ЛЕКЦИЯ 7

Методы обработки растровых изображений: преобразование цветного изображения в оттенки серого, линейное контрастирование, пороговая обработка, методы препарирования, методы выделения контуров (градиентный метод, эвристический алгоритм).

Утончение линий или задача скелетизации (алгоритм Зонга-Суэна, волновой алгоритм). Масочная фильтрация. Векторизация растровых изображений (алгоритм Хафа).

2.4. Методы обработки растровых изображений

2.4.1. Преобразование цветного изображения в «серое»

В предположении, что цвет каждого пиксела исходного цветного изображения задан в цветовой схеме RGB, то же самое изображение, исполненное в различных оттенках серого цвета, можно получить в результате преобразования:

$$R'=G'=B'=0{,}3\cdot R + 0{,}59\cdot G + 0{,}11\cdot B,$$

где (R, G, B) – исходные цветовые компоненты пиксела, (R′, G′, B′) – результирующие цветовые компоненты пиксела, соответствующие одной из градаций серого цвета.

Данный алгоритм является характерным представителем так называемых «поэлементных» алгоритмов, то есть тех, которые независимо применяются к каждому элементу (в данном случае – к пикселу) изображения. Многие из рассматриваемых ниже алгоритмов относятся к этому же типу.

2.4.2. Линейное контрастирование

Под «контрастностью» обычно понимается перепад яркостей, присутствующих на изображении. Предположим, что минимально возможное значение яркости пикселов равно L=0, а максимально возможное H=63. Изображение, не использующее всего возможного диапазона яркостей, и содержащее яркости пикселов только в пределах от min=20 до max=40, очевидно, является «малоконтрастным». Задача контрастирования заключается в «растягивании» реального динамического диапазона на всю шкалу. Контрастирование можно осуществить при помощи линейного поэлементного преобразования

$$g = af + b. \qquad (1)$$

Параметры этого преобразования a и b нетрудно определить, исходя из требуемого изменения динамического диапазона. Если в результате обработки нужно получить шкалу $[g_{min}, g_{max}]$, то, как следует из (1),

$g_{min} = af_{min} + b,$

$g_{max} = af_{max} + b.$

Отсюда

$$a = \frac{g_{max} - g_{min}}{f_{max} - f_{min}}, \quad b = \frac{g_{min} f_{max} - g_{max} f_{min}}{f_{max} - f_{min}}. \qquad (2)$$

2.4.3. Пороговая обработка

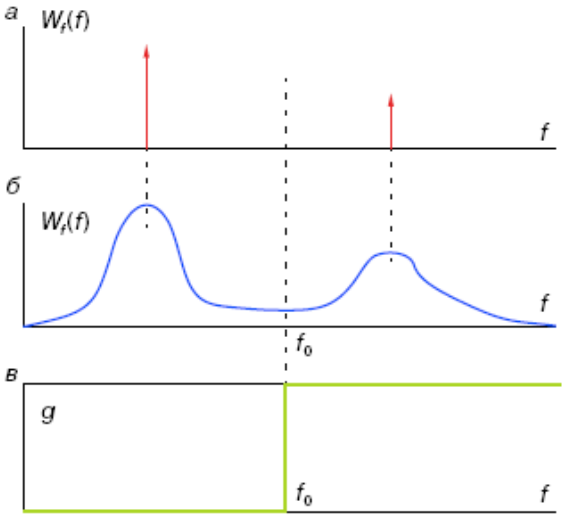
Пороговая обработка полутонового изображения заключается в разделении всех элементов изображения на два класса по признаку яркости, то есть в выполнении поэлементного преобразования вида

$$g = \begin{cases} g_{max}, & f \geq f_0, \\ g_{min}, & f < f_0, \end{cases} \qquad (3)$$

где f_0 - некоторое «пороговое» значение яркости.

При выполнении пороговой обработки основной вопрос состоит в выборе порога f_0.

Пусть полутоновое изображение содержит интересующие нас объекты одной яркости на фоне другой яркости (типичные параметры: машинописный текст, чертежи, медицинские пробы под микроскопом и т.д.). Тогда в идеале плотность распределения яркостей должна выглядеть как две дельта-функции (рис. *а*). В данном случае задача установления порога тривиальна: в качестве f_0 можно взять любое значение между «пиками».



Совместное распределение яркости двух объектов

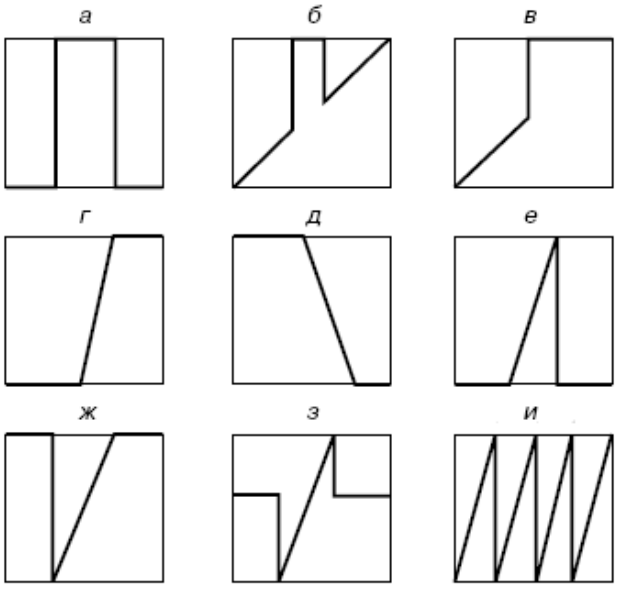
На практике, однако, встречаются определенные трудности, связанные с тем, что, во-первых, как для объектов, так и для фона характерен некоторый разброс яркостей. В результате пики функции плотности распределения «расплываются», хотя обычно ее бимодальность сохраняется (рис. *б*). В такой ситуации можно выбрать порог f_0, соответствующий положению минимума между модами, то есть использовать функцию поэлементного преобразования, показанную на рис. *в*.

2.4.4. Препарирование

Широкий класс процедур обработки изображения заключается в их препарировании, то есть в приведении к такому виду, который, возможно, весьма далек от естественного, но удобен для визуальной интерпретации или дальнейшего машинного анализа.

Частным случаем препарирования является пороговая обработка, рассмотренная выше. Перечислим некоторые другие используемые преобразования.

Очевидным обобщением пороговой обработки является преобразование яркостного среза (*а*). Оно позволяет выделить определенный интервал диапазона яркостей входного изображения. Перемещая «рабочий» интервал по шкале и меняя его ширину, можно произвести визуальный анализ отдельных изображенных объектов, различающихся по



Функции препарирования изображений

яркости. Детали, не попадающие в указанный интервал, то есть относящихся к «фону», будут подавлены. Функция (*б*) иллюстрирует вариант яркостного среза с сохранением фона. В данном случае изображение в целом сохраняется, но на нем «высвечиваются» участки, попавшие в заданный интервал яркостей. Если этот интервал примыкает к границе шкалы яркости, то получаем преобразование так называемой неполной пороговой обработки (функция (*в*)).

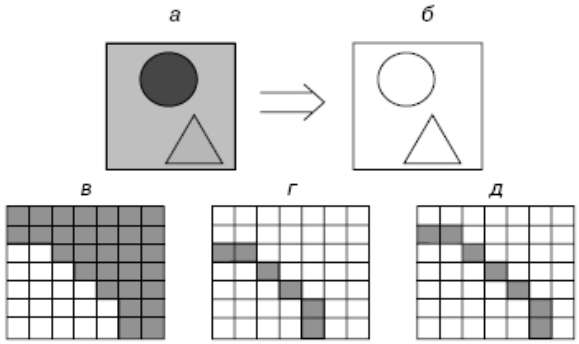
Контрастное масштабирование в своем простейшем варианте совпадает по смыслу с линейным контрастированием, рассмотренным в п.2.4.2, здесь «рабочий» интервал яркостей растягивается на весь диапазон допустимых значений (функция (*г*)). В других случаях Контрастное масштабирование может быть связано с обращением функции яркости, то есть получением «негатива» (функция (*д*)), представлением «рабочего» интервала яркостей на однородном фоне: черном (функция (*е*)), белом (функция (*ж*)) или сером (функция (*з*)) и т. д.

Пилообразное контрастное масштабирование иллюстрирует рис. и. Как показывает практика, если изображение состоит из нескольких крупных областей с медленно меняющимися (по плоскости) значениями яркости, то такое преобразование почти не разрушает целостности его восприятия и в то же время резко увеличивает контрастность плохо различимых мелких деталей.

2.4.5. Выделение контуров

Задача выделения контуров состоит в построении изображения именно границ объектов и очертаний однородных областей.

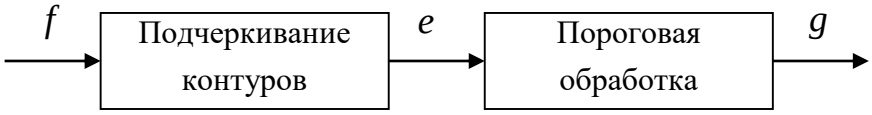
На рис. а, б показаны соответственно исходное изображение, состоящее из областей различной яркости, и его графический вариант, состоящий только из границ этих областей.



Выделение контуров

Будем называть контуром изображения совокупность его пикселов, в окрестности которых наблюдается скачкообразное изменение функции яркости. Так как цифровой обработке изображение представлена как функция целочисленных аргументов, то контуры представляются линиями шириной, как минимум, в один пиксел. При этом может возникнуть неоднозначность в определении линии контура, как это показано на рис. *г,д* для исходного изображения с перепадом яркости (рис. *в*).

Общую процедуру построения бинарного изображения границ объектов иллюстрирует схема, представленная на рисунке.



Процедура выделения контуров

Исходное изображение *f* подвергается линейной или нелинейной обработке, с тем

чтобы выделить перепады яркости. В результате этой операции формируется изображение *e*, функция яркости которого существенно отличается от нуля только в областях резких изменений яркости изображения *f*. Затем в результате пороговой обработки из изображения *e* формируется графический (контурный) препарат *g*.

Правильный выбор порога на втором этапе должен производиться из следующих соображений. При слишком высоком пороге могут появиться разрывы контуров, а слабые перепады яркости не будут обнаружены. При слишком низком пороге из-за шумов и неоднородности областей могут появиться ложные контуры. Других особенностей пороговая обработка не имеет. Поэтому обратим основное внимание на первую операцию – выделение перепадов яркости (контуров) – и рассмотрим основные методы выполнения этой операции.

Градиентный метод

Одним из наиболее простых способов выделения границ является пространственное дифференцирование функции яркости. Градиент функции яркости вычисляется по формуле

$$|\Delta f(x,y)| = \sqrt{\left(\frac{\partial f(x,y)}{\partial x}\right)^2 + \left(\frac{\partial f(x,y)}{\partial y}\right)^2}$$

и пропорционален максимальной (по направлению) скорости изменения функции яркости в данной точке и не зависит от направления контура.

Частные производные можно вычислить приближенно с помощью следующих конечных разностей:

$$\frac{\partial f(x,y)}{\partial x} \approx \frac{f(x_{i+1},y_j) - f(x_i,y_j)}{x_{i+1} - x_i}, \quad \frac{\partial f(x,y)}{\partial y} \approx \frac{f(x_i,y_{j+1}) - f(x_i,y_j)}{y_{j+1} - y_j}.$$

Т. к. $x_{i+1} - x_i = y_{j+1} - y_j = 1$, то

$$\frac{\partial f(x,y)}{\partial x} \approx f(x_{i+1},y_j) - f(x_i,y_j), \quad \frac{\partial f(x,y)}{\partial y} \approx f(x_i,y_{j+1}) - f(x_i,y_j).$$

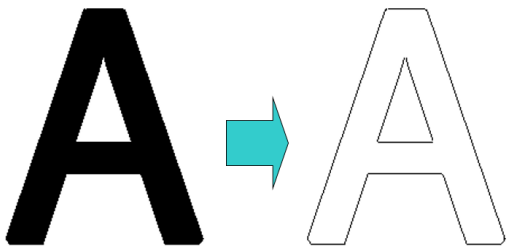
Следовательно

$$|\Delta f(x,y)| \approx \sqrt{\left(f(x_{i+1},y_j) - f(x_i,y_j)\right)^2 + \left(f(x_i,y_{j+1}) - f(x_i,y_j)\right)^2}.$$

Таким образом, операция выделения контуров заключается в выполнении нелинейной локальной обработки изображения «окно» 2×2 (без одной точки):

$$e(i,j) = \sqrt{\left(f(x_{i+1},y_j) - f(x_i,y_j)\right)^2 + \left(f(x_i,y_{j+1}) - f(x_i,y_j)\right)^2}.$$

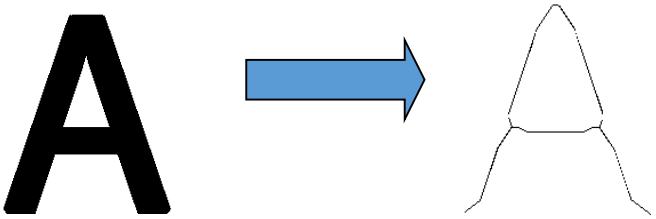
Эвристический алгоритм (от греч. *heurisco* – открывать, узнавать новое) предполагает, что для каждого пиксела *C*(*x*,*y*) с координатами *x* и *y* вводятся локальная



F1=300, *F2*=100

плотность изображения $d(x,y)=\sum_{i=-1}^{1}\sum_{j=-1}^{1}C(x+i,y+j)-C(x,y)$ и плотность локальных плотностей $D(x,y)=\sum_{i=-1}^{1}\sum_{j=-1}^{1}d(x+i,y+j)-d(x,y)$. Пиксел остается на изображении, если для него $d>F1$ и $D>F2$, где $F1$ и $F2$ – некоторые константы, подбираемые для конкретного изображения экспериментально.

2.4.6. Алгоритмы утончения линий



а) объект до утончения б) после утончения

Утончение линий

Задача «*утончения линий*» (или «*задача скелетизации*») актуальна в тех случаях, когда необходимо преобразовать произвольное монохромное изображение в штриховое, т.е. состоящее из отдельных точек и линий. Например, такая проблема встречается в процессе предварительной обработки отсканированных изображений текстов и чертежей, предназначенных для дальнейшего распознавания. Существует множество разнообразных алгоритмов, мы рассмотрим наиболее характерные из них.

Скелетом изображения будем называть множество точек, равноудаленных от границ изображения.

Итерационные алгоритмы утончения линий, предусматривают несколько последовательных циклов просмотра изображения, во время каждого из которых удаляются пикселы наружного слоя фигур. Различают *последовательные* алгоритмы, в которых важен порядок обхода пикселов изображения (например, слева направо и сверху вниз) и *параллельные*, в которых пикселы могут просматриваться в произвольном порядке. Во всех подобных алгоритмах пикселы сначала только помечаются как удаленные, а убираются с изображения только по окончании цикла сканирования.

1. Параллельный алгоритм Зонга-Суэна

В качестве примера рассмотрим.

Введем матрицу 3x3: $\begin{bmatrix} P9 & P2 & P3 \\ P8 & P1 & P4 \\ P7 & P6 & P5 \end{bmatrix}$

Накладываем матрицу на изображение, совмещая интересующий нас пиксел с $P1$.

Каждая итерация состоит из 2-х подытераций:

Подытерация 1:

Пиксел P1 удаляется из изображения, если выполняются следующие условия:

a) $2 \le B(P1) \le 6$;

b) $A(P1)=1$;

c) $P2 \times P4 \times P6 = 0$;

d) $P4 \times P6 \times P8 = 0$.

Здесь A(P1) - число конфигураций 01 в последовательности P2,P3,P4,P5,P6,P7,P8,P9, замыкая эту цепочку на P2, т.е. вокруг этого пиксела существует только один переход от 0 к 1.

$$B(P_i) = \sum_{i=2}^{9} P_i$$

Подытерация 2:

Выполняется аналогично, только

$P2 \times P4 \times P8 = 0$,

$P2 \times P6 \times P8 = 0$.

Таким образом:

Подытерация 1. Удаление точек на юго-восточной границе и северо-западных угловых точек

Подытерация 2. Удаление точек на северо-западной границе и юго-восточных угловых точек

Эти итерации мы выполняем до тех пор, пока не будет удален ни один пиксел.

Но этими условиями мы не охватываем некоторых случаев. Рассмотрим следующую картинку:

▮	▮	▮	▮	▮
1	▮	▮	▮	▮
1	0	0	▮	▮
1	p	0	▮	▮
0	1	1	1	▮

Пиксел P, если он был единицей, этими условиями не удаляется. Поэтому проводится еще одна итерация, которая устраняет подобные недочеты.

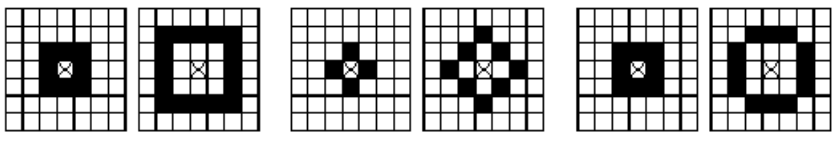
На этой итерации ищутся два единичных пиксела по вертикали или горизонтали, которые окружены нулями.

Итак, точка P удаляется, если выполняется одно из условий:

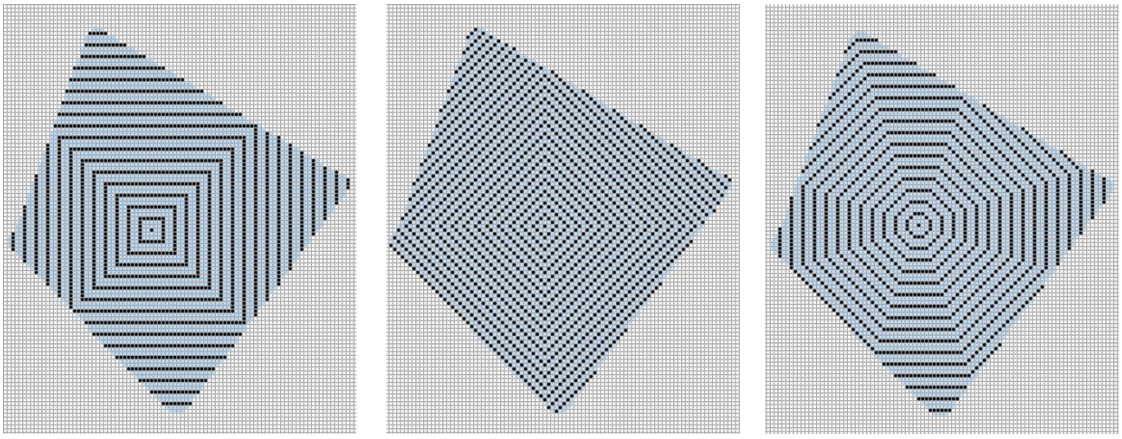
1) !P9*P4*P6=1,
2) !P5*P8*P2=1,
3) !P3*P6*P8=1,
4) !P7*P2*P4=1, (где !P9- отрицание P9).

2. Волновой алгоритм

Использует последовательную пометку пикселов, причем этот процесс напоминает распространение кругов на воде. На рис. изображены первые генерации трех видов «волны», родившейся из помеченного крестом пиксела. Под «связностью» здесь понимается количество рассматриваемых «соседей» пиксела.

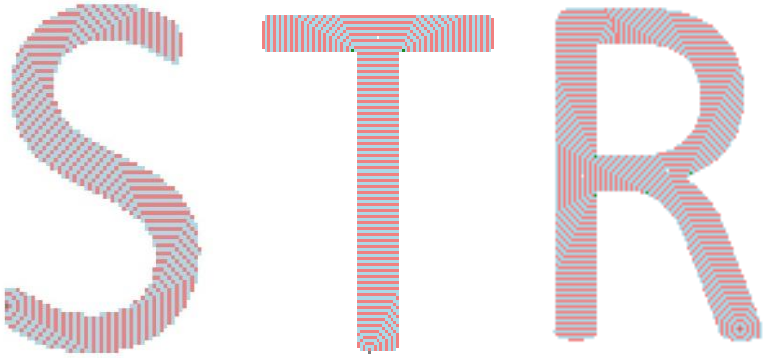


а) 8-связная б) 4-связная в) комбинированная



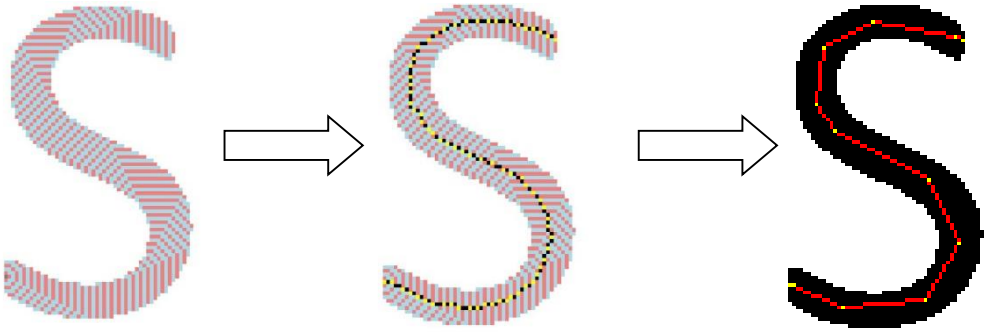
Разновидности «волны»

Выпущенная из одной точки и распространяющаяся «волна» обладает рядом замечательных свойств: довольно быстро приобретает вид «параллельных гребней», умеет «поворачивать» и «огибать препятствия».



Свойства «волны»

Алгоритм волновой скелетизации условно можно разбить на два основных этапа: этап построения первичного графа (включает в себя запуск волны, отслеживание середин генераций волны, мест разделения и затухания волны) и следующий за ним этап оптимизации графа, в результате чего отбрасывается лишняя информация. Иллюстрация работы данных этапов приведена на рисунке.

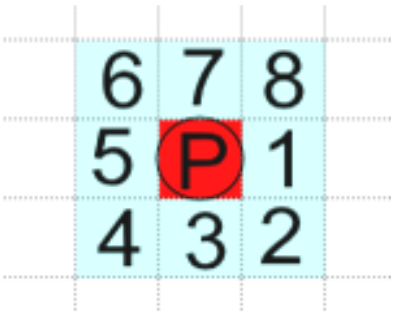


Основные этапы скелетизации: построение первичного графа и оптимизация графа

На первом этапе внутри объекта запускается сферическая волна. Генерация волны – это фронт волны. Точки, принадлежащие отдельным генерациям волны, помечаются разным цветом на изображении, и отслеживаются центральные пиксели каждой нечетной генерации, которые помещаются в первичный граф как его узлы.

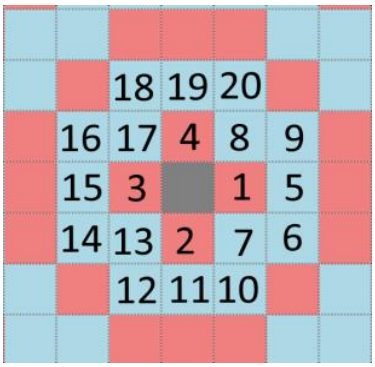
Для получения генерации сферической волны применяется попеременно 4-х и 8-ми связное распространение. На рисунке буквой P обозначен текущий пиксель, а пронумерованы его соседи для 4-х и 8-ми связных представлений, соответственно.

а) 4-х связное представление



б) 8-ми связное представление

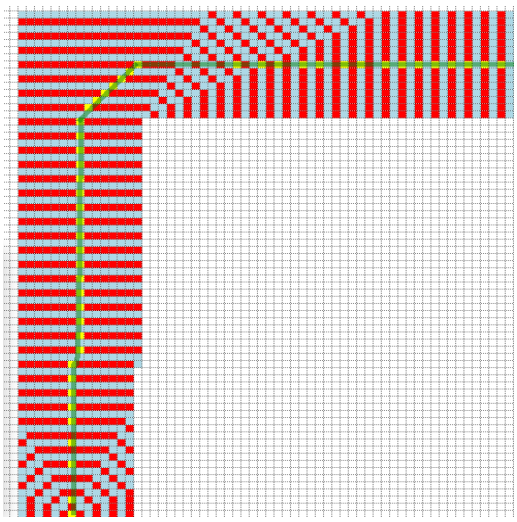
Порядок обхода пикселей



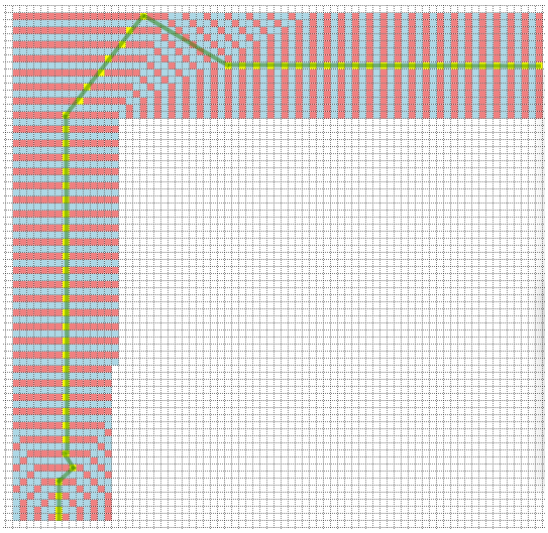
Нумерация пикселей первой и второй генерации

При запуске волны сначала применяется 4-х связное, затем 8-ми связное распространение для каждого пикселя из предыдущей 4-х связной генерации и так далее поочередно. Тогда распространение волны идёт в виде восьмиугольника.

Центральные пиксели генерации волны могут выбираться двумя разными способами: как центры отрезков, соединяющих крайние точки генерации волны либо как серединные пиксели на самой генерации волны.



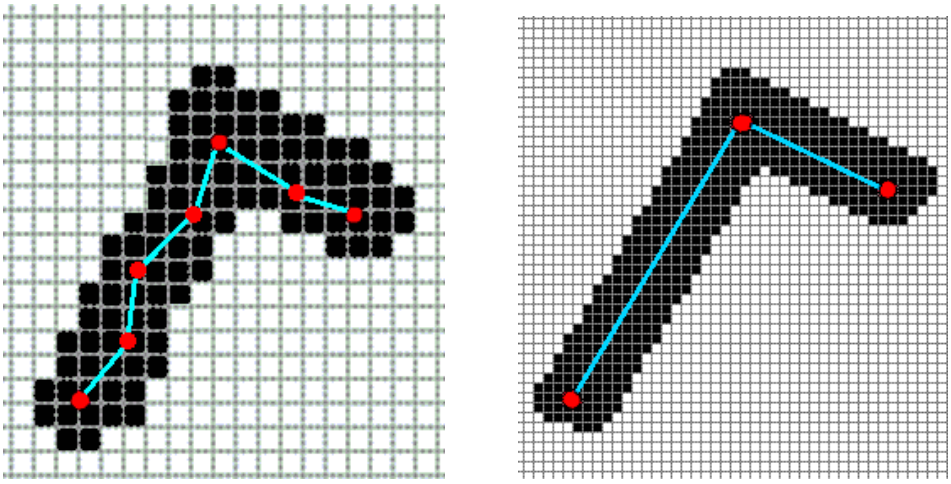
а) за точку скелета принимается середина отрезка, соединяющего крайние точки генерации волны



б) за точку скелета принимается центральный пиксель генерации волны, относительно которого число пикселей слева и справа равно

Отслеживание пути прохождения волны

Первичный граф обычно содержит большое количество избыточной информации, поэтому на этапе оптимизации графа лишние точки из него удаляются.



Оптимизация полученного скелета

2.4.7. Преобразования изображений при помощи масочных фильтров

Эти методы преобразования изображений, в отличие от поэлементных, предусматривают преобразование значения яркости пиксела, зависящее не только от него самого, но и от его «окружения». В простейшем случае в окружение включаются 8 ближайших пикселов, но может рассматриваться и более обширное «соседство».

Методы *линейной масочной фильтрации* предусматривают, что над яркостью каждого пиксела $C(X,Y)$ производится преобразование

$$C'(X,Y) = A + B \cdot \sum_{i=-N}^{N} \sum_{j=-N}^{N} C(X+i, Y+j) \cdot M(X+i, Y+j),$$

где N-количество «слоев», входящих в окружение пиксела (обычно, N=1), A и B - некие константы, M - некая матрица («маска») размерности $(2 \cdot N+1) \times (2 \cdot N+1)$. Следует иметь в виду, что при попиксельном сканировании изображения в рассмотрение должны включаться исходные, а не преобразованные значения яркостей пикселов.

1. Сглаживающие маски:

$$M_1 = \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad M_2 = \frac{1}{10}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad M_3 = \frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}. \qquad (4)$$

Коэффициенты масок нормированы, т.е. $\sum\sum_{(k_1,k_2)\in W} m(k_1,k_2) = 1$, с тем чтобы процедура подавления помех не вызывала смещения яркости исходного изображения. Маски (4) отличаются степенью сглаживания шумов (у маски M_1 она максимальная, у M_3 - минимальная). При увеличении степени сглаживания шумов происходит также подавление высокочастотной составляющей полезного изображения, что вызывает исчезновение мелких деталей и размазывание контуров. Если требуемая степень сглаживания с применением маски размера 3×3 не достигает, то следует использовать сглаживающие маски больших размеров (5×5,7×7 и т.д.).

2. Увеличение контрастности:

$$A=0,\ B=1/4,\ M=\begin{bmatrix} 0 & -1 & 0 \\ -1 & 8 & -1 \\ 0 & -1 & 0 \end{bmatrix};\ A=0,\ B=1,\ M=\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix};$$

3. Придание изображению рельефности:

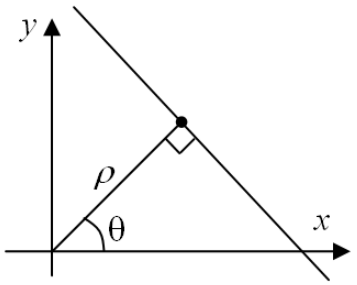
$$A=1/2\times\text{МахЯркость},\ B=1/2,\ M=\begin{bmatrix}-1 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 1\end{bmatrix}.$$

4. В качестве примера *нелинейной масочной фильтрации* приведем метод *медианной фильтрации*: все пикселы, входящие в окружение текущего (включая и его самого), сортируются по яркости, затем текущий пиксел заменяется центральным пикселом в отсортированной группе. Медианная фильтрация уменьшает резкость и позволяет избавиться от мелких деталей, например, от точечных помех.

2.4. Векторизация растровых изображений. Преобразование Хафа (*Hough*)

Элементы растеризации векторных изображений были рассмотрены выше, в разделе, посвященном построению простых геометрических фигур средствами растровой графики. Гораздо более сложна обратная операция – преобразование растровых изображений в векторные.

Рассмотрим специализированное преобразование Хафа. Согласно методике, использующей это преобразование, на монохромном растровом изображении вместо декартовой системы координат вводится полярная с координатами (ρ, θ).



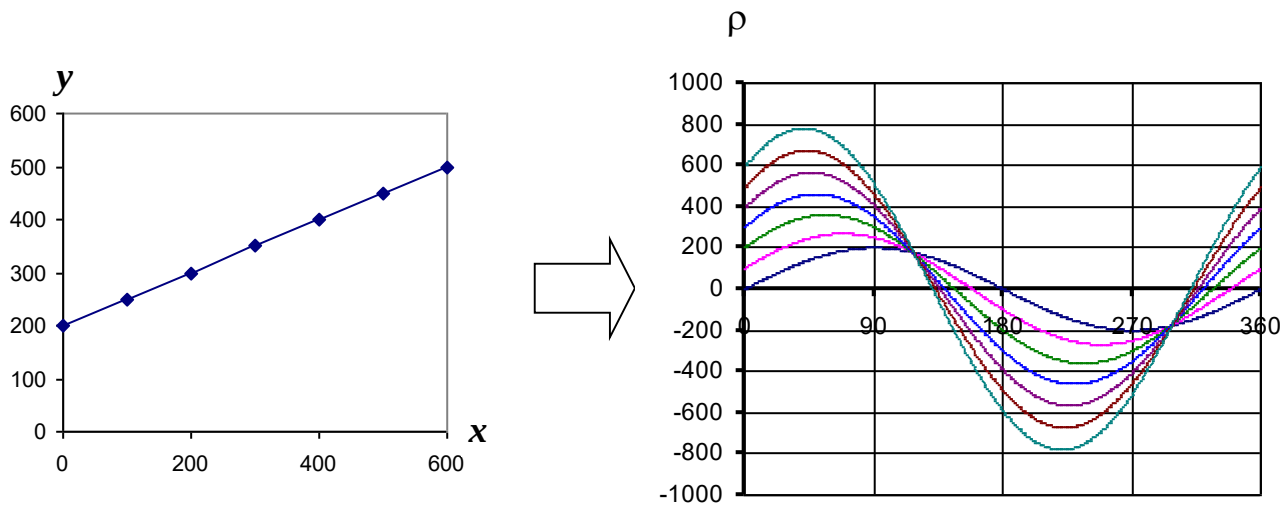
Переход к полярной системе координат

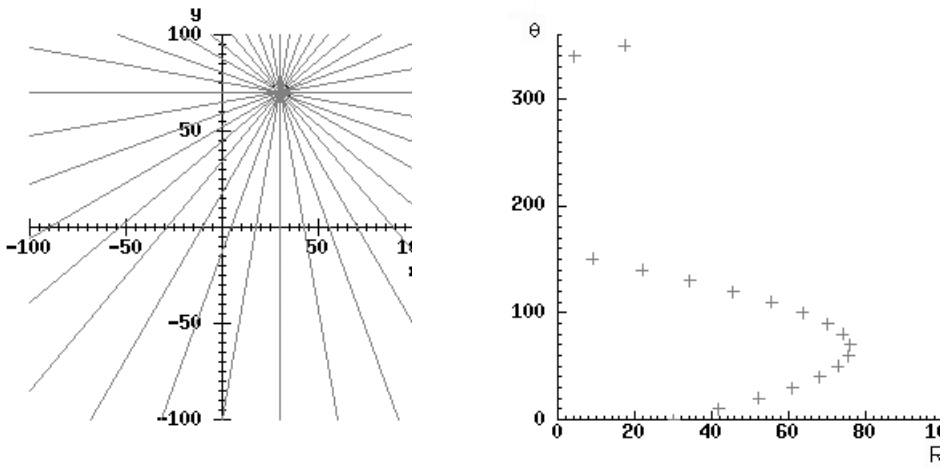
Прямая, проходящая через точку с координатами (x, y), будет определяться уравнением $x\cdot cos\theta+y\cdot sin\theta=\rho$. Здесь ρ - длина перпендикуляра, опущенного на прямую из начала координат, а θ – угол между осью x и данным перпендикуляром.

Через каждую точку (x, y) растрового изображения можно провести несколько прямых с разными ρ и θ (рис. (а)). Таким образом, каждой точке (x, y) соответствует некий набор точек в фазовом пространстве (ρ, θ), образующий синусоиду (рис. (б)).

В свою очередь, каждой точке пространства (ρ,θ) соответствует набор точек (x, y) на изображении, образующий прямую.

Каждой точке (ρ_0, θ_0) пространства (ρ, θ) можно поставить в соответствие счетчик, соответствующий количеству точек (x, y), лежащих на прямой $x\cdot cos\theta_0+y\cdot sin\theta_0=\rho_0$.





а) прямые, проходящие через общую точку

б) соответствующая прямым синусоида

Декартово и полярное пространства

Вследствие дискретности растрового представления исходного изображения, каждой прямой в координатах (x, y) будет соответствовать не одна точка, а сгущение точек в координатах (ρ,θ). Таким образом, достаточно выбрать на изображении, построенном в полярных координатах, самые «жирные пятна» и для их центров произвести преобразование в полярные координаты, получив тем самым параметры соответствующей прямой.

Таким образом алгоритм Хафа нахождения прямых на изображении включает в себя 2 этапа.

1 этап. Построение пространства Хафа (фазового пространства).

2 этап. Выделение точек пересечения синусоид (нахождение локальных максимумов в пространстве Хафа)

На первом этапе пространство Хафа задается в виде двумерного массива, заполнение которого происходит по следующему алгоритму.

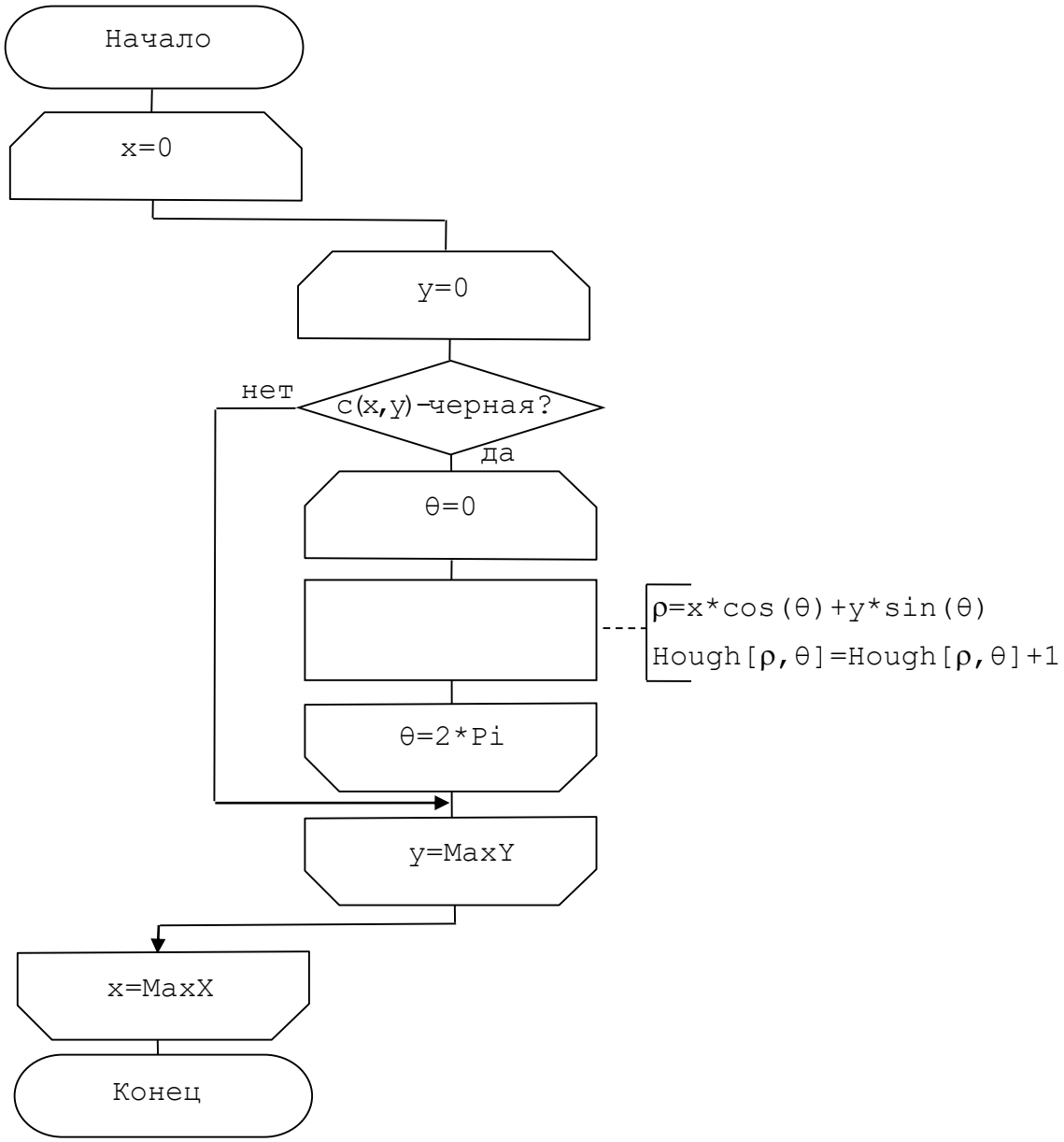
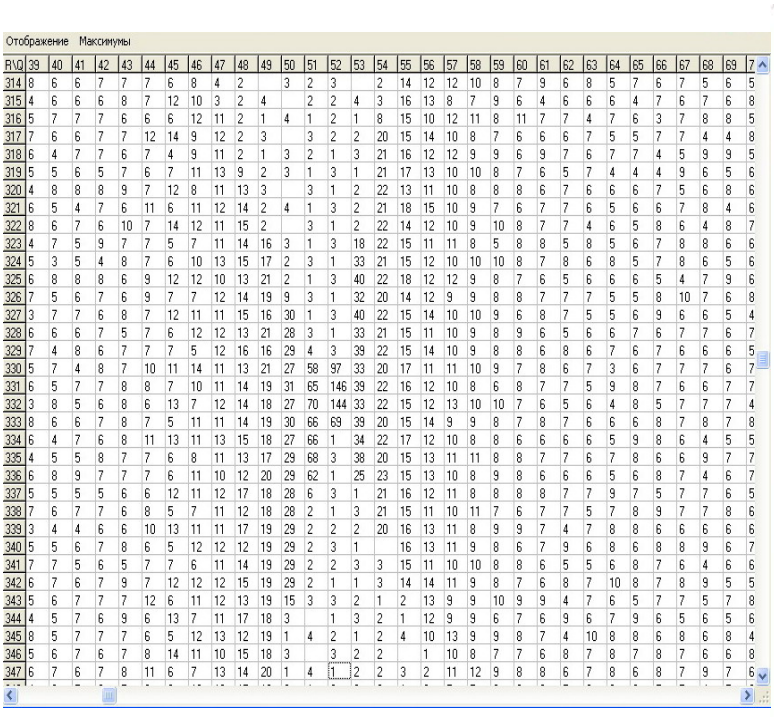
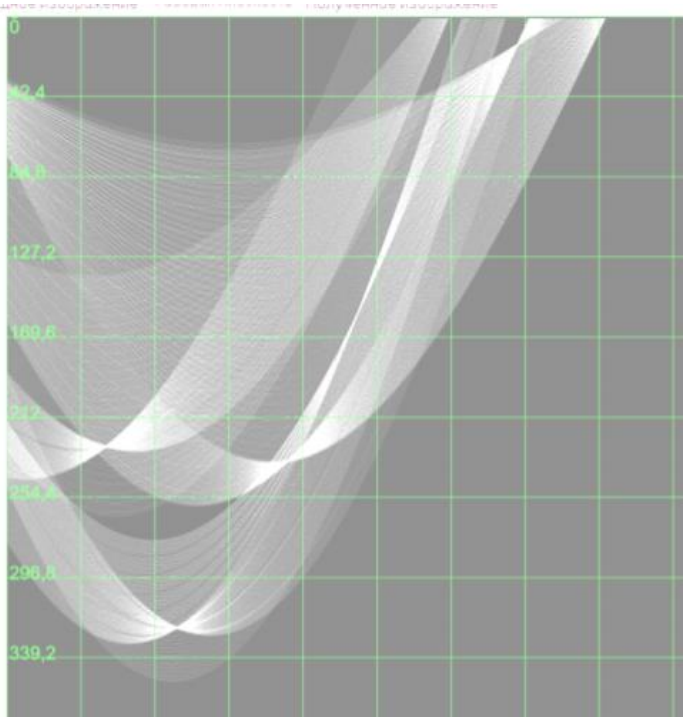


Схема алгоритма преобразования Хафа

В итоге получим массив, заполненный числами. В данном массиве на втором этапе предстоит произвести поиск локальных максимумов (точек пересечения большого количества синусоид). Координаты (ρ,θ) данных локальных максимумов и будут являться параметрами искомых прямых на изображении.


Отображение Максимумы

R\Q	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	7
314	8	6	6	7	7	7	6	8	4	2		3	2	3		2	14	12	12	10	8	7	9	6	8	5	7	6	7	5	6	5
315	4	6	6	6	8	7	12	10	3	2	4		2	2	4	3	16	13	8	7	9	6	4	6	6	6	4	7	6	7	6	8
316	5	7	7	7	6	6	6	12	11	2	1	4	1	2	1	8	15	10	12	11	8	11	7	7	4	7	6	3	7	8	8	5
317	7	6	6	7	7	12	14	9	12	2	3		3	2	2	20	15	14	10	8	7	6	6	6	7	5	5	7	7	4	4	8
318	6	4	7	7	6	7	4	9	11	2	1	3	2	1	3	21	16	12	12	9	9	6	9	7	6	7	7	4	5	9	9	5
319	5	5	6	5	7	6	7	11	13	9	2	3	1	3	1	21	17	13	10	10	8	7	6	5	7	4	4	4	9	6	5	6
320	4	8	8	8	9	7	12	8	11	13	3		3	1	2	22	13	11	10	8	8	8	6	7	6	6	6	7	5	6	8	6
321	6	5	4	7	6	11	6	11	12	14	2	4	1	3	2	21	18	15	10	9	7	6	7	7	6	5	6	6	7	8	4	6
322	8	6	7	6	10	7	14	12	11	15	2		3	1	2	22	14	12	10	9	10	8	7	7	4	6	5	8	6	4	8	7
323	4	7	5	9	7	7	5	7	11	14	16	3	1	3	18	22	15	11	11	8	5	8	8	5	8	5	6	7	8	8	6	6
324	5	3	5	4	8	7	6	10	13	15	17	2	3	1	33	21	15	12	10	10	10	8	7	8	6	8	5	7	8	6	5	6
325	6	8	8	8	6	9	12	12	10	13	21	2	1	3	40	22	18	12	12	9	8	7	6	5	6	6	6	5	4	7	9	6
326	7	5	6	7	6	9	7	7	12	14	19	9	3	1	32	20	14	12	9	9	8	8	7	7	7	5	5	8	10	7	6	8
327	3	7	7	6	8	7	12	11	11	15	16	30	1	3	40	22	15	14	10	10	9	6	8	7	5	5	6	9	6	6	5	4
328	6	6	6	7	5	7	6	12	12	13	21	28	3	1	33	21	15	11	10	9	8	9	6	5	6	6	7	6	7	7	6	7
329	7	4	8	6	7	7	7	5	12	16	16	29	4	3	39	22	15	14	10	9	8	8	6	8	6	7	6	7	6	6	6	5
330	5	7	4	8	7	10	11	14	11	13	21	27	58	97	33	20	17	11	11	10	9	7	8	6	7	3	6	7	7	7	6	7
331	6	5	7	7	8	8	7	10	11	14	19	31	65	146	39	22	16	12	10	8	6	8	7	7	5	9	8	7	6	6	7	7
332	3	8	5	6	8	6	13	7	12	14	18	27	70	144	33	22	15	12	13	10	10	7	6	5	6	4	8	5	7	7	7	4
333	8	6	6	7	8	7	5	11	11	14	19	30	66	69	39	20	15	14	9	9	8	7	8	7	6	6	6	8	7	8	7	8
334	6	4	7	6	8	11	13	11	13	15	18	27	66	1	34	22	17	12	10	8	8	6	6	6	6	5	9	8	6	4	5	5
335	4	5	5	8	7	7	6	8	11	13	17	29	68	3	38	20	15	13	11	11	8	8	7	7	6	7	8	6	6	9	7	7
336	6	8	9	7	7	7	6	11	10	12	20	29	62	1	25	23	15	13	10	8	9	8	6	6	6	5	6	8	7	4	6	7
337	5	5	5	5	6	6	12	11	12	17	18	28	6	3	1	21	16	12	11	8	8	8	8	7	7	9	7	5	7	7	6	5
338	7	6	7	7	6	8	5	7	11	12	18	28	2	1	3	21	15	11	10	11	7	6	7	7	5	7	8	9	7	7	8	6
339	3	4	4	6	6	10	13	11	11	17	19	29	2	2	2	20	16	13	11	8	9	9	7	4	7	8	8	6	6	6	6	6
340	5	5	6	7	8	6	5	12	12	12	19	29	2	3	1		16	13	11	9	8	6	7	9	6	8	6	8	8	9	6	7
341	7	7	5	6	5	7	7	6	11	14	19	29	2	2	3	3	15	11	10	10	8	8	6	5	5	6	8	7	6	4	6	6
342	6	7	6	7	9	7	12	12	12	15	19	29	2	1	1	3	14	14	11	9	8	7	6	8	7	10	8	7	8	9	5	5
343	5	6	7	7	7	12	6	11	12	13	19	15	3	3	2	1	2	13	9	9	10	9	9	4	7	6	5	7	7	5	7	8
344	4	5	7	6	9	6	13	7	11	17	18	3		1	3	2	1	12	9	9	6	7	6	9	6	7	9	6	5	6	5	6
345	8	5	7	7	7	6	5	12	13	12	19	1	4	2	1	2	4	10	13	9	9	8	7	4	10	8	8	6	8	6	8	4
346	5	6	7	6	7	8	14	11	10	15	18	3		3	2	2		1	10	8	7	7	6	8	7	8	7	8	7	6	6	8
347	6	7	6	7	8	11	6	7	13	14	20	1	4	1	2	2	3	2	11	12	9	8	8	6	7	8	6	8	7	9	7	6



Пространство Хафа

Существуют также разновидности преобразования Хафа для дуг и окружностей. Таким образом, преобразование Хафа позволяет представить монохромное штриховое изображение в виде набора формул, описывающих прямолинейные и криволинейные элементы изображения, то есть произвести векторизацию этого изображения.

ЛЕКЦИЯ 7 *Проекции: классификация, применение. Аксонометрическое и перспективное проецирование. Преобразование координат проекции в экранные координаты.*

2.6. Проецирование объемных фигур на плоскость

Большинство существующих в настоящее время устройств отображения предусматривают построение изображения на плоскости. В математическом смысле *проекции* - это преобразования точек пространства размерности *n* в точки пространства размерности меньшей, чем *n*. В компьютерной графике рассматриваются преимущественно проекции образа трехмерного пространства на двумерную картинную плоскость.

Проекция трехмерного объекта, представленного в виде совокупности точек, строится при помощи прямых проецирующих лучей, которые называются проекторами и которые выходят из центра проекции, проходят через каждую точку объекта и, пересекая картинную плоскость, образуют проекцию.

Определенный таким образом класс проекций называют *плоскими геометрическими проекциями*, поскольку проецирование в этом случае производится на *проекционную плоскость* и в качестве проекторов используются прямые. Существуют и другие проекции, в которых проецирование осуществляется на криволинейные поверхности или же проецирование осуществляется не с помощью прямых (такие проекции используются, например, в картографии).

В проецировании участвуют две системы координат:

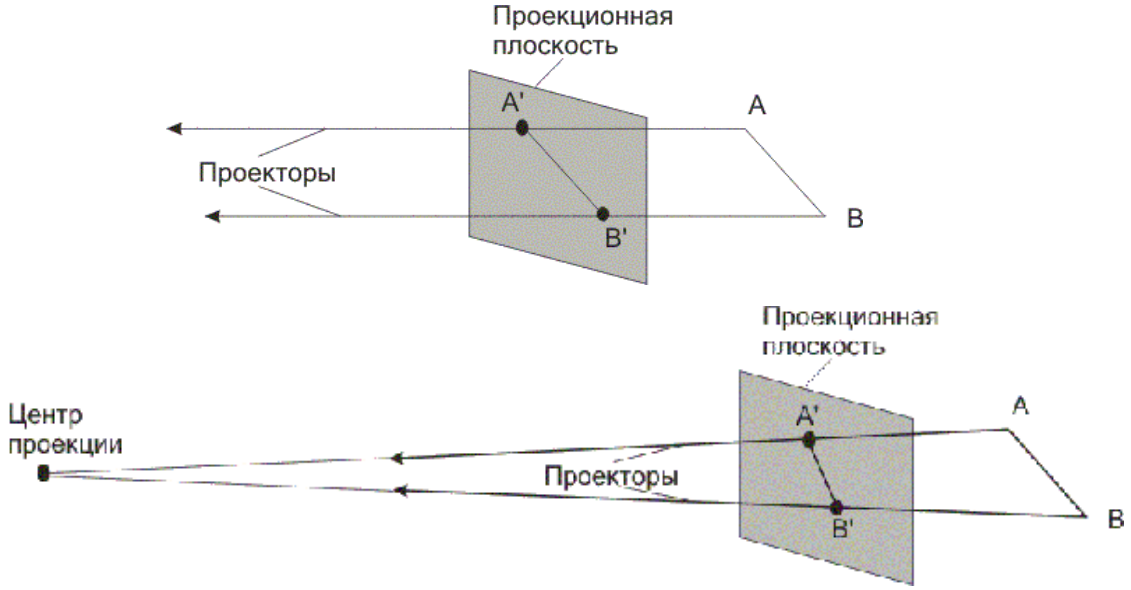
1) *мировые координаты* – описывают истинное положение объектов в пространстве;

2) *координаты проекции* – описывают изображение объекта в заданной плоскости проецирования.

Классификация проекций

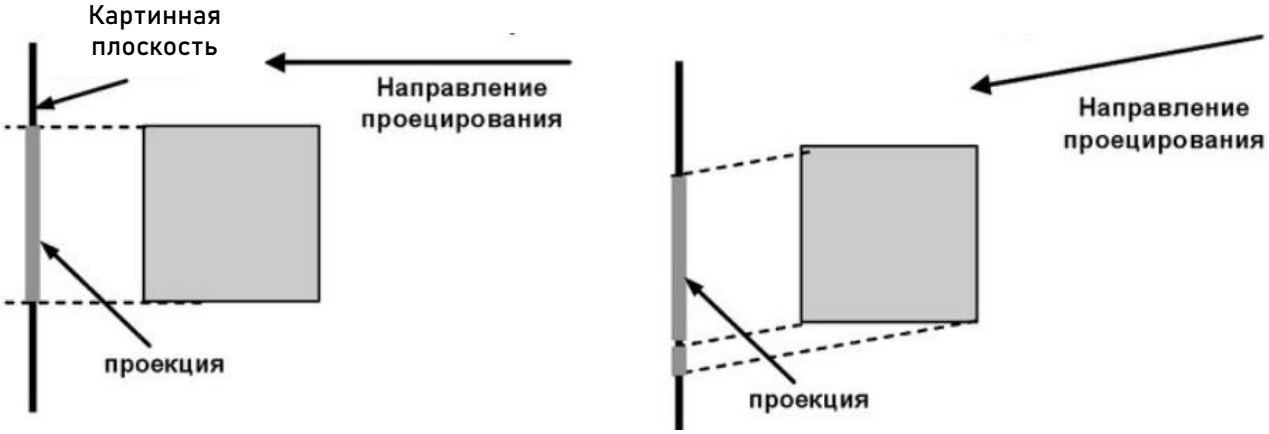
Основная задача проецирования: для точки с координатами (*x*, *y*, *z*) в пространстве мировых координат найти ее координаты (*X*, *Y*, *Z*) на плоскости проецирования.

Плоские геометрические проекции подразделяются на два основных класса: *центральные* и *параллельные*. Различие между ними определяется соотношением между центром проекции и проекционной плоскостью. Если расстояние между ними конечно, то проекция будет *центральной*, если же оно бесконечно, то проекция будет *параллельной*. Параллельные проекции названы так потому, что центр проекции бесконечно удален и все проекторы параллельны.



Параллельная и центральная (перспективная) проекции

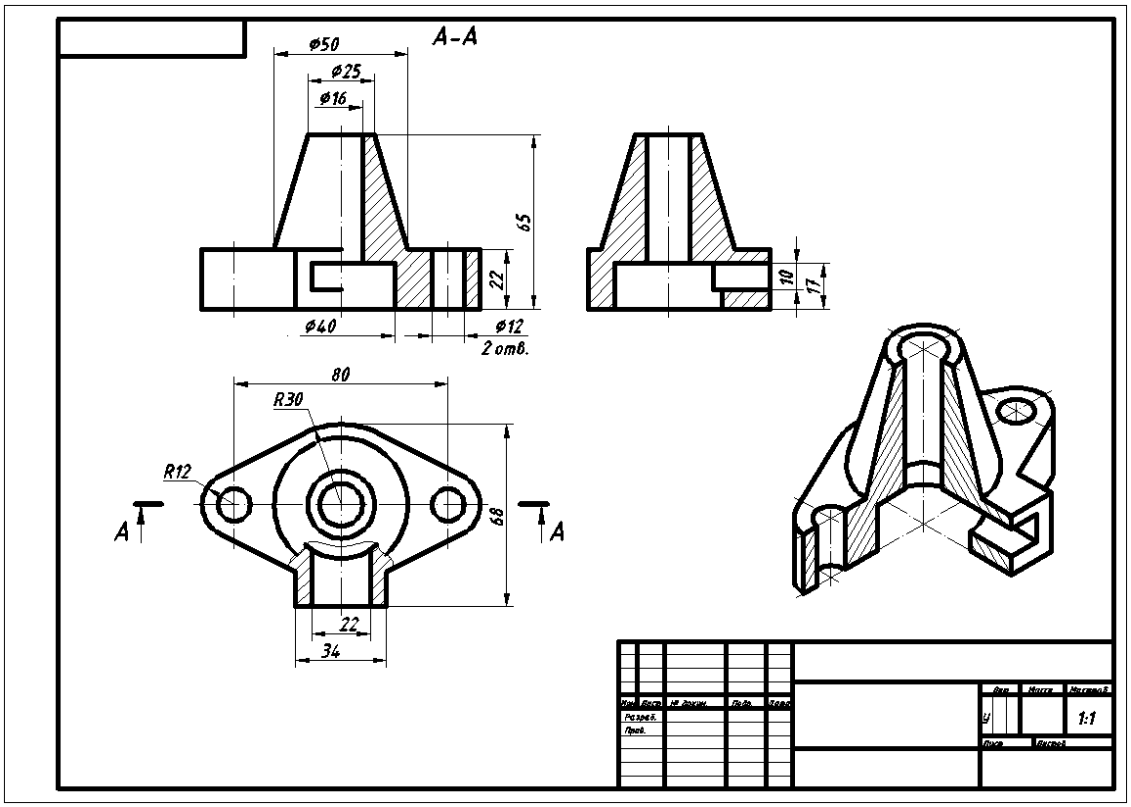
В зависимости от расположения проецирующих лучей относительно картинной плоскости (плоскости проецирования) параллельные проекции разделяются на: *прямоугольные* (проецирующие лучи перпендикулярны плоскости проецирования) и *косоугольные* (проецирующие лучи наклонены к плоскости проецирования).



Прямоугольная (ортогональная) и косоугольная проекции

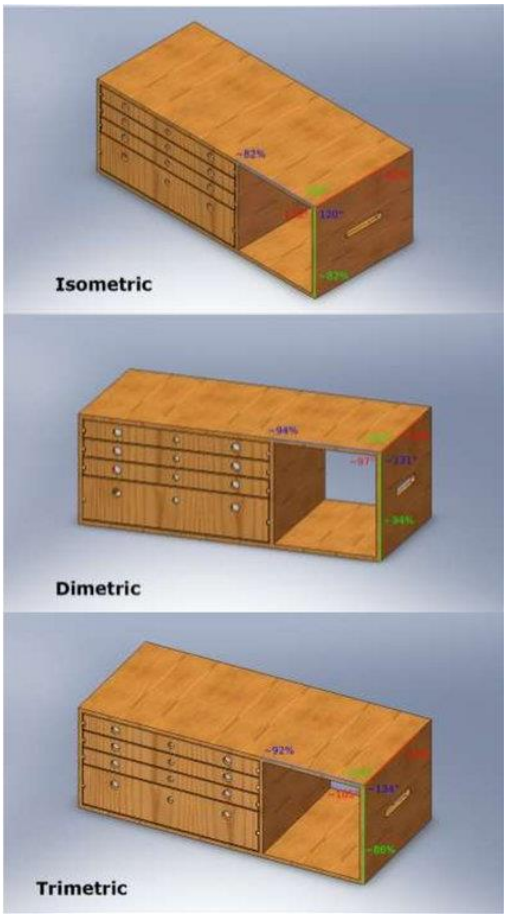
Также параллельные проекции можно разделить на два типа в зависимости от расположения плоскости проецирования относительно координатных плоскостей. Если плоскость проецирования параллельна одной из координатных плоскостей, то это

называется *проекцией вида*. Если же плоскость проецирования расположена в пространстве произвольным образом – это *аксонометрия*.



Ортографические проекции (виды) и аксонометрия

Аксонометрическими проекциями называют изображения, полученные путем проецирования параллельными лучами фигуры (предмета) вместе с осями координат на произвольно расположенную плоскость, которую называют аксонометрической (или картинной).

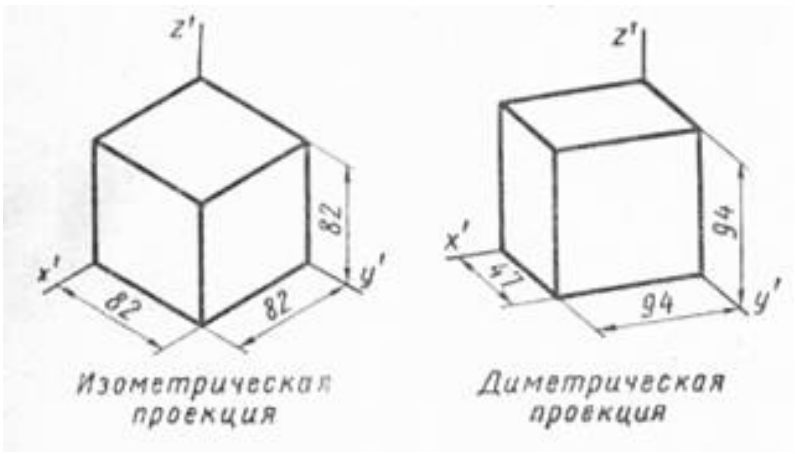


В зависимости от наклона осей координат к плоскости проецирования все аксонометрические проекции делятся на три основных вида:

1) *изометрические* (оси ***z, x*** и ***y*** наклонены одинаково);

2) *диметрические* (две оси координат имеют один и тот же наклон, а третья - другой);

3) *триметрические* (все оси имеют разный наклон.



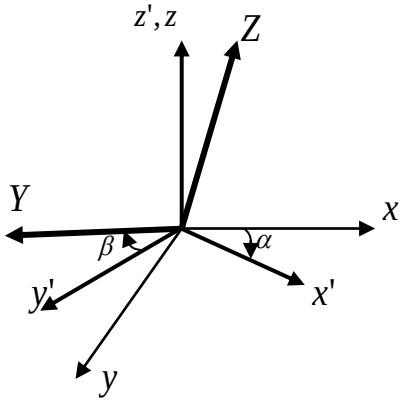
Аксонометрия (изометрия, диметрия, триметрия)

При *аксонометрическом проецировании* положение плоскости проецирования в мировых координатах задается двумя углами α и β (см. рис.):

а) поворот системы координат (x, y, z) относительно оси z мировых координат на угол α.

Матрица преобразования данного поворота $A = \begin{pmatrix} \cos\alpha & \sin\alpha & 0 & 0 \\ -\sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$.

б) поворот новой системы координат (x', y', z') относительно оси x' на угол β. Данный поворот описывается матрицей $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\beta & \sin\beta & 0 \\ 0 & -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$.



Взаимное положение систем координат в аксонометрии

Итоговое преобразование координат выражается произведением матриц:

$$B \times A = \begin{pmatrix} \cos\alpha & \sin\alpha & 0 & 0 \\ -\sin\alpha\cos\beta & \cos\alpha\cos\beta & \sin\beta & 0 \\ \sin\alpha\sin\beta & -\cos\alpha\sin\beta & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Преобразование в координатной форме:

$$\begin{cases} X = x\cos\alpha + y\sin\alpha, \\ Y = -x\sin\alpha\cos\beta + y\cos\alpha\cos\beta + z\sin\beta, \\ Z = x\sin\alpha\sin\beta - y\cos\alpha\sin\beta + z\cos\beta. \end{cases}$$

Свойства аксонометрической проекции:

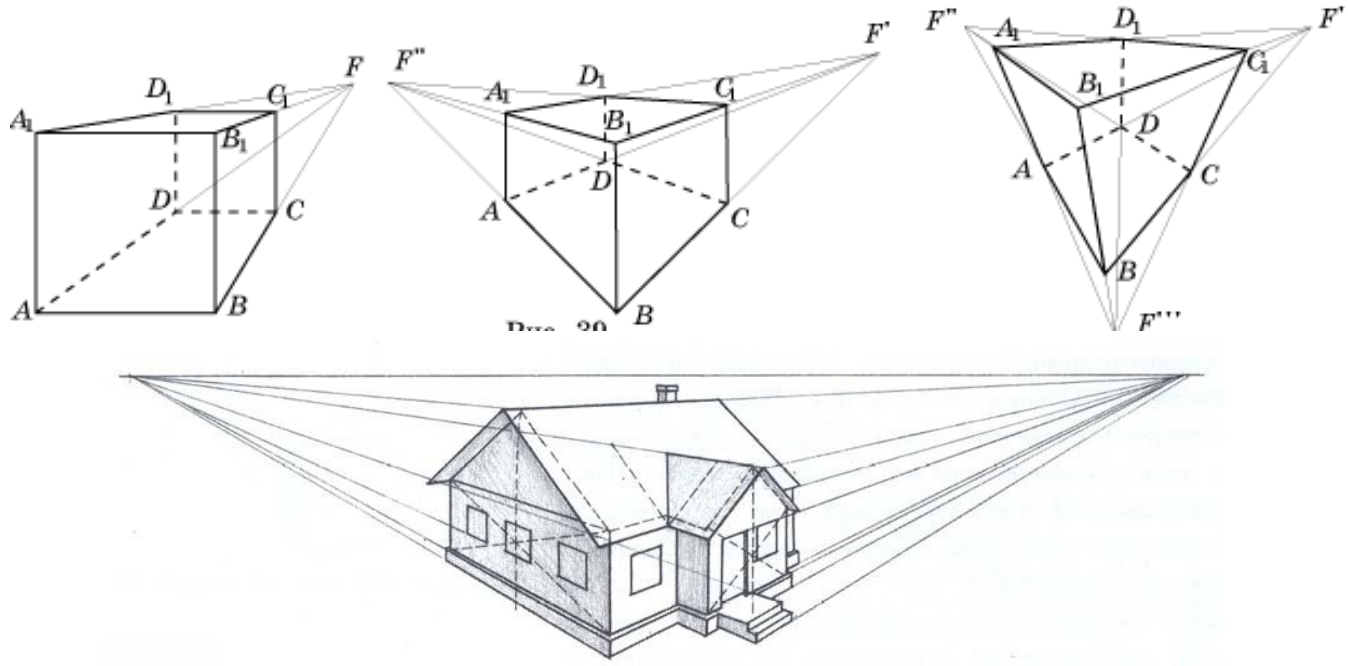
1. Аксонометрические проекции параллельных прямых параллельны между собой и в пространстве. Очевидно, что если параллельные прямые совпадут с направлением проецирования, то указанное свойство теряет смысл, так как проекциями этих прямых будут точки.

2. Отношение аксонометрических проекций отрезков, расположенных на одной прямой или параллельных прямых, равно отношению самих отрезков.

3. Если линии в пространстве пересекаются (или касаются), то и аксонометрические проекции этих линий пересекаются (или касаются).

4. В общем случае окружность изображается в аксонометрии эллипсом, в частном случае она может проектироваться на плоскость аксонометрических проекций в виде отрезка прямой или окружности.

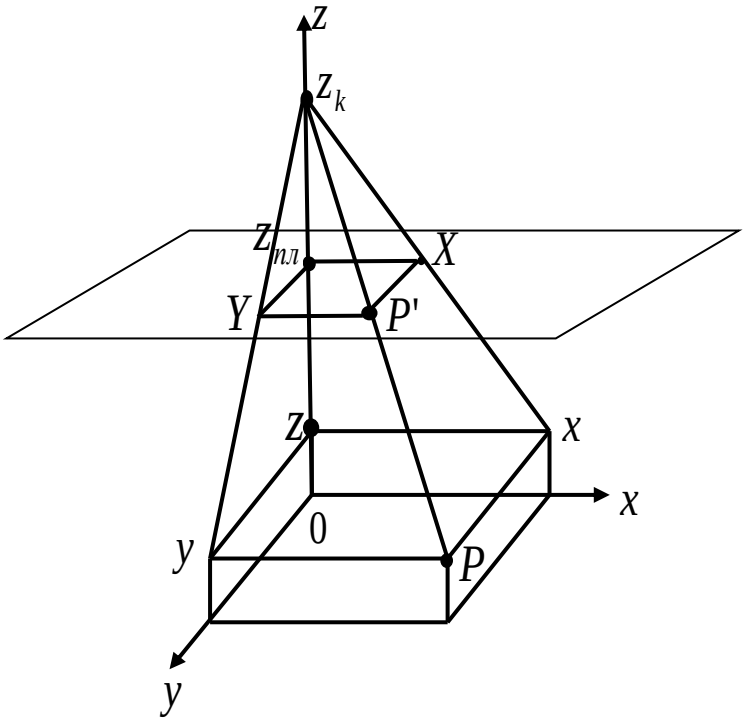
Центральные проекции классифицируются в зависимости от числа главных точек схода, которыми они обладают (т.е. от числа координатных осей, которые пересекает проекционная плоскость). В данной классификации центральные проекции бывают с *одной, двумя и тремя точками схода.*



Центральная проекция с одной, двумя и тремя точками схода

Двухточечная центральная проекция широко применяется в архитектурном, инженерном и промышленном проектировании и в рекламных изображениях, в которых вертикальные прямые проецируются как параллельные и, следовательно, не сходятся. Трехточечные центральные проекции почти совсем не используются, во-первых, потому, что их трудно конструировать, а во-вторых, из-за того, что они добавляют мало нового с точки зрения реалистичности по сравнению с двухточечной проекцией.

Перспективное (центральное) проецирование может быть представлено как последовательная комбинация: 1) аксонометрического проецирования; 2) преобразования, корректирующего линейные размеры.



Перспективное проецирование

Это преобразование зависит от z_k – мировой координаты z камеры и $z_{пл}$ – мировой координаты плоскости проецирования (см. рис.). В координатной форме выглядит следующим образом:

$$\begin{cases} X = x(z_k - z_{пл})/(z_k - z), \\ Y = y(z_k - z_{пл})/(z_k - z), \\ Z = z - z_{пл}. \end{cases}$$

Матрица преобразования

$$\begin{pmatrix} \frac{(z_k - z_{пл})}{(z_k - z)} & 0 & 0 & 0 \\ 0 & \frac{(z_k - z_{пл})}{(z_k - z)} & 0 & 0 \\ 0 & 0 & 1 & -z_{пл} \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Основные свойства перспективного преобразования:

1) прямые линии изображаются прямыми линиями;

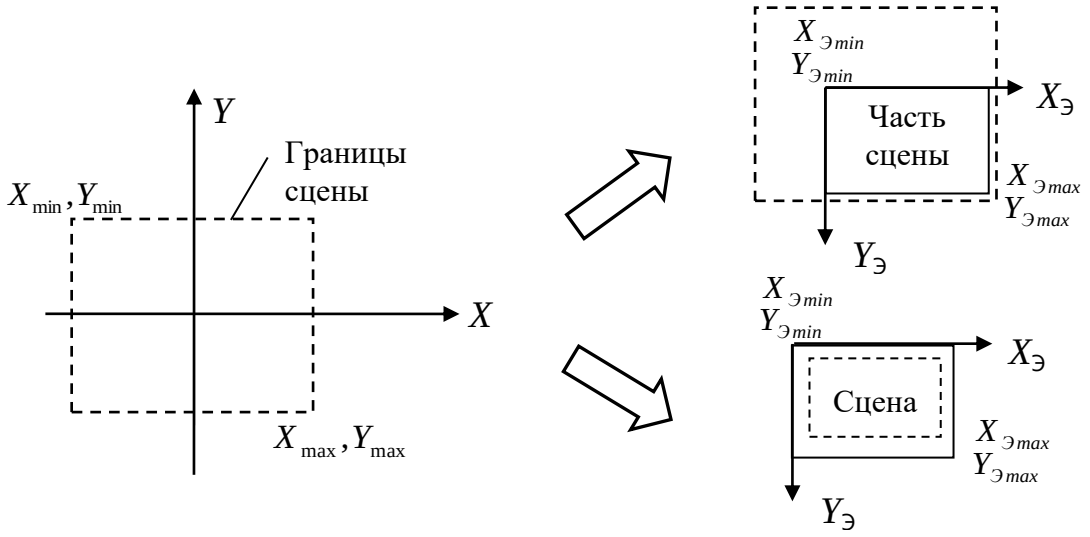
2) не сохраняются отношения длин и площадей;

3) центральные проекции любой совокупности параллельных прямых, которые не параллельны проекционной плоскости, будут сходиться в одной точке.

2.7. Отображение в окне

Пусть (X, Y, Z) – координаты проецирования;

($X_э$, $Y_э$, $Z_э$) – экранные координаты;



Отображение в окне

Для отображения сцены в окне используются два аффинных преобразования – масштабирование и перемещение.

$$\begin{cases} X_э = kX + dx, \\ Y_э = kY + dy, \\ Z_э = kZ. \end{cases}$$

Условия вписывания сцены в окно заданных размеров:

$$\begin{cases} X_{эmin} \le kX_{min} + dx, \\ Y_{эmin} \le kY_{min} + dy, \\ X_{эmax} \ge kX_{max} + dx, \\ Y_{эmax} \ge kY_{max} + dy. \end{cases}$$

$$\Rightarrow k \le min\left(\frac{X_{эmax} - X_{эmin}}{X_{max} - X_{min}}, \frac{Y_{эmax} - Y_{эmin}}{Y_{max} - Y_{min}} \right) = k_{min}.$$

Чтобы изображение в окне было максимальным, нужно, чтобы $k = k_{min}$.

Условия для *dx*, *dy*:

$X_{эmin} - kX_{min} \le dx \le X_{эmax} - kX_{max}$, $Y_{эmin} - kY_{min} \le dy \le Y_{эmax} - kY_{max}$.

Чтобы изображение располагалось по центру экрана:

$$dx = \frac{X_{эmin} - kX_{min} + X_{эmax} - kX_{max}}{2}, \; dy = \frac{Y_{эmin} - kY_{min} + Y_{эmax} - kY_{max}}{2}.$$