XY, куда падает проекция главного луча. Простыми преобразованиями можно показать, что для первой четверти справедливо

$\sin\alpha = | -X_{\text{пр}} - X_0 | / \{ (X_{\text{пр}} - X_0)^2 + (Y_{\text{пр}} - Y_0)^2 + (Z_{\text{пр}} - Z_0)^2 \}^{1/2} = A_1$ ;  $\cos\alpha = -| Y_{\text{пр}} - Y_0 | / \{ (X_{\text{пр}} - X_0)^2 + (Y_{\text{пр}} - Y_0)^2 + (Z_{\text{пр}} - Z_0)^2 \}^{1/2} = B_1$ ; для второй четверти  $\sin\alpha = -A_1$ ;  $\cos\alpha = B_1$ ; для третьей  $\sin\alpha = A_1$ ;  $\cos\alpha = B_1$ ; для четвертой  $\sin\alpha = -A_1$ ;  $\cos\alpha = -B_1$ .

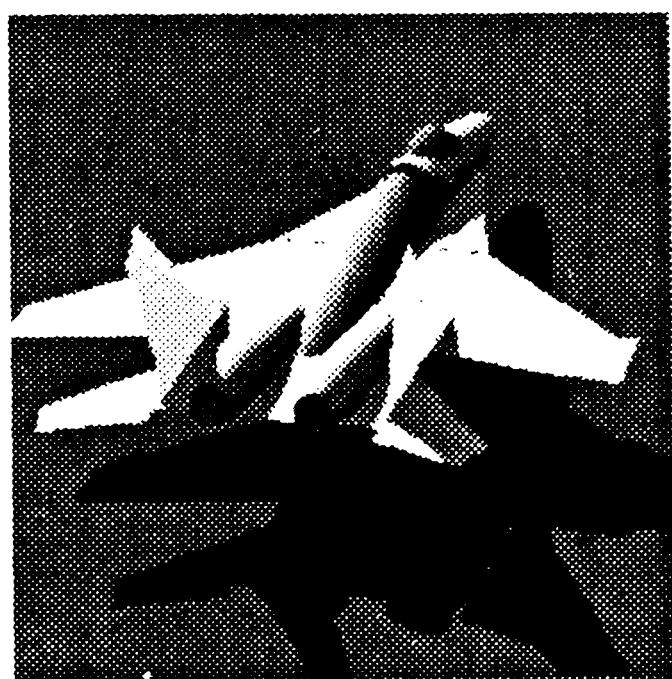
На основании этих соотношений происходит формальное заполнение матрицы  $M^{-1}$ , используемой для фиксации направления на объект, и матрицы  $M$ , используемой для сокращения объема вычислений (см. § 5.2).



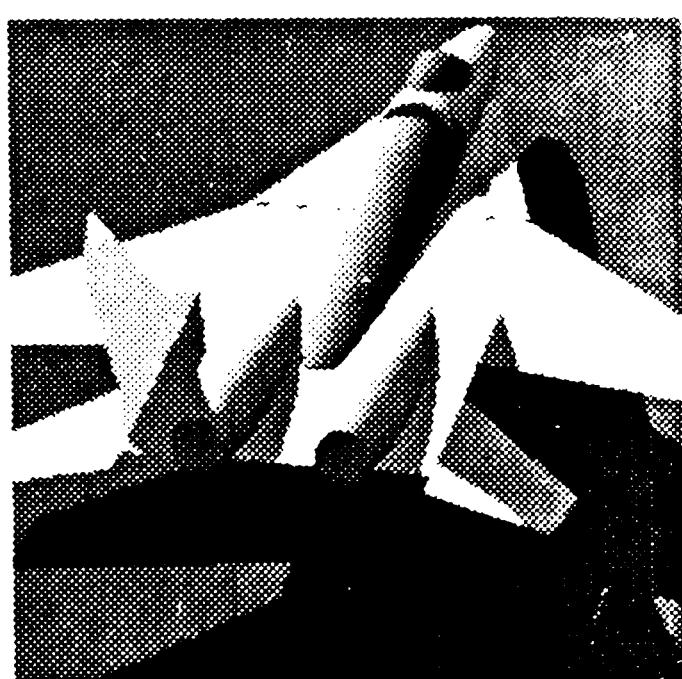
а)



б)



в)



г)

Рис. 5.1.1. Смена ракурса и масштаба изображения

На рис. 5.1.1 показан самолет-истребитель в различных ракурсах, достижение которых обеспечено выбором соот-

ветствующих элементов матрицы  $M^{-1}$ . Заметим, что при смене ракурса необходимо сохранять неизменность удаления до объекта для соблюдения равномасштабности серии изображения, как это выполнено в парах на рис. 5.1.1, а, в и на рис. 5.1.1, б, г.

## 5.2. СПЕКТР СИНТЕЗИРОВАННОГО ИЗОБРАЖЕНИЯ

Для оценки изобразительного качества синтезированного на ЭВМ изображения сравним последнее по спектру с идеальным изображением, полученным в тех же условиях наблюдения по той же сцене. Идеальное – это изображение, свободное от шумов, размытия, дискретизации, aberrаций, т.е. построенное только по законам геометрической оптики. Обозначим такое идеальное непрерывное изображение через  $I(x,y)$ , где  $x, y$  – координаты в поле изображения.

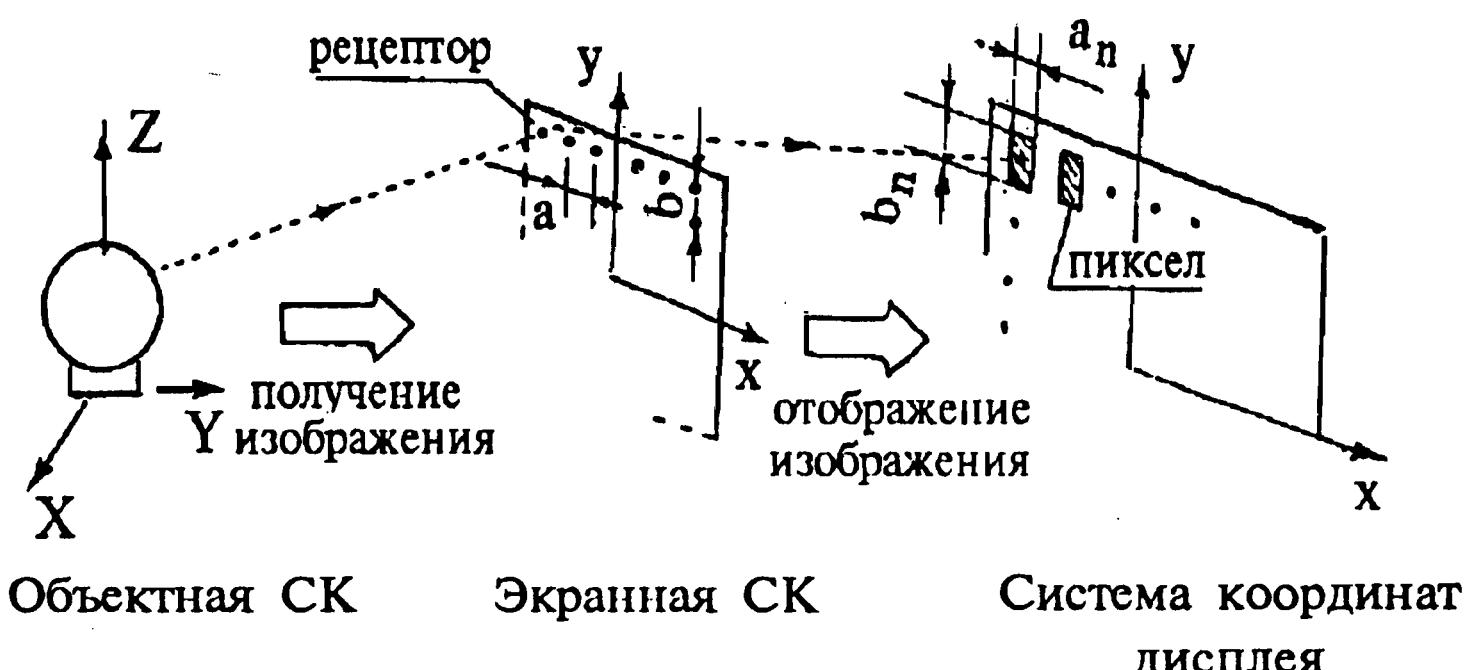


Рис. 5.2.1. Информация об объекте проходит по пути: объект – приемник – дисплей

Во всех системах трехмерной машинной графики процесс синтеза изображения имеет два этапа: запечатление изображения рецепторами и отображение содержимого рецепторов в виде реального изображения (рис. 5.2.1). Структуры приемника светового излучения части экрана и устройства визуализации могут быть совершенно различными, что следует учитывать при анализе спектра. В физических устройствах приемной частью может выступать малоформатная фотопленка, а средством отображения – огромные листы фотобумаги.

Так как на ЭВМ невозможно осуществить пространственно непрерывный процесс переноса информации от каждой точки объекта на изображение, то вне зависимости от метода построение

изображения выполняют путем выборки информации в отдельных точках. Так, при трассировании лучей в плоскость приемника передается информация, соответствующая идеальному изображению только в узлах дискретизирующей решетки. Опишем математически эту решетку по известным правилам [29]  $J(x,y) = \text{Comb}(x/a) \text{Comb}(y/b)$ , где  $a, b$  – расстояние между узлами дискретизации вдоль осей экранной системы координат  $ox, oy$  соответственно.

Тогда на приемном экране формируется дискретное изображение, которое состоит из множества регулярно расположенных точечных отсчетов и которое можно записать в виде  $I'(x,y) = I(x,y) K(x,y) J(x,y)$ , где  $K(x,y)$  – безразмерная искажающая функция, обусловленная неточностью математической модели расчета освещенности.

Изображение  $I'(x,y)$  еще нельзя физически увидеть, так как значениям освещенности соответствуют в плоскости изображения точки, площадь которых равна нулю. Поэтому каждый отсчет  $I'(x,y)$  в узле решетки распространяют на пиксел конечных размеров. На дисплее пикселом является триада цветного люминофора, на принтере – матрица точек, передающая один отсчет яркости. Примем для упрощения совпадение системы координат  $x_p, y_p$  устройства отображения и приемного экрана  $x, y$ , т.е.  $x_p = x, y_p = y$ . Для прямоугольных пикселов функция их пропускания  $W(x,y)$  может быть описана следующим образом:  $W(x_p, y_p) = W(x, y) = \text{Rect}(x/a_p) \otimes \text{Rect}(y/b_p)$ , где  $a_p, b_p$  – геометрические размеры прямоугольного пикселя вдоль осей  $ox, oy$ .

Операция распространения точечного отсчета  $I'(x,y)$  на все поле пикселя описывается оператором свертки:  $I''(x,y) = I'(x,y) \otimes W(x,y) = [I(x,y) K(x,y) \text{Comb}(x/a) \text{Comb}(y/b)] \otimes [\text{Rect}(x/a_p) \text{Rect}(y/b_p)]$ .

Тогда, используя правила фурье-преобразования стандартных функций [29], можно получить выражение, описывающее спектр синтезированного растрового изображения:  $\tilde{I}(f_x, f_y) = [\tilde{I}(f_x, f_y) \otimes \tilde{K}(f_x, f_y) \otimes \text{Comb}(af_x) \otimes \text{Comb}(bf_y)] [\text{Sinc}(a_p f_x) \text{Sinc}(b_p f_y)]$ , где  $\tilde{I}(f_x, f_y)$  – спектр идеального изображения;  $\tilde{K}(f_x, f_y)$  – спектр искажающей функции;  $(f_x, f_y)$  – пространственные частоты вдоль осей  $ox, oy$  изображения.

Сравнивая полученный спектр со спектром изображения в реальном матричном приемнике, непрерывно визирующим картину [29], убеждаемся, что первый отличается от второго только исключением операции усреднения по каждому рецептору и влиянием ошибок вычисления освещенности  $\tilde{K}(f_x, f_y)$ . В реальных устройствах в пределах поля рецептора попадает энергия от многих неодинаково освещенных точек объекта, эти субэнергии усредняются от одного отсчета. Значение последнего передается равномерным свечением всего поля соответствующего пикселя. В машинной графике в рецептор попадает информация только об освещенности одной точки объекта. Этот отсчет при отображении распространяется на все поле пикселя изображения. В некоторых случаях [65] предлагается повысить качество отображения из-за усреднения освещенности от четырех угловых точек рецептора, что приводит к четырехкратному возрастанию объемов вычислений и эквивалентно увеличению частоты дискретизации.

Спектр синтезированного изображения образован многократным повторением несколько искаженного спектра изображения, построенного по законам геометрической оптики. Соседние спектры частично перекрываются, как это наблюдается в любой дискретизирующей системе.

Таким образом, в растровой графике формируется изображение, которое по качеству отличается от изображения, построенного по законам геометрической оптики, из-за методологических погрешностей вычисления яркости и осуществления выборки, соответствующей выборке матричным приемником, непрерывно визирующими картину, без операции усреднения в пределах рецептора. В узлах дискретизации изображение идеально резкое, в нем не учитывается влияние aberrаций.

### 5.3. СОКРАЩЕНИЕ ВРЕМЕНИ СИНТЕЗА ИЗОБРАЖЕНИЙ

Существует два принципиальных направления сокращения времени синтеза изображений: устранение заведомо лишних вычислений и распараллеливание вычислений. Потребность в ускорении процесса синтеза связана с тем, что высококачественные и сложные изображения формируются на однопроцессорных машинах за десятки минут, что выходит за рамки даже самых скромных потребительских требований.

### 5.3.1. МЕТОД ОБОЛОЧЕК

Метод оболочек получил широкое распространение в системах трехмерной машинной графики как для однопроцессорных, так и многопроцессорных машин. Свое название он получил от использования простых по конструкции трехмерных выпуклых фигур-оболочек, которые охватывают объект или его составные части и позволяют сравнительно просто выявлять часть пространства, где находится объект. Известно несколько модификаций метода, например в [145] показано его применение на этапе определения видимости.

**Оболочки-многоугольники.** Сущность метода, реализующего сокращение времени вычислений изображений с тенями [21], заключается в следующем. На этапе создания математической конструкции объекта каждый его отдельный элемент (примитив или система примитивов) окружается интерактивно или автоматически [64, 127] воображаемой оболочкой – выпуклым многогранником. Число вершин  $N$  и ориентация граней выбираются из соображений надежного охватывания тела элемента оболочкой, минимального объема оболочки и минимального числа граней. Обозначим одну такую отдельную оболочку  $A^*$  и запишем в матрицу  $A$  координаты вершин оболочки  $A^*$ :

$$A = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ X_i & Y_i & Z_i & 1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ X_N & Y_N & Z_N & 1 \end{bmatrix},$$

где  $X_i, Y_i, Z_i$  – координаты  $i$ -й крайней точки выпуклой оболочки  $A^*$ ,  $i=1(1)N$ ;  $N$  – число вершин оболочки.

При автоматическом построении выпуклой оболочки в качестве формы последней чаще всего выбирают параллелепипед [60]. Основная идея определения положения и размеров параллелепипеда заключается в вычислении максимальных габаритных размеров охватывающего примитива вдоль осей  $OX$ ,  $OY$ ,  $OZ$ . Так как отрезки, определяющие максимальные габаритные размеры, могут быть не параллельны осям, то часто построенная

оболочка захватывает лишние, не принадлежащие объекту участки пространства, а это, в свою очередь, снижает эффективность метода. При "ручном" задании оболочки в виде неправильного многоугольника достигается более плотное облегание объекта, но зато требуется неавтоматизированная работа оператора. На практике эта работа сводится к выставлению в пространстве наименьшего числа точек, лежащих как можно ближе к объекту и одновременно таких, что выпуклая оболочка, построенная на этих точках, не сечет своими гранями объекта.

Координаты полученных точек и составляют матрицу А, причем порядок следования строк в ней может быть произвольным. К построенному многограннику  $A^*$  достраивают второй, так называемый теневой многогранник (оболочку). Эта теневая оболочка  $C^*$  включает все вершины оболочки  $A^*$  и дополнительные, которые находят путем пересечения лучей света, проходящих через вершины многогранника  $A^*$ , с поверхностью, на которую падают тени. Такой поверхностью в задаче моделирования фотосъемки здания является основание Земли. Так, если солнечный луч, проходящий через вершину  $i$ , имеет направляющий вектор  $(S_x, S_y, S_z)$ , то координаты точки пересечения определяются решением системы уравнений

$$\begin{cases} (X - X_i)/S_x = (Y - Y_i)/S_y = (Z - Z_i)/S_z; \\ \phi(X, Y, Z) = 0 \end{cases},$$

где  $\phi(X, Y, Z)$  – уравнения поверхности (основания), на которую падают тени.

Множество всех пересечений записывают в матрицу В:

$$B = \begin{bmatrix} X'_1 & Y'_1 & Z'_1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ X'_i & Y'_i & Z'_i & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ X'_N & Y'_N & Z'_N & 1 \end{bmatrix},$$

где  $X'_i, Y'_i, Z'_i$  – координаты точки пересечения основания с солнечным лучом, проходящим через  $i$ -ю крайнюю точку оболочки  $A^*$ , и объединяют две матрицы А и В в матрицу С размером  $2N \times 4$  элементов:  $C = A/B$ .

Заметим, что выпуклая трехмерная оболочка, построенная на точках матрицы  $A$ , полностью охватывает тело объекта, а оболочка, построенная на точках матрицы  $C$ , надежно охватывает всю область как собственной, так и падающей тени, куда бы тень ни падала — на соседние объекты или поверхность-основание. На рис. 5.3.1 графически показан объект — шар на цилиндрической подставке и оболочка-многогранник. Крайние точки оболочки  $A^+$ :  $B, C, D, E, F, G, K$ . Крайние точки теневой оболочки  $C^+$ :  $B, C, D, E, F, G, K, L, M, R$ . Проекциями вершин  $E, K, F$  на поверхность-основание вдоль солнечных лучей являются точки  $L, M, R$ . Проекции остальных вершин не показаны.

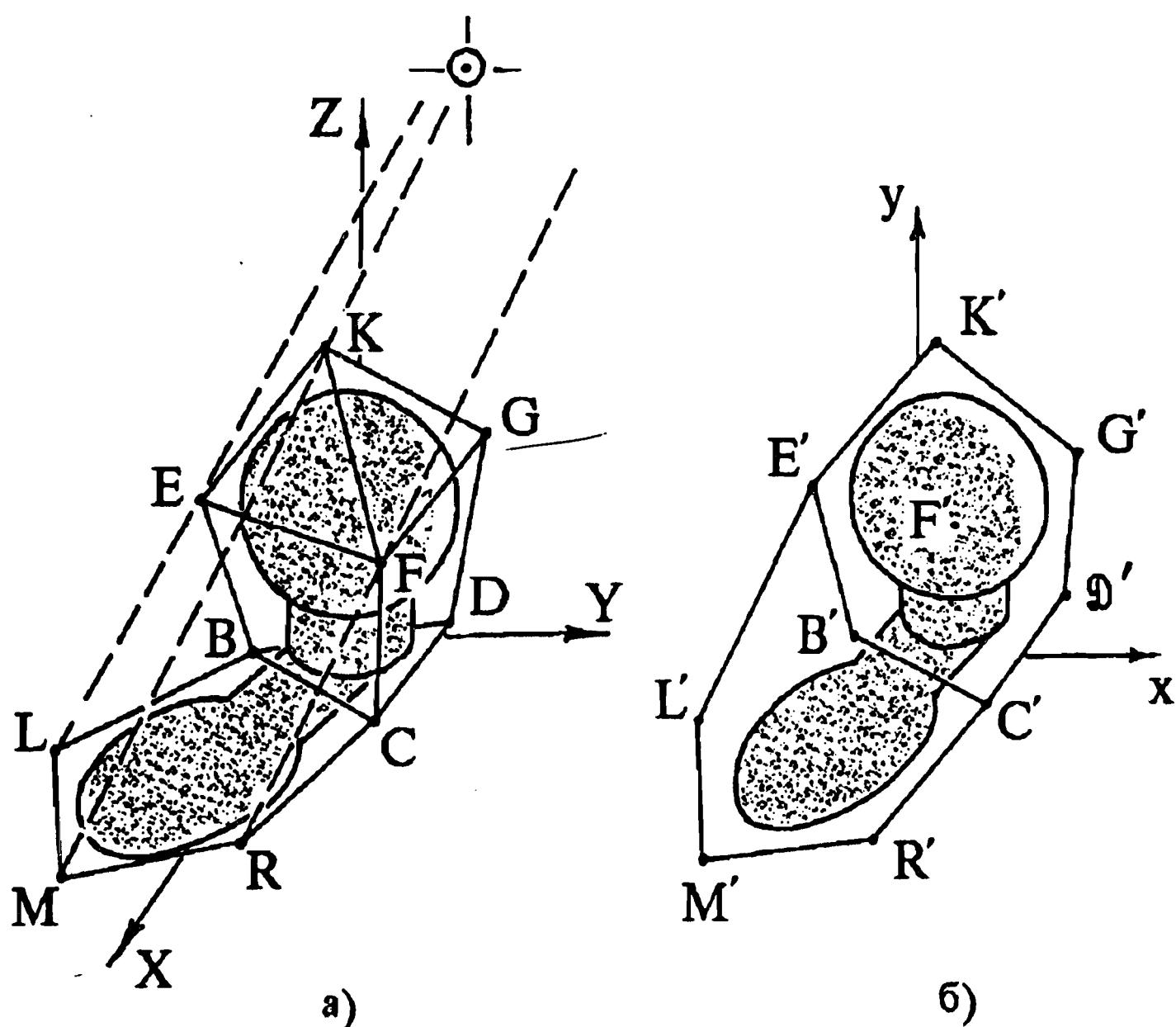


Рис. 5.3.1. Объект, его собственная и теневая оболочки (а), изображение объекта и его оболочек на экране (б)

Спроектируем точки матрицы  $C$  на экран. Для этого сначала пересчитаем их координаты из объектной в экранную систему:  $C' = CM$ , где  $C'$  — матрица размером  $2N \times 4$  элементов, представляющая запись координат точек из области в экранной

системе координат;  $M$  – матрица преобразования размером  $4 \times 4$  элемента [19, 21]. Затем определим перспективные координаты в плоскости экрана для всех точек из матрицы  $C'$ :  $x_j^C = fC'(i,1)/[f-C'(i,3)]$ ;  $y_j^C = fC'(i,2)/[f-C'(i,3)]$ , где  $f$  – фокусное расстояние наблюдательной системы;  $x_j^C, y_j^C$  – координаты изображения  $j$ -й точки матрицы  $C$ ,  $j=1(1)2N$ . Здесь предполагается, что центр проекции лежит между экраном и объектом.

Полученные значения координат изображения точек из матрицы  $C$  запишем в виде

$$C'' = \begin{bmatrix} x_1 & y_1 & \cdot \\ \vdots & \vdots & \\ x_N & y_N & \\ \hline x_{N+1} & y_{N+1} & \\ \vdots & \vdots & \\ x_{2N} & y_{2N} & \end{bmatrix} = \frac{\mathbf{A}''}{\mathbf{B}''},$$

где  $A''$  включает координаты изображений точек выпуклой оболочки  $A$ ,  $C'' - C^+$ .

Так как теневая трехмерная оболочка  $C^+$  уверенно охватывает целиком область тени объекта, то двухмерная выпуклая оболочка  $\theta$ , построенная вокруг изображений  $C''$  крайних точек оболочки  $C^+$ , надежно охватывает изображения тени от данного объекта. Аналогично можно определить область положения на изображении самого тела объекта путем построения выпуклой двухмерной оболочки  $Q$  вокруг точек множества из  $A''$ . На рис. 5.3.1,б показано изображение объекта и оболочек на экране. Крайние точки двухмерной оболочки  $Q$ :  $E', K', G', D', C', B'$ . Крайние точки двухмерной оболочки  $\theta$ :  $M', L', E', K', G', D', C', R'$ . Штрихи в обозначениях указывают, что речь идет о изображении приведенной точки, т. е.  $E'$  есть изображение  $E$ . Способ построения выпуклой оболочки вокруг множества точек на плоскости известен [17, 64, 81, 126, 127]. Так, если  $\Delta$  – оператор, выбирающий из множества точек на плоскости крайние точки выпуклой оболочки, то  $\theta = \Delta(C'')$ ,  $Q = \Delta(A'')$ .

На этапе построения самого изображения, прежде чем производить трассирование луча из каждого рецептора, последний оценивается на принадлежность внутренней области оболочек  $Q$  и  $\theta$ .

Если рецептор находится снаружи оболочки  $Q$ , то он не “видит” тело объекта, и если рецептор находится снаружи оболочки  $q$ , то он не принадлежит области тени от данного объекта. Оценивание пересечения трассирующего луча с примитивом производится только в случае принадлежности рецептора внутренней области оболочки  $Q$ , а оценивание затененности точки, видимой из рецептора, производится только в случае попадания рецептора во внутреннюю область оболочки  $\theta$ . В приложении даны описания подпрограмм AVTOZON, YNZONA, позволяющих оценивать принадлежность текущего рецептора областям  $Q$  и  $\theta$ .

Использование описанного подхода для иерархии примитивов избавляет от слепого поиска и анализа точек пересечения трассирующего луча с заведомо невидимыми примитивами и комбинациями примитивов и сокращает время вычислений в несколько раз. При этом эффект от использования оболочек становится тем ощутимее, чем больше примитивов в составе объекта и сложнее их конструкция.

Заметим, что при вычислении матрицы  $C'$  удобно использовать аналитически полученное значение матрицы  $M$  преобразования координат из объектной системы в экранную:

$$M = \begin{bmatrix} -\cos\alpha & -\sin\alpha \sin\psi & -\sin\alpha \cos\psi & 0 \\ \sin\alpha & -\cos\alpha \sin\psi & -\cos\alpha \cos\psi & 0 \\ 0 & \cos\psi & -\sin\psi & 0 \\ A & B & C & 1 \end{bmatrix},$$

где  $A = X_0 \cos\alpha - Y_0 \sin\alpha$ ;  $B = -Z_0 \cos\psi + \sin\psi(X_0 \sin\alpha + Y_0 \cos\alpha)$ ;  $C = Z_0 \sin\psi + \cos\psi(X_0 \sin\alpha + Y_0 \cos\alpha)$ . Пространственное расположение систем координат и правила отсчитывания углов приведены на рис.3.1.4, пояснения к структуре компонент – в § 3.1.

**Эллиптические оболочки.** Кроме многогранников и параллелепипедов в качестве оболочек можно применить эллипсоиды. Замечательным свойством эллипсоида среди всех поверхностей второго порядка является замкнутость поверхности, и поэтому эллипсоид может быть использован как оболочка, охватывающая объект. Изменением размера и ориентации полуосей можно достаточно плотно охватить тело практически любой конфигурации.

Уиттед [149] впервые отметил применимость таких оболочек для сокращения вычислительных затрат при трассировании лучей. Для других методов синтеза эллиптическую форму оболочек применять нецелесообразно из-за необходимости поточечного или

полигонального представления эллипсоида, что значительно сложнее, чем описание многогранника.

Основная идея предлагаемого подхода заключается в следующем. На этапе конструирования объекта отдельные его части или весь объект окружают воображаемыми оболочками – эллипсоидами. Определить видимость или затененность от эллипсоида гораздо быстрее, чем для сложного, композиционного объекта или пространственно комбинирующихся примитивов. Тогда устанавливают те рецепторы, которые “видят” эллипсоид и тень от него и все дальнейшие вычисления по наблюдению самого объекта ведут только для этих, так называемых рабочих рецепторов. Отношение общего числа рецепторов к числу рабочих приближенно показывает, во сколько раз сократились вычислительные затраты. Приближенность (оценки) связана с неплотностью прилегания оболочки к телу объекта. Рецепторы, которые “видят” промежуток между оболочкой и объектом, определяют некоторую часть ненужных вычислений, на которую уменьшается теоретический коэффициент сокращения  $N$ .

Пусть некоторый геометрически сложный объект  $O$  расположен над поверхностью падения теней  $P$  (рис. 5.3.2). Окружим объект оболочкой  $A^+$  в виде эллипсоида, описываемого функцией  $f_{\mathcal{E}}(X, Y, Z)$ . Определим видимость эллипсоида для каждого  $ij$ -го рецептора экрана  $\mathcal{E}$ . Для этого необходимо определить факт наличия решений системы следующих четырех уравнений;

$$\begin{cases} f_{\mathcal{E}}(X, Y, Z) = [X \ Y \ Z \ 1] P_{\mathcal{E}} [X \ Y \ Z \ 1]^T = 0; \\ X = X_F + l_{ij} t, \ Y = Y_F + m_{ij} t, \ Z = Z_F + n_{ij} t, \end{cases}$$

где  $P_{\mathcal{E}}$  – матрица эллипсоида;  $t$  – параметр;  $X_F, Y_F, Z_F$  – координаты центра проекции;  $l_{ij}, m_{ij}, n_{ij}$  – направляющие векторы  $ij$ -го светового луча;  $l_{ij} = X_{ij} - X_F$ ;  $m_{ij} = Y_{ij} - Y_F$ ;  $n_{ij} = Z_{ij} - Z_F$ . Первое уравнение в системе описывает эллипсоид, остальные – прямую, проходящую через  $ij$ -й рецептор  $(X_{ij}, Y_{ij}, Z_{ij})$  и центр проскции.

Нетрудно показать из решения квадратного уравнения, к которому сводится данная система, что действительные решения существуют при условии

$$(\Pi P_{\mathcal{E}} \Phi T)^2 - \Pi P_{\mathcal{E}} \Phi P_{\mathcal{E}} \Phi^T \geq 0, \quad (5.3.1)$$

где  $\Pi = [X_F \ Y_F \ Z_F \ 1]$ ,  $\Pi = [l \ m \ n \ 0]$ .

В другой форме это условие можно представить в виде [39]

$$B_0^2 - 4A_0C_0 \geq 0, \quad (5.3.2)$$

где  $A_0 = A_{ij}^2 + B_{mj}^2 + C_{nj}^2 + 2Dl_{ij}m_{ij} + 2Em_{ij}n_{ij} + 2Fn_{ij}l_{ij}$ ;  $B_0 = l_{ij}(X_F + D Y_F + F Z_F + G) + m_{ij}(D X_F + B Y_F + E Z_F + H) + n_{ij}(F X_F + E Y_F + C Z_F + J)$ ;  $C_0 = [X_F \ Y_F \ Z_F \ 1] P [X_F \ Y_F \ Z_F \ 1]^T$ ;  $A, B, \dots$  – коэффициенты матрицы квадратичной поверхности

$$P_3 = \begin{bmatrix} A & D & F & G \\ D & B & E & H \\ F & E & C & J \\ G & H & J & K \end{bmatrix}.$$

Заметим, что с вычислительной точки зрения эффективно провести проверку условия (5.3.2) в два этапа. Сначала следует вычислить только значение  $A_0$  и проверить его равенство нулю. Если  $A_0=0$ , то условие (5.3.2) выполняется автоматически и уже нет потребности вычислять  $B_0$  и  $C_0$ . Если же  $A_0 \neq 0$ , то процедуру следует провести полностью, определив  $B_0$  и  $C_0$  и результат  $B_0^2 - 4A_0C_0$ . Состояние  $C_0$  невозможно, так как это означало бы пересечение оболочки центра проекции  $F$ , которая находится снаружи оболочки. Поэтому отдельная проверка  $C_0$  на равенство нулю не производится.

Проверка условий (5.3.1), (5.3.2) является простой операцией для процессора, что обеспечивает быстрый отбор всех рецепторов, которые “видят” оболочку. Только для этих рецепторов следует вести дальнейшие вычисления по определению видимости самого тела объекта, что чувствительно сокращает вычисления.

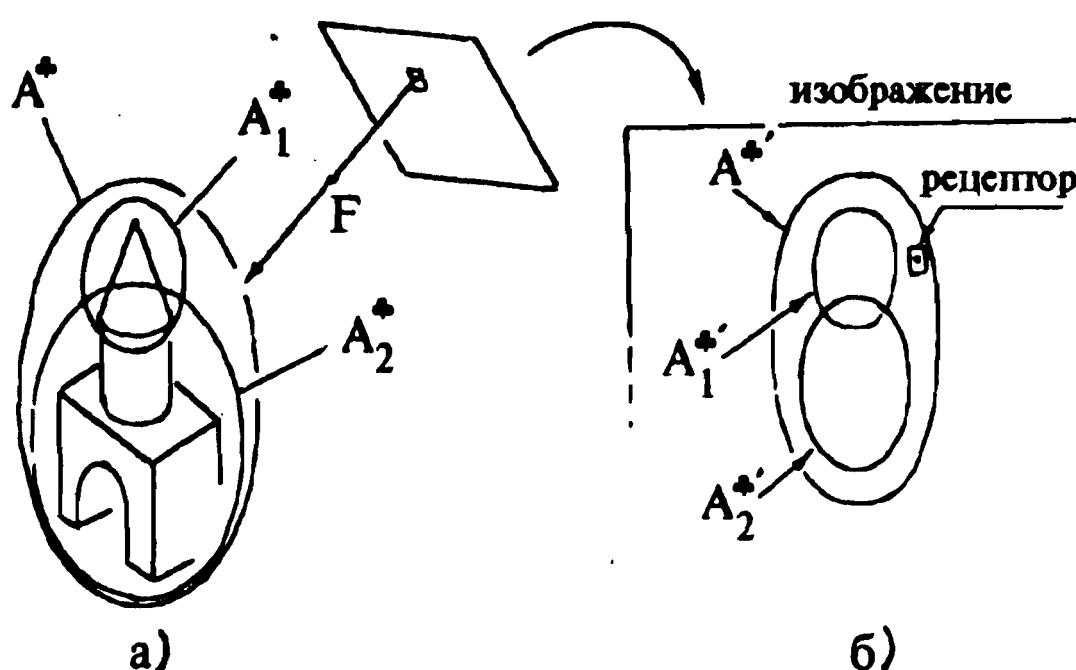


Рис. 5.3.2. Эллиптические оболочки (а) и их изображение (б)

Как и оболочки других форм, эллиптические оболочки могут составлять структуру с несколькими уровнями вложения. На

рис.5.3.2,а показана башня, оболочкой которой является  $A^+$ , купол имеет оболочку  $A_1^+$ , основание –  $A_2^+$ . На рис.5.3.2,б приведено изображение, на которое спроектированы оболочки в соответствующие двухмерные зоны –  $A^{+''}$ ,  $A_1^{+''}$ ,  $A_2^{+''}$ . Рецепторы, попадающие в зону  $A^{+''}$ , “видят” всю башню, те из них, которые попадают в зону  $A_1^{+''}-A_2^{+''}$ , – только купол, в  $A_2^{+''}-A_1^{+''}$  – только основание, в  $A_2^{+''}\&A_1^{+''}$  – либо купол, либо основание. Такое разбиение площади изображения на области позволяет сокращать объем вычислений в несколько раз.

**Оболочки-параллелепипеды.** Такая оболочка определяет габаритные размеры примитива или семейства примитивов вдоль координатных осей. Каждый примитив представляется как бы погруженным в ящик, стенки которого параллельны координатным плоскостям.

Пусть примитивы и их оболочки заданы (пересчитаны) в экранной системе координат. Как и ранее, ось о<sub>z</sub> экранной системы направим на объект, строки рецепторов разместим вдоль о<sub>y</sub>, столбцы – вдоль о<sub>x</sub>. Тогда строка рецепторов компланарна хо<sub>z</sub>, а стенки оболочек компланарны хо<sub>y</sub>, уо<sub>z</sub>, зо<sub>x</sub>.

При ортографической проекции легко установить (см.[9]) признак необходимости обработки очередным i<sub>j</sub>-м рецептором k-го примитива. Обозначим через  $y_{min_k}$ ,  $y_{max_k}$ ,  $x_{min_k}$ ,  $x_{max_k}$  максимальные ординаты и абсциссы k-й оболочки. Тогда i<sub>j</sub>-й рецептор должен обрабатывать k-й примитив, если  $y_{min_k} \leq y_{ij} \leq y_{max_k}$ ;  $x_{min_k} \leq x_{ij} \leq x_{max_k}$ . При невыполнении последнего условия k-й примитив надежно не попадает в поле зрения рецептора (рис.5.3.3).

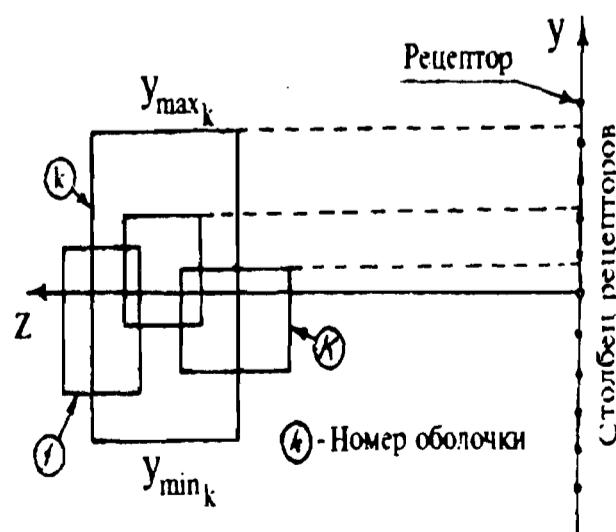


Рис. 5.3.3. Построение оболочек-параллелепипедов

Этот алгоритм может быть легко трансформирован для случая центральной проекции.

Положение граней параллелепипеда может быть определено аналитически как положение касательных плоскостей, параллельных  $xoy$ ,  $yoz$ ,  $zox$ . Это позволяет автоматически строить оболочки, что значительно повышает эффективность использования системы машинного зрения.

Покажем на примере поверхностей второго порядка правила определения габаритных точек примитивов. Под *габаритными точками* будем понимать точки касания поверхности второго порядка с касательными плоскостями, параллельными  $xoy$ ,  $yoz$ ,  $zox$ . Габаритные точки являются точками экстремума функции второго порядка  $F(x,y,z) = [x \ y \ z \ 1]P[x \ y \ z \ 1]^T$  в направлениях вдоль  $ox$ ,  $oy$ ,  $oz$ , и одновременно они лежат на самой поверхности. Поэтому координаты габаритных точек определяются решением систем следующих троек уравнений в направлении  $oz$   $dF/dx=0$ ,  $dF/dy=0$ ,  $F(x,y,z)=0$ ;  $oy$   $dF/dx=0$ ,  $dF/dz=0$ ,  $F(x,y,z)=0$ ;  $ox$   $dF/dy=0$ ,  $dF/dz=0$ ,  $F(x,y,z)=0$ . Значения частных производных определяются следующим образом в обозначениях § 2.2, 3.4.3, принятых для поверхности второго порядка:  $dF/dx=Ax+Dy+Fz+G$ ;  $dF/dy=Dx+By+Ez+H$ ;  $dF/dz=Fx+Ey+Cz+J$ . Решение каждой из приведенных систем сводится к квадратному уравнению. Если существует два действительных корня, то они определяют две габаритные точки в соответствующем направлении.

Таким образом обеспечивается автоматическое построение параллелепипедных оболочек вокруг поверхностей второго порядка. В общем случае только эллипсоид имеет две габаритные точки в каждом направлении, поверхности другого вида простираются в бесконечность, поэтому их оболочка может быть представлена как полупространство.

### 5.3.2. ВОЗМОЖНОСТИ ИСПОЛЬЗОВАНИЯ ТРАНСПЬЮТЕРОВ В МАШИННОЙ ГРАФИКЕ

Машинная графика относится к области очень смких по вычислительным затратам задач. Основное время отводится на арифметические операции, поэтому даже самые изощренные алгоритмические и программистские приемы не позволяют добиваться потребительски приемлемых результатов при использовании однопроцессорных машин. Значительно более плодотворным, хотя и более дорогим по затратам на аппаратуру, оказался подход, основанный на распараллеливании вычислений. Техническими устройствами, реализующими этот механизм,

являются матричные (векторные) процессоры [50] и транспьютеры [56, 96, 139].

Матричный процессор работает под управлением внешней однопроцессорной машины и по отношению к последней выступает как внешнее устройство. Матричный процессор очень быстро выполняет основные операции над матрицами большого размера: сложение, вычитание, умножение и т.д. Обычно этим функциям матричный процессор снабжен за счет комбинации старых или введения новых операций. Пользователь может расширить перечень с помощью комбинации старых или введения новых операций. При расчете изображения, управляющая программа, реализованная на одном процессоре, в нужные моменты пересыпает массив в матричный процессор, где в соответствии с командой выполняется та или иная операция. Результат пересыпается по каналу связи в обратном направлении.

Значительно большими возможностями в параллельной обработке обладают транспьютерные системы, которые появились в среде вычислительной техники как ответ на требования практики к мультипроцессорным структурам [119]. Данные структуры должны обеспечивать коммуникацию между процессорами как через аппаратуру, так и через программное обеспечение, они должны осуществлять программирование на языке высокого уровня, включающем элементы коммуникации между процессорами, и, наконец, каждый отдельный процессор должен обладать высокой работоспособностью, чтобы сеть процессоров не была очень большой.

Этим требованиям практически полностью удовлетворяет микропроцессор, получивший название *транспьютер*. Каждый блок в транспьютере имеет четыре очень быстрых последовательных интерфейса, которые называются звеньями. Эти звенья можно использовать для коммуникации с соседними транспьютерами или в качестве интерфейсов на остальные части всей системы. Они легко программируются благодаря некоторым особенностям математического обеспечения. Разрядность выпускаемых фирмой INMOS транспьютеров составляет 32 бита. Транспьютеры T800 и T9000 имеют математический сопроцессор, что наряду с оригинальной архитектурой обеспечивает значительно большее быстродействие даже одного такого микропроцессора, чем у известного микропроцессора Pentium. При параллельном включении транспьютеров, судя по проспектам, удается достичь быстродействия 100 млн.оп./с (восемь транспьютеров). Вычислительная мощность прямо пропорциональна числу используемых транспьютеров.

Распределение памяти между транспьютерами может быть организовано строками (столбцами) или в шахматном порядке. Строковое распределение означает, что первый транспьютер обрабатывает строки 1,  $n+1$ ,  $2n+1$ , ..., второй – 2,  $n+2$ ,  $2n+2$ , ... и т.д., где  $n$  – число транспьютеров в системе. При шахматном распределении каждый транспьютер “отвечает” за свою клетку на шахматном поле экрана. Строковое (столбцовое)

распределение позволяет более эффективно использовать ресурс системы, так как при локальных по полю изображения вычислительных перегрузках все транспьютеры используются равномерно.

Транспьютер может быть использован в качестве отдельного самостоятельного устройства, обеспечивающего производительность около 10 млн.оп./с [56]; при этом для программирования используется широкий набор стандартных высокоуровневых языков, так как архитектура транспьютера ориентирована на эффективное применение компиляции. Для полной реализации возможности объединения транспьютеров в сети матрицы целесообразно применять специализированный язык Оккам. Он специально разработан для реализации параллельной обработки, в ходе которой множество частей системы функционируют по отдельности и взаимодействуют друг с другом.

Оккам позволяет программисту представлять программу в виде параллельных процессоров, взаимодействующих друг с другом посредством пересылки сообщений по каналам обмена информацией. Ввод и вывод синхронизированы: когда процессы как ввода, так и вывода в одном и том же канале готовы к коммуникации, выходное значение переписывается из процесса вывода в процесс ввода без явного использования буфера, после чего оба процесса продолжают независимо выполняться до следующей коммуникации (если она последует).

Алгоритмы машинной графики органично приспособлены к реализации на транспьютерной сети. Например, обратное трассирование лучей обеспечивает полную независимость вычислений для каждого рецептора. Поэтому простейшим построением вычислительной системы является независимое использование отдельных транспьютеров для отдельных рецепторов, или чередующихся отдельных строк (столбцов), или полей, разбросанных в виде шахматного поля. Вычислительная нагрузка на различные части экрана рецепторов распределяется очень неравномерно. Трассирующий луч, не пересекающий объекты сцены, отрабатывается очень быстро, так как процедуры пересечения луча с примитивами завершаются по сокращенному пути из-за отсутствия решений. Поэтому транспьютеры должны быть распределены по экрану как можно равномернее. Удачным построением можно назвать столбовое (строковое) чередование, что еще удобно и с точки зрения программирования.

Использование параллельных процессоров позволяет добиваться практически приемлемых результатов по быстродействию в задачах машинной графики. Изображение 1024x1024 элементов для объекта, состоящего из 500 примитивов (около 2000 поверхностей, из них половина – второго порядка, половина – первого порядка) с тенями, тонами и зеркальностью, реализуется на сети из восьми транспьютеров T9000 за 10...15 с. Изображения полигональных полей без теней выполняются за сотые доли секунд, а полигональных сеток – в реальном масштабе времени.

## 6. ПРИКЛАДНОЕ ИСПОЛЬЗОВАНИЕ ТРЕХМЕРНОЙ МАШИННОЙ ГРАФИКИ

### 6.1. АРХИТЕКТУРНОЕ ПРОЕКТИРОВАНИЕ

В архитектурном проектировании существует задача анализа внешнего вида архитектурного сооружения до строительства в целях выбора облика здания, его согласования с ландшафтом и исторической застройкой. Внешний облик здания тесно связан с внутренней планировкой, а последняя, определяющее воздействует на интерьер. Таким образом, архитектурное проектирование представляет собой комплексную проблему в области художественного трехмерного моделирования [98]. Традиционное решение обычно связано с физическим макетированием здания из пенопласта, пластилина, гипса и других подходящих материалов. Можно назвать следующие основные недостатки этого подхода: трудоемкость изготовления, сложности корректировки модели, значительные затраты на изменение оптических свойств поверхностей, узкий диапазон изменения условий освещения, ограниченность диапазона обзора и положения точки наблюдателя. Трудоемкость изготовления обусловлена, как правило, ручным трудом в единичном уникальном производстве. Любое изменение конструкции связано с физическими переделками модели.

Как правило, физические модели наблюдают при текущем естественном или фиксированном искусственном освещении, что не дает полной картины изменения освещенности и теней в течение светового дня. Заметим, что тень является важным составным элементом облика архитектурного сооружения. Наконец, физические модели наблюдают “с высоты птичьего полета”, что не эквивалентно реальным условиям наблюдениям человеком в действительных масштабах, и, как правило, с направлением взгляда снизу-вверх или из окна. Последний недостаток очень часто проявляется в том, что эффектно смотрящееся здание в плане при физическом наблюдении человеком с поверхности Земли представляется негармоничным или ординарным.

Реальной альтернативой традиционному подходу является использование трехмерной машинной графики в качестве инструмента моделирования. Сущность метода вполне ясна: вместо физического макета создают математическую модель и затем отображают ее с помощью ЭВМ [106].

Моделирование объекта—здания осуществляют методом сплошных тел или полигональными полями. В качестве алгоритма построения изображения могут быть выбраны методы сканирующей строки, Робертса, трассирования лучей [130, 132, 136]. Применение последнего подхода оправдано только в случае очень сложных объектов с криволинейными поверхностями. Для большинства объектов эффективным является интервальный метод построчного сканирования, так как многие строительные элементы современных зданий образованы плоскостями [26]. Подавляющее большинство коммерческих программных продуктов архитектурного проектирования (Autocad, ArcInfo Tin, Syscad, Gable ...) базируются на использовании полигональных полей или сплошных тел-многогранников в качестве модели объекта и метода сканирующей строки в качестве модели визуализации. Изображения, построенные такими системами, характеризуются следующими изобразительными признаками и особенностями: псевдоцветовое окрашивание поверхностей; отсутствие собственных и падающих теней; ортогональное проецирование. Эффективность метода визуализации позволяет за несколько секунд отображать объекты, состоящие из сотен поверхностей, при использовании 386-го микропроцессора.

Применение машинной графики позволяет спрогнозировать облик и для наземного наблюдателя, и для обзора с крыши, из окна, с самолета, с соседних зданий. Можно достаточно простыми средствами изменять окраску стен и другие их оптические свойства. Возможности компьютерного синтеза изображений позволяют имитировать современные строительные материалы и моделировать, например, сплошное покрытие поверхности матированными стеклами. В таком случае корректный облик сооружения с зеркальным отражением окружающей застройки может быть построен только с помощью метода трассирования лучей.

Одним из необходимых условий для прогнозирования облика архитектурного сооружения является не только отображение формы здания, но и текстуры, цвета, отражательной способности, геометрического рисунка на поверхности, облика окон и фигурных украшений. Непосредственное математическое описание последних и синтез их изображений трудоемки и требуют значительных вычислительных мощностей. Для разрешения затруднений подобного рода без ущерба для цели самого проектирования применяют следующее упрощение. На этапе математического

моделирования отдельно описывают геометрическую форму здания с оптически однородными поверхностями без окон, плоских узоров, текстуры и других подобных деталей. Эти накладывающиеся на фасад элементы создаются отдельно в виде изображений и в оцифрованном виде вводят и хранят в памяти машины. После генерации облика здания картину стены автоматически переносят на генерированное изображение по опорным или реперным точкам. *Реперными точками* в данном случае будут называться соответствующие точки на изображении плоского узора и изображении объекта.

На рис.6.1.1 показан случай нанесения картины окон на внутреннюю поверхность цилиндрического здания. На первом этапе (а) проектировщик создает трехмерную модель здания в системе  $OXYZ$  в виде сплошного однородного тела. Художественным, фотографическим или иным методом строится плоская картина текстуры или узора фасада (рис.1.6.1,б) в некоторой прямоугольной системе  $o'x'y'$ . На этой картине выбираются опорные точки 1,2,...,10 и проставляются позиции, соответствующие им в трехмерной модели (рис.1.6.1,а).

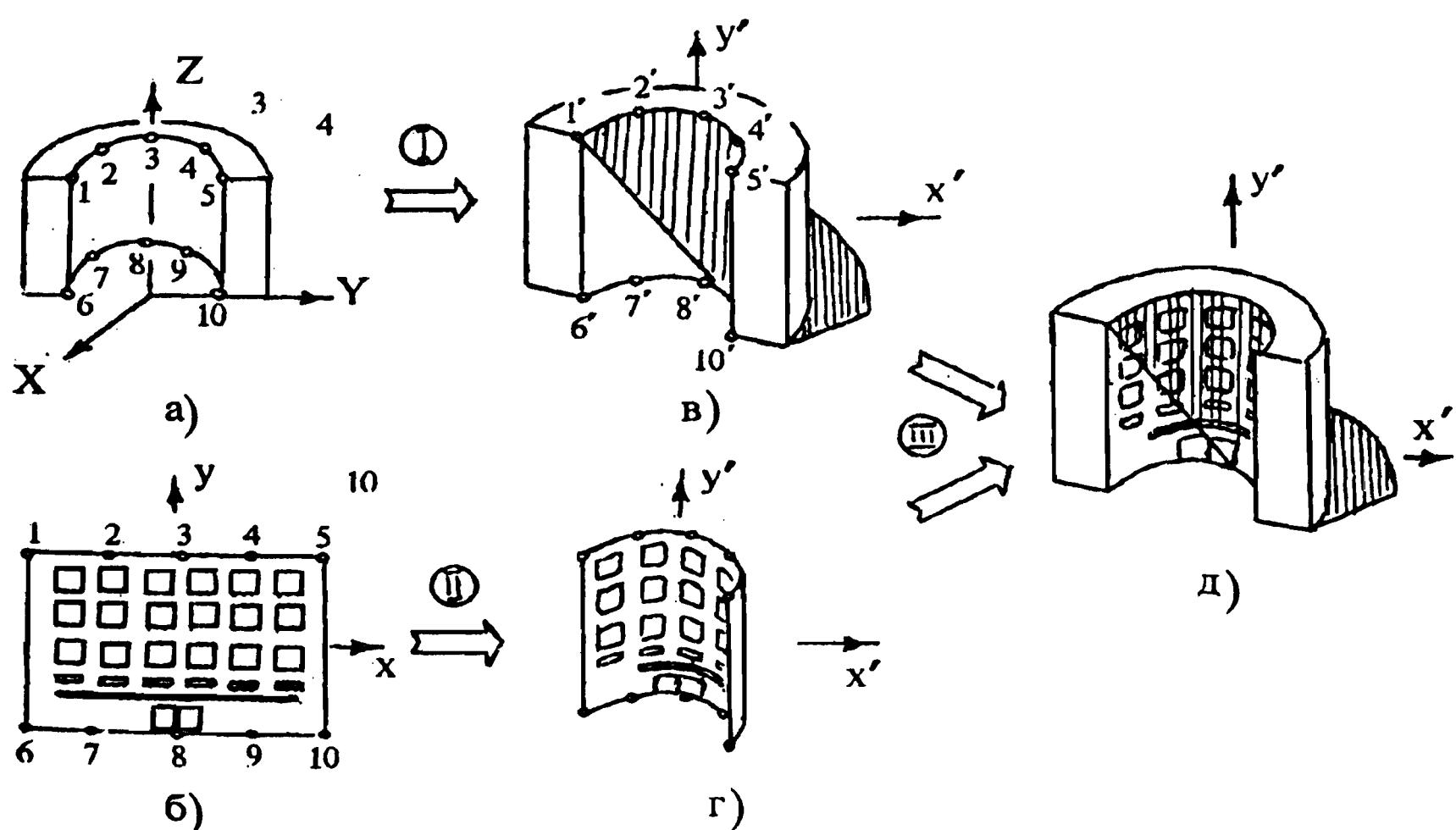


Рис. 6.1.1. Нанесение рисунка на поверхность здания методом трансформирования

Затем методами машинной графики в заданных условиях строится изображение здания в системе координат изображения  $o'x'y'$  (рис.1.6.1,в), одновременно с которым определяется положение изображений  $1', 2', \dots, 10'$  опорных точек в системе  $o'x'y'$ .

Узор (рис.1.6.1,б) трансформируют по опорным точкам ( $1, 2, \dots, 10$  в  $1', 2', \dots, 10'$ ) в положение (рис.1.6.1,г) на изображении и совмещают два изображения (рис.1.6.1,г и в) в единое целое (рис.1.6.1,д), корректное по характеру картины фасада, теням, тонам и др. Часть трансформируемого фасада может быть закрыта другими деталями; в этом случае она удаляется вручную.

Для развертывающихся поверхностей может быть построена автоматическая система перенесения узора, основанная на использовании метода трассирования лучей (рис.6.1.2). Пусть визирующий луч встречается с объектом в точке  $A(X, Y, Z)$  (рис.6.1.2,а); тогда, как ясно из рисунка, координаты  $xu$  плоской развертки фасада определяются по правилу  $y = Z; x = \pi R^{\text{arctg}(X/Y)}$ . В этих координатах определяется уровень тона предварительно построенного узла (рис.6.1.2,б), который и используется в качестве мультипликативного коэффициента яркости при вычислении освещенности. При таком подходе удается избежать осложнений при экранировании поверхностей и добиться полного автоматизма.

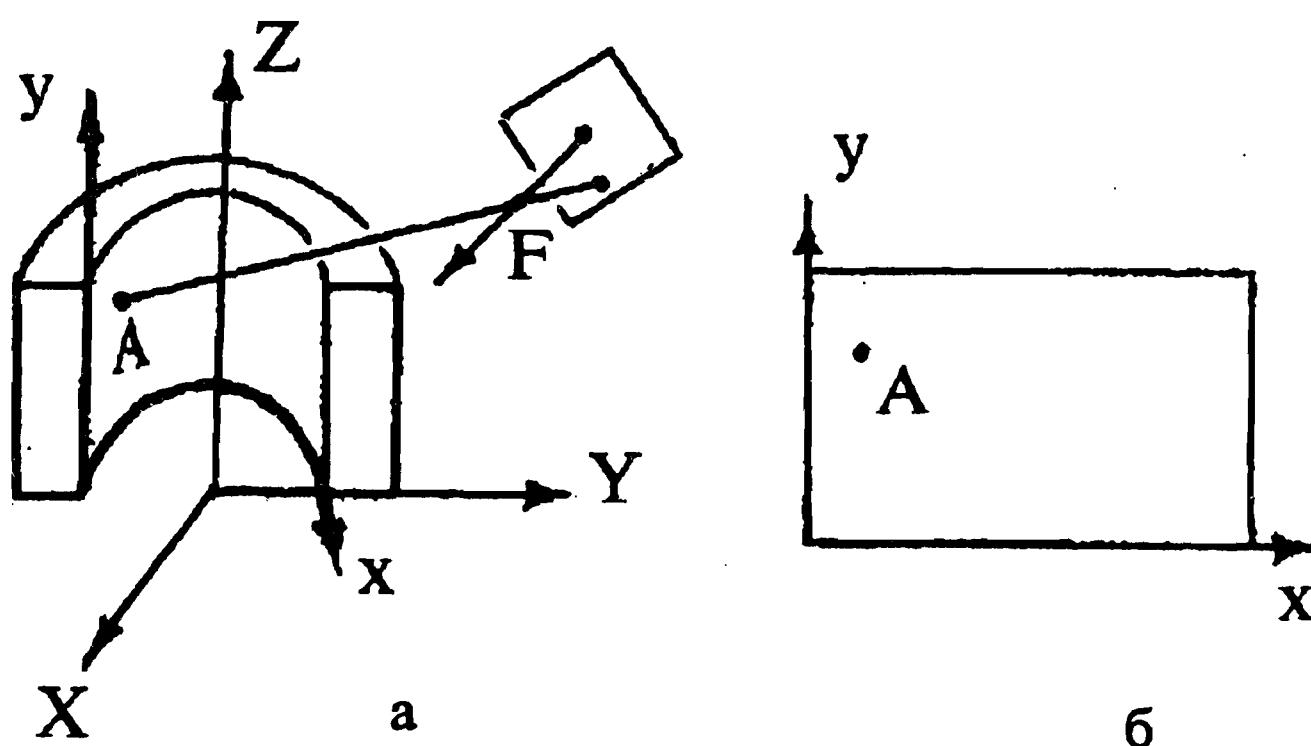


Рис. 6.1.2. Нанесение рисунка на развертывающуюся поверхность

Машинная графика является незаменимым средством архитектурного проектирования новых зданий в местах исторической застройки, где внешний вид здания должен не только разрушать

специфический стиль района, но и гармонировать с ним. Многочисленные работы по этой теме основаны на идее фотографирования района застройки с нескольких информативных ракурсов и впечатывании в полученные фотографии синтезированных на ЭВМ изображений нового здания. Анализ полученных комбинированных изображений позволяет значительно повышать объективность принятия решения в выборе облика здания. Рассмотрим этот подход более подробно.

Как и во многих других прикладных задачах, сначала создают математическую модель трехмерной математической формы и оптических свойств поверхности проектируемого объекта. Описание хранят в базе данных ЭВМ. Затем производят натурное фотографирование панорамы исторической застройки в районе спроектированного места строительства. Фотографирование выполняют с нескольких информативных позиций и каждый раз условия фотографирования вводят в память ЭВМ. К этим условиям относятся: положение камеры в системе координат места застройки, ориентация оптической оси, фокусное расстояние аппаратуры съемки, положение Солнца в системе координат места застройки и другие параметры (вид и уровень облачности, относительное отверстие объектива, удаление от объекта). Последние параметры позволяют более корректно рассчитывать освещенность изображения будущего здания, согласующуюся со световым фоном всей сцены. Заметим, что некорректный расчет по освещенности всегда может быть интерактивно исправлен на комбинированном изображении. Затем методами машинной графики синтезируют изображение проектируемого объекта. При этом объект размещают в выбранном месте в системе координат строительной площадки, а условия синтезирования выбирают такими же, какими они были в момент реальной съемки. В результате такой операции синтезированное изображение по масштабу, тонам, теням, ракурсу и т. д. полностью соответствует условиям реального фотографирования в каждой позиции наблюдения. Частота дискретизации синтезированного изображения должна быть достаточно высокой для достижения однородности по качеству машинного и реального участков результирующего (комбинированного) изображения. Критерии частоты дискретизации изложены в § 5.1. Последний этап заключается в интерактивном впечатывании синтезированного изображения объекта на поле оцифрованной фотографии района застройки.

Полученное комбинированное изображение позволяет объективно оценивать качество проектирования. В случае необходимости в модель здания могут быть внесены изменения и повторен цикл синтеза и впечатывания для достижения требуемой эстетичности конечной картины.

## 6.2. РАСПОЗНАВАНИЕ ВИДЕООБРАЗОВ

Распознавание видеообразов уже долгое время остается популярной областью [120], привлекающей возможности самых разных областей математики, средств вычислительной техники и методов программирования. Наиболее распространенными методами распознавания изображений являются экстремально-корреляционные, статистические, структурно-лингвистические, геометрических инвариантов [2]. Прежде чем показать связь этих подходов с машинной графикой, дадим их общую краткую характеристику, определим те причины, которые задерживают методы распознавания в рамках лабораторий и оставляют проблему далекой от широкого практического применения.

Идея экстремально-корреляционного решения сводится к вычислению корреляционной функции  $K(x,y)$  исходного изображения  $I(x,y)$  и изображения эталона  $\Theta(x,y)$ . Вычислительные аспекты этого процесса изучены в [42, 45]. Если в исходном изображении найдется фрагмент, идентичный  $\Theta(x,y)$ , то в этом месте  $I(x,y)$  возникнет локальный экстремум. Все поле корреляционной функции подвергают высокочастотной фильтрации для подавления шумов, размытых пиков и пороговым методом селектируют положение эталонного объекта. Недостатками экстремально-корреляционного метода являются высокая чувствительность к несовпадению масштаба, ориентации, яркости и значительный объем вычислений. Последний недостаток преодолеваются путем параллельных вычислений на множестве специализированных микропроцессоров, достигая высокой скорости из-за потери универсальности вычислителя. Метод может давать “ложные тревоги” в случае отсутствия поискового объекта в исходном изображении, и, кроме того, имеется субъективизм в выборе порога, вида высокочастотного фильтра и некоторых других параметров. Положительной чертой метода является то, что он работает непосредственно с изображением, а не с производными признаками.

Статистические методы [55] основываются на идее сходства

некоторых статистических характеристик (математического ожидания, дисперсии, моментов высшего порядка, гистограмм ...) изображения эталона и их проявлений на анализируемом изображении. Такой подход пригоден для решения задачи распознавания после того, как объект обнаружен, что само по себе остается проблемой. Кроме того, метод анализирует лишь часть содержащейся на изображении информации, игнорируя такие важные характеристики, как форма и структура. Метод оперирует небольшим числом сравнительно просто вычисляемых признаков, что делает его пригодным в частных случаях.

Сущность структурных методов заключается в выделении на распознаваемом и эталонном изображениях некоторых признаков и их связей, кодировании признаков и связей на формальном языке и решении задачи идентичности семантических образов эталона и анализируемого изображения. В качестве признаков, например, могут использоваться фрагменты контурной линии изображения объекта, такие как прямые и дуги определенных направлений. Описываемый подход инвариантен к масштабу, повороту, яркости, однако сам процесс извлечения признаков из изображения остается проблемным и обычно решается субъективным вмешательством.

Метод геометрических инвариантов оперирует такими признаками изображения, как площадь фигуры и площадь выпуклой оболочки, длина периметра, величины углов смежных контурных линий, и их соотношениями в различных сочетаниях. Для выделения этих признаков эталон и анализируемое изображение подвергают бинаризации. Метод может быть применен только в частных случаях.

Все перечисленные методы объединяет общая идея сравнения эталонного и анализируемого изображений непосредственно или через вторичные признаки. Вне зависимости от метода качество сравнения в сильной степени зависит от идентичности условий освещения и наблюдения анализируемого и эталонного изображений.

### 6.2.1. РАСПОЗНАВАНИЕ ОБРАЗОВ ПО АДАПТИВНЫМ ЭТАЛОНАМ

В реальных условиях естественного или искусственного освещения изображения трехмерных объектов подвержены значительной изменчивости в зависимости от положения источников освещения, ракурса наблюдения, параметров

съемочной аппаратуры. Например, в течение светового дня из-за перемещения Солнца один и тот же трехмерный объект характеризуется не только различной картиной полутонаов, но и разными тенями. В сценах с искусственным освещением из-за влияния нескольких источников образуются множественные падающие тени. Изменение изобразительных свойств в зависимости от положения источников освещения подробно описано и проиллюстрировано в [58,97]. Вариация ракурса приводит к проявлению различных форм и сторон трехмерного объекта.

Совершенно очевидно, что накопление эталонных изображений во всем диапазоне условий невозможно, так как учит только таких параметров, как ракурс, положение источника света, фокусное расстояние, разрешающая способность, уже выходит за рамки первых сотен изображений для одного объекта. Если учитывают  $N$  параметров ( $n = 1(1) N$ ) с выборкой  $K_n$  числа значений внутри каждого параметра, то требуемое число эталонных изображений должно достигать

$$\prod_{n=1}^N K_n$$

Основная идея компьютерного моделирования эталонных изображений сводится к хранению в памяти ЭВМ описания геометрической формы объекта поиска вместо его изображения и генерации последнего в условиях получения анализируемого изображения [18,22].

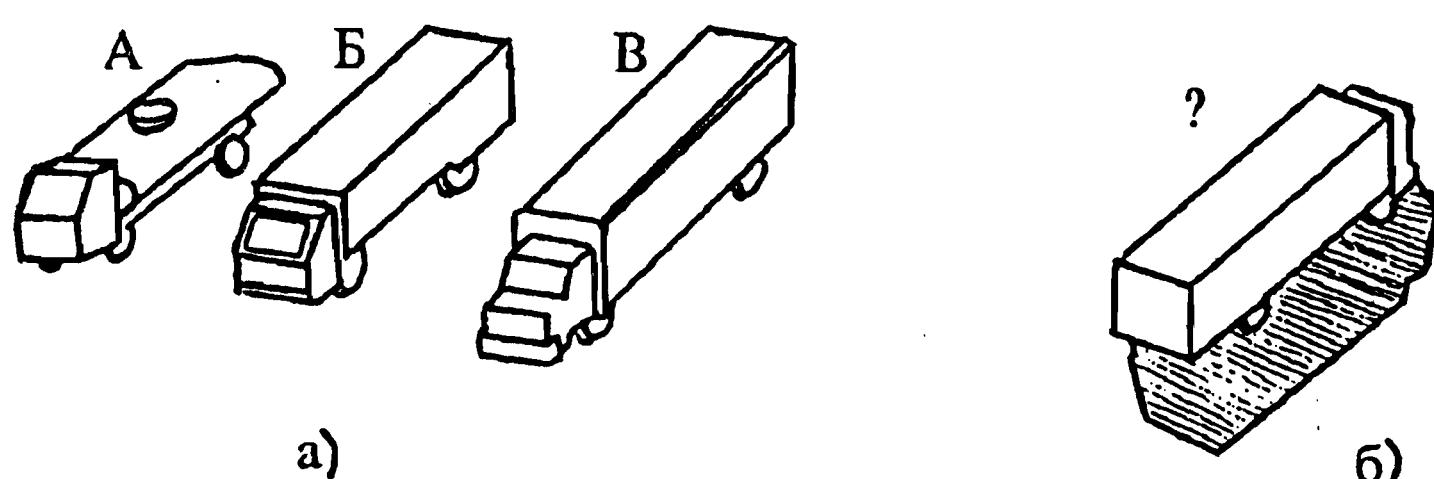


Рис. 6.2.1. Объекты, наблюдаемые в текущих условиях, (а) и объект – престендент на распознавание (б)

Рассмотрим задачу автоматической навигации самолета, заходящего на посадку, методом оптической привязки по изображению трехмерного здания, расположенного рядом с

местом посадки. Очевидно, что картина здания будет зависеть от времени дня, высоты полета, направления оптической оси, облачности и других условий, поэтому невозможно обойтись единственным эталоном, а хранение множества эталонов во всем диапазоне изменения возможных условий недостижимо из-за перегрузки памяти машины. Поэтому целесообразно на борту хранить одно описание трехмерной геометрии и оптических свойств объекта привязки вместо множества изображений-эталонов. Непосредственно перед посадкой должны быть определены условия наблюдения и освещения (высота полета, расчетное положение Солнца и т.д.). При этих условиях генерируется эталонное изображение, которое обнаруживается коррелятором по текущему изображению.

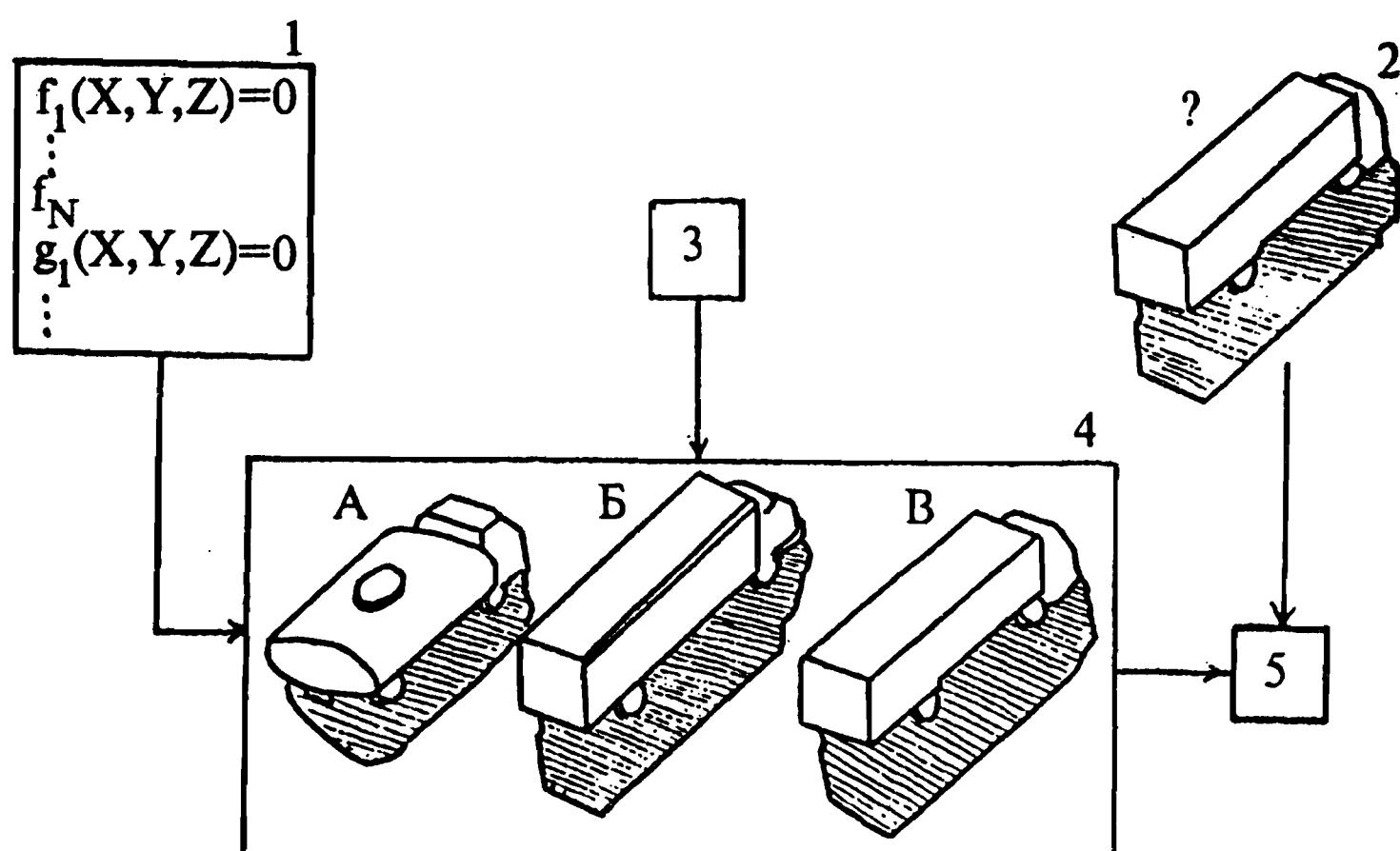


Рис. 6.2.2. Схема распознавания объекта по адаптивным эталонам

Способ распознавания трехмерного объекта (рис. 6.2.1,б) среди нескольких претендентов (рис. 6.2.1,а) проиллюстрирован на рис. 6.2.2. Предварительно и независимо от текущего наблюдения за объектом поиска в памяти ЭВМ создают банк математических описаний 1 трехмерной геометрической формы и оптических свойств всех объектов-эталонов, претендующих на распознавание. Определяют условия 2 наблюдения и освещения,

при которых получено реальное изображение 3. Эти условия передаются в систему компьютерного моделирования 4, где математическое описание каждого объекта преобразуется в изображение, но уже в условиях реально действующих факторов, т. е. тех же самых, что и у реального образа 3. В банке 4 формируется таблица эталонных изображений, имеющих ракурсы и тени, соответствующие реальному изображению, что является необходимым условием успешного автоматического распознавания, выполняемого в блоке 5.

### 6.2.2. РАСПОЗНАВАНИЕ ОБРАЗОВ В УСЛОВИЯХ НЕИЗВЕСТНОЙ ОРИЕНТАЦИИ ИХ КОРПУСОВ

Рассмотренная в § 6.2.1 задача распознавания изображений по адаптированным эталонам предполагает известную ориентацию тела обнаруживаемого объекта в системе координат, являющейся общей для объекта, приемника и источника света. В общем случае это предположение не оправдывается, так как объект на изображении может ориентироваться случайным образом по отношению к общей системе координат.

Рассмотрим постановку задачи на конкретном примере распознавания объемных образцов на ленте транспортера. Решение принимается процессором видеоробота. Исходная информация поступает от видеокамеры, которая однократно включается в том случае, когда неизвестный ей объект находится перед объективом. В качестве объектов могут быть, например, простые геометрические тела: шар, куб, конус, призма и др. Задачей робота является распознавание типа объекта и его ориентации для правильного захвата соответствующим манипулятором. Априори ориентация объектов на плоскости транспортера неизвестна. Положение и тип источника определены.

В решении этой задачи невозможно обойтись единственным эталонным изображением, адаптированным к условиям наблюдения, так как картина эталона (форма, тона, тени и др.) значительно меняется в зависимости от ориентации тела объекта. Подобные проблемы встречаются и в распознавании плоских фигур неизвестной ориентации. Для таких фигур проблема решается посредством вращения с небольшим приращением угла поворота и корреляции на каждом шаге поворота очередной картины эталона с анализируемым изображением. В описываемом выше случае недостаточно просто вращать эталонное изображение, так как при

этом не происходит требуемой перестройки тонов, теней, формы. Поэтому целесообразно вращать не изображение, а математическую модель объекта и в каждом шаге поворота синтезировать новое изображение с присущими такому положению тела объекта изобразительными свойствами. Сгенерированная серия изображений объекта одного типа проверяется путем корреляции с анализируемым изображением. Подобные операции должны быть проведены для всех типов объектов. Например, если претендентами на распознавание являются три объекта: куб, конус и призма, а поворот корпуса совершается в пределах  $360^\circ$  с шагом  $36^\circ$ , то семейство эталонных образцов должно включать 30 изображений: по 10 на куб, конус и призму. Каждое из эталонных образов-изображений должно быть подвергнуто корреляции с анализируемым изображением. Глобальный максимум среди полей всех корреляционных функций индицирует действительное положение объекта в поле обзора, а соответствующий эталонный образец указывает на тип и ориентацию.

Принцип распознавания объекта среди нескольких объектов при неизвестной ориентации корпуса объекта показан на рис.6.2.3.

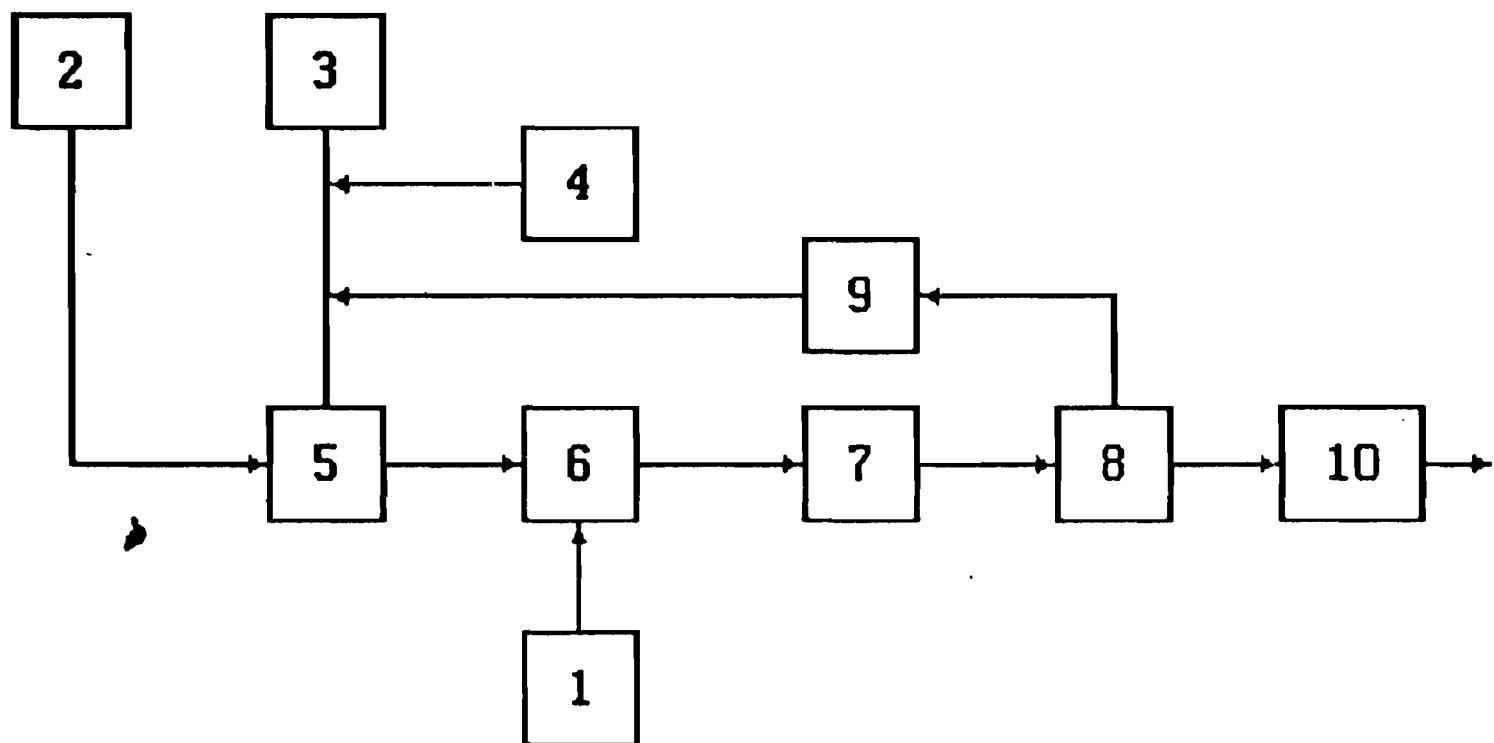


Рис. 6.2.3. Схема распознавания объекта по адаптивным эталонам в условиях неизвестной ориентации корпуса объекта

Датчик поставляет в систему исходное анализируемое изображение 1. Предварительно в памяти ЭВМ создают и хранят математические описания объектов-эталонов 2. Определяют условия наблюдения и освещения 3. Устанавливают начальное положение (азимут) тела для каждого эталонного объекта 4 и синтезируют

методами машинной графики 5 изображение каждого эталонного объекта в заданной позиции поворота. Проводят корреляцию 6 анализируемого изображения с каждым эталонным изображением, а полученные корреляционные функции сохраняют в памяти. Проверяют факт завершения полного поворота тела каждого объекта (азимут проходит полный круг) 8 и в случае его невыполнения осуществляют приращение 9 азимута на некоторый угол (обычно не более  $150^\circ$ ) и продолжают процесс в блоках 5 – 7 с наполнением новых корреляционных функций. Если полный поворот завершился, то производят выбор 10 глобального максимума среди всех сохранных в блоке 7 корреляционных функций: Этот максимум индцирует положение объекта, его тип и ориентацию.

### 6.2.3. РАСПОЗНАВАНИЕ ОБРАЗОВ ПО АДАПТИРОВАННЫМ ЭТАЛОНАМ В ПИРАМИДАЛЬНОЙ СТРУКТУРЕ

Идея пирамидально-рекурсивного представления и распознавания изображений хорошо известна [113]. Она заключается в представлении эталона и анализируемого изображения в виде структуры слоев изображений с различным разрешением. На самой вершине такой пирамиды располагается обзорное изображение, состоящее из малого числа пикселов, в основании пирамиды – исходное, наиболее детальное и соответственно состоящее из максимального числа пикселов. Освещенность каждого пикселя верхнего уровня образована усреднением освещенности групп  $2 \times 2$ ,  $3 \times 3$ , ...,  $n \times n$  пикселов ближайшего нижнего уровня. Сущность распознавания заключается в последовательной идентификации, начиная с самого верхнего уровня, соответствующих уровней пирамид анализируемого и эталонного изображений. Так как задача может быть решена на некотором среднем уровне, то уже этим достигается повышение эффективности процесса из-за сокращения вычислительных затрат.

Аналогично этому подходу при распознавании по адаптированному эталону нет необходимости генерировать этalon сразу до уровня высокого разрешения, так как это связано со значительными вычислительными затратами. Генерацию следует начинать с самого верхнего уровня будущей пирамиды, постепенно спускаясь вниз. Информация из предыдущего уровня должна полностью использоваться на последующем. После генерации очередного уровня проводится его идентификация с соответ-

вующим уровнем пирамиды анализируемого изображения. При превышении максимума корреляционной функции некоторого порога процесс прекращается. Для повышения качества распознавания значения корреляционных функций всех задействованных уровней необходимо анализировать совместно.

Важно отметить, что эффективность вычислительного процесса тесно связана с возможностью использования информации предыдущего уровня пирамиды эталона для построения последующего. Есть несколько путей реализации этого подхода. Одним из них является возможность использования пикселов изображения верхнего уровня среди вычисляемых пикселов нижнего уровня, как показано на рис.6.2.4, где 1 – значения яркости, вычисляемые для верхнего уровня; 2 – для нижнего. Другой подход предполагает когерентность соседних пикселов, поэтому обработка строится исходя из предположения: наиболее вероятно, что рецепторы 2 "видят" те же поверхности, что и ближайшие к ним рецепторы 1 из нижнего уровня (рис.6.2.4,б).

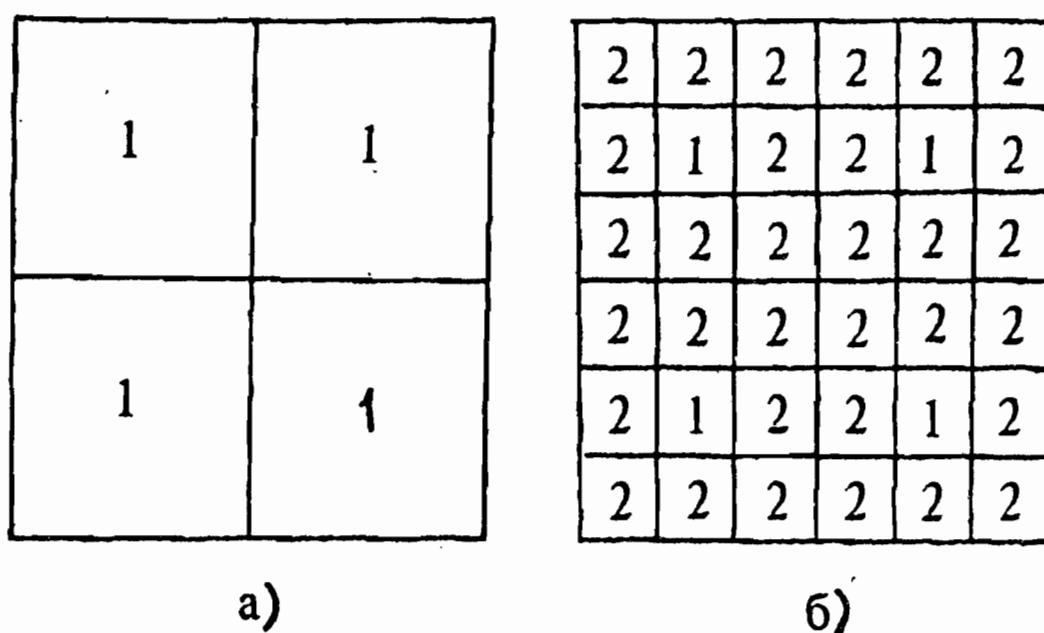


Рис. 6.2.4. Структура рецепторов соседних уровней пирамиды изображения: верхнего (а) и нижнего(б)

Схема, иллюстрирующая способ распознавания образов в пирамidalной структуре, показана на рис.6.2.5.

Анализируемое изображение 1 является объектом, на котором осуществляется поиск образца по эталону. Пирамида анализируемого изображения 2 представляет серию изображений с ранжированным уровнем загрузления. Обычный путь получения пирамиды – загрузление каждого текущего уровня методом “сворачивания” четверок 2x2 смежных пикселов в один.

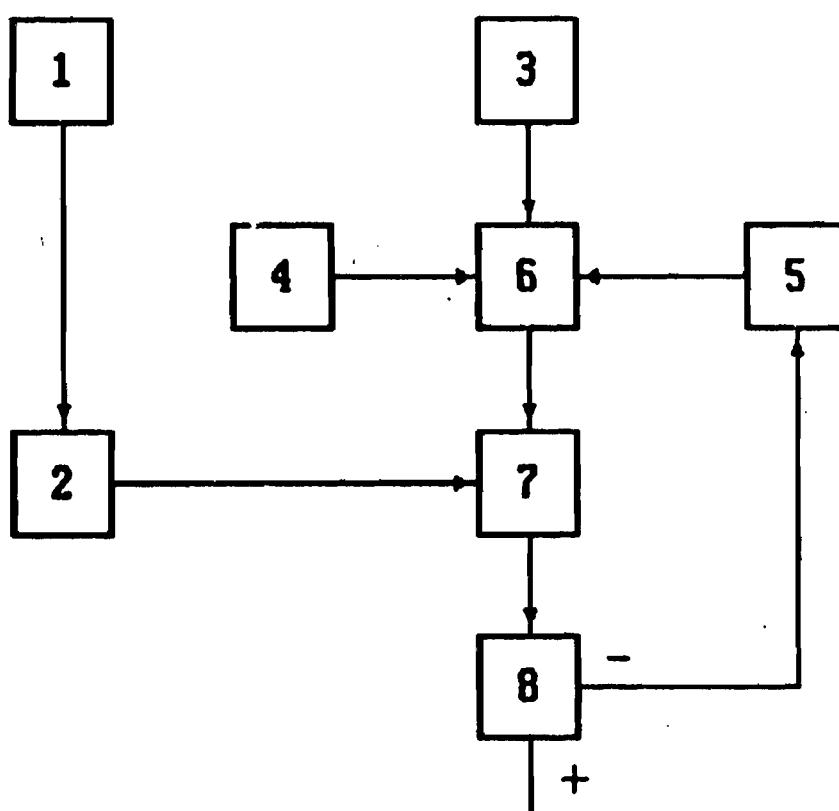


Рис. 6.2.5. Схема распознавания объекта по адаптивным эталонам в пирамидальной структуре

Математическое описание формы эталона 3 представляет собой формализованную модель трехмерной геометрии и оптических свойств поверхности эталонного образца. Условия наблюдения и освещения 4 – это информация о положении источников света, положении и ориентации датчика, с помощью которого было получено анализируемое изображение, а также фокусное расстояние аппаратуры датчика, коэффициент поглощения среды и др. Установка номера 5 пирамиды сводится к установке степени дискретизации синтезируемого эталона в соответствии со степенями дискретности уровней пирамиды анализируемого изображения.

При первом обращении устанавливается самый грубый (верхний) уровень, соответствующий разрешению верхнего уровня пирамиды анализируемого изображения. При втором обращении устанавливается степень разрешения второго сверху уровня пирамиды анализируемого изображения и т.д. Компьютерный синтез 6 – генерация эталонного изображения методом машинной графики в установленных условиях наблюдения и освещения и при выбранном разрешении (номере уровня). Особенность этого процесса заключается в использовании информации о предыдущем уровне эталона посредством указанного выше метода (рис.6.2.4). Корреляция заключается в определении корреляционной функции

соответствующих уровней эталонного и анализируемого изображений. Блок анализа 8 осуществляет идентификацию положения объекта совместным анализом всех корреляционных функций предыдущих уровней. Если решение не может быть принято (знак “-” на рис.6.2.5), то генерируется следующий уровень эталона, иначе – выдается сообщение о положении объекта.

#### 6.2.4. РАСПОЗНАВАНИЕ ОБРАЗОВ ПО ЭТАЛОНAM, АДАПТИРОВАННЫМ ПО УСЛОВИЯМ НАБЛЮДЕНИЯ И КАЧЕСТВУ

Приведенный способ определения типа объекта предусматривает использование в качестве эталонных достаточно высоко-качественных синтезированных изображений, которые отличаются от идеальных, построенных по законам геометрической оптики, в основном влиянием процессов выборки [29]. При высокой частоте дискретизации синтезированные изображения практически близки к идеальным по качеству. В реальных условиях изображения, полученные физическими датчиками, характеризуются размытием и шумами. Поэтому даже после приведения эталонных изображений к условиям реального наблюдения по ракурсу и теням остается рассогласование по качеству. Это рассогласование может вносить существенное затруднение в процесс распознавания. Однако отмеченный недостаток можно устранить, дополнив приведенную схему определения типа введением “испорченного” изображения эталона. Причем идеальный эталон надо испортить точно до уровня изобразительного качества реального изображения.

Известно [36], что в фотосистемах снижение качества изображения по отношению к идеальному обусловлено размытием из-за влияния звеньев: оптической среды, объектива, пленки и шумов гранулярности фотопленки. В оптико-электронных системах [5,6,45] кроме размытия в оптической среде и объективе имеют место выборка изображения приемником светового излучения и шум приемника. Зарегистрированное на носителе изображение  $\hat{I}(x,y)$  связано с идеальным  $I(x,y)$  соотношением

$$\hat{I}'(x,y) = I(x,y) \otimes h(x,y) + \hat{N}(x,y), \quad (6.2.1)$$

где  $h(x,y)$  – результирующая функция рассеивания системы, состоящей из  $n$  звеньев, каждое из которых имеет функцию рассеивания  $h_i(x,y)$ ;  $\hat{N}(x,y)$  – шум изображения;  $\hat{\cdot}$  – знак случайности;  $i = 1(1)n$ ;  $h(x,y) = h_1(x,y) \otimes h_2(x,y) \otimes \dots \otimes h_n(x,y)$  [36].

В [36,45,61] показано, что флюктуационный шум зернистости фотопленки, так же как шум, вносимый большинством датчиков видеосигнала, может быть описан случайной функцией, подчиняющейся нормальному закону распределения. Для осуществления операции (6.2.1) необходимо определить конкретный вид импульсного отклика  $h(x,y)$  и числовые характеристики закона распределения случайной функции  $N^*(x,y)$ . Известно [36], что единственным параметром, полностью определяющим данный (нормальный) закон распределения, служит дисперсия  $D_{N^*}$ . Для нахождения  $h(x,y)$  сначала определяют случайную функцию рассеивания  $h(x)$  по зашумленному реальному изображению в соответствии с методикой измерения передаточных характеристик [36] по изображению сигнала яркости.

Считая двухмерную функцию  $h(x,y)$  симметричной относительно оси ординат, по измерительной одномерной функции  $h(x,y)$  можно найти сечение двухмерной функции рассеивания  $h(r)$  [36]:

$$h(\rho) = -1/(\pi x) \frac{d}{dx} \int_{\rho}^{\infty} h(x)(x^2 - r^2)^{-0.5} x dx, \text{ где } \rho = (x^2 + y^2)^{0.5}.$$

Определив значения функции рассеивания в ряде дискретных точек, расстояния между которыми выбраны равными шагу дискретизации идеального изображения, сформируем матричную модель импульсного отклика  $H(l_1, l_2)$ , где  $l_1 = 1(1)L$ ,  $l_2 = 1(1)L$ , а  $L \times L$  – размер квадратной матрицы. Если цифровое представление идеального изображения есть матрица  $I$  размера  $N \times N$ , то операцию свертки можно описать в виде [45]

$$Q(m_1, m_2) = \sum_{n_1=1}^{m_1} \sum_{n_2=1}^{m_2} I(n_1, n_2) H(m_1 - n_1 + 1, m_2 - n_2 + 1),$$

где  $m_1, m_2 = 1, 2, \dots, M$ ;  $M$  – размер обработанного изображения  $Q$ ;  $M = N + L - 1$ .

Так как характер воздействия шумовых помех одинаков по полю изображения, то можно считать шумовую функцию  $N(x,y)$  стационарной и эргодической. Оценки числовых характеристик такой функции могут быть определены как среднее по пространственному интервалу  $S$  наблюдения одной реализации [53]:

$$\tilde{M}_{\hat{B}} = \tilde{M}[\hat{B}(x,y)] = (1/S) \iint_{x,y \in S} B(x,y) dx dy;$$

$$\tilde{D}_{\hat{N}} = \tilde{D}[\hat{N}(x,y)] = (1/S) \iint_{x,y \in S} (B(x,y) - \tilde{M}_{\hat{B}})^2 dx dy,$$

где  $B(x,y)$  – яркость реального изображения в пределах поля  $S$ ;  $\tilde{M}_{\hat{B}}$  – оценка математического ожидания яркости;  $\tilde{D}_{\hat{N}}$  – оценка дисперсии шумовой составляющей яркости. Поле  $S$  должно быть выбрано на реальном изображении таким, чтобы флюктуации оптической плотности на нем обусловливались только шумами, а не проявлялись из-за наличия в поле наблюдения объектов с различной яркостью. Приближенная оценка дисперсии  $\tilde{D}_{\hat{N}}$  по цифровому изображению находится из множества  $N$  отсчетов  $I_i$ ,  $i = 1(1)N$ :

$$\tilde{D}_{\hat{N}} = 1/N \sum_{i=1}^N (I_i - 1/N \sum_{i=1}^N I_i)^2,$$

где отсчеты взяты по полю постоянной фоновой яркости, близкой по значению к средней яркости обрабатываемого сюжета.

Заметим, что модель изображения с шумом идентична реальному изображению только в статистическом смысле, поэтому сравнение эталонного изображения, на котором шум смоделирован, с реальным зашумленным изображением может привести к некорректным результатам. Знание законов распределения и числовых характеристик не позволяет имитировать уникальную, конкретную реализацию случайной функции шума. Следовательно, эталонное изображение с реализацией шума будет в большей степени отличаться от реального зашумленного изображения, чем эталонное изображение без шума от того же реального зашумленного изображения. Поэтому в системе распознавания эталонное изображение должно быть адаптировано по параметру размытия, но не шума. Модели шума могут быть использованы для научных исследований при моделировании систем наблюдения.

Поэтому основная идея распознавания по адаптированным эталонам с учетом оптического качества сводится к априорному или апостериорному определению функции рассеивания реального изображения и внесению соответствующего размытия в смоделированные методом машинной графики эталоны.

Изображения, полученные физическими датчиками, реально представляют собой функции оптической плотности или напряжения в системе координат кадра. Приведем для случая моделирования фотосистемы формулы из [36], позволяющие вычислять значения оптической плотности преобразованного изображения по найденной в системе машинной графики освещенности или яркости  $B(x,y)$  объекта в системе координат кадра  $oxy$ . Освещенность геометрического изображения вычисляется по правилу  $E(x,y; \lambda) = =[\pi K_B(x,y) t_{sf}(\lambda) t_{o.c}(\lambda) c^4(x,y)/4K^2] [B(x,y;\lambda) t_a(x,y;\lambda) + B_D(x,y;\lambda)]$ , где  $\lambda$  – длина волны;  $x,y$  - координаты (здесь и далее в круглых скобках до знака ";" проставлены переменные, после знака ";" – параметры);  $B_D(x,y;\lambda)$  – яркость дымки в оптической среде, заданная в системе координат  $oxy$  кадра;  $t_{sf}(\lambda)$ ,  $t_{o.c}(\lambda)$ ,  $t_a(\lambda)$  – монохроматические коэффициенты светопропускания фильтра, оптической системы и оптической среды;  $K_B$  – коэффициент виньетирования;  $K$  – диафрагменное число объектива;  $c^4(x,y) = \cos^4\beta(x,y)$  ( $\beta$  – текущий угол поля изображения).

Освещенность размытого изображения определяется выражением  $E'(x,y;\lambda) = E(x,y;\lambda) \otimes h(x,y;\lambda)$ , где  $h$  – результирующая функция рассеивания всех звеньев.

Зная выдержку  $t_e(x,y)$  и монохроматические характеристические кривые  $D(x,y;\lambda) = D(\lg H(x,y;\lambda))$ , можно определить значение экспозиции  $H(x,y;\lambda) = E'(x,y;\lambda)t_e(x,y)$  и оптической плотности  $D(x,y)$ . Окончательное выражение  $\hat{D}(x,y) = D(\lg H(x,y;\lambda)) + + \Delta \hat{D}(x,y)$ , где  $\Delta \hat{D}(x,y)$  – флюктуация оптической плотности, обусловленная шумом гранулярности фотопленки.

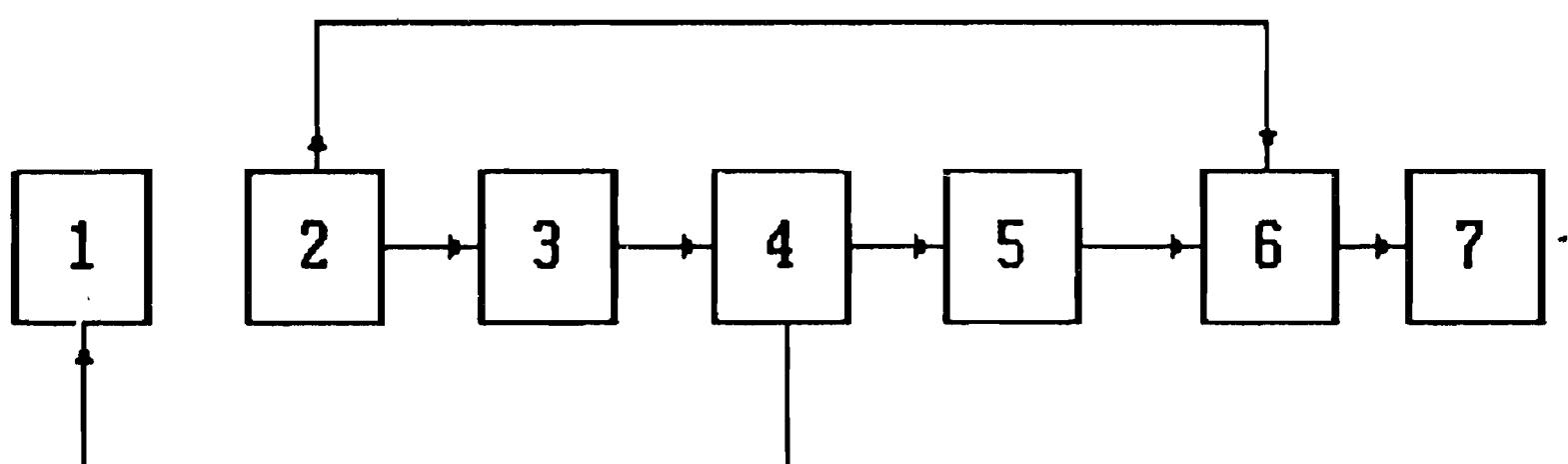


Рис. 6.2.6. Схема распознавания объекта при адаптации эталона по изобразительным признакам и качеству

Структурная схема процесса определения типа объектива с введением преобразования эталонов по качеству приведена на рис.6.2.6: 1 – банк данных математического описания трехмерной геометрической формы и оптических свойств типов объектов, претендующих на распознавание; 2 – изображение, полученное реальным физическим датчиком; 3 – определение условий наблюдения и освещения; 4 – синтез эталонных изображений всех типов объектов установленного класса в реальных условиях наблюдения и освещения; 5 – преобразование синтезированных изображений до уровня качества реального изображения; 6 – сравнение каждого эталонного синтезированного изображения с реальным; 7 – определение типа.

### 6.3. АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ В МАШИНОСТРОЕНИИ

Машиностроительное проектирование стало одной из основных прикладных областей, использующих возможности машинной графики [110]. Практически все трудоемкие ручные операции проектировщика: создание основных видов детали, сборочного чертежа, изометрии, сечений – выполняются с высокой эффективностью автоматизированными компьютерными системами [33]. На потребительском рынке широкое распространение получили коммерческие программные продукты и системы, такие как Autocad, Syscad, Verscad, Robocad, P-Cad, Conception 3D [128], которые могут быть использованы для автоматизации конструирования и технологической подготовки производства машиностроительных деталей. Специализированные системы состоят из нескольких программных модулей.

Модель конструирования в двухмерном пространстве обеспечивает конструктору необходимый сервис для построения контуров машиностроительных деталей с использованием традиционной техники, принятой в машиностроительном черчении. Конструктор работает в одной, двух или трех произвольно выбранных проекциях, используя язык, максимально приближенный к лексике данной предметной области. Реализуемые с помощью языка операции приводят к построению в памяти ЭВМ двухмерного образа и трехмерной модели конструируемой детали. Полученная трехмерная модель изображается как в виде проволочной модели, так и в виде модели твердого тела. В результате конструирования в памяти ЭВМ создается модель изделия, необходимая для всех

остальных процессов, связанных с доведением этого изделия до производства. В случае необходимости выпуска чертежей система позволяет передавать полученные данные в подсистему документирования, обеспечивающую оформление чертежно-конструкторской документации в соответствии с различными стандартами.

Модуль параметрического конструирования позволяет использовать технику задания параметров проектируемого изделия, формировать каталоги унифицированных стандартных и нормированных деталей. При формировании таких каталогов конструктор может использовать различные приемы задания параметров изделия в явном или неявном виде. Обычно предусматривается возможность формировать таблицы параметров конструируемых деталей. При этом выбор соответствующей детали обеспечивается выбором необходимой строки сформированной таблицы. Модуль параметрического конструирования позволяет обеспечивать параметризацию не только двухмерных изображений, но и моделей изделий в трехмерном пространстве.

Метод трехмерного проектирования позволяет, используя технику создания модели изделий, принятую в модуле двухмерного конструирования, сформировать модель твердого тела, которая может отображаться на экране графического дисплея в виде произвольного вида детали из любой точки. Полученная модель может быть отображена с удалением видимых линий, она может быть раскрашена, на ней могут быть совершены операции визуализации произвольного сечения [68].

Модель изделия, полученная на этапе конструирования, автоматически передается в подсистему технологической подготовки производства. Подсистема технологической подготовки позволяет автоматизировать процессы подготовки лент для станков с ЧПУ от 2,5 до 5 координат в различных технологиях. Существуют программы, обеспечивающие подготовку лент для станков с ЧПУ для технологии вырубных и гибочных штампов, точения, сверления, фрезерования на 2,5 – 5 координатных станках, а также для других технологий механообработки. Настройка на конкретный станок осуществляется с помощью подключения постпроцессора.

Подсистема расчета по методу конечных элементов позволяет рассчитывать динамические, статистические, прочностные характеристики спроектированных изделий, при этом сетка для расчета по методу конечных элементов обычно генерируется автоматически.

Такие специализированные системы ориентированы на

автоматическое проведение всех необходимых массоинерционных расчетов деталей: центра тяжести, объема, площади объема, момента инерции и т.п. Коммерческие программные продукты ориентированы на создание библиотек элементов, из которых формируются детали конкретного класса. После разработки такой библиотеки пользователь может применять ее для компилирования из элементов библиотеки деталей соответствующего класса.

Системы машиностроительного проектирования обычно реализуют на рабочей станции, использующей 486-й микропроцессор или RISC-процессор и скоростной дисплейный процессор, имеющий высокую интерактивность производимых процедур (производительность около 10 тысяч векторов в секунду).

Получают все более широкое распространение транспьютерные системы машиностроительного проектирования [139] в конфигурации: любой персональный компьютер плюс транспьютерный продукт. Компьютер используется только в качестве интерфейса, поэтому его быстродействие совершенно некритично для обеспечения процесса проектирования.

Как в других прикладных областях, так и в машиностроительном проектировании желательна мгновенность построения изображения. Реальный масштаб времени может быть достигнут только при использовании дорогостоящей аппаратуры, поэтому в практике часто применяется двухступенчатый процесс проектирования – отображение. На первом этапе конструктор оперирует объемными примитивами, которые отображаются не в естественном полутонаром виде, а в форме проволочной модели. Так как даже самый простой персональный компьютер быстро изображает проволочные фигуры, то процесс конструирования происходит без задержек. На любом промежуточном или оконечном этапе проектирования объект, отображаемый проволочными примитивами, может быть переведен в нормальную полутонарую форму с сечениями, в изометрии, с разверткой и другими эффектами. Эта вторая ступень процесса занимает значительно больше времени, чем первая, но порождает качественное изображение. Таким образом достигается в целом высокая эффективность процесса конструирования.

#### 6.4. ВОССТАНОВЛЕНИЕ ФОРМЫ СКРЫТЫХ ОБЪЕКТОВ В МЕДИЦИНСКИХ ИССЛЕДОВАНИЯХ

В практике медицинских исследований и диагностике существует задача визуального наблюдения органов, закрытых

мягкими тканями. Единственным инструментом получения информации об исследуемом объекте являются рентгеновские лучи и иногда ультразвуковые волны. С помощью методов томографии [57] удается восстановить послойную форму "среза" объекта. Информация о серии срезов достаточна для синтеза на компьютере изображения скрытого объекта в любом ракурсе. Сущность способа заключается в следующем. При получении томограмм, достаточно часто и равномерно пересекающих объект от его начала до конца в выбранном направлении, автоматически запоминают в компьютере пространственное положение каждого среза и линию внешнего контура в пределах отдельного среза. Трехмерную форму объекта представляют в виде стопки поставленных друг на друга обобщенных вертикальных цилиндров. Образующая каждого цилиндра равна по длине расстоянию между соседними срезами, а направляющая соответствует линии внешнего контура объекта в срезе. Направляющие всех цилиндров параллельны между собой и перпендикулярны плоскости, в которой вращается рентгеновская пушка (рис.6.4.1). Форму направляющей линии обычно не удается описать аналитически, поэтому эту замкнутую кривую разбивают на прямолинейные отрезки, составляющие ломаную замкнутую линию. Обобщенный цилиндр, соответствующий определенному срезу томограммы, аппроксимируют многогранником. Стопка поставленных друг на друга многогранников служит математической моделью объекта, закрытого от непосредственного наблюдения. После получения модели производят ее визуализацию методами машинной графики.

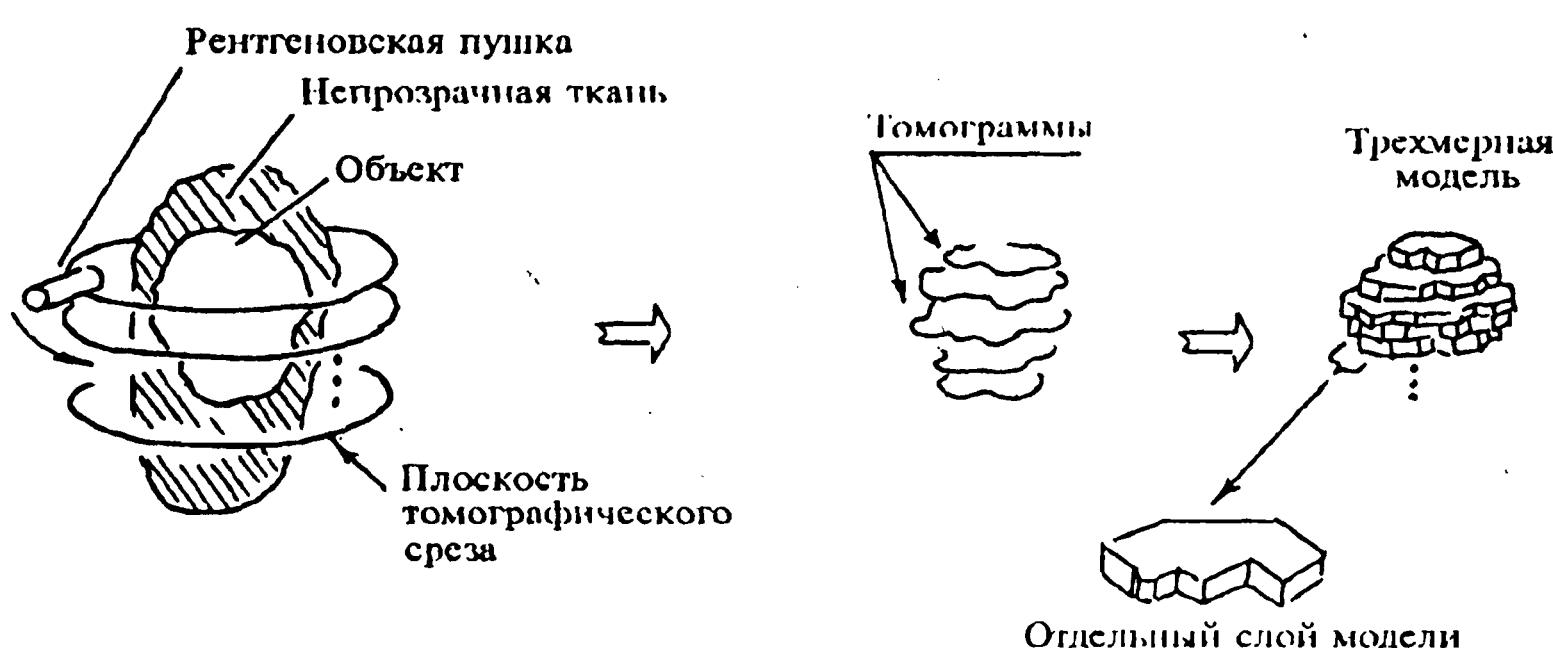


Рис. 6.4.1. Рентгеноскопирование объекта и восстановление его формы

Так как основная задача такого медицинского исследования сводится к наиболее полной передаче формы объекта через изображение в различных ракурсах, то это изображение должно быть построено без теней и зеркальных бликов. Поэтому наиболее приемлемым методом отображения является метод изображения полигонального поля. Объект представляется в виде совокупности примитивов – многогранников, а поверхность каждого из них, в свою очередь, из прямоугольников, образующих замкнутую ленту, и двух торцевых многоугольников.

Для изображения такого полигонального поля применяют ортогональную проекцию, что и вычислительно эффективно, и приемлемо с потребительской точки зрения.

Рассмотрим отдельную модель томографического среза в виде матрицы размера  $2N \times 4$ :

$$T = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ X_2 & Y_2 & Z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ X_N & Y_N & Z_N & 1 \\ \vdots & \vdots & \vdots & \vdots \\ X_{2N} & Y_{2N} & Z_{2N} & 1 \end{bmatrix},$$

где  $X, Y, Z$  – координаты вершин многоугольника в объектной системе координат, т.е. в системе координат томографа.

Первые  $N$  точек лежат на верхней плоскости модели томографического среза, вторые  $N$  точек – соответственно на нижней. Точки на каждой плоскости расположены последовательно одна за другой, точно так же они помещены и в своих половинах матрицы  $T$ . Первый боковой прямоугольник образует точки 1, 2,  $N+1, N+2$ ; второй 3, 4,  $N+3, N+4$ ; ...  $i$ -й  $2i-1, 2i, N+2i-1, N+2i$ ;  $N$ -й  $N, 1, 2N, N+1$ , где  $i = 1(1)N-1$ .

Спроектируем эти точки объекта на изображение:

$$T' = \begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_j & y_j & z_j & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_{2N} & y_{2N} & z_{2N} & 1 \end{bmatrix} = TM,$$

где  $x_j, y_j, z_j$  – координаты  $j$ -й точки в экранной системе координат;

$j = 1(1)2N$ ;  $M$  – матрица преобразования координат, связывающая экранную и объектную системы (см. § 3.1).

Таким образом, экран покрывают множеством точек, соответствующих положению на изображении вершин выпуклых многогранников. Объединение точек  $2i-1, 2i, N+2i-1, N=2i$  отрезками прямых в такой последовательности является границей изображения  $i$ -го бокового прямоугольника в томографическом срезе. Изображения отдельных прямоугольников будут конкурировать между собой на видимость. Оценка видимости эффективнее всего в данном случае осуществляется методом сканирующей строки (см. § 4.2). В результате воздействования этого алгоритма для каждого отрезка сканирующей строки становится известна информация о номере видимой плоскости – прямоугольника или номере торцевой плоскости, ограничивающей томографический срез. Для вычисления освещенности изображения на каждом участке сканирующей линии необходимо определить нормальный вектор к поверхностям каждого  $i$ -го прямоугольника и к поверхностям торцевых плоскостей.

Каждую  $i$ -ю плоскость можно описать тремя точками под номерами  $2i-1, 2i, N+2i-1$  в матрице  $T$ . Известными правилами [23] определяется нормаль к плоскости, проходящая через эти точки:

$$\begin{aligned} \mathbf{N}_i = i & \begin{bmatrix} Y_{2i-1} & Z_{2i-1} & 1 \\ Y_{2i} & Z_{2i} & 1 \\ Y_{N+2i-1} & Z_{N+2i-1} & 1 \end{bmatrix} + j \begin{bmatrix} Z_{2i-1} & X_{2i-1} & 1 \\ Z_{2i} & X_{2i} & 1 \\ Z_{N+2i-1} & X_{N+2i-1} & 1 \end{bmatrix} + \\ & + k \begin{bmatrix} X_{2i-1} & Y_{2i-1} & 1 \\ X_{2i} & Y_{2i} & 1 \\ X_{N+2i-1} & Y_{N+2i-1} & 1 \end{bmatrix}. \end{aligned}$$

Нормаль к торцевой плоскости проходит через три любые точки, записанные в верхнюю половину матрицы  $T$  (например, первую, вторую, третью), соответственно нормаль к торцевой плоскости

$$\begin{aligned} \mathbf{N}_T = i & \begin{bmatrix} Y_1 & Z_1 & 1 \\ Y_2 & Z_2 & 1 \\ Y_3 & Z_3 & 1 \end{bmatrix} + j \begin{bmatrix} Z_1 & X_1 & 1 \\ Z_2 & X_2 & 1 \\ Z_3 & X_3 & 1 \end{bmatrix} + \\ & + k \begin{bmatrix} X_1 & Y_1 & 1 \\ X_2 & Y_2 & 1 \\ X_3 & Y_3 & 1 \end{bmatrix}. \end{aligned}$$

Так как при восстановлении томографической трехмерной картины нет теней и необходимости передачи зеркальных эффектов, то направление нормального вектора безразлично.

Освещенность вычисляется по правилу  $E = a E_0 \cos(N, K_z)$ , где  $N$  – нормаль к изображаемой поверхности;  $K_z$  – орт оси  $oz$  экранной системы координат;  $E_0$  – освещенность, порождаемая абсолютно белой матовой поверхностью;  $a$  – коэффициент яркости, выбранный для изображаемого тела.

Изображения, восстановленные из томограмм по такому алгоритму или подобному ему, имеют ряд специфических отличительных признаков: освещенность округлых предметов плавно падает до нуля к краям объекта; ближайшая к наблюдателю точка на круглом предмете является самой яркой; отсутствует зеркальность в тени, как это бывает на фотоизображениях, связанных с помощью фотовспышки; при низком разрешении томограммы или недостаточной частоте срезов по предмету на изображении выделяются грани срезов томограмм и их послойная структура.

Несмотря на значительное число обрабатываемых точек, алгоритм оказывается вычислительно эффективным, так как организация матрицы  $T$  позволяет достаточно просто определять точки, относящиеся к одной плоскости. Кроме того, здесь используется одна-единственная операция над примитивами (томографическими срезами) – логическое сложение.

Возможны другие модификации алгоритма, в том числе с добавлением информации о форме объекта от взаимно перпендикулярных томографических сечений и более сложного полигонального поля, аппроксимирующего объект.

## 6.5. ВИДЕОТРЕНАЖЕРЫ И ИМИТАТОРЫ СЛОЖНЫХ СЦЕН

Имитационные тренажеры находят в последнее время все более широкое применение в авиации и космонавтике – областях, где проведение физического полетного обучения летчика дорого, а иногда и невозможно [10, 24, 66]. Примером таких систем является парный тренажер воздушного боя. Место каждого пилота снабжено имитационным экраном вместо лобового стекла, а также рычагами управления, которые являются устройствами изменения условий наблюдения. В память ЭВМ заложены данные о геометрических и оптических свойствах ландшафта, технических характеристиках самолета. Во время “боя” пилот перемещает рычаги управления,

информация поступает в ЭВМ, в режиме реального времени переформируется картина обзора. Поражение цели также может быть смоделировано. Другой пример – обучение космонавта операциям по выводу на орбиту спутника из грузового отсека возвращаемого транспортного корабля. Это ответственная операция, выполняемая механическим манипулятором; поэтому человек еще на земле должен получить навыки управления манипулятором.

Физический имитатор в наземных условиях неэффективен, так как мешающее действие силы тяжести сводит на нет смысл эксперимента. Поэтому задачу решают следующим образом. В память ЭВМ вводят математическое описание интерьера грузового отсека, спутника, манипулятора. Управление манипулятором осуществляется путем визуального контроля по экрану имитационной обстановки. Каждое воздействие на рычаги управления сообщается машине как изменение пространственного расположения объектов сцены.

Важнейшим условием эффективности подобных тренажеров является мгновенность реакции синтеза изображения на изменение параметров наблюдения. Поэтому применяют матричные процессоры с быстродействием в несколько миллиардов операций в секунду, ликвидируют ряд визуальных эффектов: тени, зеркальность и др., и используют рекурсивные процедуры синтеза изображения.

Основная идея последних связана с тем, что при небольшом изменении условий наблюдения наиболее вероятно, что любой из “рецепторов” видит ту же поверхность (но другую точку на этой поверхности), что и до изменения. Это позволяет не “вслепую” перебирать поверхности в процессе определения видимой точки, а сразу начинать с наиболее вероятной. Истинность предположения определяется следующим критерием: если в начальных условиях receptor “видит” точку ( $X_1, Y_1, Z_1$ ) поверхности и после изменения условий световой луч из рецептора пересекается с той же поверхностью в точке ( $X_2, Y_2, Z_2$ ) и  $[(X_1 - X_2)^2 + (Y_1 - Y_2)^2 + (Z_1 - Z_2)^2]^{0.5} \leq d$ , где  $d$  – порог, то точка ( $X_2, Y_2, Z_2$ ) есть видимая после изменения условий.

### 6.5.1. ТРЕНАЖЕР ДЛЯ ИМИТАЦИИ ПОЛЕТА НАД РЕАЛЬНОЙ МЕСТНОСТЬЮ

Возможности машинной графики позволяют решать проблемы имитации полета на самолете над реальной местностью, что необходимо как для обучения летчиков ориентированию в

неизвестной для них обстановке, так и для автоматической навигации летательных аппаратов при беспилотной посадке.

При моделировании геометрических форм на реальной местности можно выделить три класса образований: рельеф в виде тонкой изогнутой или изломанной поверхности; постройки трехмерной формы и растительность типа массива леса или отдельных деревьев, трехмерными формами которых нельзя пренебречь. Соответственно каждому из этих образований присущи специфические оптические свойства. Местность покрыта "узором" дорог, полей, озер, которые в свою очередь имеют определенные индикаторы рассеивания, коэффициенты яркости и другие признаки. Исследования [36] показывают, что большинство покрытий природного происхождения можно считать диффузными отражателями и соответственно охарактеризовать единственным параметром – коэффициентом яркости  $a(X_m, Y_m)$ , где  $X_m, Y_m$  – координаты местности в системе координат сцены. Коэффициент яркости меняется с течением времени, что связано с деятельностью человека (распашка полей, строительство дорог), вегетативной деятельностью растений (сброс листвы) и выпадением, таянием снегов. Оптические характеристики построек можно также считать близкими к диффузным и достаточно стабильными. Лесные массивы кроме сезонных изменений отражательных свойств характеризуются и наличием теней, что также следует учитывать.

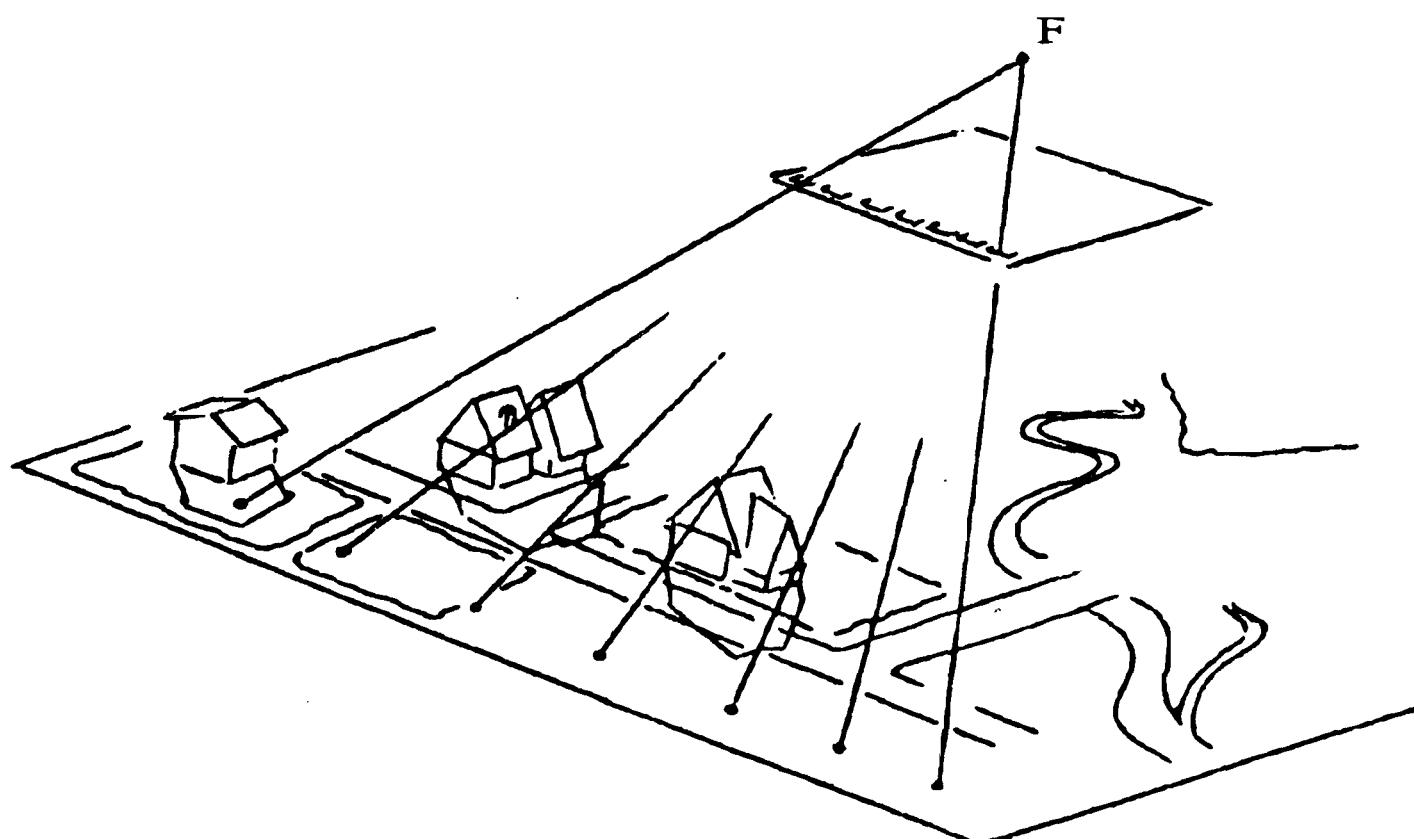


Рис. 6.5.1. Изображение построек и рисунка покрытий на местности методом трассирования лучей

Теоретически возможно создать полную математическую модель местности и построек, а затем методами машинной графики синтезировать изображение сцены в текущих условиях наблюдения и освещения. Такое изображение может быть использовано не только в видеотренажерах, но и в качестве эталонного при распознавании образов. Однако построение модели будет тогда исключительно трудоемким, а время синтеза недопустимо большим. При относительно плоской местности имеется возможность значительно сократить вычислительную емкость задачи из-за использования планового фотоизображения самой местности.

Основная идея такого решения заключается в следующем. Модель сцены представляют в виде однородной по отражательным свойствам плоскости и трехмерными конструкциями зданий, расположенных на этой плоскости. В заданных условиях наблюдения (модель ракурса) и освещения (модель Солнца) производится отслеживание (трассирование) каждого визирующего луча (см.рис.6.5.1). Если луч “видит” трехмерный объект, то вычисление освещенности видимой точки осуществляется по правилу  $E = E_{\max} a(X, Y, Z) \cos \alpha$ , где  $E_{\max}$  – максимальное значение освещенности, достигаемое на изображении белой диффузно отражающей поверхности с коэффициентом яркости, равным единице, и с нормалью к поверхности, направленной на источник света – Солнце;  $a$  – коэффициент яркости поверхности;  $\alpha$  – угол между нормалью к видимой поверхности объекта и направлением на Солнце.

Если же луч визирования “видит” поверхность местности в точке  $(X_m, Y_m)$ , то освещенность рассчитывается с учетом действительной отражательной способности, информация о которой извлекается из фотографии местности:  $E = E_{\max} a_m(X_m, Y_m) \cosh h$ , где  $a_m$  – коэффициент яркости местности;  $\cosh h$  – высота стояния Солнца [36].

Пользуясь расчетными формулами освещенности для диффузного объекта, возможно восстановление значения  $a_m$  по всей сцене на основании анализа аэрофотографии района. Так как на фотографии запечатлена не собственно освещенность, а производная величина – оптическая плотность, то требуются дополнительные расчеты с корректными данными для перехода от оптической плотности к освещенности и коэффициенту яркости.

Исходную аэрофотографию “оцифровывают” (рис.6.5.2), представляя в виде матрицы коэффициентов яркости, координатно

наложенной на местность. Если размер изображения в масштабе местности составляет  $X_{\max} Y_{\max}$  единиц измерения, а число строк I и столбцов J, на которые разбивают исходное изображение, где строки располагаются вдоль оси OY, то номер элемента матрицы,  $ij$  в которую упал луч визирования, можно определить следующим образом:  $i = \inf(1 + I/2 - (I-1)Y_m/Y_{\max})$ ;  $j = \inf(1 + J/2 + (J-1)X_m/X_{\max})$ , где  $X_m, Y_m$  – координаты пересечения визирующего луча с плоскостью местности.

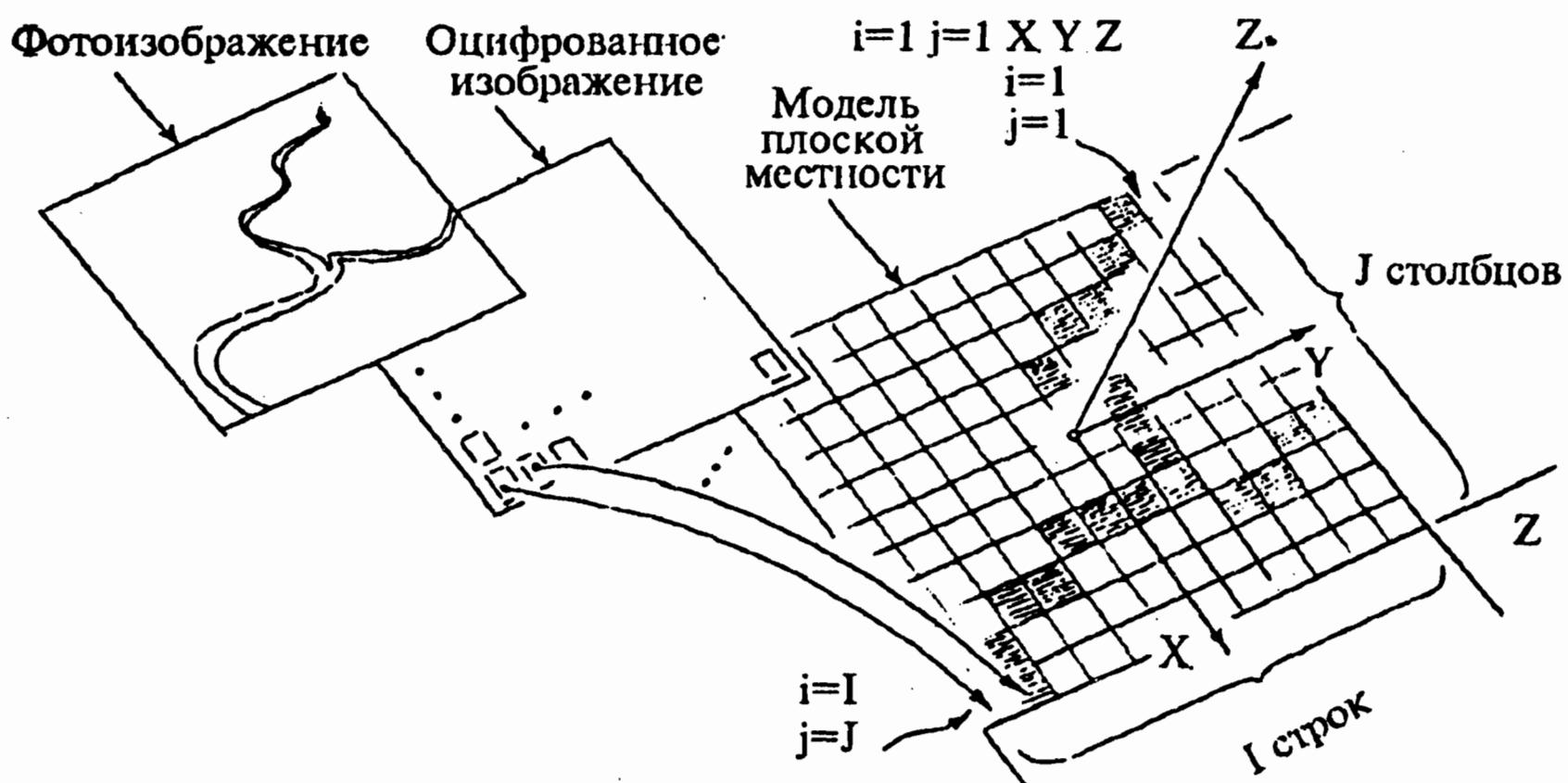


Рис. 6.5.2. Преобразование фотоизображения местности

Так как отдельные участки местности на исходной фотографии могут быть затенены падающими тенями, то для участков информации о коэффициенте яркости местности должна быть извлечена из снимка, сделанного при другом местном времени, когда исследуемый участок не затенен.

Таким образом, общее решение задачи сводится к следующему. Предварительно на район имитационного полета получают несколько разновременных аэрофотоизображений. С помощью этих изображений определяют коэффициент яркости местности, разбитой на отдельные участки, в пределах которой яркость считается постоянной. С помощью информации со снимков определяют геометрические формы и размеры зданий и создают математическую модель трехмерных конструкций в системе

координат сцены. Помещают поле, геометрически соответствующее матрице коэффициентов яркости, в систему координат сцены. В каждой позиции наблюдения определяют трассирование лучей через рецепторы и центр проекции до пересечения со сценой. При пересечении луча с объектом освещенность определяется на основании коэффициента яркости видимой поверхности объекта, при пересечении луча с поверхностью местности сначала определяется номер элемента матрицы коэффициентов яркости местности, а затем вычисляется освещенность на основании содержания этого элемента матрицы. При этом может быть достигнута проработка освещенности затененных участков местности.

Высокочастотное геометрически и оптически корректное моделирование площадных объектов на поверхности Земли представляет значительную сложность и требует больших трудозатрат. Например, известны концепции моделирования и соответствующая трудоемкость описания современного города с плотной застройкой [106]. Работа выполнялась с целью создания видеоимитатора, в том числе для архитектурного проектирования. Вся сцена города Глазго разбивалась на три составляющие: поверхность Земли, сеть дорог и трехмерные здания. Формы поверхности Земли вводились путем оцифрования данных с карты в масштабе 1:10 000 на площади 64 кв.м. Все дороги были оцифрованы в виде единого изображения, причем предполагалось, что они все лежат в одной плоскости. Затем этот узор дорог “опускался” на поверхность Земли, порождал комбинацию рельеф – дороги. Наибольшую трудность по вводу исходных данных представили трехмерные здания. В центре города площадью 6 кв. км сосредоточено около 2500 зданий. Их плановые формы извлекались из аэрофотографий, а высоты определялись по теням из анализа стереоизображений и проектной документации. Ввод данных производился практически вручную, что вылилось в трудозатраты около 1 чел./г. Трехмерная модель здания накладывалась на модель Земли и дорог, создавая полную модель сцены.

Имитаторы, использующие принцип трассирования лучей, работают сравнительно медленно, поэтому их реализация возможна только на суперЭВМ или многопроцессорных комплексах.

Метод сканирующей строки является наиболее эффективным для получения изображения объектов, составленных из многоугольников. Подавляющее большинство искусственных сооружений

допускает достаточно простую интерпретацию плоскостями. Например, практически все здания современных городов представлены многогранниками.

Рассмотрим вопросы моделирования фотоподобных изображений плоского ландшафта с расположеннымными на нем трехмерными зданиями. Предварительно на район местности получают несколько плановых фотоизображений с различными направлениями падающих теней (рис.6.5.3,а,б).

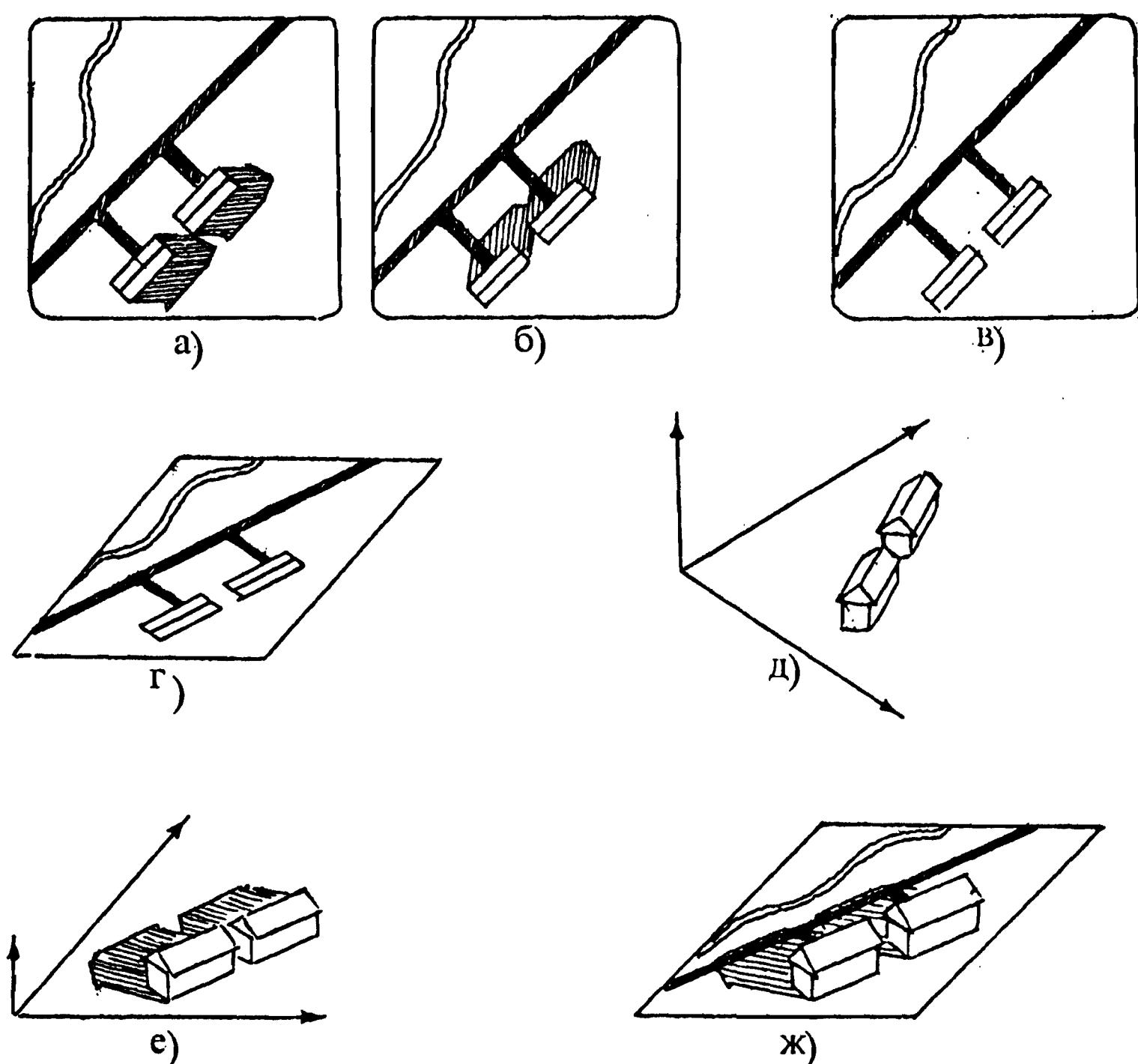


Рис. 6.5.3. Преобразование фотоизображений для синтеза картины методом сканирующей строки

Затем в интерактивном режиме средства цифровой обработки изображений из нескольких исходных плановых изображений создают одно бесстеневое (рис.6.5.3,в) путем локального замещения

затененных участков освещенными. В соответствии с требуемыми условиями наблюдения и освещения, как это требуется для имитации обзора сцены, бестеневое изображение (рис.6.5.3,в) подвергают геометрическому трансформированию [30] и фотометрической коррекции (рис.6.5.3,г). В результате создаются изображения, отличающиеся от требуемого только искажениями изображений трехмерных объектов и теней от них. По исходным фотоизображениям, другим перспективным фотоизображениям создают модель трехмерных объектов сцены (рис.6.5.3,д) – многоугольников и многогранников, при этом высотные размеры зданий определяют по теням, по перспективным изображениям вертикальных стенок и стереометодом [30]. В требуемых условиях наблюдения и освещения, которые были уже однажды использованы для трансформирования, методом сканирующей строки строят синтезированное изображение (рис.6.5.3,е) трехмерных объектов, отдельно храня участки изображений объектов с собственными тенями и участки падающих теней. Так как изображения (рис.6.5.3,г,е) порождены при одинаковых и тех же условиях, то они геометрически совпадают, и задача сводится к их комбинированию. Последнее производится путем замещения участков мощности участками изображений объектов с собственными тенями и понижением яркости изображения местности в зонах падающих теней. После такого комбинирования достигается окончательный результат (рис.6.5.3,ж).

На рис. 6.5.4–6.5.6 приведены результаты компьютерного моделирования изображений застройки на фоне конкретной местности. Изображения на рис. 6.5.4 и 6.5.5 получены методом трассирования лучей с использованием процедур, приведенных в приложении. Эта сцена использовалась на практике для прогнозирования места размещения планируемой ТЭС и устройства коммуникаций. Третья сцена (рис. 6.5.6) получена методом сканирующей строки. Технология ее производства включает следующие этапы: получение аэрофотоизображения, определение формы зданий и их высот, интерактивное удаление изображений зданий с оцифрованного аэрофотоизображения, построение математической модели зданий, создание модели сцены из зданий и естественного ландшафта, компьютерное моделирование синтезированного изображения. На практике последняя сцена использовалась для анализа радиовидимости системы сотовой связи, при этом анализировалось не изображение, а объемная модель.

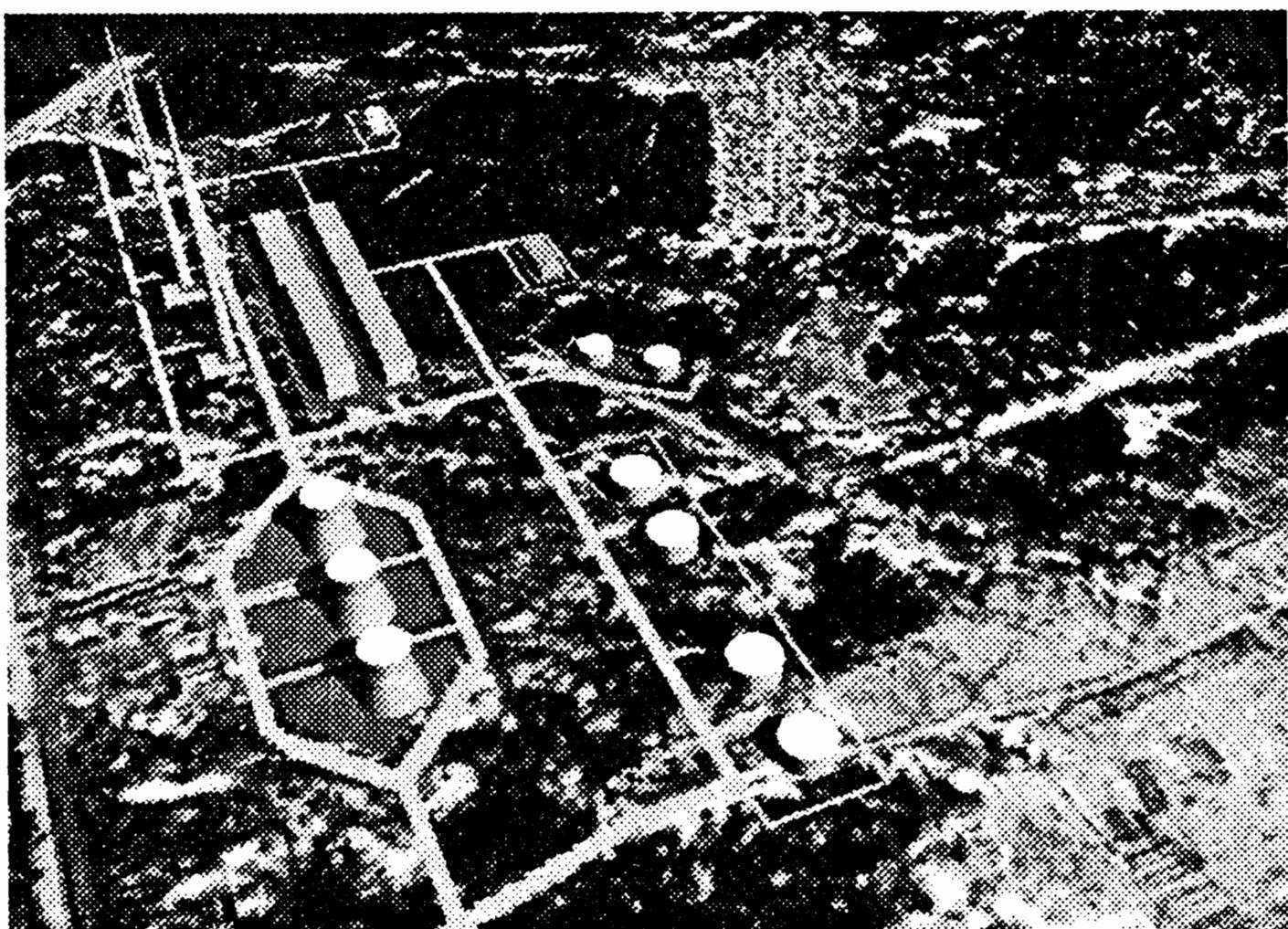


Рис. 6.5.4. Синтезированное изображение местности и предприятия  
(ракурс № 1)

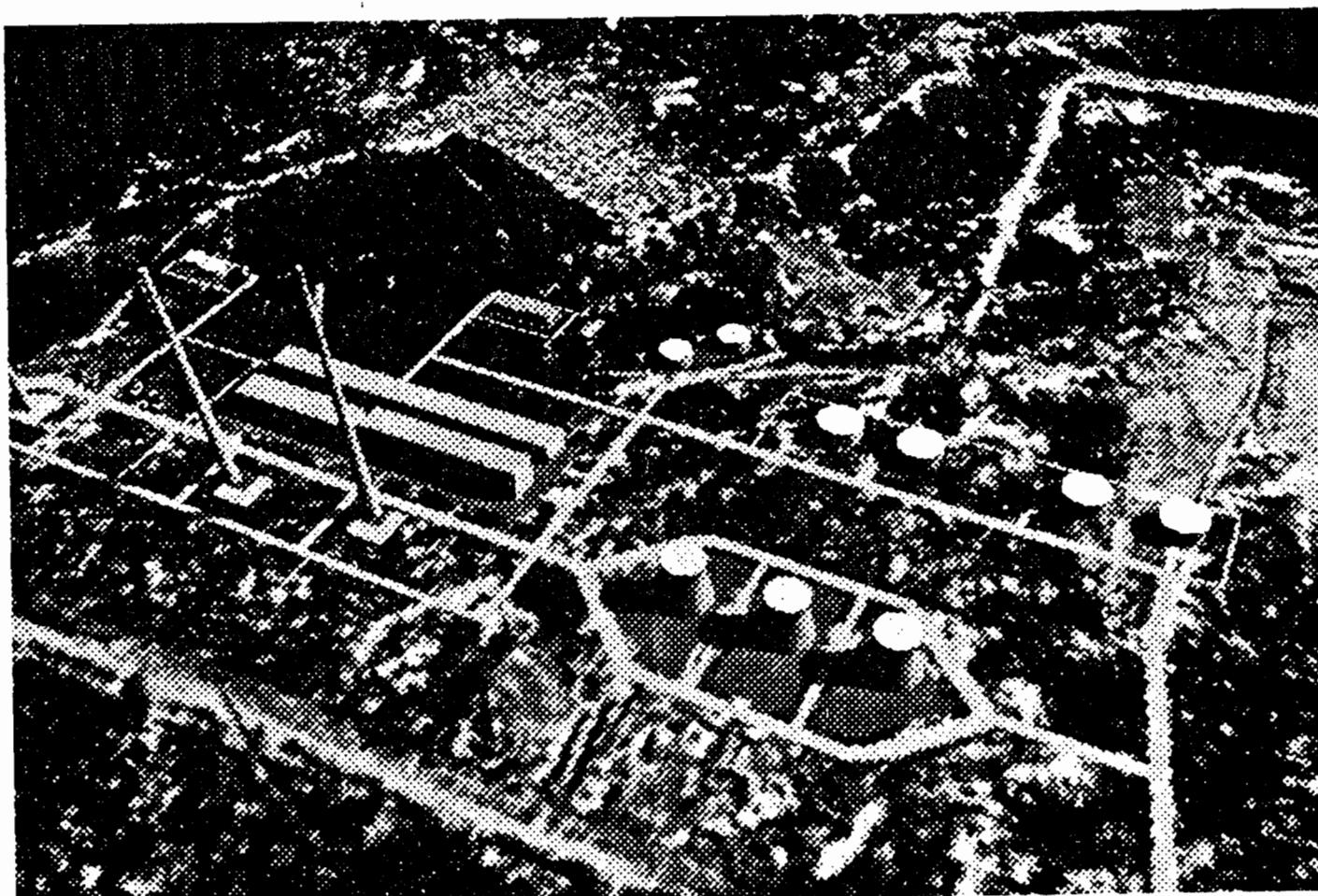


Рис. 6.5.5. Синтезированное изображение местности и предприятия  
(ракурс № 2)

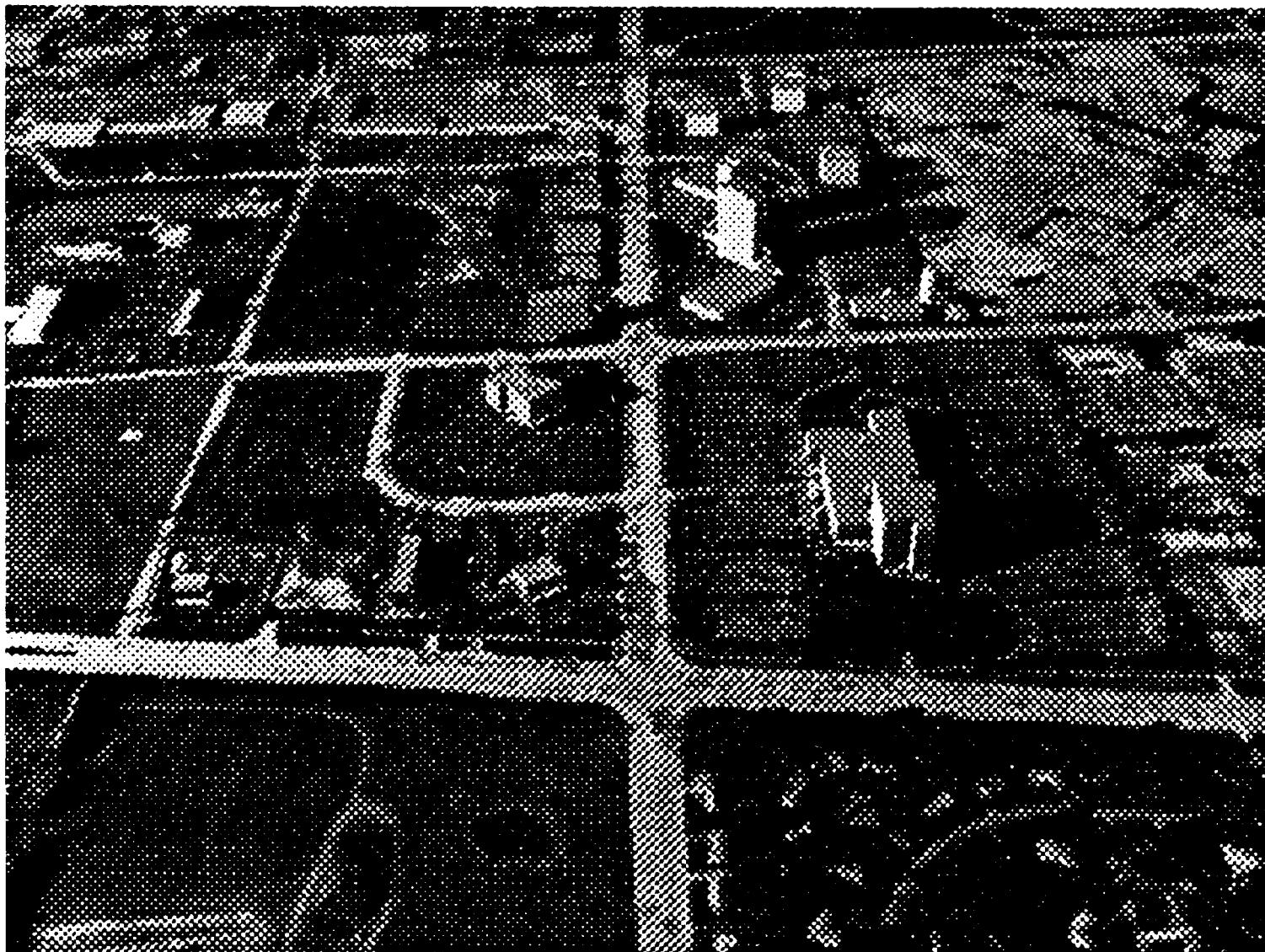


Рис. 6.5.6. Синтезированное изображение местности и построек

### 6.5.2. КОМПЬЮТЕРНЫЙ ИМИТАТОР ИЗОБРАЖЕНИЙ ОБЛАКОВ

Для решения задач видеораспознавания в естественных условиях, в видеотренажерах для летчиков и мультипликации могут применяться методы компьютерного синтеза облаков с соответствующими изобразительными признаками: плотностью, размытостью краев, прозрачностью. Рассмотрим в качестве примера моделирование кучевых облаков. В основе математического описания трехмерного тела облака сложной конфигурации лежит его аппроксимация набором эллиптических поверхностей различных размеров (рис.6.5.7).

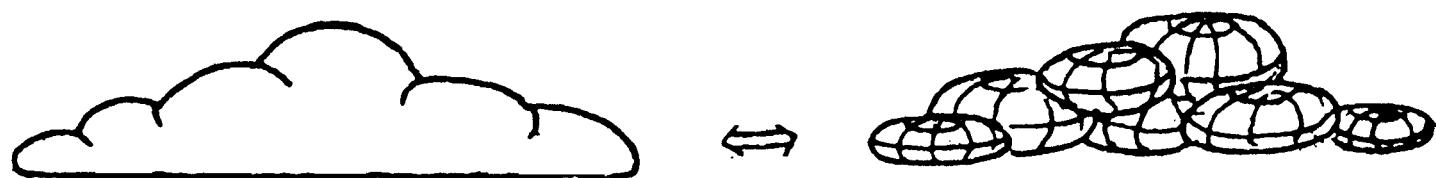


Рис. 6.5.7. Имитация кучевых облаков эллипсоидами

Разумеется, такая модель слишком далека от реальности, но в последующем она будет постепенно доведена до вполне приемлемых показателей. Общие принципы моделирования приведены в [94, 114]. Опишем их, вводя дополнительные визуальные эффекты, а именно проработку яркости на теле облака в области падающих и собственных теней и формирование падающих теней. В качестве базовой используем метод трассирования лучей. В правой прямоугольной системе координат местности на выбранной оператором высоте разместим эллиптическое тело облака (рассмотрим один эллипсоид для простоты изложения, комбинация эллипсоидов обрабатывается аналогично). Уравнение, описывающее поверхность эллипсоида, представлено в виде

$$Q(X, Y, Z) = [X \ Y \ Z \ 1] Q[X \ Y \ Z]^T = 0, \text{ где}$$

$$Q = \begin{bmatrix} Q_1 & 0,5Q_4 & 0,5Q_6 & 0,5Q_7 \\ Q_4 & Q_2 & 0,5Q_6 & 0,5Q_7 \\ Q_6 & 0,5Q_5 & Q_3 & 0,5Q_9 \\ Q_7 & 0,5Q_8 & 0,5Q_9 & Q_0 \end{bmatrix},$$

причем знак функции  $Q(X, Y, Z)$  выберем таким, чтобы значение функции для любой точки внутренней области эллипсоида было положительным.

Трассирующий луч, исходящий из некоторого рецептора и имеющий запись в параметрическом виде

$$X = X_F + lt, \quad Y = Y_F + mt, \quad Z = Z_F + nt, \quad (6.5.1)$$

ориентирован случайным образом относительно эллиптического облака. Из центра  $(X_0, Y_0, Z_0)$  эллипсоида опустим перпендикуляр  $n$  на прямую (6.5.1); координаты точки пересечения  $(X_\perp, Y_\perp, Z_\perp)$  трассирующего луча с перпендикуляром  $n$  можно найти, решив систему двух уравнений:

$$\begin{cases} l(X - X_0) + m(Y - Y_0) + n(Z - Z_0) = 0; \\ (X - X_F)/l = (Y - Y_F)/m = (Z - Z_F)/n, \end{cases}$$

где первое уравнение описывает плоскость, перпендикулярную прямой (6.5.1). После преобразования получим  $X = X_F + lU$ ;  $Y = Y_F + mU$ ;  $Z = nU$ , где  $U = \{Z_F(l^2/n - m^2/n) + l(X_0 - X_F) + m(Y_0 - Y_F) + Z_0n\} / [l^2 + n^2 + m^2]$ . Определим значение функции  $Q(X, Y, Z)$  в точке  $(X, Y, Z)$ :  $Q(X, Y, Z) = [X \ Y \ Z \ 1] Q[X \ Y \ Z \ 1]^T$ . Если  $Q(X, Y, Z) < 0$ , то луч не “видит” облако и следует находить его перемещение с поверхностью Земли (объектами или небесной сферой). В противном случае луч пересекает облако и должна быть определена

функция прозрачности  $a$  в точке  $(X_a, Y_a, Z_a)$ . Сформируем нормированную функцию эллипсоида  $g(X, Y, Z)$  по правилу  $g(X, Y, Z) = Q(X, Y, Z)/Q(X_0, Y_0, Z_0)$ . В таком случае в центре эллипсоида  $g(X_0, Y_0, Z_0) = 1$ , на поверхности эллипсоида в точке  $(X_\infty, Y_\infty, Z_\infty)$  –  $g(X_\infty, Y_\infty, Z_\infty) = 0$  и за пределами эллипсоида функция  $g$  отрицательна.

Координаты видимой точки  $(X_B, Y_B, Z_B)$  определяют путем решения системы:  $Q(X, Y, Z) = 0$   $(X - X_F)/l = (Y - Y_F)/m = (Z - Z_F)/n$  и выбора из возможных решений ближайшей точки к точке фокуса. Особенности решения пересечений эллипсоида и прямой обсуждались в § 3.4.4, 5.3.1.

Таким образом, для каждого трассирующего луча определяется видимая точка  $(X_B, Y_B, Z_B)$  и нормированная функция  $g(X_\perp, Y_\perp, Z_\perp)$ , показывающая удаление трассирующего луча от центра эллипсоида.

В [94] приводится правило определения освещенности на поверхности эллиптического тела облака  $E = (1-a)\{(1-t)(1-s)E_d + sE_s + tT(X_\perp, Y_\perp, Z_\perp)\} + aE_0$ , где  $E_d = E_d(X_B, Y_B, Z_B)$  – освещенность видимой точки поверхности облака, определяемая диффузными свойствами;  $E_s = ES(X_B, Y_B, Z_B)$  – освещенность видимой точки  $(X_B, Y_B, Z_B)$  на поверхности облака, определяемая зеркальными свойствами;  $E_0$  – освещенность участка поверхности, на который падает трассирующий луч после прохождения облака (поверхность Земли или небо);  $a$  – функция пропускания света сквозь облако, определяемая текстурными свойствами и удалением трассирующего луча от центра облака,  $a = [0, 1]$ . Для абсолютно прозрачных участков  $a = 1$ ;  $t = [0, 1]$  – доля освещенности, определяемая текстурными свойствами, при  $t = 0$  текстура не учитывается, при  $t = 1$  – учитывается в полной мере.

Коэффициент  $a$  называют еще *коэффициентом прозрачности*. Он тесно связан с текстурными свойствами, визуально воспринимаемыми как характерное изменение плотности и проявление размытости. В качестве псевдослучайной функции, имитирующей текстуру трехмерного облака, обычно используется [94] следующая функция:

$$T(X, Y, Z) = k \sum_{i=1}^n [C_i \sin(f_{X_i} X + \phi'_{X_i}) + T_0] \sum_{i=1}^n [C_i \sin(f_{Y_i} Y + \phi'_{Y_i}) + T_0],$$

где  $X, Y, Z$  – координаты некоторой точки на поверхности облака;  $T_0, k$  – постоянные коэффициенты.

Параметры  $C_1, f_{X_1}, f_{Y_1}$  выбираются в зависимости от типа и особенностей облака, относительно них строятся следующие члены:

$$f_{X_{i+1}} = 2f_{X_i}, f_{Y_{i+1}} = 2f_{Y_i}, C_{i+1} = 2^{0.5}C_i.$$

Фазы  $j'_{X_i}, j'_{Y_i}$  рассчитываются по правилу

$$j'_{X_i} = 1/2\pi \sin(1/2f_{Y_i} Y) + \pi \sin(f_{X_i} Z/2);$$

$$j'_{Y_i} = 1/2\pi \sin(1/2f_{Y_i} Y) + \pi \sin(f_{X_i} Z/2).$$

Параметр  $T_0$  определяет контраст текстурной функции  $T(X, Y, Z)$ , коэффициент  $k$  подбирается таким образом, чтобы обеспечить максимальное значение  $T(X, Y, Z)$  равным единице.

Определив текстурную функцию, построим функцию прозрачности:  $a = 1 - \{T(X_\perp, Y_\perp, Z_\perp) - T_1 - (T_2 - T_1)[1 - g(X_\perp, Y_\perp, Z_\perp)]\}/D$ , где  $T_1$  – пороговое значение текстурной функции в центре эллипсоида;  $T_2$  – пороговое значение текстурной функции на границе эллипсоида;  $D$  – диапазон изменения значений текстурной функции, в пределах которого прозрачность  $a$  меняется от 0 до 1.

В целях проработки полутеней в формулу вычисления  $E$  добавим член, определяющий подсветку от надоблачных слоев атмосферы:  $E = (1-a)\{(1-t)[(1-s)IE_d + (1-l)E_H + sE_S] + tT(X_\perp, Y_\perp, Z_\perp)\} + aE_0$ , где  $1-l$  – доля освещенности на поверхности облака из-за подсветки от надоблачных слоев атмосферы;  $E_H$  – освещенность видимой точки  $(X_B, Y_B, Z_B)$  облака из-за подсветки.

Для моделирования перистых облаков их тело выбирается плоским, а не эллиптическим и функция прозрачности  $a$  строится по правилу  $a = 1 - (T(X_B, Y_B, Z_B) - T_1)/D$ , где  $T_1$  – пороговое значение. Если  $T(X_B, Y_B, Z_B) < T_1$ , то  $a = 0$ , если иначе, то значение  $a$  вычисляется.

Указанный подход применяется для расчета освещенности в рецепторах. Если же рецептор видит поверхность Земли, то следует определить яркость на земной поверхности из-за возможной падающей от облака тени. В соответствии с правилами трассирования лучей проведем прямую их точки видимости  $(X_B, Y_B, Z_B)$  на поверхность Земли в направлении Солнца и оценим возможность пересечения этой прямой с эллиптическим телом облака. Если пересечений нет, то освещенность в рецепторе определяется освещенностью земной поверхности. Если пересечение

есть, то следует определить функцию непрозрачности облака на пути рассматриваемого солнечного луча по приведенным выше правилам. Освещенность  $E_T$  земной поверхности, имевшей исходную бесстеневую освещенность  $E_S$ , определяется по правилу  $E_T = E_S a + E$ , где  $E$  – освещенность земной поверхности в случае полной непрозрачности ( $a = 0$ ) облака и создаваемая переотражением света от пространственно распределенных источников.

## 6.6. РЕКЛАМА И МУЛЬТИПЛИКАЦИЯ

Визуальная привлекательность синтезированных изображений с первых шагов развития машинной графики была использована в интересах рекламы [98]. С эстетической точки зрения машинная графика выделяется цветовыми эффектами, объединением фотографичности изображения и необычности движения предметов, возможностями отображения динамики деформации объекта [93]. Классическим примером такой рекламы является отображение текста из объемных букв (обычно название фирмы или товара). Каждую букву динамически перемещают в пространстве, что создает привлекательную картину. Известна целая серия видеоклипов с двигающимися роботами, летающими самолетами и т.д. С технической точки зрения в такие системы вводят цветность, зеркальность, меняют центральную проекцию на ортогональную, иногда устраняют тона за счет псевдоцветов и не моделируют тени, что позволяет сэкономить машинное время и придать картине необычность, оригинальность. Например, переход к ортогональной проекции позволяет устранять единственную нелинейную операцию при определении видимой точки. Так как остальные операции линейны, то их реализуют очень быстро на матричном процессоре. Машинная объемная графика стала инструментом автоматизации основного процесса мультиплексионной технологии – изображения перемещающихся фигур и предметов. Математическими методами трудно передать мимику, выражение лица, но для некоторых простых сюжетов удастся получить синтетические изображения, полностью подготовленные машиной.

Обычно математическими методами описывают интерьер квартиры, сцену зданий и др., моделируют автоматически серию их изображений в меняющихся в соответствии с сюжетом условиях наблюдения, а главных героев прорисовывают вручную на фоне

синтезированных картин. В то же время успехи вычислительной техники позволили произвести и полностью синтетические мультифильмы, где героями являются стилизованные насекомые, роботы, куклы и даже люди, “списанные” с натуры [72, 140]. В отличие от тренажеров системы графики для мультипликации не требуют выполнения режима реального времени в процессе синтеза изображений. Обычным показателем является соотношение – на 10 секунд демонстрации мультфильма 5 часов работы машины при подготовке кадров. Графика в мультипликации используется и как генератор положения не всего объекта, а лишь нескольких характерных контуров и точек, все остальное дорисовывает художник.

**Оживление изображений.** Методами машинной графики можно создать не только статические фигуры, в том числе человеческую фигуру и лицо конкретного человека, но и передать динамику походки, мимику лица. С потребительской точки зрения такие приложения используются в рекламе, мультипликации, компьютерных видеоклипах. Например, создан фильм на основе компьютерного моделирования, где воссозданы известные киноактеры, по сценарию они сидят за столом, разговаривают, улыбаются, поднимают предметы сервировки [72, 112].

Моделирование трехмерного человеческого тела в машинной графике занимает около 15 лет. Кроме тела моделировались лицо и руки. Основная идея моделирования связана с созданием последовательности изображений, которые впоследствии можно соединить в один мультфильм. В идеале перед системой компьютерного моделирования человека ставятся следующие задачи [78, 82]: внешняя похожесть на конкретного человека; имитация поведения, походки, манеры, мимики и разговорной мимики; обеспечение естественности деформации тела при движении.

Достижение этих требований обеспечивается в свою очередь: точной передачей формы, цвета, текстуры, отражательных свойств; моделированием перемещений различных частей тела при движении; моделированием челюстно-лицевой мимики; обеспечением связности, синхронности движущихся частей.

В трехмерной графике оживление моделируется двумя основными способами: по опорным кадрам и алгебраически. Суть оживления по опорным кадрам заключается в автоматической генерации промежуточных кадров, которые словно заключены в интервале между опорными кадрами. Такой подход имеет прямую

аналогию в математике – интерполяция по опорным точкам. Описанная техника, в свою очередь, делится на две ветви: генерация промежуточных кадров в результате интерполяции ключевых кадров [140]; интерполяция параметров модели объекта.

При интерполяции параметров оператор создает опорные кадры на языке значений параметров формы объектов, затем параметры интерполируются с некоторым шагом. На каждом шаге приращения строится свое изображение.

В алгебраическом оживлении движение частей описывается математическими моделями, которые отражают физические законы перемещения человеческой фигуры.

При моделировании живого человека ключевое значение имеют модели походки, действий и разговора. Моделирование походки соответствует задаче перемещения из одной точки в другую с заданными характеристиками. Необходимо передать траекторию, скорость и ускорение каждой части, например руки, ноги, головы. Моделирование действия соответствует задаче взятия предмета в точке А, перемещению его в точку В по естественной траектории на другой предмет. Моделирование разговора связано с выполнением команды на произнесение конкретной фразы. Само акустическое звучание передается синтезатором речи, а мимика – синтезатором формы. Система должна анализировать фразу, разделять ее на звуки и находить соответствующее выражение лица для этих звуков.

Выражение лица физически определяется напряжением мускулов, состоянием рта, губ, глаз. После нахождения соответствия фраза – звуки – выражение лица система должна разработать расписание во времени для генерации различных выражений лица и синхронизировать его с работой синтезатора речи.

# ПРИЛОЖЕНИЕ. ОПИСАНИЕ ПРИКЛАДНЫХ ПОДПРОГРАММ

(Подпрограммы даны в алфавитном порядке)

## Подпрограмма ALFF

1. Вычисление косинуса угла между двумя векторами, заданными проекциями ( $s_1, s_2, s_3$ ), ( $n_1, n_2, n_3$ ), где  $s_1$  – проекция на ось  $OX$ ,  $s_2$  –  $OY$ ,  $s_3$  –  $OZ$ ;  $n_1, n_2, n_3$  – соответственно.

2. Обращение к подпрограмме:

$\text{ALFF}(s_1, s_2, s_3, n_1, n_2, n_3, a);$

3. Описание параметров, пояснения:

$s_1, s_2, s_3$  – real, проекции первого вектора на оси  $OX, OY, OZ$ ;

$n_1, n_2, n_3$  – real, проекции второго вектора на оси  $OX, OY, OZ$ ;

$a$  – real, значение косинуса угла между векторами.

4. Тело подпрограммы:

procedure ALFF(  $s_1, s_2, s_3, n_1, n_2, n_3$  : Re; var  $a$  : Re);

begin

$a := (s_1 * n_1 + s_2 * n_2 + s_3 * n_3) / \sqrt{(s_1^2 + s_2^2 + s_3^2) * (n_1^2 + n_2^2 + n_3^2)}$ ;

$(Sqr(n_1) + Sqr(n_2) + Sqr(n_3)))$ ;

end; { – ALFF – }

## Подпрограмма ALFO

1. Формирует матрицу преобразований

$$M^{-1} = \begin{bmatrix} -\cos(\alpha) & \sin(\alpha) & 0 & 0 \\ -\sin(\alpha)\sin(\beta) & -\sin(\beta)\cos(\alpha) & \cos(\beta) & 0 \\ -\sin(\alpha)\cos(\beta) & -\cos(\alpha)\cos(\beta) & -\sin(\beta) & 0 \\ X_0 & Y_0 & Z_0 & 1 \end{bmatrix}$$

по известным координатам ( $X_0, Y_0, Z_0$ ) центра снимка и координатам ( $X_{\text{пр}}, Y_{\text{пр}}, Z_{\text{пр}}$ ) точки прицеливания на объекте.

2. Вызов подпрограммы:

$\text{ALFO}(XYZ0, XYZ1, M);$

3. Описание параметров:

$XYZ0$  – array [1..3] of real, координаты центра снимка записаны в матрице  $XYZ0$  в последовательности  $X_0, Y_0, Z_0$ ;

$XYZ1$  – array [1..3] of real, координаты точки прицеливания записаны в матрице  $XYZ1$  в последовательности  $X_{\text{пр}}, Y_{\text{пр}}, Z_{\text{пр}}$ ;

$M$  – array [1..4, 1..4] of real, искомая матрица преобразования.

4. Тело подпрограммы:

procedure ALFO(  $XYZ0, XYZ1$  : mr\_3; var  $M$  : mr\_4\_4);

label

1,2;

var

$sf, cf, sa, ca, x, y, xg, yg$  : Re;

$ib, i, j$  : Int;

begin

```

for i:=1 to 4 do
  for j:=1 to 4 do M[i,j]:=0;
  for ib:=1 to 3 do
    begin
      M[ib,ib]:=-1;
      M[4,ib]:=XYZ0[ib];
    end;
    if (XYZ0[1]=XYZ1[1])AND(XYZ0[2]=XYZ1[2]) then goto 1;
    sf:=0;
    for j:=1 to 3 do
      sf:=sf+Sqr(XYZ0[j]-XYZ1[j]);
      sf:=Abs(XYZ0[3]-XYZ1[3])/Sqrt(sf);
      cf:=Sqrt(1-Sqr(sf));
      x:=XYZ1[1]-XYZ0[1];y:=XYZ1[2]-XYZ0[2];
      xr:=Abs(x)/Sqrt(Sqr(x)+Sqr(y));yr:=Abs(y)/Sqrt(Sqr(x)+Sqr(y));
      if (x<=0)AND(y<0) then
        begin sa:=xr;ca:=yr; goto 2; end;
      if (x<0)AND(y>0) then
        begin sa:=xr;ca:=-yr; goto 2; end;
      if (x>=0)AND(y>0) then
        begin sa:=-xr;ca:=-yr; goto 2; end;
      sa:=-xr;ca:=yr;
2:   M[1,1]:=-ca; M[1,2]:=sa;   M[2,1]:=-sa*sf; M[2,2]:=-ca*sf;
      M[2,3]:=cf; M[3,1]:=-sa*cf; M[3,2]:=-ca*cf; M[3,3]:=-sf;
1:
      M[4,4]:=1;
end;{ — ALFO — }

```

**Подпрограмма ALLIPS**  
(обращается к подпрограмме KVADR)

1. Обеспечивает решение системы из двух уравнений вида

$$(X - X_{11})/(X_{11} - X_f) = (Y - Y_{11})/(Y_{11} - Y_f) = (Z - Z_{11})/(Z_{11} - Z_f); \\ (X - X_0)^{**2}/a^{**2} + (Y - Y_0)^{**2}/b^{**2} + (Z - Z_0)^{**2}/c^{**2} - 1 = 0.$$

Первое уравнение описывает прямую, проходящую через точки  $(X_{11}, Y_{11}, Z_{11})$  и  $(X_f, Y_f, Z_f)$ , второе – эллипсоид.

2. Обращение к подпрограмме:

ALLIPS(T, A11, AL, ANS2, INDEX);

3. Описание параметров:

TF – array [1..3] of real, TF = [ Xf, Yf, Zf ];

A11 – array [1..3] of real, A11 = [ X11, Y11, Z11 ];

AL – array [1..2, 1..3] of real;

ANS2 – array [1..2, 1..3] of real;

INDEX – integer, индикатор,

INDEX =  $\begin{cases} -1, & \text{если нет решений;} \\ 0, & \text{единственное решение;} \\ 1, & \text{два решения различных.} \end{cases}$

4. Тело подпрограммы:

procedure ALLIPS( T,A11 : mr\_3; AL : mr\_2\_3;

var Ans2 : mr\_2\_3; var Index : Int);

var

```

xz,yz,yy,xx,aa,ba,da : Re;
i,j : Int;
{Прим.: mr_3 - array[1..3] of real,
mr_2_3 -array[1..2,1..3] of real,
Re- real,
Int - integer.}
begin
  for i:=1 to 2 do
    for j:=1 to 3 do
      Ans2[i,j]:=0;
  xx:=A11[3]-T[3];
  xz:=(A11[1]-T[1])/xx;
  yz:=(A11[2]-T[2])/xx;
  xx:=A11[1]-AL[1,1]; yy:=A11[2]-AL[1,2];
  AA:=Sqr(AL[2,3]*AL[2,1]*yz)+Sqr(AL[2,1]*AL[2,2])+  

    Sqr(AL[2,3]*AL[2,2]*xz);
  BA:=2*(-A11[3]*Sqr(AL[2,3]*AL[2,1]*yz)
    +Sqr(AL[2,3]*AL[2,1])*yz*yy  

    -AL[1,3]*Sqr(AL[2,1]*AL[2,2])
    -A11[3]*Sqr(AL[2,2]*AL[2,3]*xz)
    +Sqr(AL[2,2]*AL[2,3])*xz*xx);
  DA:=Sqr(AL[2,3]*AL[2,1]*(A11[3]*yz-yy))
    +Sqr(AL[2,3]*AL[2,2]*(A11[3]*xz-xx))
    +Sqr(AL[2,1]*AL[2,2])*(Sqr(AL[1,3])-Sqr(AL[2,3]));
  KVADR(aa,ba,da,Index,Ans2[1,3],Ans2[2,3]);
  if Index=2 then Exit;
  Ans2[1,1]:=(Ans2[1,3]-A11[3])*xz+A11[1];
  Ans2[1,2]:=(Ans2[1,3]-A11[3])*yz+A11[2];
  Ans2[2,1]:=(Ans2[2,3]-A11[3])*xz+A11[1];
  Ans2[2,2]:=(Ans2[2,3]-A11[3])*yz+A11[2];
end;{ — ALLIPS — }

```

**Подпрограмма ALLOID**  
(обращается к ALLIPS, NA8FIR, SOME, PLOS)

1. Вычисляет ближайшую к некоторой точке ( $X_F, Y_F, Z_F$ ) точку пересечения прямой и эллипса, ограниченного двумя отсекающими плоскостями. Эллипсоид имеет главные оси, параллельные координатным осям. Положение отсекающих плоскостей произвольное.

2. Обращение к подпрограмме:

ALLOID( $T, F, ALL, FS, XYZ, IND, NORM$ );

3. Описание параметров, пояснения:

$T$  – array[1..3] of real. Координаты первой точки, через которую проходит прямая;

$F$  – array[1..3] of real. Координаты второй точки, через которую проходит прямая;

$ALL$  – array[1..2,1..3] of real. Это матрица параметров в уравнении эллипса:  

$$(X-X_0)^2/a^2 + (Y-Y_0)^2/b^2 + (Z-Z_0)^2/c^2 - 1 = 0$$

$$ALL = \begin{bmatrix} a & b & c \\ X_0 & Y_0 & Z_0 \end{bmatrix};$$

FS – array[1..4,1..2] of real. Матрица коэффициентов плоскостей, ограничивающих тело эллипсоида:  $A1*X+B1*Y+C1*Z+D1=0$  и  $A2*X+B2*Y+C2*Z+D2=0$ . Нормальный вектор к плоскостям должен быть направлен внутрь изображаемой части эллипсоида.

$$FS = \begin{bmatrix} A1 & A2 \\ B1 & B2 \\ C1 & C2 \\ D1 & D2 \end{bmatrix};$$

XYZ – array[1..3] of real. Координаты точки решения;

IND – integer, индикатор,

$$IND = \begin{cases} 1, & \text{если решение единственное;} \\ 0, & \text{если решений нет или их бесконечно много.} \end{cases}$$

NORM – array[1..3] of real. Матрица проекций нормального вектора к поверхности дважды усеченного эллипсоида в точке решения. Нормаль направлена внутрь тела.

4. Тело подпрограммы:

Procedure ALLOID( T,F : mr\_3; ALL : mr\_2\_3; FS : mr\_4\_2;  
var XYZ : mr\_3; var Ind : Int; var NORM : mr\_3);

var

P1 : mr\_SM\_3;

ANS : mr\_2\_3;

KTO,KAK : mr\_4\_1;

i,j : Int;

{Прим.: mr\_3 – array[1..3] of real,  
mr\_2\_3 – array[1..2,1..3] of real,  
mr\_4\_2 – array[1..4,1..2] of real,  
mr\_SM\_3 – array[1..4,1..3] of real,  
mr\_4\_1 – array[1..4,1] of real,  
Int – integer.}

begin

for i:=1 to 4 do KAK[i,1]:=0;  
KTO:=KAK;KTO[3,1]:=1;KTO[4,1]:=-1;

ALLIPS(F,T,ALL,ANS,Ind);

if Ind=2 then Exit;

for i:=1 to 3 do

begin

P1[1,i]:=ANS[1,i];

P1[2,i]:=ANS[2,i];

end;

for i:=1 to 4 do KAK[i,1]:=FS[i,1];

PLOS(T,F,KAK,XYZ,Ind);

for i:=1 to 3 do P1[3,i]:=XYZ[i];

if Ind=0 then Exit;

for i:=1 to 4 do KAK[i,1]:=FS[i,2];

PLOS(T,F,KAK,XYZ,Ind);

for i:=1 to 3 do P1[4,i]:=XYZ[i];

if Ind=0 then Exit;

for i:=1 to 4 do KAK[i,1]:=0;

for i:=1 to 2 do

```

if ((FS[1,1]*P1[i,1]+FS[2,1]*P1[i,2]+FS[3,1]*P1[i,3]+FS[4,1])>=0)
    AND
((FS[1,2]*P1[i,1]+FS[2,2]*P1[i,2]+FS[3,2]*P1[i,3]+FS[4,2])>=0)
    then KAK[i,1]:=1;
for i:=3 to 4 do
begin
    j:=5-i;
if ((FS[1,j]*P1[i,1]+FS[2,j]*P1[i,2]+FS[3,j]*P1[i,3]+FS[4,j])>=0)
    AND
((Sqr((P1[i,1]-ALL[1,1])/ALL[2,1])+
  Sqr((P1[i,2]-ALL[1,2])/ALL[2,2])+
  Sqr((P1[i,3]-ALL[1,3])/ALL[2,3])-1)<=0)
    then KAK[i,1]:=1;
end;
i:=4;
NA8FIR(P1,KTO,KAK,F,I,J);
if J=0 then
begin
    Ind:=0;
    Exit;
end;
for i:=1 to 3 do XYZ[i]:=P1[J,i];
Ind:=1;
if KTO[J,1]=1 then
begin
    for i:=1 to 3 do NORM[i]:=FS[i,1];
    Exit;
end;
if KTO[J,1]=-1 then
begin
    for i:=1 to 3 do NORM[i]:=FS[i,2]; Exit;
end;
for i:=1 to 3 do
    NORM[i]:=- (XYZ[i]-ALL[1,i])/Sqr(ALL[2,i]);
end;{ — ALLOID — }

```

### Процедура AVTOZON

- Формирует направляющие вектора отрезков прямых, соединяющих крайние точки двухмерной плоской выпуклой оболочки. Направляющие вектора ориентированы так, что достигается положительность значений функций каждой прямой внутри зоны. Операция производится над несколькими оболочками.
- Обращение к подпрограмме:

AVTOZON ( SKR, N, M, \_KR );

- Описание параметров, пояснения:

SKR – аттау [1..SM, 1..2, 1..MKT] of real. SKR – трехмерный массив, объединяющий MKT плоскостей, на каждой из которых задано SM точек посредством двух координат X, Y. Значения SM, MKT определяются вызывающим модулем. Порядок следования точек в SKR должен быть однозначным ( по или против часовой

стрелки внутри каждой зоны);

$N$  – integer, число обрабатываемых крайних точек каждой зоны;

$M$  – integer, число обрабатываемых плоскостей;

Прим.:  $N \leq SM$ ,  $M \leq MKT$ .  $_KR$  – array [1..SM, 1..2, 1..MKT] of real,  $_KR$  – трехмерный массив направляющих векторов-отрезков, соединяющих крайние точки.

Проекции направляющих векторов выбраны так, чтобы обеспечить для любой прямой, проходящей через точки  $i + 1$ ,  $i$ , положительность значений функции этой прямой во внутренней точке ( $X_b$ ,  $Y_b$ ,  $Z_b$ ) зоны:  $(X_b - X_{i+1})(Y_{i+1} - Y_i) - (X_{i+1} - X_i)(Y_b - Y_{i+1}) > 0$ .

4. Тело подпрограммы:

```
procedure AVTGZON( SKR : mr_SM_2_MKT; N,M : Int;
  var _KR : mr_SM_2_MKT); var i,j,k : Int;
  { Прим. mr_SM_2_MKT – array[1..SM,1..2,1..MKT] of real,
    Int – integer}
begin for i:=1 to (N-1) do
  for j:=1 to 2 do
    for k:=1 to M do
      _KR[i,j,k]:=SKR[i+1,j,k]-SKR[i,j,k];
  for j:=1 to 2 do
    for k:=1 to M do
      _KR[N,j,k]:=SKR[1,j,k]-SKR[N,j,k];
  for i:=1 to M do
    if ((SKR[3,1,i]-SKR[1,1,i])*_KR[1,2,i]
      -(SKR[3,2,i]-SKR[1,2,i])*_KR[1,1,i])<0
    then
      for j:=1 to N do
        for k:=1 to 2 do
          _KR[j,k,i]:=-_KR[j,k,i];
end;{ – AVTOZON – }
```

### Подпрограмма BETTA

1. BETTA рассчитывает косинус угла зеркальности, который определяется как угол между зеркально отраженным от поверхности лучом и направлением на приемник света.

2. Обращение к подпрограмме:

BETTA(n1,n2,n3,s1,s2,s3,r1,r2,r3,q);

3. Описание параметров, пояснения:

$n1, n2, n3$  – real, проекции нормали к поверхности соответственно на оси OX, OY, OZ. Нормаль – внешняя;

$s1, s2, s3$  – real, проекции вектора, направленного из точки поверхности на источник света;

$r1, r2, r3$  – real, проекции вектора, направленного из точки поверхности на приемник света;

$q$  – real, косинус угла зеркальности.

4. Тело подпрограммы:

```
procedure BETTA( n1,n2,n3,s1,s2,s3,r1,r2,r3 : Re;var q : Re );
begin
```

$q:=(s1*n1+s2*n2+s3*n3)/(Sqr(n1)+Sqr(n2)+Sqr(n3));$

$q:=(r1*(2.0*n1*q-s1)+r2*(2.0*n2*q-s2)+r3*(2.0*n3*q-s3))/$

```

Sqr((Sqr(r1)+Sqr(r2)+Sqr(r3))*  

(Sqr(2.0*n1*q-s1)+Sqr(2.0*n2*q-s2)+Sqr(2.0*n3*q-s3)));  

end;{ — ВЕТТА — }

```

**Подпрограмма CILIND**  
(обращается к PLOS, KVADR, NA8FIR, SOME, PRESS)

1. Подпрограмма вычисляет координаты точки пересечения прямой с поверхностью тела, ограниченного круговым цилиндром и торцевыми плоскими кругами. Из всех возможных пересечений выбирается ближайшая точка к некоторой критериальной ( $X_f, Y_f, Z_f$ ), лежащей на прямой. Вычисляются параметры ненормированного нормально-го вектора, ориентированного внутрь тела. Положение цилиндра – произвольное и задается двумя осевыми точками на торцах.

2. Обращение к подпрограмме:

CILIND(F, S, R, T11, TF, GOT, WHAT, NORA);

3. Описание параметров, пояснения:

F – array [1..3] of real. F – матрица координат X, Y, Z осевой точки цилиндра на его первом торце;

S – array [1..3] of real. F – матрица координат X, Y, Z осевой точки цилиндра на его втором торце;

R – real. R – радиус цилиндра;

T11 – array [1..3] of real. T11 – матрица координат X11, Y11, Z11 первой точки, через которую проходит прямая;

TF – array [1..3] of real. TF – матрица координат X11, Y11, Z11 второй (критериальной) точки, через которую проходит прямая;

GOT – array [1..3] of real. GOT – матрица искомых координат точки, решения X, Y, Z; WHAT – integer, индикатор. WHAT = 1, если решение единствено, WHAT = 0, если решений нет или их бесконечно много;

NORA – array [1..3] of real. В этой матрице записаны проекции ненормированной нормали к поверхности тела в точке решения ( GOT ) соответственно вдоль осей OX, OY, OZ. Нормаль направлена внутрь тела.

4. Ограничения и области допустимых значений.

Состояние  $Z_{11} \geq Z_f$  (  $T11[3] \geq TF[3]$  ) недопустимо.

5. Тело подпрограммы:

```

procedure CILIND(F,S : mr_3 ;R : Re; T11,TF: mr_3;
                  var GOT : mr_3; var WHAT : Int; var NORA : mr_3);
var
  ABC : mr_3;
  XZ,YZ,XX,YY,Lola,Stom,A,B,C : Rc;
  P1 : mr_SM_3;
  WHO,KAK,FIR,SEK : mr_4_1;
  N,I,J,Jop : Int;
{Прим.: mr_3 – array[1..3] of real,
  mr_SM_3 – array[1..4,1..3] of real,
  mr_4_1 – array[1..4,1] of real,
  Re – real,
  Int – integer.}
begin
  N:=4;
  WHO[1,1]:=0;WHO[2,1]:=0;WHO[3,1]:=1;WHO[4,1]:=1;

```

```

for i:=1 to 3 do ABC[i]:=S[i]-F[i];
Lola:=0;
for i:=1 to 3 do Lola:=Lola+Sqr(ABC[i]);
Lola:=Sqrt(Lola);
for i:=1 to 3 do ABC[i]:=ABC[i]/Lola;
XX:=T11[1]-F[1]; YY:=T11[2]-F[2];
XZ:=(T11[1]-TF[1])/(T11[3]-TF[3]);
YZ:=(T11[2]-TF[2])/(T11[3]-TF[3]);
Lola:=ABC[1]*XZ+ABC[2]*YZ+ABC[3];
Stom:=-T11[3]*(ABC[1]*XZ+ABC[2]*YZ)-ABC[3]*F[3]-
    ABC[1]*XX+ABC[2]*YY;
A:=Sqr(XZ)+Sqr(YZ)+1-Sqr(Lola);
if A=0 then
begin
    WHAT:=0;
    Exit;
end;
B:=2*(-Sqr(XZ)*T11[3]+XZ*XX-Sqr(YZ)*T11[3]+
    YZ*YY-F[3]-Lola*Stom);
C:=Sqr(XZ*T11[3]-XX)+Sqr(YZ*T11[3]-YY)+
    Sqr(F[3])-Sqr(Stom)-Sqr(R);
KVADR(A,B,C,WHAT,P1[1,3],P1[2,3]);
if WHAT=2 then
begin
    WHAT:=0;
    Exit;
end;
for i:=1 to 3 do
    FIR[i,1]:=ABC[i];
    FIR[4,1]:=-ABC[1]*F[1]-ABC[2]*F[2]-ABC[3]*F[3];
for i:=1 to 4 do SEK[i,1]:=-FIR[i,1];
    SEK[4,1]:=ABC[1]*S[1]+ABC[2]*S[2]+ABC[3]*S[3];
    PLOS(T11,TF,FIR,GOT,WHAT);
if WHAT=0 then Exit;
for i:=1 to 3 do P1[3,i]:=GOT[i];
    PLOS(T11,TF,SEK,GOT,WHAT);
for i:=1 to 3 do P1[4,i]:=GOT[i];
for i:=1 to 2 do
begin
    DIRECT(T11,TF,P1[i,3],GOT);
    for j:=1 to 3 do P1[i,j]:=GOT[j];
end;
for i:=1 to 2 do
if ((P1[i,1]*FIR[1,1]+P1[i,2]*FIR[2,1]+
    P1[i,3]*FIR[3,1]+FIR[4,1])>=0)
    AND
    ((P1[i,1]*SEK[1,1]+P1[i,2]*SEK[2,1]+
    P1[i,3]*SEK[3,1]+SEK[4,1])>=0)
then KAK[i,1]:=1
else KAK[i,1]:=0;

```

```

for i:=3 to 4 do
if (Sqr(P1[i,1]-F[1])+Sqr(P1[i,2]-F[2])+Sqr(P1[i,3]-F[3])
    -Sqr(ABC[1]*(P1[i,1]-F[1])+ABC[2]*(P1[i,2]-F[2])+
    ABC[3]*(P1[i,3]-F[3]))-Sqr(R)) <= 0
    then KAK[i,1]:=1
    else KAK[i,1]:=0;
    NA8FIR(P1,WHO,KAK,TF,N,Jop);
if Jop=0 then
begin
    WHAT:=0;
    Exit;
end;
for i:=1 to 3 do GOT[i]:=P1[Jop,i];
WHAT:=1;
if WHO[Jop,1]=1 then
begin
    NORA:=ABC;
    Exit;
end;
Lola:=0;
for i:=1 to 3 do
    Lola:=Lola+ABC[i]*(GOT[i]-F[i]);
for i:=1 to 3 do
    NORA[i]:=GOT[i]-F[i]-ABC[i]*Lola;
end;{ — CILIND — }

```

### Подпрограмма DIRECT

1. По заданному уравнению прямой и известной координате Z некоторой точки на прямой находит две остальные координаты точки. Прямая задается посредством координат двух известных точек ( $X_1, Y_1, Z_1$ ) и ( $X_2, Y_2, Z_2$ ).

2. Обращение к подпрограмме:

DIRECT(A, B, Z, XYZ);

3. Описание параметров:

A – array [1..3] of real, матрица координат первой точки ( $X_1, Y_1, Z_1$ ), через которую проходит прямая;

B – array [1..3] of real, матрица координат второй точки ( $X_2, Y_2, Z_2$ ).

Z – real, известная координата Z на прямой;

XYZ – array[1..3], матрица координат точки на прямой.

4. Ограничения и область допустимых значений:  $Z_1 \neq Z_2$ .

5. Тело подпрограммы:

```

procedure DIRECT(A,B : mr_3; Z : Re; var XYZ : mr_3);
{ Прим.: mr_3 – array[1..3] of real, Re – real, Int – integer}
var
    i : Int;
begin
    XYZ[1]:=(Z-A[3])*(A[1]-B[1])/(A[3]-B[3])+A[1];
    XYZ[2]:=(Z-A[3])*(A[2]-B[2])/(A[3]-B[3])+A[2];
    XYZ[3]:=Z;
end;{ — DIRECT — }

```

## Подпрограмма E2

1. Вычисляет освещенность точки поверхности, обладающей диффузными и зеркальными (блковыми) свойствами при освещении точечным источником (Солнцем) и полусферическим облучателем (небом).

2. Обращение к подпрограмме:

E2(b,ca,cb,n,cg,a,r,ed,ez,et,t);

3. Описание параметров, пояснения:

$b$  – real, освещенность точки поверхности,  $b = [0,1]$ ;

$ca$  – real, косинус угла между внешней нормалью к поверхности и направлением на Солнце;

$cb$  – real, косинус угла между направлением на приемник света и зеркально отраженным лучом;

$n$  – integer, показатель ширины диаграммы зеркального отражения,  $n = [1,200]$ ;

$cg$  – real, косинус угла между внешней нормалью к поверхности и направлением вертикально вверх (в зенит неба);

$a$  – real, коэффициент яркости поверхности (диффузные свойства),  $a = [0,1]$  ;

$r$  – real, коэффициент зеркального отражения поверхности,  $r = [0,1]$ ;

$ed$  – real, относительный диапазон освещенности поверхности, обладающей только диффузными свойствами ( $a = 1$ );

$ez$  – real, относительный диапазон освещенности поверхности, обладающей только зеркальными свойствами ( $r = 1$ );

$et$  – real, относительный диапазон освещенности в тени;

$t$  – real, признак освещенности точки поверхности,  $t = 1$  – на свету,  $t = 0$  – в тени.

Примечание:  $ed,ez,et$  нормируются друг к другу так, чтобы  $ed + ez + et = 1$ .

4. Тело подпрограммы:

```
procedure E2(var b : Re;ca,cb: Re;n: Int;cg,a,r,ed,ez,et,t : Re);
```

```
function Step( cb : Re; n : Int ) : Re;
```

```
var
```

```
    c : Re;
```

```
begin
```

```
    c:=1; if n>0 then
```

```
        repeat
```

```
            c:=c*cb; Dec(n); until n=0; Step:=c; end;
```

```
begin
```

```
    if cb>0.05 then b:=(t*ez*r*Step(cb,n)+a*et*0.5*(1+cg)+ed*ca*a*t)
```

```
        else b:=a*et*0.5*(1+cg)+ed*ca*a*t;
```

```
end;{ – E2 – }
```

## Подпрограмма KONYs (обращается к PICA, DIRECT, NA8FIR)

1. Определяет ближайшую к фокусу (XF,YF,ZF) точку пересечения прямой (светового луча) и конуса, ограниченного двумя плоскостями, а также нормальный вектор в установленной точке.

2. Обращение к подпрограмме

KONYs(T,F,KON,FS,XYZ,Ind,Norm);

## 3. Описание параметров, пояснения:

**T** – array[1..3] of real.  $T = [X_{11} \ Y_{11} \ Z_{11}]$  – матрица координат первой точки, через которую проходит прямая светового луча;

**F** – array[1..3] of real.  $F = [X_F \ Y_F \ Z_F]$  – матрица координат второй точки, через которую проходит прямая светового луча;

**KON** – array[1..2,1..3] of real.

$$KON = \begin{bmatrix} A & B & C \\ X_0 & Y_0 & Z_0 \end{bmatrix}$$

– матрица коэффициентов уравнения конуса (см. описание в подпрограмме PICA);

**FS** – array[1..4,1..2] of real.

$$FS = \begin{bmatrix} A_1 & A_2 \\ B_1 & B_2 \\ C_1 & C_2 \\ D_1 & D_2 \end{bmatrix}$$

– коэффициенты уравнений первой  $A_1 \cdot X + B_1 \cdot Y + C_1 \cdot Z + D_1 = 0$  и второй  $A_2 \cdot X + B_2 \cdot Y + C_2 \cdot Z + D_2 = 0$  плоскостей, секущих конус, причем нормаль к поверхности плоскостей должна быть направлена внутрь видимой части конуса;

**XYZ** – array [1..3] of real, координаты решения;

**IND** – integer, индикатор,

**IND** = 1, если решение единственное;  
0, если решений нет;

**NORM** – array [1..3] of real. Матрица проекций нормального вектора N на оси OX,OY,OZ.  
Нормаль внутренняя.

## 4. Ограничения и области допустимых значений.

Проверка содержательной части входных параметров отдельно не производится.

## 5. Тело подпрограммы:

```
procedure KONYS( T,F : mr_3; KON : mr_2_3; FS : mr_4_2;
```

```
    var XYZ : mr_3; var Intd : Int; var NORM : mr_3);
```

```
var P : mr_SM_3;
```

```
    KTO,KAK : mr_4_1;
```

```
    i,j,k : Int;
```

{Прим.: mr\_3 – array[1..3] of real, mr\_2\_3 – array[1..2,1..3]

of real, mr\_4\_2 – array[1..4,1..2] of real, mr\_SM\_3

– array[1..4,1..3] of real, mr\_4\_1 – array[1..4,1]

of real, Int – integer.}

```
begin for i:=1 to 4 do KAK[i,1]:=0;
```

```
    KTO:=KAK;
```

```
    KTO[3,1]:=1;KTO[4,1]:=1;
```

```
    PICA(T,F,KON,P[1,3],P[2,3],Ind);
```

```
    if Ind=0 then Exit;
```

```
    for i:=1 to 2 do
```

```
        begin
```

```
            DIRECT(T,F,P[i,3],XYZ);
```

```
            for j:=1 to 3 do P[i,j]:=XYZ[j];
```

```
        end;
```

```
    for i:=1 to 2 do
```

```
        begin
```

```
            for j:=1 to 4 do KAK[j,1]:=FS[j,i];
```

```
            PLOS(T,F,KAK,XYZ,Ind);
```

```

if Ind=0 then Exit;
j:=i+2;
for k:=1 to 3 do P[j,k]:=XYZ[k];
end;
for i:=1 to 4 do KAK[i,1]:=0;
for i:=1 to 2 do
  if (FS[1,1]*P[i,1]+FS[2,1]*P[i,2]+FS[3,1]*P[i,3]+FS[4,1]>=0)
AND (FS[1,2]*P[i,1]+FS[2,2]*P[i,2]+FS[3,2]*P[i,3]+FS[4,2]>=0)
    then KAK[i,1]:=1;
for i:=3 to 4 do
  if (Sqr(P[i,1]-KON[1,1])/KON[2,1] +
      Sqr(P[i,2]-KON[1,2])/KON[2,2] +
      Sqr(P[i,3]-KON[1,3])/KON[2,3])<=0
    then
      KAK[i,1]:=1;
i:=4;
NA8FIR(P,KTO,KAK,F,i,j);
if j=0 then
begin
  Ind:=0;
  Exit;
end;
for i:=1 to 3 do XYZ[i]:=P[j,i];
Ind:=1;
if KTO[j,1]<>0 then
  for i:=1 to 3 do NORM[i]:=FS[i,1]
else for i:=1 to 3 do NORM[i]:=(XYZ[i]-KON[1,i])/KON[2,i];
end;{ — KONYA — }

```

### Подпрограмма KOROB (обращается к PLOS, SOME, YESNO)

1. Определяет координаты точки пересечения прямой и выпуклого шестиугранника, а также вектор внешней нормали в точке пересечения. Прямая задается двумя точками ( $X_1, Y_1, Z_1$ ) и ( $X_2, Y_2, Z_2$ ). Среди всех возможных пересечений выбирается ближайшее к точке ( $X_2, Y_2, Z_2$ ). Шестиугранник задается матрицей KUBB, где каждый столбец описывает отдельную грань (плоскость), а в свою очередь плоскость описывается четырьмя коэффициентами A, B, C, D в уравнении  $AX + BY + CZ + D = 0$ . Знаки коэффициентов предварительно выбраны так, чтобы нормаль к каждой грани была направлена внутрь шестиугранника.

Область допустимых значений:  $Z_1 \neq Z_2$ .

2. Обращение к подпрограмме:

KOROB(T1, T2, KUBB, XYZ, NORM, ANT);

3. Описание параметров:

T1 – array [1..3] of real, T1[X1, Y1, Z1] – матрица координат первой точки, через которую проходит прямая;

T2 – array [1..3] of real, T2[X2, Y2, Z2] – матрица координат второй точки, через которую проходит прямая;

KUBB – array [1..4, 1..6] of real, матрица, описывающая шестиугранник;

XYZ – array [1..3] of real, матрица координат (X, Y, Z) ближайшего к точке ( $X_2, Y_2,$

Z2) пересечения прямой и шестигранника;

NORM – array [1..3] of real, матрица проекций вектора внешней нормали в точке пересечения на оси OX, OY, OZ;

ANT – integer, индикатор,

$$\text{ANT} = \begin{cases} 1, & \text{если решение единственное;} \\ 0 & \text{в остальных случаях.} \end{cases}$$

4. Тело подпрограммы:

```
procedure KOROB( T1,T2 : mr_3; KUBB : mr_4_6;
                  var XYZ,NORM : mr_3; var Ant : Int);
```

var

SOV : array[1..6,1..3] of Re;

BUF : mr\_SM\_3;

WHO : array[1..6] of Int;

KUBA: mr\_4\_1;

i,j,k,jor,an1,nom : Int;

{Прим.: mr\_3 – array[1..3] of real,  
 mr\_4\_6 – array[1..4,1..6] of real,  
 mr\_SM\_3 –array[1..4,1..3] of real,  
 mr\_4\_1 – array[1..4,1]of real,  
 Int – integer, Re – real.}

begin

j:=0;

for i:=1 to 6 do

begin

    KUBA[1,1]:=KUBB[1,i];

    KUBA[2,1]:=KUBB[2,i];

    KUBA[3,1]:=KUBB[3,i];

    KUBA[4,1]:=KUBB[4,i];

    PLOS(T1,T2,KUBA,XYZ,ANT);

    if Ant=1 then

        begin

            YESNO(XYZ,KUBB,An1);

            if An1=1 then

                begin

                    j:=j+1;

                    WHO[j]:=i;

                    for k:=1 to 3 do

                        SOV[j,k]:=XYZ[k];

                end;

            end;

        end;

    if j=0 then

        begin

            Ant:=0;                      Exit;

        end;

    Ant:=1;

    for i:=1 to 4 do

        for k:=1 to 3 do

            BUF[i,k]:=SOV[i,k];

    SOME(BUF,T2,j,Jor);

```

for i:=1 to 3 do
  XYZ[i]:=SOV[Jor,i];
j:=WHO[Jor];
for i:=1 to 3 do
  NORM[i]:=KUBB[i,j];
end;{ — KOROB — }

```

**Подпрограмма KRILo**  
(обращается к PLOS, YNZONA)

1. Определяет координаты и нормальный вектор в точке пересечения луча и плоского выпуклого многоугольника. Многоугольник задается уравнением плоскости и списком (по или против часовой стрелки) координат крайних точек, причем берутся только две координаты X и Y, или Y и Z, или Z и X.

2. Обращение к подпрограмме:

KRILo(P2, NOR\_, T, F, P, YN, XYZ,Prck);

3. Описание параметров:

P2 – array [1..SM\_2, 1..2] of real, матрица координат крайних точек, число столбцов 2, строк – любое ( N ).

$$P2 = \begin{bmatrix} X_1 & Y_1 \\ \vdots & \vdots \\ \vdots & \vdots \\ X_n & Y_n \end{bmatrix} \quad \text{или} \quad \begin{bmatrix} Y_1 & Z_1 \\ \vdots & \vdots \\ \vdots & \vdots \\ Y_n & Z_n \end{bmatrix} \quad \text{или} \quad \begin{bmatrix} Z_1 & X_1 \\ \vdots & \vdots \\ \vdots & \vdots \\ Z_n & X_n \end{bmatrix} .$$

Точки записаны по (против) часовой стрелке обхода вершин многоугольника;  
NOR\_ – array [1..SM\_2, 1..2], матрица проекций нормали к отрезкам, соединяющим две соседние вершины многоугольника. Число столбцов 2, строк N. В первой строке записан вектор, соединяющий первую и вторую точки из P2. Оси, на которые проецируется вектор, соответствуют осям, предусмотренным в P2, т. е.

$$\text{если } P2 = \begin{bmatrix} X & Y \\ \vdots & \vdots \\ \vdots & \vdots \end{bmatrix}, \text{ то } NOR_ = \begin{bmatrix} \text{Пр}X & \text{Пр}Y \\ \vdots & \ddots \\ \vdots & \vdots \end{bmatrix};$$

$$\text{если } P2 = \begin{bmatrix} Y & Z \\ \vdots & \vdots \\ \vdots & \vdots \end{bmatrix}, \text{ то } NOR_ = \begin{bmatrix} \text{Пр}Y & \text{Пр}Z \\ \vdots & \ddots \\ \vdots & \vdots \end{bmatrix};$$

$$\text{если } P2 = \begin{bmatrix} Z & X \\ \vdots & \vdots \\ \vdots & \vdots \end{bmatrix}, \text{ то } NOR_ = \begin{bmatrix} \text{Пр}Z & \text{Пр}X \\ \vdots & \ddots \\ \vdots & \vdots \end{bmatrix};$$

T – array [1..3] of real, координаты центра рецептора, T = [ X Y Z ];

F – array [1..3] of real, координаты точки фокуса, F = [ X<sub>f</sub> Y<sub>f</sub> Z<sub>f</sub> ];

P – array [1..4] of real, описание плоскости, в которой лежит многоугольник, где

A<sub>X</sub> + B<sub>Y</sub> + C<sub>Z</sub> + D = 0;

YN – integer, индикатор,

$$YN = \begin{cases} 1, & \text{есть решение,} \\ 0, & \text{нет решения;} \end{cases}$$

XYZ – array [1..3] of real, координаты точки решения пересечения прямой и многоугольника;

Proek – integer, указатель наименования координат крайних точек,

если Proek =  $\begin{cases} 1, & \text{если точка задана координатами } X, Y; \\ 2, & \text{если точка задана координатами } Y, Z; \\ 3, & \text{если точка задана координатами } Z, X. \end{cases}$

#### 4. Тело подпрограммы:

```
procedure KRILO(P2,NOR_ : mr_SM_2; N : Int; T,F : mr_3; P : mr_4_1;
```

```
var YN : Int; var XYZ : mr_3; Proek : Int );
```

```
var
```

```
T2 : array[1..2] of Re;
```

```
i,j,k1,k2 : Int;
```

{Прим.: mr\_3 – array[1..3] of real,

mr\_4\_1 – array[1..4,1]of real,

mr\_SM\_2 – array[1..SM,1..2]of real,

Int – integer, Re – real.}

```
begin
```

```
if Proek=1 then begin k1:=1;k2:=2; end;
```

```
if Proek=2 then begin K1:=2;K2:=3; end;
```

```
if Proek=3 then begin k1:=3;K2:=1; end;
```

```
PLOS(T,F,P,XYZ,YN);
```

```
if YN=0 then Exit;
```

```
T2[1]:=XYZ[k1];T2[2]:=XYZ[k2];
```

```
YNZONA(P2,NOR_,N,T2[1],T2[2],YN);
```

```
end;{ – KRILO – }
```

### Подпрограмма KVADR

1. Решает квадратное уравнение общего вида

$$A*X^{**2}+B*X+C=0.$$

2. Обращение к подпрограмме:

```
KVADR(A,B,C,Ind,X1,X2);
```

3. Описание параметров, пояснения:

A – real, коэффициент A.

B – real, коэффициент B.

C - real, коэффициент C.

A1 – real, значение первого решения.

A2 – real, значение второго решения.

Ind – real, индикатор:

Ind =  $\begin{cases} 2, & \text{если нет решений;} \\ 0, & \text{если имеется одно решение } A1 = A2; \\ 1, & \text{если имеется два решения.} \end{cases}$

## 4. Тело подпрограммы:

```

procedure KVADR(A,B,C : Re; var Intd : Int; var X1,X2 : Re);
var
  V : Re;
begin
  V:=Sqr(B)-4*A*C;
  if ( V<0 )or( A=0 ) then
    begin
      Intd:=2;
      Exit;
    end
    else
  begin
    V:=Sqrt(V);
    X1:=(-b+V)/(2*A);
    X2:=(-B-V)/(2*A);
  end;
  if V=0 then Intd:=0
  else Intd:=1;
end;{ — KVADR — }

```

## Подпрограмма MMGG

1. Вычисляет произведение двух матриц  $A[K,L]*B[L,M]$ . Результат помещается в матрицу  $C[K,M]$ .

2. Обращение к подпрограмме:

MMGG(A,B,K,L,M,C);

3. Описание параметров, пояснения:

$A$  – array[1..K,1..L] of real. Левый множитель матричного произведения  $A*B=C$ ;

$B$  – array[1..L,1..M] of real. Правый множитель матричного произведения  $A*B=C$ ;

$C$  – array[1..K,1..M] of real. Результат матричного произведения  $A*B=C$ ;

$K,L,M$  – byte, размерности матриц. Максимальное значение каждой размерности – 256.

4. Ограничения и области допустимых значений:

Размерности матриц должны соответствовать описаниям, максимальные значения  $L$ ,  $M$ ,  $K$  – 256.

5. Тело подпрограммы:

```
procedure MMGG(MA,MB : mr_SM_SM; k,l,m : By;var MC : mr_SM_SM);
```

var

  i,j,n : Int;

{Прим.: mr\_SM\_SM – array[1..SM,1..SM] of real,

  by – byte,

  Int – integer.}

begin

  for i:=1 to k do

    for j:=1 to m do

      MC[i,j]:=0;

  for i:=1 to k do

    for j:=1 to m do

```

for n:=1 to l do
    MC[i,j]:=MC[i,j]+MA[i,n]*MB[n,j];
end;{ — MMGG — }

```

**Подпрограмма NA8FIR**  
(обращается к PRESS, SOME)

1. Эта подпрограмма из множества точек, которые не все принадлежат объемной фигуре, выбирает сначала принадлежащие поверхности фигуры, а из последних – ближайшую точку по отношению к некоторой заданной (критериальной). В качестве объемных фигур подразумевается эллипсоид или цилиндр, ограниченные двумя плоскостями.

2. Обращение к подпрограмме:

NA8FIR(T, WHO, KAK, TF, N, JOP);

3. Описание параметров:

T – array [1..SM\_3, 1..3] of real, исходная матрица точек пересечения луча со всеми поверхностями, ограничивающими объемную фигуру.

$$T = \begin{bmatrix} X & Y & Z \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix}.$$

Число столбцов в матрице T равно 3. В первую и вторую строку записывают координаты пересечения с первой и второй плоскостью. После выполнения процедуры матрица преобразуется;

WHO – array [1..4] of real, матрица имеет столько же строк, что и T: например, если WHO[1, 1] = 1, то в первой строке матрицы T записано решение, принадлежащее некоторой поверхности тела, если WHO[1, 1] = 0, то решения нет и строка из T имеет машинное заполнение;

KAK – array [1..4] of real, матрица имеет столько же строк, что и T. KAK определяет принадлежность точки – строки из T поверхности объемного тела. Например, если KAK[1, 1] = 1, то первая точка из T лежит на поверхности тела, если KAK[1, 1] = 0, то точка лежит вне тела, но на продолжении одной из его боковых поверхностей;

TF – array [1..3] of real, точка фокуса. TF = (Xf, Yf, Zf);

N – integer, на входе – размерность первых матриц T, WHO, KAK по числу строк, на выходе – число рабочих строк тех же матриц. Рабочие строки – это те, которые соответствуют точкам, лежащим на поверхности тела;

JOP – integer, выходной показатель, JOP = 0, если решений нет, JOP = 1 – решение находится в первой строке преобразованной матрицы T, JOP = N – решение находится в N –й строке преобразованной матрицы T.

4. Тело подпрограммы:

```

procedure NA8FIR( var T: mr_SM_3; var WHO,KAK : mr_4_1;
                  TF : mr_3; var N,Jop : Int);
label
  1;
var
  ii,jj,i,k,j : Int;
  BUF : mr_SM_3;
{Прим.: mr_3 – array[1..3] of real,
  mr_SM_3 –array[1..4,1..3] of real,

```

```

mr_4_1 - array[1..4,1]of real,
Int - integer.}

begin
  Jop:=0;
  if SUMMR_4_1(KAK)=0 then Exit;
  k:=1;
1: for i:=k to N do
  if KAK[i,1]=0 then
    begin
      PRESSMR_4_1(KAK,i,N);N:=N+1;
      PRESSMR_4_1(WHO,i,N);N:=N+1;
      for ii:=1 to 4 do
        for jj:=1 to 3 do
          BUF[ii,jj]:=T[ii,jj];
      PRESSMR_SM_3(BUF,i,N);
      for ii:=1 to 4 do
        for jj:=1 to 3 do
          T[ii,jj]:=BUF[ii,jj];
      goto 1;
    end
    else
    begin
      k:=k+1;
      goto 1;
    end;
  for i:=1 to 4 do
    for j:=1 to 3 do
      BUF[i,j]:=T[i,j];
  SOME(BUF,TF,N,Jop);
  for i:=1 to 4 do
    for j:=1 to 3 do
      T[i,j]:=BUF[i,j];
end;{ — NA8FIR — }
function SUMMR_4_1( Mat : mr_4_1):Re;
var
  i,j : Int;
  S   : Re;
begin
  S:=0;
  for i:=1 to 4 do
    S:=S+Mat[i,1];
  SUMMR_4_1:=S;
end;{ — SUMMR_4_1 — }
procedure PRESSMR_SM_3(var A : mr_SM_3; L : Int; var N : Int);
var
  i,j : Int;
begin
  N:=N-1;
  for i:=1 to N do
    if i>=L then

```

```

A[i]:=A[i+1];
end;{ — PRESSMR_SM_3 — }
procedure PRESSMR_4_1(var A : mr_4_1; L : Int; var N : Int);
var
  i,j : Int;
begin
  N:=N-1;
  for i:=1 to N do
    if i>=L then
      A[i,1]:=A[i+1,1];
end;{ — PRESSVEC4_1 — }

```

**Подпрограмма OPTIMA**  
(обращается к ОВМЕН, PRESS)

1. Обеспечивает построение выпуклой двухмерной оболочки—многоугольника над множеством точек на плоскости.
2. Обращение к подпрограмме:

OPTIMA(A,n,Indic);

3. Описания параметров, пояснения:

A – array[1..NT,1..2] of real. Массив точек, над которыми строится выпуклая оболочка.

$$A = \begin{bmatrix} X_1 & Y_1 \\ \vdots & \vdots \\ \vdots & \vdots \\ X_{NT} & Y_{NT} \end{bmatrix}.$$

Значение SM определяет число строк в матрице A. Порядок следования строк – произвольный. На выходе A преформируется так, что первые N строк описывают вершины выпуклой оболочки по или однозначно против часовой стрелки; N – integer. На входе это число первых строк – точек матрицы A, над которыми строится оболочка,  $N \leq SM$ . На выходе первые N строк в матрице A по/против час. стрелки описывают вершины выпуклой оболочки;

Indic – integer. Индикатор: I=1, если построение оболочки прошло успешно; I=0, если над фигурой невозможно построить выпуклую оболочку (точки принадлежат прямой).

4. Ограничения и области допустимых значений:

Значения A и n после обращения к подпрограмме изменяются.

5. Тело подпрограммы:

```

procedure OPTIMA( var A : mr_SM_2; var n,Indik : Int);
label
  1,4,5;
var
  i,j,r,s,k,m,ans,stand : Int;
{Прим.: mr_SM_2 – array[1..SM,1..2] of real,
 int – integer.}
begin
  indik:=1;

```

```

s:=0; j:=1;
1: k:=0;
  for i:=2 to n do
    begin
      m:=0;
      for r:=2 to n do
        begin
          ans:=SIGN((A[r,1]-A[j,1])*(A[i,2]-A[j,2])-  

                     (A[i,1]-A[j,1])*(A[r,2]-A[j,2]));
          if ans=0 then goto 4;
          m:=m+1;
          if m=1 then stand:=ans;
          if ans<>stand then goto 5;
        end;
        if m=0 then goto 5;
        s:=s+1; k:=i;
        OBMEN(A,n);
        if s=n then Exit;
        goto 1;
      end;
      if k=0 then
        begin
          PRESSMR_SM_2(A,j,n);
          if n=2 then
            begin
              Indik:=0;
              Exit;
            end;
            goto 1;
          end;
        end;
      end;{ — OPTIMA — }
procedure PRESSMR_SM_2(var A : mr_SM_2; L : Int; var N : Int);
var
  i,j : Int;
begin
  N:=N-1;
  for i:=1 to N do
    if i>=L then
      A[i]:=A[i+1];
end;{ — PRESSMR_SM_2 — }
procedure OBMEN( var A : mr_SM_2; var nr : Int);
var
  B : array[1..2] of Re;
  i,j : Int;
begin
  for i:=1 to 2 do B[i]:=A[1,i];
  for i:=1 to nr-1 do
    for j:=1 to 2 do
      A[i,j]:=A[i+1,j];
  for i:=1 to 2 do

```

```

A[nr,i]:=b[i];
end;{ — OBMEN — }

```

### Подпрограмма PICA

(обращается к подпрограмме KVADR)

1. Определяет координаты Z1 и Z2 двух точек пересечения прямой с конусом: F(X, Y, Z) = 0, где  $F(X, Y, Z) = (X - X_0)^2/A + (Y - Y_0)^2/B + (Z - Z_0)^2/C$ , одно из трех чисел A, B, C равно -1, а остальные два – положительны.

2. Обращение к подпрограмме:

PICA(T1, T2, KON, Z1, Z2, INDI);

3. Описание параметров:

T1 – array [1..3] of real, T1 = [X11, Y11, Z11] – матрица координат первой точки (центра рецептора), через которую проходит прямая (световой луч);

T2 – array [1..3] of real, T2 = [Xf, Yf, Zf] – матрица координат второй точки (точки фокуса), через которую проходит прямая (световой луч);

KON – array [1..2, 1..3] of real, см. выше уравнение конуса. В первой строке X0, Y0, Z0, во второй – A, B, C соответственно;

Z1 – real, координата первого решения;

Z2 – real, координата второго решения;

INDI – integer, индикатор, если INDI = 1, если есть решения, INDI = 0, если решений нет или их бесконечно много.

4. Тело подпрограммы:

```

procedure PICA( T1,T2 : mr_3; KON : mr_2_3;
    var Z1,Z2 : Re; var Indi : Int);

```

var

XYZ : mr\_3;

NORM : array[1..2] of Re;

A : Int;

begin

NORM[1]:=(T1[3]-T2[3]);

NORM[2]:=(T1[2]-T2[2])/NORM[1];

NORM[1]:=(T1[1]-T2[1])/NORM[1];

XYZ[1]:=KON[2,2]\*KON[2,3]\*Sqr(NORM[1])+

KON[2,1]\*KON[2,3]\*Sqr(NORM[2])+

KON[2,1]\*KON[2,2];

XYZ[2]:=2\*(KON[2,2]\*KON[2,3]\*NORM[1]\*((T1[1]-KON[1,1])-

T1[3]\*NORM[1])+KON[2,1]\*KON[2,3]\*NORM[2]\*((T1[2]-

KON[1,2])-T1[3]\*NORM[2])-KON[2,1]\*KON[2,2]\*KON[1,3]);

XYZ[3]:=KON[2,1]\*KON[2,3]\*Sqr(T1[3]\*NORM[2]-(T1[2]-KON[1,2]))+

KON[2,2]\*KON[2,3]\*Sqr(T1[3]\*NORM[1]-(T1[2]-KON[1,1]))+

KON[2,1]\*KON[2,2]\*Sqr(KON[1,3]);

KVADR(XYZ[1],XYZ[2],XYZ[3],A,Z1,Z2);

if A=2 then Indi:=0

else Indi:=1;

end;{ — PICA — }

### Процедура PLOS

(Обращается к процедуре DIRECT)

1. Определяет координаты пересечения прямой и плоскости. Прямая задается двумя

точками с координатами соответственно  $X_1, Y_1, Z_1$  и  $X_2, Y_2, Z_2$ . Плоскость задается параметрами  $A, B, C, D$  в уравнении  $AX + BY + CZ + D = 0$ . Область допустимых значений:  $Z_1 \neq Z_2$ .

2. Обращение к подпрограмме:

PLOS(T1, T2, P, XYZ, FLAG);

3. Описание параметров:

$T1$  - array [1..3] of real,  $T1$  – матрица координат первой точки ( $X_1, Y_1, Z_1$ ), через которую проходит прямая;

$T2$  – array [1..3] of real,  $T2$  – матрица координат второй точки ( $X_2, Y_2, Z_2$ ), через которую проходит прямая;

$P$  – array [1..4] of real,  $P$  – матрица-столбец, элементы которой  $A, B, C, D$  (сверху вниз);

$XYZ$  – array [1..3] of real,  $XYZ$  – матрица координат точки пересечения прямой и плоскости, где координаты соответственно  $X, Y, Z$ ;

$FLAG$  – integer, индикатор,  $FLAG = 1$ , если пересечение единствено,  $FLAG = 0$ , если решений нет или их бесконечное множество.

4. Тело подпрограммы:

```
procedure PLOS( T1,T2 : mr_3; P : mr_4_1; var XYZ : mr_3;
               var Flag : Int);
var
  Znam : Re;
{Прим.: mr_3 – array[1..3] of real,
  mr_4_1 – array[1..4,1]of real,
  Int – integer.}
begin
  Znam:=P[1,1]*(T1[1]-T2[1])+P[2,1]*(T1[2]-T2[2])+P[3,1]*
    (T1[3]-T2[3]);
  if Znam=0 then
    begin
      Flag:=0;           Exit;
    end;
  Flag:=1;
  XYZ[3]:=T1[3]+(T1[3]-T2[3])*
    (-P[1,1]*T1[1]-P[2,1]*T1[2]-P[3,1]*T1[3]-P[4,1])/Znam;
  DIRECT(T1,T2,XYZ[3],XYZ);
end;{ – PLOS – }
```

### Подпрограмма PL3T

1. Определяет коэффициенты уравнения плоскости  $AX + BY + CZ + D = 0$  по координатам трех точек.

2. Обращение к подпрограмме:

PL3T(T0, T1, T2, P);

3. Описание параметров:

$T0$  – array [1..3] of real, в этой матрице записаны координаты  $X, Y, Z$  первой точки;

$T1$  – array [1..3] of real, соответственно для второй точки;

$T2$  – array [1..3] of real, соответственно для третьей точки;

$P$  – array [1..4] of real, матрица-столбец коэффициентов  $A, B, C, D$  уравнения плоскости.

4. Тело подпрограммы:

```

procedure PL3T( A,B,C : mr_3; var P : mr_4_1);
var
  i : Int;
{Прим.: mr_3 - array[1..3] of real,
  mr_4_1 - array[1..4,1]of real,
  Int - integer.}
begin
  for i:=1 to 3 do
    begin
      B[i]:=B[i]-A[i];
      C[i]:=C[i]-A[i];
    end;
  P[1,1]:=B[2]*C[3]-C[2]*B[3];
  P[2,1]:=-B[1]*C[3]+C[1]*B[3];
  P[3,1]:=B[1]*C[2]-C[1]*B[3];
  P[4,1]:=-A[1]*P[1,1]+A[2]*P[2,1]-A[3]*P[3,1];
end;{ - PL3T - }

```

**Процедура PLOS**  
(Обращается к процедуре DIRECT)

1. Определяет координаты пересечения прямой и плоскости. Прямая задается двумя точками с координатами соответственно  $X_1, Y_1, Z_1$  и  $X_2, Y_2, Z_2$ . Плоскость задается параметрами  $A, B, C, D$  в уравнении  $AX + BY + CZ + D = 0$ . Область допустимых значений:  $Z_1 \neq Z_2$ .

2. Обращение к подпрограмме:

PLOS(T1, T2, P, XYZ, FLAG);

3. Описание параметров:

$T_1$  – array [1..3] of real,  $T_1$  – матрица координат первой точки ( $X_1, Y_1, Z_1$ ), через которую проходит прямая;

$T_2$  – array [1..3] of real,  $T_2$  – матрица координат второй точки ( $X_2, Y_2, Z_2$ ), через которую проходит прямая;

$P$  – array [1..4] of real,  $P$  – матрица-столбец, элементы которой  $A, B, C, D$  (сверху вниз);

$XYZ$  – array [1..3] of real,  $XYZ$  – матрица координат точки пересечения прямой и плоскости, где координаты соответственно  $X, Y, Z$ ;

$FLAG$  – integer, индикатор,  $FLAG = 1$ , если пересечение единственное,  $FLAG = 0$ , если решений нет или их бесконечное множество.

4. Тело подпрограммы:

```

procedure PLOS( T1,T2 : mr_3; P : mr_4_1; var XYZ : mr_3;
                var Flag : Int);
var
  Znam : Re;
{Прим.: mr_3 - array[1..3] of real,
  mr_4_1 - array[1..4,1]of real,
  Int - integer.}
begin
  Znam:=P[1,1]*(T1[1]-T2[1])+P[2,1]*(T1[2]-T2[2])+P[3,1]*
    (T1[3]-T2[3]);

```

```

if Znam=0 then
begin
  Flag:=0;
  Exit;
end;
Flag:=1;
XYZ[3]:=T1[3]+(T1[3]-T2[3])*
(-P[1,1]*T1[1]-P[2,1]*T1[2]-P[3,1]*T1[3]-P[4,1])/Znam;
DIRECT(T1,T2,XYZ[3],XYZ);
end;{ — PLOS — }

```

## Подпрограммы PRESSMR\_SM\_3, PRESSMR\_4\_1, PRESSMR\_SM\_2

Это служебные подпрограммы, используемые, в частности, подпрограммой ОРТИМА. См. описание подпрограммы ОРТИМА.

### Подпрограмма SOME

1. Среди множества точек, лежащих на одной прямой, выбирает одну, которая является ближайшей к некоторой критериальной точке, также лежащей на той же прямой. Исходное множество точек записано в матрицу размером  $N \times 3$ , где  $N$  – целое число. Каждая строка этой матрицы – координаты  $X$ ,  $Y$ ,  $Z$  одной из точек. Из всех  $N$  точек может анализироваться лишь  $L$  первых точек ( $L \leq N$ ).

2. Обращение к подпрограмме:

SOME(AB, TF, MAB, JOP);

3. Описание параметров:

$AB$  – array [1.. $N$ , 1..3] of real, размерность этой матрицы определяется описанием вызывающего модуля.  $AB$  – это матрица  $N \times 3$  координат  $N$  точек;

$TF$  – array [1..3] of real,  $TF$  – матрица-строка координат  $X$ ,  $Y$ ,  $Z$  критериальной точки;

$MAB$  – integer, число рабочих строк в матрице  $AB$ , рабочие строки – это те  $L$  первых строк, среди которых имеется решение;

$JOP$  – integer, номер строки в матрице  $AB$ , который соответствует искомой точке.

4. Тело подпрограммы:

procedure SOME(AB : mr\_SM\_3; TF : mr\_3; mab : Int; var Jop : Int);

var

plo,pla : Re;  
 i : Int;

begin

Jop:=1;

pla:=Abs(TF[3]-AB[1,3]);

for i:=2 to mab do

begin

plo:=Abs(TF[3]-AB[i,3]);

if plo<pla then

begin

pla:=plo;

Jop:=i;

end;

```

end;
end;{ — SOME — }

```

### Подпрограмма YESNO

1. Определяет факт принадлежности некоторой точки ( X,Y,Z ) поверхности или внутренней области выпуклого шестиугранника, нормали к плоскостям-граням которого направлены внутрь тела.

2. Обращение к подпрограмме:

YESNO ( A, PL, KRIT );

3. Описание параметров, пояснения:

A — array [1..3] of real. A — матрица координат X, Y, Z точки, проверяемой на принадлежность к шестиуграннику;

PL — array [1..4,1..6] of real. PL — матрица, описывающая шестиугранник:

$$PL = \begin{bmatrix} A1 & A2 \dots A6 \\ B1 & B2 \dots B6 \\ C1 & C2 \dots C6 \\ D1 & D2 \dots D6 \end{bmatrix}$$

Каждый столбец описывает соответствующие коэффициенты i - й (i=1(1)6) плоскости в ее уравнениях  $A_iX + B_iY + C_iZ + D = 0$ .

KRIT — integer, индикатор, KRIT = 1, если точка принадлежит шестиуграннику, KRIT = = 0 — если не принадлежит.

4. Ограничения и области допустимых значений: точка считается лежащей вне шестиугранника, если значение функции ближайшей плоскости в этой точке менее 0,003. Значение подобрано экспериментально, оно отлично от нуля для компенсации неточности представления чисел в ЭВМ.

5. Тело подпрограммы:

```
procedure YESNO( A : mr_3; PL : mr_4_6; var Krit : Int);
```

var

B : Re;

i : Int;

{Прим.: mr\_3 — array[1..3] of real,  
mr\_4\_6 — array[1..4,1..6] of real,  
Re — real,  
Int — integer.}

begin

Krit:=1;

for i:=1 to 6 do

begin

B:=PL[1,i]\*A[1]+PL[2,i]\*A[2]+PL[3,i]\*A[3]+PL[4,i];

if B < -0.003 then

begin

Krit:=0;

Exit;

end;

end;

end;{ — YESNO — }

## Подпрограмма YNZONA

1. Определяет факт принадлежности заданной точки внутренней области выпуклой двухмерной зоны, заданной координатами крайних точек.

2. Обращение к подпрограмме:

**YNZONA(A,NOR\_,N,X,Y,YN);**

3. Описание параметров, пояснения:

**A** – array[1..2,1..NN] of real. Матрица крайних точек выпуклой двухмерной зоны. В каждой строке записаны два числа: соответственно координаты X, Y точки. В матрице до строки N включительно записаны координаты крайних точек, начиная со строки N+1 – машинное заполнение. N<=NN;

**NOR\_** – array[1..2,1..NN] of real. Матрица проекций направляющих векторов прямых, соединяющих 1 и 2, 2 и 3,...,N и N+1 точки. Первый столбец матрицы – проекция на ось OX, второй – на OY. N<=NN;

**N** – integer, число первых обрабатываемых строк матриц A,NOR\_;

**X** – real, координата X анализируемой точки;

**Y** – real, координата Y анализируемой точки;

**YN** – integer, индикатор. YN = 1, если анализируемая точка внутри зоны, YN =0 – если вне зоны.

4. Ограничения и области допустимых значений:

Точка считается лежащей вне зоны, если значение функции  $(X - X_i)(Y_i - Y) - (Y - Y_i)(X_i - X)$  ближайшей прямой, проходящей через i - ю и i+1 - ю вершины, меньше 0,003. Значение подобрано экспериментально, оно отлично от нуля для компенсации неточности представления чисел в ЭВМ.

5. Тело подпрограммы:

```

procedure YNZONA( A,NOR_ : mr_SM_2; N : Int; X,Y : Re;
                   var YN : Int);
var
  i : Int;
{Прим.: mr_SM_3 -array[1..2,1..3] of real,
  Int – integer, Re – real.}
begin
  YN:=1;
  for i:=1 to N do
    if ((X-A[i,1])*NOR_[i,2]-(Y-A[i,2])*NOR_[i,1])<-0.003 then
      begin
        YN:=0;
        Exit;
      end;
  end;{ – YNZONA – }

```

## СПИСОК ЛИТЕРАТУРЫ

- 1.Альберт Дж. Теория сплайнов и ее приложения: Пер. с англ.–М.: Наука, 1971.–14 с.
- 2.Анисимов Б.В., Курганов В.Д., Злобин В.К. Распознавание и цифровая обработка изображений.– М.: Высшая школа, 1983.–295 с.
- 3.Арнхейм Р. Искусство и зрительное восприятие: Пер. с англ.–М.: Прогресс, 1971.–402 с.
- 4.Бабенко В.С. Оптика телевизионных устройств.–М.: Радио и связь, 1982.–356 с.
- 5.Батраков А.С., Летуновский А.В., Минеев В.И. Элементы оптико-электронных приборов.–Л.: МО, 1984.–192 с.
- 6.Батраков А.С., Летуновский А.В. Телевизионные системы.–Л.: МО, 1984.–192 с.
- 7.Бегунов Б.Н., Заказнов Н.П. Теория оптических систем.– М.: Машиностроение, 1973.–431 с.
- 8.Борн М., Вольф Э. Основы оптики.–М.: Наука, 1973.–855 с.
- 9.Бобков В.А., Кислюк О.С., Хамидулин А.В. Формирование растровых графических изображений пространственных объектов, построенных методом конструктивной геометрии // Программирование.–1989.–№ 3.–С. 88-92.
- 10.Башков Е.А., Казак А.Б. Генераторы изображения для авиатренажеров // Зарубежная радиоэлектроника.–1984.–№ 8.–С. 60-68.
- 11.Выгодский В. М. Справочник по высшей математике. –М.: Наука, 1980.–680 с.
- 12.Гилой В. Интерактивная машинная графика: Пер. с англ. – М.: Мир, 1981.–380 с.
- 13.Гонин Г.Б. Космические съемки Земли.–Л.: Недра, 1989.–380 с.
- 14.Гранант Д.Д. Роль моделей зрения человека в обработке изображений // ТИИЭР.–1981.–№ 5.–С. 65-77.
- 15.Гуревич М.М. Фотометрия.–Л.: Энергоатомиздат, 1983.–272 с.
- 16.Демидович Б.П., Марон И.А. Основы вычислительной математики.– М.: ГИФМЛ, 1963.–659 с.
- 17.Демьянов В.Ф., Малоземов В.Н. Введение в минимакс.–М.: Наука, 1972.–368 с.
- 18.Иванов В.П. Компьютерный синтез изображений в задаче видеораспознавания образцов // Тез. докл. Всесоюз. конф. “Оптическое изображение и регистрирующие среды”.–Л., 1990.–Ч.2.–С. 127,128.
- 19.Иванов В.П., Батраков А.С. Синтез изображений объектов сложной формы методом трассирования лучей // Программирование.–1989.–№ 2.–С. 70-75.
- 20.Иванов В.П., Сечко А.М., Желаннов С.А. Полутоновое иллюстрирование трехмерных функций с помощью ЭВМ // Программирование.–1987.–№ 1.–С.51-55.
- 21.Иванов В.П., Желаннов С.А., Сечко А.М. Совершенствование модели формирования тени для машинной графики // Электронное моделирование.–1988.–№ 3.–С. 81-85.
- 22.Иванов В.П., Якунченков С.И., Михоларкин Ю.А. Приведение опорных изображений к виду, удобному для распознавания в задаче автоматического анализа дорожной обстановки // Тез. докл. IV Всесоюз. конф. “Математические методы распознавания образов” (Рига, октябрь 1989 г.).–Рига, 1989.–Ч.4.–С. 67,68.

- 23.Корн Г., Корн Т. Справочник по математике: Пер. с англ.—М.: Наука, 1984.—831с.
- 24.Ковалев А.М., Талныкин Э.А. Машинный синтез визуальной обстановки // Автометрия.—1984.—№ 4.—С. 78-83.
- 25.Коган В.Ф. Основы теории поверхностей в тензорном изложении.—М.; Л.: Физматгиз, 1947,1948.—267 с.
- 26.Кондратьев В.С. Геометрическое обеспечение решения задачи по построению чертежей архитектурных форм // Инженерно-технические вопросы архитектурного проектирования.—1985.—С. 239-243.
- 27.Клименко С.В., Kochin B.H. Об одном способе изображения поверхностей в машинной графике // Программирование.—1981.—№ 2.—С. 68-71.
- 28.Ландсберг Г.С. Оптика.—М.: Наука, 1976.—927 с.
- 29.Ллойд Дж. Системы тепловидения: Пер. с англ.— М.: Мир, 1972.—414 с.
- 30.Лобанов А. Н. Аэрофотоподография.—М.: Недра, 1978.—576 с.
- 31.Машинная графика и ее приложения: Сб. науч. тр. АН СССР. / Под ред. А.М. Мацокина.—Новосибирск: АН СССР и СО АН, 1983.—137 с.
- 32.Тезисы докладов V Всесоюзной конференции по проблемам машинной графики "Машинная графика'89".—Новосибирск: АН СССР и СО АН, 1989. —176 с.
- 33.Тезисы докладов IV Всесоюзной конференции по проблемам машинной графики "Машинная графика'89".—Серпухов: НС по КП "Кибернетика", 1987.—195с.
- 34.Машинная графика и ее приложения: Сб. статей /Под ред. В.Л. Перчука.—Владивосток: Дальневосточный научный центр АН СССР, 1975. —Вып. 24. —137 с.
- 35.Машинная графика и вычислительная геометрия в задачах машиностроения / Под ред. О.М. Белоцерковского.—М.: НС по КП "Кибернетика" при АН СССР 1989.—178 с.
- 36.Мельканович А.Ф. Фотографические средства и их эксплуатация.—Л.: МО, 1984.—514 с.
- 37.Международный светотехнический словарь/ Под ред. Д.Н. Лазарева.—М.: Русский язык, 1979.—132 с.
- 38.Мишев Д. Дистанционные исследования Земли из космоса.—М.: Мир, 1985.—232 с.
- 39.Моденов П.С. Аналитическая геометрия.—М.: МГУ, 1969.—698 с.
- 40.Норден А.П. Теория поверхностей.—М.: Госмехиздат, 1956.—259 с.
- 41.Ньюмен У., Спрулл Р. Основы интерактивной машинной графики: Пер. с англ.—М.: Мир, 1976.—573 с.
- 42.Очин Е.Ф. Вычислительные системы обработки изображений.—М.: Энергоатомиздат, 1989.—133 с.
- 43.Пайтген Х.О., Рихтер П.Х. Красота фракталов: образы комплексных динамических сцен // В мире науки.—1990.—№3.—С. 133,134.
- 44.Погорелов А.В. Дифференциальная геометрия.—М.: Высшая школа, 1969.—365 с.
- 45.Прэтт У. Цифровая обработка изображений: Пер. с англ.—М.: Мир, 1982.—Кн.1.—310 с.; Кн.2. — 473 с.
46. Прэтт У., Фожра О., Гагалович А. Применение моделей стохастических текстур для обработки изображений // ТИИЭР.—1981.—№5.—С. 54–64.

- 47.Проектирование оптических систем: Пер. с англ. / Под ред. Р. Шеннона. –М.: Мир, 1983.–430 с.
- 48.Роджерс Д., Адамс Дж. Математические основы машинной графики: Пер. с англ.–М.: Машиностроение, 1980.–240 с.
- 49.Роджерс Д. Алгоритмические основы машинной графики: Пер. с англ.– М.: Мир, 1989.–504 с.
- 50.Семерджян М.А., Налбандян Ж.С., Гаспарян Л.Х. Принципы работы и системы ПО МП ЕС 2700.–М.: Наука, 1988.–224 с.
- 51.Смирнов С.А. Проективная геометрия.– М.: Недра, 1976.–176 с.
- 52.Статистические методы обработки результатов наблюдений / Р.М. Юсупов, А.Я. Иоффе; Под ред. Р.М. Юсупова.–Л.: МО, 1984.–563 с.
- 53.Талныкин Э.А. Внутренний язык для описания визуальных моделей // Автометрия.–1985.–№ 4.–С. 44-49.
- 54.Фокс А., Пратт М. Вычислительная геометрия: Пер. с англ.–М.: Мир, 1982.–304 с.
- 55.Фукунага К. Введение в статистическую теорию распознавания образов: Пер. с англ.–М.: Наука.–1979.–204 с.
- 56.Фрир Дж. Построение вычислительных систем на базе перспективных микропроцессоров: Пер. с англ.–М.: Мир, 1990.–414 с.
- 57.Хермен Г. Восстановление изображений по проекциям: Пер. с англ.–М.: Мир, 1983.–349 с.
- 58.Хорн Б.К. Зрение роботов: Пер. с англ.–М.: Мир, 1989.–488 с.
- 59.Уокер С., Гурд Д.П., Дроник Е.А. Интерактивная машинная графика: Пер. с англ.–М.: Машиностроение, 1980.–171 с.
- 60.Фоли Дж., А. вэн Дэм. Основы интерактивной машинной графики: Пер. с англ. – М.: Мир, 1985.– Т.1.–367 с.; Т.2.–368 с.
- 61.Якушенков Ю.Г. Теория и расчет оптико-электронных приборов. – М.: Сов. радио, 1980.–390 с.
- 62.Advances in computer graphics / Ed. by G.Enderle, M.Grave, F.Lillehagen.–Berlin: Springer-Verlag.–512 p.
- 63.Advanced computer graphics: Proc. of Computer Graphics Tokyo'86 / Ed. by T.L. Kunii.–Tokyo: Springer, 1986.–504 p.
- 64.Aki S.G., Toussaint G.T. A fast convex hull algorithm // Information Processing Letters 7. –1978.–P. 1234-1243.
- 65.Appel A. Some techniques for shading machine rendering of solids // SJCC.–Washington, 1968.– D.,C.– P. 37-45.
- 66.Artwick B.A. Applied concepts in microcomputer Graphics // Prentice Hall, Inc. Eglewood Cliffs.– New Jersey, 1984.–374 p.
- 67.Atherton P., Weiler K., Greenberg D. Polygon shadow generation // Computer Graphics.–1977.–Vol. 11, №. 2.–P. 242-247.
- 68.Bishop A. Computer simulation of the real world for the mechanical and industrial designer // Computer Graphics'83: Proc. of the Intern. Conf. (London, 83).–Pinner: Online publ., 1983–XIV.–776 p.
- 69.Blinn J.F. A scan-line algorithm for computer display of parametrically defined surfaces // Computer Graphics.–1978.–Vol.12.–P. 345.
- 70.Blinn J.F. Simulation of wrinkled surfaces // Computer Graphics.–1978.–Vol.12, №3, 1978.–P. 286-292.

- 71.Blinn J., Newell M. Texture and reflection in computer generated images // Communication of ACM.-1976.-Vol.19. № 10.-P. 542-547.
- 72.Boisvert D., Mgnehat-Thalmann S., Thalmann D. An integrated control view of synthetic actors: theoretical foundations of computer graphics and CAD / Ed. by R.A.Earshaw.-Berlin: Springer-Verlag, 1988.-P. 277-283.
- 73.Bouknight W.K., Kelley K.C. An algorithm for producing halftone computer graphics presentations with shadows and movable light sources // Spring Joint Conf. / AFIPS Press .- 1970.-P. 1-10.
- 74.Bouville C., Brusq R. Generating high quality pictures by ray-tracing // Computer Graphics Forum.-1985.-Vol.4, № 2.-P. 567-579.
- 75.Braid I.C. The synthesis of solids bounded by many faces // Communication of the ACM.-1975.-Vol.18, № 4.-P. 209-215.
- 76.Braid I.C. Geometric modelling // Advances in Computer Graphics. - Berlin: Springer-Verlag, 1981.-P. 425-362.
- 77.Bret M. Images de synthese.-Paris: Bo-pre, 1989.-317 p.
- 78.Burtnik N., Wein M. Computer generated key-frame animation // Journal of SMPTE.-1980.-P. 149-153.
- 79.Catmull E. Computer display of curved surfaces // Proc. IEEE. Conf. on Computer Graphics Pattern Recognition and Data Structure ( N.Y., May 1975).-New York, 1975.-P. 456-459.
- 80.Colonna J. From display of computer results to artistic creation // Computer Graphics'83: Proc. of the Intern. Conf. (London, 83). -Pinner: Online publ., 1983-XIV.-776 p.
- 81.Computational geometry / Ed. by G. Toussaint.-Amsterdam: North Holland, 1985.-459 p.
- 82.Computer Animation CG'87: Proc. of the Conf. Held at Computer Graphics 87 (London, Oct. 87.)- London: Online publ., -1987.-VII.-221 p.
- 83.Computer Graphics'81 : Proc. of the Intern. Conf. - Northwood(UK): Online publ., -1981.-545 p.
- 84.Computer Graphics'82 : Proc. of the Intern. Conf. - Northwood(UK): Online publ.,-1982.-XVIII.-392 p.
- 85.Computer Graphics'83: Proc. of the Intern. Conf. (London, 83). -Pinner: Online publ.,-1983.-XIV.-776 p.
- 86.Computer graphics society. Techniques for computer graphics / Ed. by R.A. Earnshaw, D.F. Rogers.-New York: Springer, 1987.-VIII.-512 p.
- 87.Computer Graphics 1987: Proc. of CG Intern. Conf.'87 / Ed. by T.L. Kunii.-Tokyo : Springer, 1987.-IX.-490 p.
- 88.Computer graphics: Visual technology and art. // Proc. of Computer Graphics Tokyo'85 / Ed. by T.L. Kunii. -Tokyo: Springer, 1985.-VI.-382 p.
- 89.Computer graphics technology & systems. CG'87: Proc. of the Conf. ( London, Oct. 1987).-London: Online publ., 1987.-219 p.
- 90.Computer graphics. Theory and applications: Proc. of Intergraphics' 83 / Ed. by T.L. Kunii.-Tokyo: Springer-Verlag, 1983.-530 p.
- 91.Chasen S.H. Geometrical principles and procedures for computer graphics applications.-Englewood Cliffs(N.J.): Prentice Hall, 1975.-345 p.
- 92.Crow F.C. Algorithms for computer graphics // Computer Graphics.-1977.-Vol.11, № 2.-P. 242-247.

93. Daniel L., Ruan M. Computer-aided graphics and design. — New York, 1985. — 400 p.
94. Gardner G.Y. Visual simulation of clouds // Computer Graphics. — 1985. — Vol. 19, № 3. — P. 297-303.
95. Goldstein R.A. 3-D visual simulation // Simulation. — 1971. — № 1. — P. 25-31.
96. Hauber G., Guthseel P. Transputer. Systemarchitektur und Maschinensprache. — Munchen: Franzis, 1990. — 215 p.
97. Ikeuchi K., Kanade T. Modeling sensors and applying sensor model to automatic generation of object recognition program // Proc. Image Understanding Workshop ( San Mateo, California, Apr. 1988). — 1988. — Vol. 2. — P. 697-708.
98. Jankel A., Morton R. Creative Computer Graphics. — Cambridge: Univercity Press, 1984. — 143 p.
99. Jarke J.V. Bicubic patches for approximating non-rectangular control-point meshes / / Computer Aided Geometric Design. — 1986. — Vol. 3, № 1. — P. 456-459.
100. Kajiya J.T. Ray tracing parametric patches // Computer Graphics (SIGGRAPH'82). — 1982. — Vol. 16. — P. 245-254.
101. Kay D., Greenberg D. Transparency for computer synthesized images // Computer Graphics. — 1982. — Vol. 13, № 2. — P. 158-164.
102. Kilgour A.C. Techniques for modelling and displaying 3D scenes // Advances in Computer Graphics. — New York: Springer-Verlag, 1989. — 1123 p.
103. Levin J. A Parametric algorithm for drawing pictures of solid objects composed of quadric surfaces // Communication of the ACM. — 1976. — Vol. 19, № 10. — P. 555-563.
104. Levner G., Tassinari P., Marini D. A simple general methods for ray tracing bicubic surfaces // Theoretical Foundations of Computer Graphics and CAD . — New York: Springer-Verlag, 1988. — P. 805-820.
105. Lord E.A., Wilson C.B. The mathematical description of shape and form. — Chichester: Harwood, 1984. — 260 p.
106. Maver T.W. Flying dreams: visualization & animation in architecture // Computer Graphics Technology & Systems. CG'87: Proc. of the Conf. (London, Oct. 1987), — London: Online publ., — 1987. — 211 p.
107. Mahl R. Visible surface algorithm for quadric patches // IEEE Transactions on Computers. — Jan. 1972. — Vol. C-21. — P. 1-4.
108. Marshall R.G. Computer graphics and application. — New Jersey: Prentice Hall, 1987. — 454 p.
109. McGregor J., Watt A. The art of graphics for the IBM PC. — Wokingham: Addison-Wesley, 1986. — X. — 454 p.
110. Mechanical Engineering. — 1985. — Vol. 107, № 4. — 76 p.
111. Mittelman P. Computer graphics at MAGI // Computer Graphics'83: Proc. of the Intern. Conf. (London, 1983). — Pinner: Online publ., — 1983. — XIV. — 776 p.
112. Mgnenant-Thalmann N, Thalmann D. The direction of synthetic actors in the film "Redez-vous a montreal" // Computer Graphics and Applications. — 1987. — Vol. 7, № 12. — P. 9-19.
113. Multiresolution image processing and analysis / Ed. by A. Rozenfeld. — Berlin: Springer, 1984. — 385 p.
114. Nelson L. Max light diffusion through clouds and haze // Computer Visison, Graphics and Image Processing. — 1986. — Vol. 33, № 3. — P. 19-29.
115. New advances in computer graphics: Proc. of CG International'89 / Ed. by R.A. Earnshaw, B. Wyvill. — New York: Springer, 1989. — X. — 718 p.

116. Newell M.E., Newell R.G., Saucha T.L. A new approach to the shaded picture problem // Proc. of ACM Nat. Conf.-1972.-443 p.
117. Nishita T., Okamura I., Nakamae E. Shading models for point and linear sources // ACM Transactions on Graphics.-1987. -Vol. 4, № 2.-P. 344-349.
118. Page I. A dual-parallel processor for graphics & vision // Computer Graphics Technology & Systems. CG'87: Proc. of the Conf. (London, Oct. 1987).-London: Online publ.,- 1987.-P. 63-71.
119. Parallel computer vision / Ed. by Leonard Uhr. -Orlando, Florida: Academic Press. -1987.-303 p.
120. Parcer C.S. Understanding computers and data processing: Today and Tomorrow.- CBS College publishing, Holt, Rinehart and Winston. The Dryden Press, 1987.- 535 p.
121. Pigel L. Key development in computer-aided geometric design // Computer Aided Design. -1989.-Vol.21, № 5.-P. 63-71.
122. Pentland A.P. From pixels to predicates. - Norwood, New Jersey: Ablen Publishing Corp., 1986.-398 p.
123. Ponce J., Healey G. Using generic geometric and physical models for representing solids // Image Understanding Workshop: Proc. of a Workshop Held at Cambridge (Massachusetts, April 88). -Massachusetts, 1988.-Vol.1.-488 p.
124. Porter T. Spherical Shading // Computer Graphics (SIGGRAPH'78). -1978.-Vol.12.-P. 282-285.
125. Pratt M.J. Interactive geometric modeling for integrated CAD/CAM // Advances in Computer Graphics / Ed. by G. Enderle, M. Grave, F. Lillehagen.-Berlin: Springer-Verlag, 1986.-P. 363-380.
126. Preparata F.P., Hohg S.J. Convex hulls of finite sets in two and three dimensions // Communications of the ACM. -1977.-№ 20.-P. 345-349.
127. Preparata F.P., Shamos M.I. Computational geometry. An introduction.-New York: Springer, 1985.-390 p.
128. Product Design & Development. -1989.-Vol. 44, № 6.-123 p.
129. Requicha A.G. Representations of rigid solids-theory, method and systems // Computing Surveys. -1980.-Vol.12, № 4.-P.437-464.
130. Resch R.D. The topological design of sculptural and architectural systems.-New York: AFIPS Press, 1973.- P. 643-650.
131. Rogers D.F., Earnshaw R.A. Techniques for computer graphics.-Berlin: Springer-Verlag, 1987.-512 p.
132. Saouma U.E. Computer graphics aided design of reinforced concrete buildings // Concrete Intern. Des. Construction.-1985.- Vol.7, № 6.-P. 345-349.
133. Selim G. Aki. Optimal parallel algorithms for selection, sorting and computing convex hulls // Computational Geometry / Ed. by G. Toussaint. -Amsterdam: North Holland, 1985.-567p.
134. Simon J.S. From pixels to features.-Amsterdam: North Holland, 1989.-345 p.
135. Sutherland I.E., Sproull R.F., Schumacher R.A. A Characterization of ten hidden-surface algorithms // Computing Surveys.-1974. -Vol.6, № 1.-P. 1-55.
136. Tacala T. Industrial design based on geometric intentions // Theoretical Foundations of Computer Graphics and CAD / Ed. by R.A. Earnshaw.-Tokyo: Springer-Verlag, 1988.-P.953-963.
137. Theoretical foundations of computer graphics and CAD / Ed. by R.A. Earnshaw.-New York: Springer-Verlag, 1988.-1242 p.

- 
138. Toth D.L. On ray tracing parametric surfaces // SIGGRAPH'85.- 1985.-Vol.19, № 3.-P. 171-178.
139. Transputer applications / Ed. by G. Harp.-London: Pitman, 1989.-273 p.
140. Upson C. Large scale digital scenes simulation for feature films commercial productions // Comput. FX'84: Comput. Animations and Digit Eff. Proc. Conf. ( London, Oct. 1984).-London: Pinner, 1984.-P. 21-30.
141. Wallace V.L. The semantics of graphics input devices // Computer Graphics. -1976.-Vol.10, № 1.-P. 61-65.
142. Watkins G.S. A real-time visible surface algorithm // University of Utah Computer Science Dept. Tech. Report.- June 1970.-125 p.
143. Westphal H., Nagel H. Toward the derivation of 3-D descriptions from image sequences for nonconvex moving objects // Comp. Vision, Graphics and Image Processing.- -1986.-Vol.34, № 3.-P. 302-320.
144. Whitted T. A scan-line algorithm for computer display of curved surfaces // Computer Graphics.-1978.-Vol.12. -324 p.
145. Whitted T. An improver illumination model for shaded display // Communication of the ACM. -1980.-Vol.23, № 6.-P. 343-349.
146. Yamaguchi F. Curves and surfaces in computer aided geometric aided design.-Berlin: Springer, 1988.-378 p.
147. Yang C.G. Illumination models for generating images of curved surfaces // CAD.- -1987.-Vol.19, № 10.-544 p.

# О ГЛАВЛЕНИЕ

Введение.....	3
<b>Глава 1. Физические процессы формирования изображений и их моделирование.....</b>	<b>5</b>
1.1. Модели в машинной графике.....	5
1.2. Источники освещения.....	7
1.2.1.Источники искусственного освещения.....	8
1.2.2. Характеристики излучения источников простой геометрической формы.....	10
1.2.3. Освещенность, создаваемая точечными и протяженными источниками.....	12
1.3. Естественное освещение.....	14
1.4. Моделирование отраженного излучения объектов.....	21
1.5. Проекции в трехмерной машинной графике и их математические модели.....	25
1.5.1. Параллельная проекция.....	27
1.5.2. Центральная (перспективная) проекция.....	29
1.5.3. Описание преобразований с использованием однородных координат.....	30
1.5.4. Проекции при различных видах съемки.....	32
1.6. Текстура изображения.....	35
1.7. Моделирование энергетических преобразований в системах формирования изображений.....	37
1.8. Синтез изображений с учетом специфики их формирования в реальных условиях.....	40
<b>Глава 2. Описание геометрических форм.....</b>	<b>44</b>
2.1. Описание поверхностей.....	44
2.2. Общие характеристики основных типов поверхностей.....	48
2.3. Модели объектов и их классификация.....	55
<b>Глава 3. Синтез изображений методом обратного трассирования лучей.....</b>	<b>60</b>
3.1. Модель приемника света и его размещение в пространстве.....	62
3.2. Модель объекта.....	65
3.2.1. Примитивы – базовые строительные элементы объекта.....	65
3.2.2. Пространственные комбинации примитивов.....	69
3.2.3. Формализованная математическая модель объекта.....	70
3.2.4. Взаимное положение точки и объекта.....	73
3.3. Определение видимых и затененных точек.....	76
3.3.1. Пересечения светового луча с примитивом.....	76
3.3.2. Пересечения светового луча с комбинацией примитивов.....	79
3.3.3. Определение затененных точек.....	82
3.4. Изображение основных примитивов методом трассирования лучей.....	84
3.4.1. Плоские многоугольники.....	84
3.4.2. Выпуклые многогранники.....	94
3.4.3. Квадратичные поверхности.....	99
3.4.4. Бикубические поверхности.....	103
3.5. Изображение зеркальных и преломляющих поверхностей.....	108
3.6. Особенности определения освещенности.....	112

---

<b>Глава 4. Математические методы синтеза изображений путем проектирования точек и линий на экран.....</b>	115
4.1. Трассирование лучей от объекта к экрану.....	115
4.1.1. Основные принципы.....	115
4.1.2. Модель объекта.....	116
4.1.3. Определение видимых точек.....	117
4.1.4. Определение освещенности.....	118
4.1.5. Изображение поточечно описанных объектов.....	118
4.2. Метод сканирующей строки.....	121
4.2.1. Модель объекта из многоугольников.....	122
4.2.2. Модель приемника.....	123
4.2.3. Изображение многоугольников на экране.....	124
4.2.4. Изображение криволинейных поверхностей.....	127
4.3. Алгоритм Робертса.....	129
4.3.1. Модель объекта.....	129
4.3.2. Синтез изображения.....	130
<b>Глава 5. Организация синтеза изображений.....</b>	132
5.1. Управление процессом синтеза.....	132
5.2. Спектр синтезированного изображения.....	135
5.3. Сокращение времени синтеза изображений.....	137
5.3.1. Метод оболочек.....	138
5.3.2. Возможности использования транспьютеров в машинной графике.....	146
<b>Глава 6. Прикладное использование трехмерной машинной графики.....</b>	149
6.1. Архитектурное проектирование.....	149
6.2. Распознавание видеообразов.....	154
6.2.1. Распознавание образов по адаптивным эталонам.....	155
6.2.2. Распознавание образов в условиях неизвестной ориентации их корпусов.....	158
6.2.3. Распознавание образов по адаптированным эталонам в пирамидальной структуре.....	160
6.2.4. Распознавание образов по эталонам, адаптированным по условиям наблюдения и качеству.....	163
6.3. Автоматизированное проектирование в машиностроении.....	167
6.4. Восстановление формы скрытых объектов в медицинских исследованиях.....	169
6.5. Видеотренажеры и имитаторы сложных сцен.....	173
6.5.1. Тренажер для имитации полета над реальной местностью.....	174
6.5.2. Компьютерный имитатор изображений облаков.....	182
6.6. Реклама и мультипликация.....	186
<b>Приложение. Описания прикладных программ.....</b>	189
<b>Список литературы.....</b>	215



Профессиональная  
издательская система.

### Не нужна:

- для написания любовных писем;
- для составления бухгалтерских отчетов;
- для ведения дневников;
- тем, кто без ума от Windows

$$\cos \frac{1}{2}\phi = \sqrt{\frac{1}{2} \left( 1 + \sqrt{(k_x^2 + k_y^2)^2 + 4k_z^2} \right)}$$

A diagram illustrating a curved surface in three-dimensional space. The surface is represented by a series of small squares. A point on the surface is labeled with a coordinate system. Curvatures \$k\_x\$ and \$k\_y\$ are indicated along the principal axes of the surface's local coordinate system. A differential area element \$dA\$ is shown at a point on the surface. Below the surface, a coordinate system \$(x, y)\$ is shown, and the corresponding curvilinear coordinates \$(\varphi, \psi)\$ are also indicated.

### Абсолютно необходима:

- для создания оригинал-макетов научных и художественных изданий на любых носителях (бумага, пленка, Postscript- и TIFF-файлы) на уровне мировых стандартов качества;
- для набора и верстки научных трудов с большим количеством математических и химических формул;
- для ведения деловой переписки, создания бланков, форм;
- для подготовки рекламы с уникальными спецэффектами и тремя системами цветodelения.

Система "All-in-One" – мощный текстовый редактор (включая ТЕХ в режиме WYSIWYG), редактор векторной графики, работа с растровой графикой популярных форматов, полный набор типографских функций для страниц любого формата, а также печать на любых устройствах вывода (включая фотонабор) – раз в 25 быстрее, чем из-под Windows.

Продажа, сертифицированное фирмой-производителем обучение и поддержка осуществляется учебным издательским центром – тел. (095) 137 8709, E-mail [mc@netserv2.free.net](mailto:mc@netserv2.free.net)