

Любая комбинация  $N$  последовательных АП

$$p_0 \xrightarrow{A_1, B_1} p_1 \xrightarrow{A_2, B_2} \dots \xrightarrow{A_N, B_N} p_N,$$

где  $p_1 = p_0 A_1 + B$ ,  $p_2 = p_1 A_2 + B_2 = p_0 A_1 A_2 + B_1 A_2 + B_2$  и по индукции

$$p_N = p_0 \prod_{i=1}^N A_i + \sum_{i=1}^N B_i \prod_{j=i+1}^N A_j = p_0 A + B, \quad (2.5, a)$$

эквивалентна *одному* АП  $p_0 \xrightarrow{A, B} p_N$  с матрицами преобразования

$$A = A_1 A_2 \dots A_N, \\ B = (\dots (B_1 A_2 + B_2) A_3 + \dots + B_{N-1}) A_N + B_N. \quad (2.5, б)$$

Этот алгоритм использует операции сложения и умножения матриц. Он достаточно трудоемок в расчете матрицы  $B$ .

Комбинация  $N$  преобразований вектора  $V$  согласно (2.4) и (2.5) также не зависит от наличия в этих преобразованиях переносных составляющих:

$$V_N = V \prod_{i=1}^N A_i. \quad (2.6)$$

С целью упрощения и унификации записи геометрических преобразований вводится понятие *расширенного пространства* (РП)  $R^{n+1}$ , в котором  $(n + 1)$ -я координата равна фиксированному числу — единице или нулю. Координаты точки  $\tilde{p} = [p \ 1]$  или вектора  $\tilde{V} = [p \ 0]$  в РП часто называют *однородными*. Наиболее удачное, по мнению автора, объяснение этому спорному термину приведено в [35].

Преобразование (2.1) точки в РП принимает вид

$$[p \ 1] = [p \ 1] \begin{bmatrix} A & O_{n \times 1} \\ B & 1 \end{bmatrix} \text{ или } \tilde{p}' = \tilde{p} C, \text{ где } C = \begin{bmatrix} A & O_{n \times 1} \\ B & 1 \end{bmatrix} \in R^{(n+1) \times (n+1)} \quad (2.7)$$

есть матрица преобразования в РП с обратной матрицей

$$C^{-1} = \begin{bmatrix} A^{-1} & O_{n \times 1} \\ -BA^{-1} & 1 \end{bmatrix},$$

описывающей *обратное преобразование* в РП  $\tilde{p} = \tilde{p}' C^{-1}$ , равносильное (2.2).

Цепочка из  $N$  преобразований в РП

$$\tilde{p}_0 \xrightarrow{C_1} \tilde{p}_1 \xrightarrow{C_2} \dots \xrightarrow{C_N} \tilde{p}_N$$

эквивалентна одному преобразованию с матрицей

$$C = \prod_{i=1}^N C_i. \quad (2.8)$$

По сравнению с алгоритмом (2.5) матрица преобразования в расширенном пространстве рассчитывается единообразно — с помощью только перемножения матриц. Унификация — основное *преимущество* выполнения АП в однородных координатах, позволяющее без лишних раздумий правильно записать и быстро вычислить матрицу сложного преобразования  $C$  как произведение матриц элементарных преобразований  $C_i$ . Несмотря на то, что для этого требуется выход в пространство  $R^{n+1}$ , количество арифметических операций в общем итоге даже снижается, да и программирование (2.8) гораздо проще, чем (2.5, б).

К *недостаткам* преобразований в расширенном пространстве можно отнести:

- избыточную размерность расширенных матриц и векторов;
- фиксированные элементы последнего столбца  $[0 \dots 0 \ 1]^T$ , потребляющие машинные ресурсы;
- необходимость контроля и коррекции фиксированных элементов после матричных операций в связи с возможным накоплением погрешностей машинного округления.

В отличие от (2.3), одновременное *групповое преобразование* точек может быть выполнено по более простой матричной формуле

$$\begin{bmatrix} \tilde{p}'_1 \\ \dots \\ \tilde{p}'_m \end{bmatrix} = \begin{bmatrix} \tilde{p}_1 \\ \dots \\ \tilde{p}_m \end{bmatrix} C. \quad (2.9)$$

Преобразование *вектора*  $V = p_2 - p_1$  согласно (2.4) и (2.7) выполняется в РП следующим образом:

$$[V' \ 0] = [V \ 0] \begin{bmatrix} A & O_{n \times 1} \\ B & 1 \end{bmatrix} \text{ или } \tilde{V}' = \tilde{V} C. \quad (2.10)$$

Смещение произвольной точки  $p$  при ее преобразовании по (2.1) равно

$$d(p) = p' - p = p(A - E_n) + B.$$

Точку  $p_H$ , остающуюся в процессе АП *неподвижной*, можно найти, решая уравнения  $p_H = p_H A + B$  или  $d(p_H) = O_n$ :

$$p_H = B(E_n - A)^{-1}. \quad (2.11)$$

Условие существования точки  $p_H$  — невырожденность матрицы  $E_n - A$ . Точка, неподвижная при всех допустимых значениях матриц  $A$  и  $B$ , называется *центром* преобразования.

Невырожденное преобразование прямой и плоскости, описываемых в неявной форме векторным уравнением  $\tilde{p}F^T = 0$ , изменяет их матрицу  $F$  следующим образом:

$$\tilde{p}'F'^T = \tilde{p}CF^T = \tilde{p}F^T = 0 \Rightarrow F = F'C^T, \quad \boxed{F' = F(C^T)^{-1}}. \quad (2.12)$$

Читателю предлагается самостоятельно убедиться, что перенос плоскости из *примера 1.10*, имеющей вектор коэффициентов неявной формы  $F = [6 \ 3 \ 2 \ -6]$ , в начало координат с помощью любой из матриц

$$C_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -2 & 0 & 1 \end{bmatrix}, \quad C_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -3 & 1 \end{bmatrix}$$

даст вектор  $F' = [6 \ 3 \ 2 \ 0]$ .

Важным свойством линейных преобразований является равенство модулей определителей  $\|A\| = \|C\|$  матриц АП изменению площади (в  $\mathbb{R}^2$ ) и объема (в  $\mathbb{R}^3$ ) преобразуемого объекта, что позволяет быстро вычислять эти важнейшие параметры геометрических фигур по известным их исходным значениям и матрицам преобразований.

Знание координат  $(n+1)$ -ой пары точек образа и прообраза, не лежащих в пространстве  $\mathbb{R}^{n-1}$ , позволяет вычислить матрицы  $C$  и  $C^{-1}$  методом парных точек:

$$C = \begin{bmatrix} \tilde{p}_1 \\ \dots \\ \tilde{p}_{n+1} \end{bmatrix}^{-1} \begin{bmatrix} \tilde{p}'_1 \\ \dots \\ \tilde{p}'_{n+1} \end{bmatrix}, \quad C^{-1} = \begin{bmatrix} \tilde{p}'_1 \\ \dots \\ \tilde{p}'_{n+1} \end{bmatrix}^{-1} \begin{bmatrix} \tilde{p}_1 \\ \dots \\ \tilde{p}_{n+1} \end{bmatrix}. \quad (2.13)$$

Здесь обязательно нужно контролировать последние столбцы полученных матриц.

Часто бывает удобнее оставить объект неподвижным, а преобразования подвергать систему координат. Это пассивное АП описывается матрицей  $C_k = C^{-1}$ , обратной к матрице  $C$  такого же активного преобразования объекта. Дополним метод парных точек (2.13) формулами расчета матриц прямого и обратного преобразований СК

$$C_k = \begin{bmatrix} \tilde{p}'_1 \\ \dots \\ \tilde{p}'_{n+1} \end{bmatrix}^{-1} \begin{bmatrix} \tilde{p}_1 \\ \dots \\ \tilde{p}_{n+1} \end{bmatrix}, \quad C_k^{-1} = \begin{bmatrix} \tilde{p}_1 \\ \dots \\ \tilde{p}_{n+1} \end{bmatrix}^{-1} \begin{bmatrix} \tilde{p}'_1 \\ \dots \\ \tilde{p}'_{n+1} \end{bmatrix}, \quad (2.14)$$

в которых  $\tilde{p}_1 + \tilde{p}_{n+1}$  и  $\tilde{p}'_1 + \tilde{p}'_{n+1}$  — расширенные формы одноименных точек в исходном и конечном положениях СК. После вычисления матрицы  $C_k$  любая точка  $p_i$  имеет в новой СК координаты  $\tilde{p}'_i = \tilde{p}_i C_k$ , а любой вектор  $V = p_2 - p_1$  преобразуется в вектор

$$\tilde{V}' = \tilde{V} C_k. \quad (2.15)$$

## 2.1.2. Элементарные аффинные преобразования

Все множество аффинных преобразований может быть сведено к следующим *элементарным преобразованиям* (ЭП).

### 2.1.2.1. Перенос

*Перенос* (transfer) всех точек объекта на вектор  $p$  в неподвижной СК (преобразования  $I$  на рис. 2.2, а и б) описывается уравнением

$$p' = p + \Delta p$$

и матрицами переноса

$$A = E_n, B = \Delta p, T(\Delta p) = \begin{bmatrix} E_n & O_{n \times 1} \\ \Delta p & 1 \end{bmatrix}, T(-\Delta p) = \begin{bmatrix} E_n & O_{n \times 1} \\ -\Delta p & 1 \end{bmatrix} = T^{-1}(\Delta p). \quad (2.16)$$

Переносное смещение  $d = p$  не зависит от положения точки. Неподвижная точка согласно (2.11) удовлетворяет уравнению

$$p_n O_{n \times n} = \Delta p.$$

При ненулевом переносе  $p \neq O_n$  она отсутствует, а при  $p = O_n$  все точки пространства неподвижны. Определитель матрицы переноса  $|T| = 1$ .

### 2.1.2.2. Масштабирование

*Масштабирование* (растяжение, сжатие, отражение) вдоль осей координат описывается координатными уравнениями

$$x' = m_x x, y' = m_y y, z' = m_z z$$

с коэффициентами  $m_x$ ,  $m_y$  и  $m_z$  — диагональными элементами матриц масштабирования:

$$A = \text{diag} \left[ \underbrace{m_x \quad m_y}_{n=2} \quad m_z \right], B = O_n, M(m_x, m_y, m_z) = \begin{bmatrix} m_x & 0 & 0 \\ 0 & m_y & 0 \\ 0 & 0 & m_z \end{bmatrix}. \quad (2.17)$$

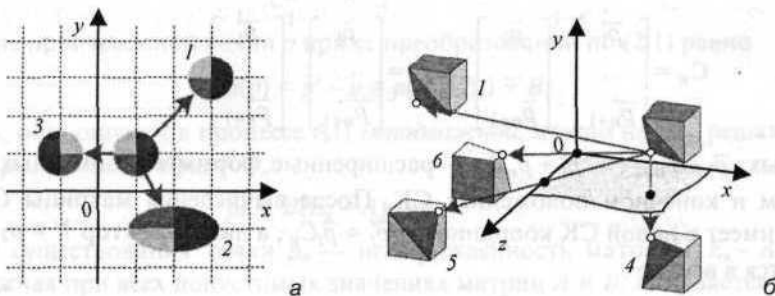


Рис. 2.2

При  $m_\omega = -1$  точка отражается вдоль оси  $\omega \in \{x, y, z\}$  от координатной плоскости  $\omega = 0$ . Если все коэффициенты  $m_x = m_y = m_z = -1$ , то точка *кососимметрично* отражается относительно начала СК. На рис. 2.2 изображены операции:

- масштабирования 2 в  $R^2$  матрицей  $M(2, -1)$ ;
- отражения 3 в  $R^2$  вдоль оси  $x$  матрицей  $M(-1, 1)$ ;
- отражения 4 в  $R^3$  вдоль оси  $y$  (от плоскости  $xz$ ) матрицей  $M(1, -1, 1)$ ;
- отражения 5 в  $R^3$  от оси  $z$  матрицей  $M(-1, -1, 1)$ ;
- отражения 6 в  $R^3$  от начала координат матрицей  $M(-1, -1, -1)$ .

Неподвижная точка, подвергаемая масштабированию, удовлетворяет уравнению

$$p_H \cdot \text{diag}[-m_x \quad 1 - m_y \quad 1 - m_z] = O_n.$$

При  $m_\omega = 1$  неподвижны все точки на оси  $\omega$ . Центр масштабирования находится в начале координат, т. к. только точка  $p_H = O_n$  остается неподвижной при произвольных масштабных коэффициентах  $m_\omega$ .

Масштабирование с нечетным числом отрицательных коэффициентов изменяет топологические свойства преобразуемых объектов. В частности, направление обхода вершин полигона при таком масштабировании меняется на противоположное. Возможность подобных эффектов необходимо предусматривать в алгоритмах, правильность работы которых зависит от направления обхода. Примером может служить задача определения видимости плоской грани полиэдра по направлению ее рассчитываемой нормали, зависящему от направления обхода вершин полигона грани. Если оно положительно и нормаль внешняя, то грань видима. Зеркальное отражение полигона этой грани имеет отрицательное направление обхода и внутреннюю нормаль, рассчитываемую по тем же формулам, что и нормаль самой грани. Таким образом, отраженная грань, реально видимая в зеркале, будет идентифицирована программой как невидимая и не отобразится на экране.

Определитель матрицы масштабирования равен  $|M| = m_x m_y m_z$ . У матрицы отражения с коэффициентами  $m_\omega = \pm 1$  он равен  $|M| = (-1)^k$ , где  $k$  — число инвертируемых координат преобразуемой точки. Например, при отражении вдоль *одной* оси получим  $|M| = -1$ . В результате инвертируются направления обхода всех отражаемых полигонов без изменения их площадей, а также объемов полиэдров.

При неравных масштабных коэффициентах преобразование масштабирования изменяет углы между векторами, площади, объемы и форму геометрических фигур. Неравномерное растяжение или сжатие вдоль осей координат — простейший способ получения эллипса из окружности и эллипсоида из сферы.

### 2.1.2.3. Сдвиг

*Сдвиг (shift)* вдоль направлений осей СК или их комбинации описывается следующими координатными уравнениями и *матрицами сдвига* в  $R^2$  и  $R^3$ :

$$\begin{cases} x' = x + s_{xy}y \\ y' = y + s_{yx}x \end{cases} \Rightarrow S_2 = \begin{bmatrix} 1 & s_{yx} \\ s_{xy} & 1 \end{bmatrix}; \quad (2.18, a)$$

$$\begin{cases} x' = x + s_{xy}y + s_{xz}z \\ y' = s_{yx}x + y + s_{yz}z \\ z' = s_{zx}x + s_{zy}y + z \end{cases} \Rightarrow S_3 = \begin{bmatrix} 1 & s_{yx} & s_{zx} \\ s_{xy} & 1 & s_{zy} \\ s_{xz} & s_{yz} & 1 \end{bmatrix}. \quad (2.18, б)$$

Параметры сдвига задаются *недиагональными* элементами матриц  $S_2$  и  $S_3$ . Коэффициент матрицы сдвига  $s_{ij}$  есть единица смещения вдоль оси  $i$ , пропорционального  $j$ -ой координате.

Сдвиговое смещение

$$d = p(S_n - E_n) \quad (2.19)$$

зависит от положения точки  $p$ , что и отличает его от переноса. Неподвижная точка удовлетворяет уравнению

$$p_n(S_n - E_n) = O_n.$$

Если в матрице  $S_n$  все недиагональные элементы  $\omega$ -ой строки — нулевые, то при сдвиге неподвижны все точки, лежащие на оси  $\omega$ . Две нулевые строки матрицы  $S_n - E_n$  дают множество неподвижных точек на координатной плоскости. Центр сдвига при произвольных параметрах матрицы  $S_n$  находится в начале координат.

Важно отметить, что если в матрице сдвигового преобразования элемент  $s_{ij} \neq 0$  не единственен, то векторы смещения (2.19) не параллельны, а сдвиг осуществляется одновременно вдоль нескольких осей СК, и его нельзя заменить последовательностью одноосевых сдвигов:

$$\begin{bmatrix} 1 & 0 \\ s_{xy} & 1 \end{bmatrix} \begin{bmatrix} 1 & s_{yx} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & s_{yx} \\ s_{xy} & 1 + s_{xy}s_{yx} \end{bmatrix} \neq \begin{bmatrix} 1 & s_{yx} \\ s_{xy} & 1 \end{bmatrix}.$$

Преобразование сдвига может иметь вырожденную матрицу с нулевым определителем, например,

$$S_2 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad S_3 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & -1 & 1 \end{bmatrix}.$$

Размерность пространства, натянутого на точки сдвинутого в  $R^n$  объекта, равна рангу матрицы  $S_n$ . Следовательно, при  $\text{rang}(S_n) < n$  преобразование сдвига не является аффинным.

На рис. 2.3, а приведены примеры преобразований сдвига на плоскости с матрицами (2.18, а) и сдвиговыми смещениями (2.19):

- отрезка ( $I$ ) матрицей

$$S_2 = \begin{bmatrix} 1 & 0 \\ -2 & 1 \end{bmatrix} -$$

с параллельными векторами смещений  $d = [-2y \ 0] \parallel x^\circ$  и множеством неподвижных точек, лежащих на оси  $x$ ;

- квадрата (2) и круга (3) матрицей

$$S_2 = \begin{bmatrix} 1 & 1 \\ -2 & 1 \end{bmatrix}$$

с непараллельными векторами смещений  $d = [-2y \ x]$ .

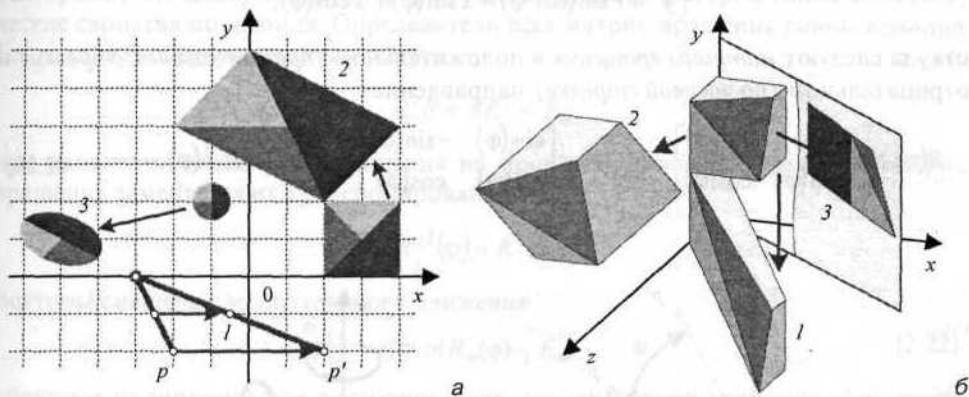


Рис. 2.3

На рис. 2.3, б выполнены преобразования объемной фигуры в пространстве:

- сдвиг (1) вдоль оси  $y$  матрицей

$$S_3 = \begin{bmatrix} 1 & -2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

с параллельными векторами смещений  $d = [0 \ -2x \ 0] \parallel y^\circ$  и множеством неподвижных точек, лежащих в плоскости  $xz$ ;

- сдвиг (2) матрицей

$$S_3 = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & -1 & 1 \end{bmatrix}$$

с векторами  $d = [-y \ -z \ -z \ 0]$ ;

- сдвиг (3) вырожденной матрицей

$$S_3 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

ранга 2, в результате чего все точки прообраза лежат в плоскости  $x-z = 0$ .

### 2.1.2.4. Вращение

*Вращение (rotation)* точки на плоскости  $xy$  вокруг начала координат (рис. 2.4, а) описывается следующими координатными уравнениями:

$$\begin{cases} x' = r \cos(\alpha + \varphi) = x \cos(\varphi) - y \sin(\varphi); \\ y' = r \sin(\alpha + \varphi) = x \sin(\varphi) + y \cos(\varphi), \end{cases}$$

откуда следуют *матрицы вращения* в положительном (*против часовой стрелки*) и отрицательном (*по часовой стрелке*) направлениях:

$$R(\varphi) = \begin{bmatrix} \cos(\varphi) & \sin(\varphi) \\ -\sin(\varphi) & \cos(\varphi) \end{bmatrix}, \quad R(-\varphi) = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{bmatrix} = R^T(\varphi) = R^{-1}(\varphi). \quad (2.20)$$

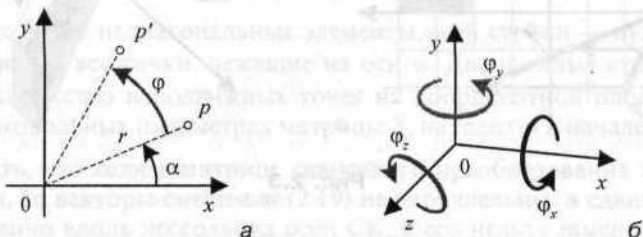


Рис. 2.4

При вращении точки в пространстве вокруг осей *правой* системы координат (положительные направления вращений удовлетворяют правилу буравчика и показаны на рис. 2.4, б) матрицы вращений получим из соответствующих уравнений:

$$\square x' = x, y' = y \cos(\varphi_x) - z \sin(\varphi_x), z' = y \sin(\varphi_x) + z \cos(\varphi_x),$$

$$R_x(\varphi_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi_x) & \sin(\varphi_x) \\ 0 & -\sin(\varphi_x) & \cos(\varphi_x) \end{bmatrix}, \quad R_x(-\varphi_x) = R_x^{-1}(\varphi_x); \quad (2.21, a)$$

$$\square x' = x \cos(\varphi_y) + z \sin(\varphi_y), y' = y, z' = -x \sin(\varphi_y) + z \cos(\varphi_y),$$

$$R_y(\varphi_y) = \begin{bmatrix} \cos(\varphi_y) & 0 & \sin(\varphi_y) \\ 0 & 1 & 0 \\ -\sin(\varphi_y) & 0 & \cos(\varphi_y) \end{bmatrix}, \quad R_y(-\varphi_y) = R_y^{-1}(\varphi_y); \quad (2.21, б)$$

$$\square x' = x \cos(\varphi_z) - y \sin(\varphi_z), y' = x \sin(\varphi_z) + y \cos(\varphi_z), z' = z,$$



$$R_z(\varphi_z) = \begin{bmatrix} \cos(\varphi_z) & \sin(\varphi_z) & 0 \\ -\sin(\varphi_z) & \cos(\varphi_z) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad R_z(-\varphi_z) = R_z^{-1}(\varphi_z). \quad (2.21, \text{в})$$

Преобразование вращения не изменяет углы между синхронно вращающимися векторами, площади, объемы и форму геометрических фигур, а также топологические свойства полигонов. Определители всех матриц вращения равны единице. Благодаря свойству *ортогональности*

$$R^T R = R R^T = E_n$$

при изменении знака угла вращения на противоположный обращение матриц вращения заменяется их транспонированием:

$$R^{-1}(\varphi) = R^T(\varphi).$$

Векторы смещения вращательного движения

$$d = p(R_\omega(\varphi) - E_n) \quad (2.22)$$

образуют цилиндрическое векторное поле, ось симметрии которого  $\omega$  содержит множество неподвижных точек. Центр всех вращений на произвольные углы вокруг любых осей координат располагается в начале СК.

Строго говоря, вращение не является элементарной операцией, т. к. его можно представить последовательностью масштабирования и сдвига:

$$\begin{bmatrix} \cos(\varphi) & \sin(\varphi) \\ -\sin(\varphi) & \cos(\varphi) \end{bmatrix} = \begin{bmatrix} \cos(\varphi) & 0 \\ 0 & \cos(\varphi) \end{bmatrix} \begin{bmatrix} 1 & \operatorname{tg}(\varphi) \\ -\operatorname{tg}(\varphi) & 1 \end{bmatrix}.$$

Тем не менее ввиду простоты понимания, распространенности и важности места вращательного движения в системе геометрических преобразований целесообразно считать вращение элементарным, а не сложным преобразованием.

### 2.1.2.5. Табличный расчет тригонометрических функций

Весьма важным в практике применения матриц вращения является *быстрый* расчет тригонометрических функций угла и наоборот — угла по значениям его тригонометрических функций. Актуальность этой проблемы объясняется присутствием преобразования вращения практически в каждом сложном преобразовании и высокой частотой выполнения этой операции, в том числе и во вложенных циклах. Использование встроженных в языки программирования функций  $\cos$  и  $\sin$ , вычисляемых суммированием рядов Тейлора, конечно, заманчиво, но непрактично из-за их невысокого быстродействия.

Еще меньшая скорость сходимости у обратных тригонометрических функций, из которых во встроженной библиотеке обычно имеется только функция  $\arctg$ . От-

существующие функции  $\arcsin$  и  $\arccos$  можно вычислить либо создав *собственные варианты* этих функций

$$asin(x) = \text{if} \left( |x| < 1: \arctg \left( \frac{\sqrt{1-x^2}}{x} \right), \text{sgn}(x) \cdot \frac{\pi}{2} \right), \quad (2.23, a)$$

$$acos(x) = \text{if} \left( x: \arctg \left( \frac{\sqrt{1-x^2}}{x} \right) + \{x < 0\} \cdot \pi, \frac{\pi}{2} \right), \quad (2.23, б)$$

либо суммируя до сходимости по заданной погрешности  $\epsilon$  соответствующие ряды Тейлора:

$$\begin{aligned} \arcsin(x) &= x + \frac{1}{2 \cdot 3} x^3 + \frac{3}{2 \cdot 4 \cdot 5} x^5 + \frac{3 \cdot 5}{2 \cdot 4 \cdot 6 \cdot 7} x^7 + \dots = \\ &= x + \sum_{k=1}^{\infty} \frac{x^{2k+1}}{2k+1} \prod_{i=2}^{2k} i^{(-1)^{i-1}}, \end{aligned}$$

$$\arccos(x) = \frac{\pi}{2} - \arcsin(x).$$

Оба ряда у границ области  $|x| \leq 1$  сходятся очень медленно (более 4000 итераций) и с недопустимо большой ошибкой (до 0.0086 рад, т. е.  $0.5^\circ$ ) при абсолютной погрешности  $\epsilon = 10^{-6}$ .

В настоящее время наиболее быстрым и популярным считается *табличный метод* вычисления тригонометрических функций с кусочно-линейной интерполяцией между узлами сетки. Суть его в том, что при запуске прикладной программы *один раз* с высокой точностью вычисляется  $N + 1$  узловое значение наиболее востребованной в машинной графике монотонно убывающей функции  $c(x) = \cos(x)$  в интервале  $x \in [0, \pi/2]$  с шагом  $\Delta x = \pi/2N$ . Таким образом, в оперативной памяти компьютера создается  $2 \times (N + 1)$ -матрица (таблица) узлов сетки

$$x_i = i\Delta x, c_i = \cos(x_i) \quad \forall i = \overline{0, N}.$$

При вычислении *тригонометрических функций*  $\cos(x)$  и  $\sin(x)$  произвольный аргумент  $x$  циклическим смещением

$$x = x - 2\text{sgn}(x)\pi$$

приводится в интервал  $[-\pi, \pi]$ . Далее используются известные тождества, сужающие интервал до  $[0, \pi/2]$ :

$$\bullet \quad 0 \leq x < \pi/2: \quad \cos(x) = c(x), \quad \sin(x) = c(\pi/2 - x); \quad (2.24, a)$$

$$\bullet \quad \pi/2 \leq x < \pi: \quad \cos(x) = -c(\pi - x), \quad \sin(x) = c(x - \pi/2); \quad (2.24, б)$$

$$\bullet \quad -\pi/2 \leq x < 0: \quad \cos(x) = c(-x), \quad \sin(x) = -c(\pi/2 + x); \quad (2.24, в)$$

$$\bullet \quad -\pi \leq x < -\pi/2: \quad \cos(x) = -c(\pi + x), \quad \sin(x) = -c(-\pi/2 - x). \quad (2.24, г)$$

Вычисляя значение  $c(\xi)$  по аргументу  $0 < \xi < \pi/2$ , определим индекс  $i = [\xi / \Delta x]$ , при котором  $x_i < \xi < x_{i+1}$  (рис. 2.5). Тогда приближенное линейно интерполированное значение функции равно

$$c(\xi) \approx c_i + (c_{i+1} - c_i) \left( \frac{\xi}{\Delta x} - i \right).$$

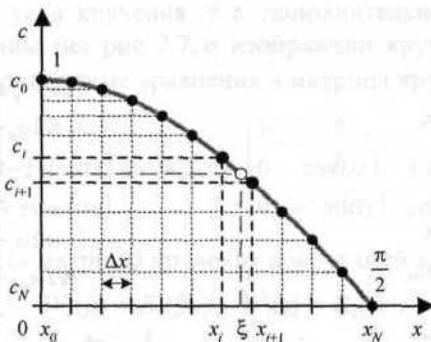


Рис. 2.5

Допустим, составлена таблица  $\{x_i, c_i\}$  с крупным шагом  $\Delta x = 0.01745$  рад =  $1^\circ$  и задан аргумент  $x = -14.2$  рад. Сместив его на два периода, получим

$$x = -14.2 + 4\pi = -1.6336 \text{ рад} = -93.6^\circ.$$

Вычислим по (2.24, z) функции  $\cos(x)$  и  $\sin(x)$ :

- $\xi = \pi + x = 1.508$  рад  $\Rightarrow i = 86$ ,  $c_{86} = 0.06976$ ,  $c_{87} = 0.05234$ , откуда

$$-c(\xi) = -0.062789,$$

в то время как  $\cos(x) = -0.069792$ . Абсолютная и относительная погрешности интерполяции составили  $2.25 \cdot 10^{-6}$  и  $3.59 \cdot 10^{-5}$ ;

- $\xi = -\pi/2 - x = 0.06283$  рад  $\Rightarrow i = 3$ ,  $c_3 = 0.99863$ ,  $c_4 = 0.99756$ , откуда

$$-c(\xi) = -0.99799,$$

тогда как  $\sin(x) = -0.998027$ . Абсолютная и относительная погрешности интерполяции составили  $3.648 \cdot 10^{-5}$  и  $3.655 \cdot 10^{-5}$ .

Расчет обратных тригонометрических функций  $0 < \arccos(a) < \pi$  (рис. 2.6, а) и  $-\pi/2 < \arcsin(a) < \pi/2$  (рис. 2.6, б) аргумента  $-1 < a < 1$  начинается с поиска индекса  $i$ , для которого  $c_{i+1} < |a| < c_i$ . С помощью линейно-интерполяционной функции

$$L(a, i) = x_i + \frac{a - c_i}{c_{i+1} - c_i} \Delta x$$

получим приближенные значения

$$\arccos(a) \approx \begin{cases} L(a, i) & \text{при } a \geq 0; \\ \pi - L(-a, i) & \text{при } a < 0; \end{cases} \quad (2.25, a)$$

$$\arcsin(a) \approx \begin{cases} \pi/2 - L(a, i) & \text{при } a \geq 0; \\ L(-a, i) - \pi/2 & \text{при } a < 0. \end{cases} \quad (2.25, б)$$

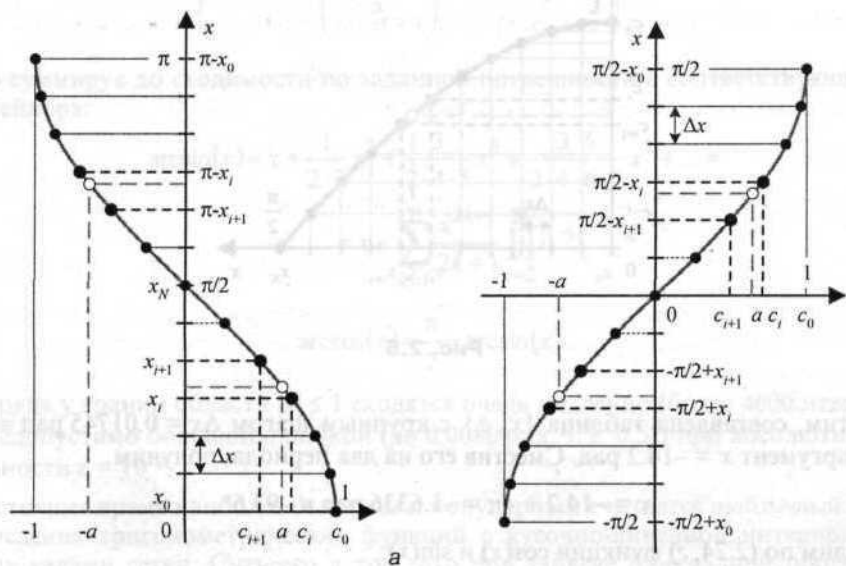


Рис. 2.6

Вычислим по таблице  $\{x_i, c_i\}$ ,  $i = \overline{0, 90}$ , функции  $\arccos(a)$  и  $\arcsin(a)$  для  $a = -0.98$ . Так как  $c_{11} = 0.982$  и  $c_{12} = 0.978$ , то  $i = 11$ . Тогда по (2.25) находим

$$\arccos(a) \approx \pi - L(-a, i) = 2.941445 \text{ рад};$$

$$\arcsin(a) \approx L(-a, i) - \pi/2 = -1.370648 \text{ рад}.$$

Точные значения углов равны, соответственно, 2.941258 рад и -1.370461 рад. Даже при такой крупной сетке, построенной с шагом  $1^\circ$ , максимальная абсолютная погрешность вычисления обратных тригонометрических функций на границах интервала  $-1 < a < 1$  составила  $4.767 \cdot 10^{-4}$  рад или  $0.03^\circ$ .

В заключение приведем примеры *нелинейных элементарных преобразований*.

**Пример 2.1.** Получить матрицы кручения и изгиба вокруг координатных осей.

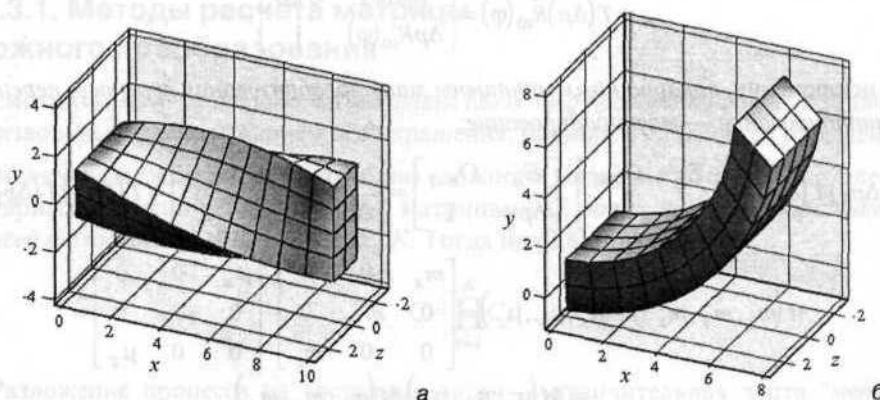
**Решение.** Деформация *кручения* характеризуется линейно возрастающим углом вращения слоев материала, ортогональных оси кручения, при удалении от неподвижного основания. Кручение так же соотносится с вращением, как сдвиг с переносом.

Вывод матрицы кручения вокруг оси  $x$  аналогичен выводу матрицы (2.21, я). Пусть  $\alpha$  — погонный угол кручения, т. е. дополнительное вращение, приходящееся на единицу длины (на рис. 2.7, а изображено кручение параллелепипеда при  $\alpha = 9^\circ$ ). Тогда координатные уравнения и матрица кручения имеют вид

$$\begin{cases} x' = x \\ y' = y \cos(\alpha x) - z \sin(\alpha x) \\ z' = y \sin(\alpha x) + z \cos(\alpha x) \end{cases} \Rightarrow W_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha x) & \sin(\alpha x) \\ 0 & -\sin(\alpha x) & \cos(\alpha x) \end{bmatrix} = R_x(\alpha x).$$

Аналогично получаются матрицы кручения вокруг осей  $y$  и  $z$ .

$$W_y(\beta) = R_y(\beta y), \quad W_z(\gamma) = R_z(\gamma z).$$



**Рис. 2.7**

Кручение превращает прямую  $p_0 + Vi$ , не ортогональную оси кручения, в *винтовую кривую*. Следовательно, это нелинейное преобразование не является аффинным.

В отличие от кручения при деформации *изгиба* угол поворота точки вокруг оси изгиба пропорционален ее удалению. Матрицы преобразования изгиба вокруг координатных осей, зависящие от погонных углов  $\alpha$ ,  $\beta$  и  $\gamma$  (на рис. 2.7, б изображен изгиб параллелепипеда вокруг оси  $z$  с коэффициентом  $\gamma = 4.5^\circ$ ), равны

$$D_x(\alpha) = R_x\left(\alpha\sqrt{y^2 + z^2}\right), \quad D_y(\beta) = R_y\left(\beta\sqrt{x^2 + z^2}\right), \quad D_z(\gamma) = R_z\left(\gamma\sqrt{x^2 + y^2}\right).$$

Изгиб превращает прямую линию  $p_0 + Vt$ , не параллельную оси изгиба, в *спиральную кривую* (см. рис. 1.31, а). Следовательно, и это нелинейное преобразование не является аффинным.

### 2.1.3. Сложные аффинные преобразования

*Сложным* (комбинированным) называется преобразование, реализующее цепочку не менее, чем из двух элементарных преобразований. Отметим, что почти все *аффинные преобразования не коммутативны*, т. е. *результат цепочки преобразований зависит от порядка их выполнения*. Для цепочки из двух АП с матрицами  $C_1$  и  $C_2$  некоммутативность означает  $C_1 C_2 \neq C_2 C_1$ . Например, комбинации вращения объекта вокруг оси  $\omega \in \{x, y, z\}$  на угол  $\varphi$  и переноса на вектор  $p$  имеют разные матрицы сложного АП:

$$\begin{aligned} \tilde{R}_\omega(\varphi)T(\Delta p) &= \begin{bmatrix} R_\omega(\varphi) & O_{n \times 1} \\ O_{1 \times n} & 1 \end{bmatrix} \begin{bmatrix} E_n & O_{n \times 1} \\ \Delta p & 1 \end{bmatrix} = \begin{bmatrix} R_\omega(\varphi) & O_{n \times 1} \\ \Delta p & 1 \end{bmatrix} \neq \\ &\neq T(\Delta p)\tilde{R}_\omega(\varphi) = \begin{bmatrix} R_\omega(\varphi) & O_{n \times 1} \\ \Delta p R_\omega(\varphi) & 1 \end{bmatrix}. \end{aligned}$$

Как исключение, *попарно коммутативны лишь преобразования перенос - перенос и масштабирование - масштабирование*:

$$\begin{aligned} T(\Delta p_1)T(\Delta p_2) &= \begin{bmatrix} E_n & O_{n \times 1} \\ \Delta p_1 & 1 \end{bmatrix} \begin{bmatrix} E_n & O_{n \times 1} \\ \Delta p_2 & 1 \end{bmatrix} = \begin{bmatrix} E_n & O_{n \times 1} \\ \Delta p_1 + \Delta p_2 & 1 \end{bmatrix} = T(\Delta p_2)T(\Delta p_1), \\ M(m_x, m_y, m_z)M(\mu_x, \mu_y, \mu_z) &= \begin{bmatrix} m_x & 0 & 0 \\ 0 & m_y & 0 \\ 0 & 0 & m_z \end{bmatrix} \begin{bmatrix} \mu_x & 0 & 0 \\ 0 & \mu_y & 0 \\ 0 & 0 & \mu_z \end{bmatrix} = \\ &= M(\mu_x, \mu_y, \mu_z)M(m_x, m_y, m_z). \end{aligned}$$

При выполнении преобразований **не теряйте из виду систему координат!** Например, вращения куба вокруг оси  $x$ , а затем вокруг оси  $y$ , *оставшейся неподвижной*, дают матрицу  $R_x R_y$  и конечное положение куба, показанное в конце цепочки 1 - 2 на рис. 2.8. Если забыть направление оси  $y$  и второе вращение куба сделать вокруг *отклоненного ребра*, которое вначале было вертикальным ((цепочка 1 - 3)), то матрица полного АП с учетом вращения системы координат вокруг оси  $x$ , поворота куба и возврата в исходное состояние будет равна

$$R_x(R_x^{-1}R_yR_x) = R_yR_x \neq R_xR_y.$$

Разные конечные положения кубов наглядно доказывают, что преобразования вращения не коммутативны.

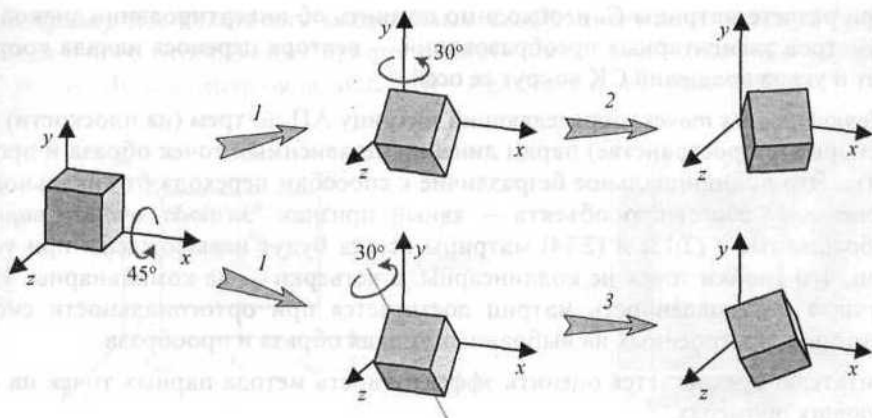


Рис. 2.8

### 2.1.3.1. Методы расчета матрицы сложного преобразования

Рассмотрим два метода расчета матрицы сложного преобразования  $C$  в задачах с произвольным расположением осей вращения, прямых и плоскостей отражения.

*Метод разложения* (декомпозиции) сложного движения объекта на  $N$  элементарных движений, описываемых матрицами  $C_i$  относительно координатных осей и плоскостей неподвижной СК. Тогда по (2.8) получим

$$C = \prod_{i=1}^N C_i.$$

Разложение процесса на составные части — отличительная черта "методов серого ящика", в которых форма (начальное и конечное состояние объекта) наполняется содержанием (промежуточными состояниями и способами их достижения) не единственно возможным образом. Задача и искусство программиста состоит в оптимизации состава ЭП.

Полезным приемом, часто облегчающим выбор ЭП, является временный перевод системы координат матрицей  $C_K$  в такое положение, где оси вращения, прямые и плоскости отражения становятся координатными. В измененной СК выполняются необходимые элементарные АП с общей матрицей  $Q$ . Восстановление системы координат в исходное положение описывается матрицей  $C_K^{-1}$ . Таким образом, сложное АП имеет матрицу

$$C = C_K Q C_K^{-1}.$$

При расчете матрицы  $C_k$  необходимо помнить об инвертировании знаков параметров элементарных преобразований — вектора переноса начала координат и углов вращений СК вокруг ее осей.

*Метод парных точек*, определяющий матрицу АП по трем (на плоскости) или четырем (в пространстве) парам линейно независимых точек образа и прообраза. Это принципиальное безразличие к способам перехода от начального к конечному состоянию объекта — явный признак "метода черного ящика". Обращаемые в (2.13) и (2.14) матрицы всегда будут невырождены при условии, что тройки точек не коллинеарны, а четверки — не компланарны. Наилучшая обусловленность матриц достигается при ортогональности систем векторов, построенных на выбранных точках образа и прообраза.

Читателю предлагается оценить эффективность метода парных точек на следующих примерах.

*Пример 2.2.* Рассчитать матрицу преобразования треугольника  $p_1p_2p_3$  в треугольник  $p_4p_5p_6$  (рис. 2.9).

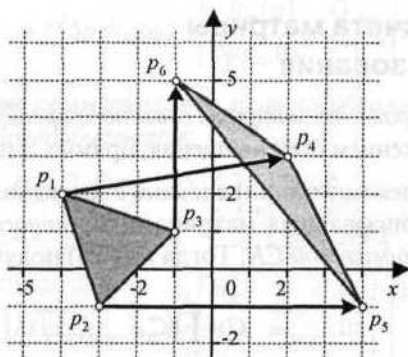


Рис. 2.9

*Решение.* Подставив вершины треугольников в (2.13), сразу получаем необходимую матрицу АП:

$$C = \begin{bmatrix} -4 & 2 & 1 \\ -3 & -1 & 1 \\ -1 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 2 & 3 & 1 \\ 4 & -1 & 1 \\ -1 & 5 & 1 \end{bmatrix} = \begin{bmatrix} -1.375 & 1.25 & 0 \\ -1.125 & 1.75 & 0 \\ -1.25 & 4.5 & 1 \end{bmatrix}.$$

Вот и все. А теперь попытайтесь угадать последовательность и параметры элементарных преобразований, переводящих треугольник  $p_1p_2p_3$  в треугольник  $p_4p_5p_6$ , вычислить матрицы этих преобразований, сравнить их произведение с вышенайденным результатом — и вы почувствуете разницу методов!



Пример 2.3. Рассчитать масштабный коэффициент  $\mu_3$  и матрицу  $C_3$  пропорционального отображения прямоугольного окна  $\{x_{\min} \div x_{\max}, y_{\min} \div y_{\max}\}$  в плоскости  $xy$  МСК в центр окна ЭСК  $\{left \div right, up \div down\}$  (рис 2.10).

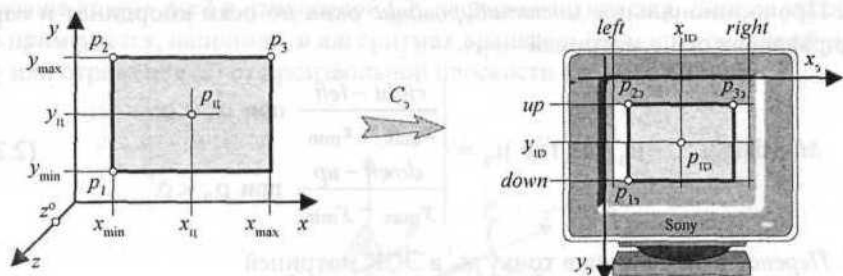


Рис. 2.10

Решение. Обозначим коэффициенты прямоугольности окон в МСК и ЭСК:

$$\rho = \frac{y_{\max} - y_{\min}}{x_{\max} - x_{\min}}, \quad \rho_3 = \frac{down - up}{right - left}$$

У пропорциональных окон коэффициенты прямоугольности  $\rho$  и  $\rho_3$  равны, а окно МСК отображается во все окно ЭСК. В этом случае матрица  $C_3$  наиболее просто находится методом парных точек (2.13) при  $n = 2$ . Хотя задача решается на плоскости, но с целью использования ее результата для преобразований в пространстве выберем четыре пары некопланарных точек в МСК и ЭСК:

$$\begin{aligned} p_1 = [x_{\min} \quad y_{\min} \quad 0] &\rightarrow p_{13} = [left \quad down \quad 0], \\ p_2 = [x_{\min} \quad y_{\max} \quad 0] &\rightarrow p_{23} = [left \quad up \quad 0], \\ p_3 = [x_{\max} \quad y_{\max} \quad 0] &\rightarrow p_{33} = [right \quad up \quad 0], \\ z^0 &\rightarrow z^0_3. \end{aligned}$$

Тогда

$$\left\{ \begin{aligned} C_3 &= \begin{bmatrix} x_{\min} & y_{\min} & 0 & 1 \\ x_{\min} & y_{\max} & 0 & 1 \\ x_{\max} & y_{\max} & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} left & down & 0 & 1 \\ left & up & 0 & 1 \\ right & up & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \\ \mu_3 &= \frac{right - left}{x_{\max} - x_{\min}} = \frac{down - up}{y_{\max} - y_{\min}}. \end{aligned} \right. \quad (2.26)$$

В случае непропорциональных окон получим матрицу  $C_3$  методом разложения, выполнив цепочку элементарных преобразований исходного окна в МСК.

Шаг 1. *Перенос* центра окна  $p_u$  в начало МСК матрицей

$$T_1 = T \left( \begin{bmatrix} -x_u & -y_u & 0 \end{bmatrix} \right), \text{ где } x_u = \frac{x_{\min} + x_{\max}}{2}, y_u = \frac{y_{\min} + y_{\max}}{2}. \quad (2.27, a)$$

Шаг 2. Пропорциональное *масштабирование* окна по осям координат и изменение направления оси  $y$  матрицей

$$M = \text{diag}[\mu_3 \quad -\mu_3 \quad 1], \text{ где } \mu_3 = \begin{cases} \frac{\text{right} - \text{left}}{x_{\max} - x_{\min}} & \text{при } \rho_3 \geq \rho; \\ \frac{\text{down} - \text{up}}{y_{\max} - y_{\min}} & \text{при } \rho_3 < \rho. \end{cases} \quad (2.27, б)$$

Шаг  $T_2 = T \left( \begin{bmatrix} x_{\text{цз}} & y_{\text{цз}} & 0 \end{bmatrix} \right), \text{ где } x_{\text{цз}} = \frac{\text{left} + \text{right}}{2}, y_{\text{цз}} = \frac{\text{up} + \text{down}}{2}. \quad (2.27, в)$

Матрица пол  $C_3 = T_1 \tilde{M} T_2 = \begin{bmatrix} \mu_3 & 0 & 0 & 0 \\ 0 & -\mu_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x_{\text{цз}} - \mu_3 x_u & y_{\text{цз}} + \mu_3 y_u & 0 & 1 \end{bmatrix}. \quad (2.27, г)$

Коэффициенты этой матрицы дают *прямые уравнения пересчета координат точки из МСК в ЭСК* (координата  $z_3$  вектора  $\tilde{p}_3 = \tilde{p} C_3$  при этом не используется):

$$x_3 = x_{\text{цз}} + \mu_3(x - x_u), y_3 = y_{\text{цз}} - \mu_3(y - y_u). \quad (2.28)$$

Описанный алгоритм гарантирует, что любое изображение в прямоугольном окне без искажений отобразится в любом желаемом окне дисплея, заполнив его максимально полностью и симметрично относительно центра.

Если при расчете масштабного коэффициента  $\mu_3$  по (2.27, б) заранее неизвестны границы окна МСК, целиком вмещающего сцену, можно задать значение  $\mu_3 = 40$  из тех соображений, что один сантиметр экрана (40 пикселей) соответствует одной единице длины в пространстве МСК. Наконец, можно выбрать единицу длины в МСК равной размеру пиксела экрана 0.25 мм. Тогда  $\mu_3 = 1$ , что сильно упрощает матрицу (2.27, г) и координатные уравнения (2.28):

$$C_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x_{\text{цз}} - x_u & y_{\text{цз}} + y_u & 0 & 1 \end{bmatrix}, x_3 = x_{\text{цз}} - x_u + x, y_3 = y_{\text{цз}} + y_u - y. \quad (2.29)$$

□

Сравним методы вычисления матрицы сложного АП на примере весьма распространенной операции *изменения пространственной ориентации системы координат*, заключающейся в совмещении начала СК с точкой  $o$ , а какой-либо координатной оси, допустим,  $z$  — с направлением заданного вектора  $V$   $O_3$  (рис. 2.11). Положение других осей в измененной СК не регламентируется. Данное преобразование применяется, например, в алгоритмах вращения (1) вокруг заданной прямой  $\{o, V\}$  или отражения (2) от произвольной плоскости вдоль ее нормали  $V$ .

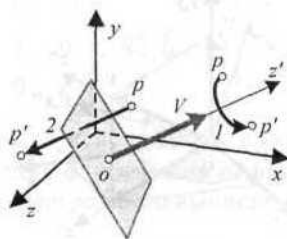


Рис. 2.11

Решение задачи достигается за два преобразования.

1. *Перенос* начала координат в точку  $o$ , описывается матрицей  $T(-o)$ .
2. *Поворот* оси  $z$  в направлении вектора  $V$ , выходящего из начала новой СК.

Обозначим нерасширенную матрицу этого преобразования  $A(V)$ .

Расширенная матрица полного преобразования СК равна

$$C(o, V) = T(-o) \tilde{A}(V).$$

Рассмотрим различные методы вычисления матрицы поворота  $A(V)$ .

### Метод разложения

Алгоритм, основанный на методе разложения, рассматривается во многих учебниках по машинной графике [5, 26, 33, 34] и состоит в нахождении проекции  $v = [V_x \ V_z]$  вектора  $V$  на координатную плоскость  $xz$  (у разных авторов варианты алгоритма отличаются выбором плоскости проецирования) и выполнении *двух вращений СК*  $xuz$  (рис. 2.12):

- первое вращение вокруг оси  $y$  на угол  $\alpha$ , у которого

$$\cos(\alpha) = \frac{V_z}{|v|}, \quad \sin(\alpha) = \frac{V_x}{|v|}, \quad \text{где } |v| = \sqrt{V_x^2 + V_z^2}.$$

Система координат после этого занимает положение  $x'y'z'$ , в котором ось  $z'$  совпадает с направлением вектора  $v$ ;

- второе вращение вокруг *новой* оси  $x'$  на угол  $-\beta$ , у которого

$$\cos(\beta) = \frac{|v|}{|V|}, \quad \sin(\beta) = \frac{V_y}{|V|}.$$

Система координат после этого занимает требуемое положение  $x''y''z''$ , в котором ось  $z''$  совпадает с направлением вектора  $V$ .

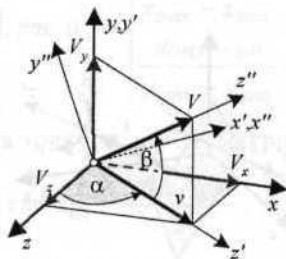


Рис. 2.12

Таким образом, матрица поворота СК по методу разложения равна

$$A_p(V) = R_y(-\alpha) R_x(\beta). \quad (2.30)$$

Всегда ли верно это решение? Очевидно, что оно не является таковым при  $|v| = 0$ , когда  $V \parallel y^0$ . В этом случае угол  $\alpha$  не определен, а совмещение орта  $z^0$  с вектором  $V$  достигается не за два, а за *одно вращение* СК матрицей

$$A_p(V) = R_x(\text{sgn}(V_x)90^\circ).$$

Однако, и при  $V \neq y^0$  у метода (2.30) имеются скрытый *недостаток* и *особенности* программной реализации, которые даже не обозначены и не обсуждаются в перечисленных выше учебниках:

- при расчете матриц вращения по (2.21) возможно использование неверных аргументов, вычисленных с помощью функции  $\arccos$  в интервале  $[0^\circ, 180^\circ]$ , в то время как углы изменяются в интервалах

$$-180^\circ \leq \alpha \leq 180^\circ, \quad -90^\circ \leq \beta \leq 90^\circ.$$

Так же некорректно определение угла  $\alpha$  (но не  $\beta$ !) функцией  $\arcsin$  в интервале  $[-90^\circ, 90^\circ]$ . Следовательно, угол  $\alpha$  должен вычисляться с учетом знака его синуса, т. е. функцией (1.31) с векторными аргументами  $[V_x \ V_y]$  и  $[0 \ 1]$ . И так, в (2.30) углы вращений находятся следующим образом:

$$\alpha = \text{if} \left( |v|: \text{ang}(v, y^0), \ 0 \right), \quad \beta = \arcsin \left( \frac{V_y}{|V|} \right); \quad (2.31)$$

- поскольку матрицам вращения требуются, собственно, не сами углы, а их косинусы и синусы, которые в рассматриваемой задаче явно выражены через элементы вектора  $V$ , то практичнее создать *модификации матриц вращения* (2.20) и (2.21) с двумя аргументами  $c$  и  $s$  — косинусом и синусом угла вращения:

$$R2(c,s) = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}, R2_x(c,s) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c & s \\ 0 & -s & c \end{bmatrix}, \tag{2.32}$$

$$R2_y(c,s) = \begin{bmatrix} c & 0 & -s \\ 0 & 1 & 0 \\ s & 0 & c \end{bmatrix}, R2_z(c,s) = \begin{bmatrix} c & s & 0 \\ -s & c & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Эти матрицы рекомендуется использовать даже в тех случаях, когда исходным является угол вращения  $\varphi$ : вычислив *один раз*  $c = \cos(\varphi)$  и  $s = \sin(\varphi)$ , подставить их в (2.32) вместо *двукратного* вычисления в (2.20) и (2.21);

- прямая подстановка

$$\cos(\alpha) = \frac{V_z}{|V|}, \sin(\alpha) = \frac{V_x}{|V|}, \cos(\beta) = \frac{|V|}{|V|}, \sin(\beta) = \frac{V_y}{|V|}$$

в матрицы вращений (2.30) и их *ручное* перемножение приводит к весьма громоздкой записи матрицы  $A_p(V)$ , которую автор здесь сознательно не приводит, дабы не искушать читателя этим *порочным* методом разработки алгоритмов компьютерной графики. Во-первых, в процессе ручных выводов очень легко наделать ошибок. Во-вторых, вместо элегантного перемножения матриц в одной строке программы поэлементное описание матрицы  $A_p(V)$  требует много строк — это путь к разбуханию программного кода, источник дополнительных ошибок и затрат времени на отладку программы.

Матрицы некоторых сложных преобразований на плоскости, вычисленные методом разложения, приведены в табл. 2.1.

Таблица 2.1

Преобразование	Матрица преобразования
Масштабирование от точки $o$	$T(-o)\tilde{M}(m_x, m_y)T(o) = \begin{bmatrix} m_x & 0 & 0 \\ 0 & m_y & 0 \\ (1-m_x)o_x & (1-m_y)o_y & 1 \end{bmatrix}$
Отражение от точки $o$	$T(-o)\tilde{M}(-1,-1)T(o) = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 2o_x & 2o_y & 1 \end{bmatrix}$

Таблица 2.1 (продолжение)

Преобразование	Матрица преобразования
Отражение от прямой $\{o, V\}$ $\cos(\alpha) = \frac{V_y}{ V }, \sin(\alpha) = \frac{-V_x}{ V }$	$T(-o)\tilde{R}(-\alpha)\tilde{M}(-1,1)\tilde{R}(\alpha)T(o) =$ $= \begin{bmatrix} 2\sin^2(\alpha) - 1 & -2\cos(\alpha)\sin(\alpha) & 0 \\ -2\cos(\alpha)\sin(\alpha) & 1 - 2\sin^2(\alpha) & 0 \\ X & Y & 1 \end{bmatrix}$ $X = 2\cos(\alpha)(o_x \cos(\alpha) + o_y \sin(\alpha))$ $Y = 2\sin(\alpha)(o_x \cos(\alpha) + o_y \sin(\alpha))$
Отражение от прямой $y = ax + b$ $o = by^0,$ $\alpha = \arctg(a),  a  < \infty$	$T(-by^0)\tilde{R}(-\alpha)\tilde{M}(1,-1)\tilde{R}(\alpha)T(by^0) =$ $= \begin{bmatrix} (1-a^2)/(1+a^2) & 2a/(1+a^2) & 0 \\ 2a/(1+a^2) & (a^2-1)/(1+a^2) & 0 \\ -2ab/(1+a^2) & 2b/(1+a^2) & 1 \end{bmatrix}$
Отражение от прямой $x = a$	$T(-ax^0)\tilde{M}(-1, 1)T(ax^0) = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 2a & 0 & 1 \end{bmatrix}$
Отражение от прямой $y = b$	$T(-by^0)\tilde{M}(1, -1)T(by^0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 2b & 1 \end{bmatrix}$
Отражение от прямой $y = x$	$\tilde{R}(45^\circ)\tilde{M}(-1, 1)\tilde{R}(-45^\circ) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
Отражение от прямой $y = -x$	$\tilde{R}(-45^\circ)\tilde{M}(-1, 1)\tilde{R}(45^\circ) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
Вращение вокруг точки $o$ на угол $\varphi$	$T(-o)\tilde{R}(\varphi)T(o) = \begin{bmatrix} \cos(\varphi) & \sin(\varphi) & 0 \\ -\sin(\varphi) & \cos(\varphi) & 0 \\ X & Y & 1 \end{bmatrix}$ $X = o_x(1 - \cos(\varphi)) + o_y \sin(\varphi)$ $Y = o_y(1 - \cos(\varphi)) - o_x \sin(\varphi)$

Таблица 2.1 (окончание)

Преобразование	Матрица преобразования
Сдвиг вдоль прямой $\{o, V\}$ $\cos(\alpha) = \frac{V_y}{ V }, \sin(\alpha) = \frac{-V_x}{ V }$	$T(-o)\tilde{R}(-\alpha) \begin{bmatrix} 1 & 0 & 0 \\ s_{xy} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tilde{R}(\alpha)T(o)$

Матрицы сложных преобразований в пространстве приведены в табл. 2.2.

Таблица 2.2

Преобразование	Матрица преобразования
Масштабирование от точки $o$	$T(-o)\tilde{M}(m_x, m_y, m_z)T(o) =$ $= \begin{bmatrix} m_x & 0 & 0 & 0 \\ 0 & m_y & 0 & 0 \\ 0 & 0 & m_z & 0 \\ (1-m_x)o_x & (1-m_y)o_y & (1-m_z)o_z & 1 \end{bmatrix}$
Отражение от точки $o$	$T(-o)\tilde{M}(-1, -1, -1)T(o) = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 2o_x & 2o_y & 2o_z & 1 \end{bmatrix}$
Отражение от прямой $\{o, V\}$ $\cos(\alpha) = \frac{V_z}{v}, \sin(\alpha) = \frac{V_x}{v},$ $\cos(\beta) = \frac{v}{ V }, \sin(\beta) = \frac{V_y}{ V },$ $v = \sqrt{V_x^2 + V_z^2}$	$V \# y^0:$ $T(-o)\tilde{R}_y(-\alpha)\tilde{R}_x(\beta)\tilde{M}(-1, -1, 1)\tilde{R}_x(-\beta)\tilde{R}_y(\alpha)T(o)$
	$V \parallel y^0:$ $T(-o)\tilde{M}(-1, 1, -1)T(o) = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 2o_x & 0 & 2o_z & 1 \end{bmatrix}$

Таблица 2.2 (продолжение)

Преобразование	Матрица преобразования
	$V \parallel x^0:$ $T(-o)\tilde{M}(1, -1, -1)T(o) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 2o_y & 2o_z & 1 \end{bmatrix}$ $V \parallel z^0:$ $T(-o)\tilde{M}(-1, -1, 1)T(o) = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 2o_x & 2o_y & 0 & 1 \end{bmatrix}$
<p>Отражение от плоскости <math>\{o, N\}</math></p> $\cos(\alpha) = \frac{N_z}{v}, \quad \sin(\alpha) = \frac{N_x}{v},$ $\cos(\beta) = \frac{v}{ N }, \quad \sin(\beta) = \frac{N_y}{ N },$ $v = \sqrt{N_x^2 + N_z^2}$	$N \# y^0:$ $T(-o)\tilde{R}_y(-\alpha)\tilde{R}_x(\beta)\tilde{M}(1, 1, -1)\tilde{R}_x(-\beta)\tilde{R}_y(\alpha)T(o)$
	$N \parallel y^0:$ $T(-o)\tilde{M}(1, -1, 1)T(o) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 2o_y & 0 & 1 \end{bmatrix}$
	$N \parallel x^0:$ $T(-o)\tilde{M}(-1, 1, 1)T(o) = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 2o_x & 0 & 0 & 1 \end{bmatrix}$



Таблица 2.2 (продолжение)

Преобразование	Матрица преобразования
	$N \parallel z^0:$ $T(-o)\tilde{M}(1, 1, -1)T(o) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 2o_z & 1 \end{bmatrix}$
<p>Вращение вокруг прямой <math>\{o, V\}</math> на угол <math>\varphi</math></p> <p><math>\cos(\alpha) = \frac{V_z}{v}, \sin(\alpha) = \frac{V_x}{v},</math></p> <p><math>\cos(\beta) = \frac{v}{ V }, \sin(\beta) = \frac{V_y}{ V },</math></p> <p><math>v = \sqrt{V_x^2 + V_z^2}, s_o = \text{sgn}(V_o)</math></p>	$V \# y^0:$ $T(-o)\tilde{R}_y(-\alpha)\tilde{R}_x(\beta)\tilde{R}_z(\varphi)\tilde{R}_x(-\beta)\tilde{R}_y(\alpha)T(o)$
	$V \parallel y^0:$ $T(-o)\tilde{R}_y(s_y, \varphi)T(o) =$ $= \begin{bmatrix} \cos(\varphi) & 0 & -s_y \sin(\varphi) & 0 \\ 0 & 1 & 0 & 0 \\ s_y \sin(\varphi) & 0 & \cos(\varphi) & 0 \\ o_x(1 - \cos(\varphi)) - & 0 & o_z(1 - \cos(\varphi)) + & 1 \\ -s_y o_z \sin(\varphi) & & + s_y o_x \sin(\varphi) & \end{bmatrix}$ $V \parallel x^0:$ $T(-o)\tilde{R}_x(s_x, \varphi) \dots T(o) =$ $= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\varphi) & s_x \sin(\varphi) & 0 \\ 0 & -s_x \sin(\varphi) & \cos(\varphi) & 0 \\ 0 & o_y(1 - \cos(\varphi)) + & o_z(1 - \cos(\varphi)) - & 1 \\ + s_x o_z \sin(\varphi) & & - s_x o_y \sin(\varphi) & \end{bmatrix}$

Таблица 2.2 (окончание)

Преобразование	Матрица преобразования
	$V \parallel z^0:$ $T(-\alpha)\tilde{R}_z(s_z\varphi)T(\alpha) =$ $= \begin{bmatrix} \cos(\varphi) & s_z \sin(\varphi) & 0 & 0 \\ -s_z \sin(\varphi) & \cos(\varphi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ +s_z o_y \sin(\varphi) & -s_z o_x \sin(\varphi) & 0 & 1 \end{bmatrix}$

### Экономичный метод разложения

*Экономичный вариант* метода разложения заключается в вычислении вектора нормали  $N = z^0 \times V \in xy$  и двух вращений системы координат (рис. 2.13):

- на угол  $\alpha$  вокруг оси  $z$ ;
- *кратчайшего* вращения на угол  $\beta$  вокруг новой оси  $y'$ .

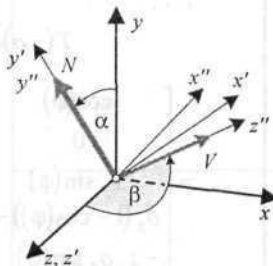


Рис. 2.13

Судя по интервалам изменения углов  $-180^\circ < \alpha < 180^\circ$  и  $0^\circ < \beta < 180^\circ$ , формулы их расчета и матрицы поворота следующие:

$$\alpha = \text{if}(|N|: \text{ang}(y^0, N), 0), \beta = \arccos(z^0 \circ \bar{V}), A_{\text{sp}}(V) = R_z(-\alpha) R_{y'}(-\beta). \quad (2.33)$$

### Метод парных точек

Для получения 3 x 3-матрицы  $A(V)$  **методом парных точек** найдем в пространстве *три* неколлинеарные точки, в которые перейдут концы ортов  $x^0$ ,  $y^0$  и  $z^0$  после

поворота системы координат (рис. 2.14). Одна из этих точек на конце вектора  $V$  — прообраз  $z^0$ . В общем случае при  $V \neq z^0$  выберем прообразы векторов  $y^0$  и  $x^0$  на концах векторов нормали  $N = z^0 \times V \in xy$  и бинормали  $W = N \times \bar{V}$ . Длины векторов  $N$  и  $W$  равны — это необходимо для неискаженного преобразования углов из СК  $x'y'z'$  в СК  $xuz$ . Подставляя точки образа  $\{x^0, y^0, z^0\}$  и прообраза  $\{W, N, V\}$  в нерасширенный вариант (2.14), получим

$$A_n(V) = \begin{bmatrix} W \\ N \\ V \end{bmatrix}^{-1} \begin{bmatrix} x^0 \\ y^0 \\ z^0 \end{bmatrix} = \begin{bmatrix} W \\ N \\ V \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} N \times \bar{V} \\ N \\ V \end{bmatrix}^{-1}.$$

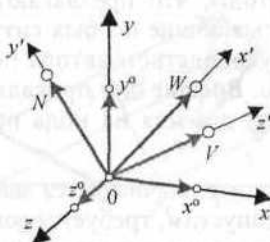


Рис. 2.14

Особый случай — параллельность векторов  $V \parallel z^0$ , при которой  $z^0 \times V = O_{z^0}$ . Специальный выбор векторов

$$N = V_z y^0, \quad W = N \times \bar{V} = |V_z| x^0$$

обеспечивает ортогональность, равномерность и правую ориентацию тройки  $\{W, N, V\}$  базисных векторов системы  $x'y'z'$  как при  $V \uparrow \uparrow z^0$ , так и при  $V \uparrow \downarrow z^0$ . Эти три свойства необходимы для правильного отображения в СК  $xuz$  как величины, так и знака угла вращения, выполняемого в СК  $x'y'z'$ .

Обобщая сказанное, запишем алгоритм расчета матрицы поворота системы координат методом парных точек в следующем виде:

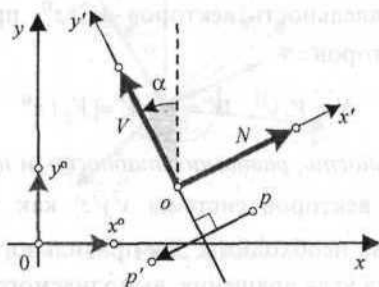
$$N = \text{if} \left( |z^0 \circ \bar{V}| < 1; z^0 \times V, V_z y^0 \right), \quad A_n(V) = \begin{bmatrix} N \times \bar{V} \\ N \\ V \end{bmatrix}^{-1}. \quad (2.34, a)$$

Метод парных точек позволяет сразу найти матрицу полного преобразования системы координат, равную  $T(-o)A_p(V)$ , без разложения на промежуточные преобразования:

$$N = \text{if} \left( \left| z^o \circ \bar{V} \right| < 1: z^o \times V, V_z y^o \right), C_n(o, V) = \begin{bmatrix} o + N \times \bar{V} & 1 \\ o + N & 1 \\ o + V & 1 \\ o & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.34, \theta)$$

Сравнение методов (2.30), (2.33) и (2.34) по критериям понятности, простоты вывода, краткости записи, вычислительной эффективности ставит впереди всех, по мнению автора, *метод парных точек*. К сожалению, все рассмотренные методы страдают условностью, т. е. зависимостью алгоритмов от особых случаев ориентации вектора  $V$ . Видимо, единый алгоритм для произвольных положений этого вектора отсутствует, а те методы, что предлагаются в учебной литературе — лишь частные случаи, не учитывающие особых ситуаций. Опыт курсового проектирования студентов под руководством автора показывает, что эти ситуации встречаются не так уж и редко. Внешне они проявляются во внезапном завершении программы из-за попытки деления на ноль при обращении вырожденной матрицы.

Гораздо проще, чем в 3d-варианте, вычисляется *матрица преобразования системы координат на плоскости*. Допустим, требуется совместить начало координат с точкой  $o$ , а оси  $y$  придать направление вектора  $V$  (рис. 2.15). Такое преобразование полезно делать при расчете отражения и масштабирования относительно произвольной прямой  $\{o, V\}$ .



Приведем без лишних рассуждений очевидные формулы расчета матрицы преобразования системы координат  $C_x$ :

- по *методу разложения*

$$C_x = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x^o & y^o \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1}. \quad (2.35)$$

Рис. 2.15

• по методу парных точек

$$N = VR(-90^\circ) = [V_y \quad -V_x], \quad C_{\Pi}(o, V) = \begin{bmatrix} o+N & 1 \\ o+V & 1 \\ o & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.36)$$

**Пример 2.4.** Построить отражение треугольника на плоскости с вершинами  $p_1 = [0 \ 0]$ ,  $p_2 = [0 \ 1]$  и  $p_3 = [2 \ 0]$  от прямой линии, проходящей через точки  $p_4 = [3 \ 0]$  и  $p_5 = [2 \ 2]$  (рис. 2.16).

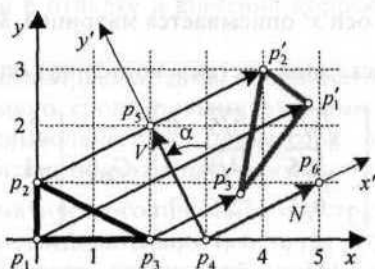


Рис. 2.16

**Решение.** Вычислим направляющий вектор и нормаль заданной прямой линии:

$$V = p_5 - p_4 = [-1 \ 2], \quad N = VR(-90^\circ) = [2 \ 1].$$

Последовательность действий для построения отражений  $p'_i$  исходных точек  $p_i$  ( $i = 1, 2, 3$ ) может быть, например, следующей.

Шаг 1. Получим по (2.35) матрицу преобразования системы координат

$$C_{\text{кр}} = T(-p_4)\tilde{R}(-\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -3 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sqrt{4/5} & -\sqrt{1/5} & 0 \\ \sqrt{1/5} & \sqrt{4/5} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \sqrt{4/5} & -\sqrt{1/5} & 0 \\ \sqrt{1/5} & \sqrt{4/5} & 0 \\ -\sqrt{36/5} & -\sqrt{9/5} & 1 \end{bmatrix}$$

методом разложения на два элементарных преобразования: перенос начала координат на вектор  $p_4$  и вращение на угол  $\alpha$ , у которого

$$\cos(\alpha) = \frac{V_y}{|V|} = \sqrt{4/5}, \quad \sin(\alpha) = \frac{-V_x}{|V|} = \sqrt{1/5}.$$

Второй вариант матрицы преобразования системы координат получим методом парных точек (2.35) по двум тройкам точек  $\{x^0, y^0, O_2\}$  и  $\{p_6, p_5, p_4\}$ :

$$C_{\text{кп}} = \begin{bmatrix} 5 & 1 & 1 \\ 2 & 2 & 1 \\ 3 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 5.4 & -0.2 & 0 \\ -0.2 & 0.4 & 0 \\ -1.2 & 0.6 & 1 \end{bmatrix}.$$

Читателя не должно смущать несовпадение полученных матриц: видимо, матрица  $C_{\text{кр}}$  эквивалентна иной, чем в матрице  $C_{\text{кп}}$ , последовательности элементарных преобразований, которые, по сути дела, нас не интересуют. Главная цель — получение матрицы преобразования системы координат — достигнута с помощью метода парных точек быстрее и проще.

**Шаг 2.** Отражение вдоль оси  $x'$  описывается матрицей  $M = \text{diag}[-1 \ 1]$ .

**Шаг 3.** Восстановление исходной системы координат описывается матрицами

$$C_{\text{кр}}^{-1} = \begin{bmatrix} \sqrt{4/5} & \sqrt{1/5} & 0 \\ -\sqrt{1/5} & \sqrt{4/5} & 0 \\ 9/5 & 12/5 & 1 \end{bmatrix}, \quad C_{\text{кп}}^{-1} = \begin{bmatrix} 2 & 1 & 0 \\ -1 & 2 & 0 \\ 3 & 0 & 1 \end{bmatrix}.$$

**Шаг 4.** Матрица сложного АП

$$C = C_{\text{кр}} \tilde{M} C_{\text{кр}}^{-1} = C_{\text{кп}} \tilde{M} C_{\text{кп}}^{-1} = \begin{bmatrix} -0.6 & -0.8 & 0 \\ -0.8 & 0.6 & 0 \\ 4.8 & 2.4 & 1 \end{bmatrix}$$

одинакова для обоих методов расчета матрицы изменения СК, что подтверждает вывод о неединственности способа выполнения сложного преобразования.

**Шаг 5.** Координаты отраженных точек рассчитаем групповым методом (2.8):

$$\begin{bmatrix} p'_1 & 1 \\ p'_2 & 1 \\ p'_3 & 1 \end{bmatrix} = \begin{bmatrix} p_1 & 1 \\ p_2 & 1 \\ p_3 & 1 \end{bmatrix} C = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} -0.6 & -0.8 & 0 \\ -0.8 & 0.6 & 0 \\ 4.8 & 2.4 & 1 \end{bmatrix} = \begin{bmatrix} 4.8 & 2.4 & 1 \\ 4 & 3 & 1 \\ 3.6 & 0.8 & 1 \end{bmatrix}. \quad \square$$

### 2.1.3.2. Кинематический метод построения объектов

Аффинные преобразования — основа *кинематического метода построения параметрических объектов* (кривых и поверхностей), генерирующего состояния объекта из заданных начальных состояний точки, отрезка, окружности как функции некоторого параметра, имеющего физический смысл времени движения. Конечным результатом преобразований являются не уравнения объекта, а его изображение на экране. Построение изображения эффективнее выполнять не по сложным аналитическим уравнениям, а непосредственно по обобщенной векторной