Модуль 2 «Математические методы, модели и алгоритмы компьютерной геометрии»
Лекция 6 «Двухмерные геометрические преобразования»

к.ф.-м.н., доц. каф. ФН-11, Захаров Андрей Алексеевич,
ауд.:930а(УЛК)
моб.: 8-910-461-70-04,
email: azaharov@bmstu.ru

МГТУ им. Н.Э. Баумана

16 октября 2023 г.

Введение

Преобразованием на плоскости f будем называть любое отображение плоскости на себя $f : \mathbb{R}^2 \to \mathbb{R}^2$. Преобразование называется *линейным*, если для любых двух векторов $\mathbf{u}$ и $\mathbf{v}$ и любого вещественного α справедливы следующие равенства:

$$f(\mathbf{u} + \mathbf{v}) = f(\mathbf{u}) + f(\mathbf{v}); \qquad f(\alpha\mathbf{u}) = \alpha f(\mathbf{u}).$$

Для любого линейного преобразования всегда найдется такая матрица, что данное преобразование можно представить как умножение вектора координат на эту матрицу, т.е. $f(\mathbf{u}) = \mathbf{A}\mathbf{u}$.
Если для всех точек плоскости $f(g(\mathbf{u})) = g(f(\mathbf{u})) = \mathbf{u}$, то преобразование g называется *обратным* к преобразованию f и обозначается как f^{-1}. Само преобразование в этом случае называется *обратимым*.
Произвольное линейное преобразование *невырожденно* тогда и только тогда, когда его матрица обратима. При этом обратное преобразование также является линейным и в качестве его матрицы выступает матрица, обратная к матрице исходного преобразования.
Примерами линейных преобразований являются преобразования масштабирования, вращения и сдвига. Для них непосредственно можно записать матрицу преобразования $\mathbf{A}$.

Введение

Преобразование называется *аффинным*, если оно имеет следующий вид:

$$f(\mathbf{u}) = \mathbf{A}\mathbf{u} + \mathbf{v},$$

где $\mathbf{A}$ — матрица и $\mathbf{v}$ — вектор.
Примером аффинного преобразования является преобразование перемещения. Для преобразования перемещения матрицу преобразования можно записать только вводя дополнительную размерность пространства и перейдя к однородным координатам.

Перемещение

Перемещение отдельной точки выполняется путём добавления постоянных смещений к её координатам, в результате чего получается точка с новыми координатами. По сути, исходная точка перемещается по прямой линии в новое положение.
Чтобы переместить точку в двухмерном пространстве, к исходным координатам (x, y) добавляются приращения t_x и t_y, в результате чего получается новая точка с координатами (x', y'):

$$x' = x + t_x, \qquad y' = y + t_y. \tag{1}$$

Пара приращений (t_x, t_y) называется *вектором перемещения*.

Перемещение

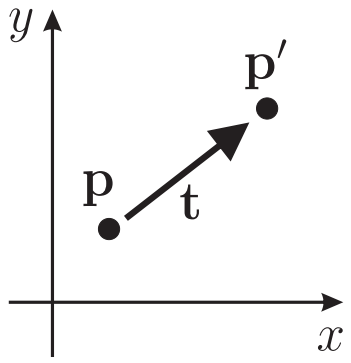


Рис.: Перемещение точки из положения $\mathbf{p}$ в положение $\mathbf{p}'$ с использованием вектора перемещения $\mathbf{t}$

Уравнения перемещения (1) можно записать как одно матричное уравнение, в котором будут фигурировать приведённые ниже векторы-столбцы, представляющие координаты точек и вектор перемещения:

$$\mathbf{p} = \begin{bmatrix} x \\ y \end{bmatrix}, \qquad \mathbf{p}' = \begin{bmatrix} x' \\ y' \end{bmatrix}, \qquad \mathbf{t} = \begin{bmatrix} t_x \\ t_y \end{bmatrix}. \tag{2}$$

Такие обозначения позволяют записать уравнения двухмерного перемещения в матричной форме:

$$\mathbf{p}' = \mathbf{p} + \mathbf{t}. \tag{3}$$

Перемещение

Перемещение является *жёстким преобразованием*, передвигающим объекты без деформации. То есть все точки объекта перемещаются на одну и ту же величину. Чтобы переместить отрезок прямой, преобразование (3) применяется к обеим концам отрезка, после чего между ними проводится линия. Многоугольник перемещается аналогично: к каждой вершине прибавляется вектор перемещения, а затем многоугольник строится заново с использованием нового набора вершин. Чтобы изменить положение круга или эллипса, перемещаются координаты центра, а затем фигура перерисовывается в новом положении. Для сплайновой кривой перемещаются точки, определяющие её траекторию, а затем восстанавливаются точки, лежащие между ними.

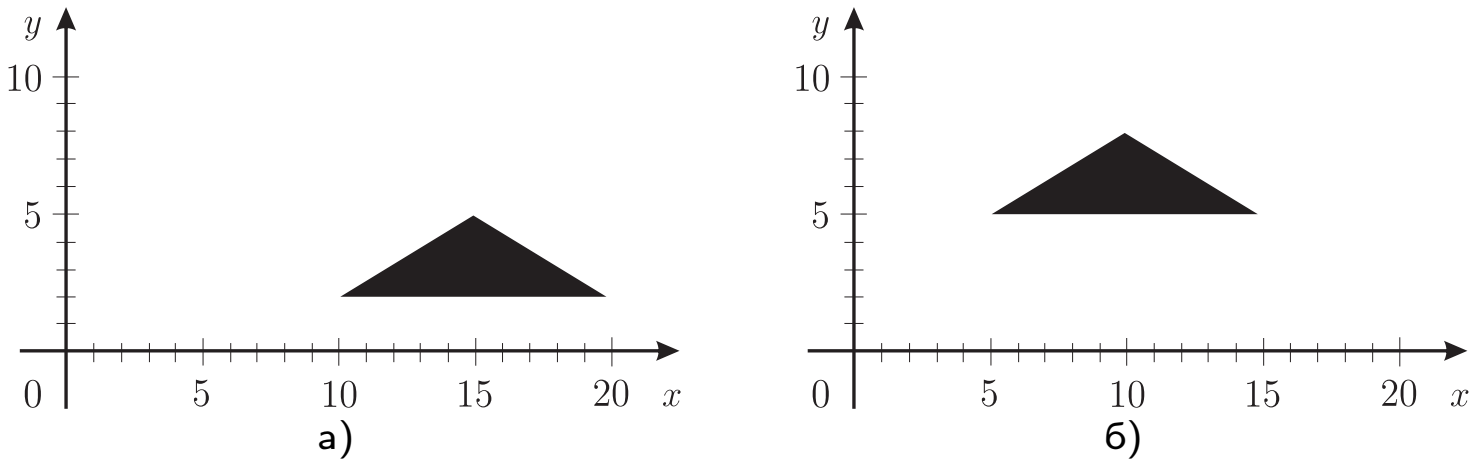


Рис.: Перемещение треугольника из положения а) в положение б) с использованием вектора перемещения (-5;3)

Поворот

Двухмерный поворот объекта — это перемещение объекта по круговой траектории на плоскости xy. В этом случае объект поворачивается относительно оси вращения, перпендикулярной плоскости xy (параллельной координатной оси z). Параметрами двухмерного поворота являются угол поворота θ и точка (x_r, y_r), называемая *центром поворота* (или *центром вращения*), вокруг которой поворачивается объект. Центр поворота является точкой пересечения оси вращения с плоскостью xy. Положительный угол θ определяет направление против часовой стрелки вокруг центра поворота, а при отрицательном угле объект поворачивается по часовой стрелке.

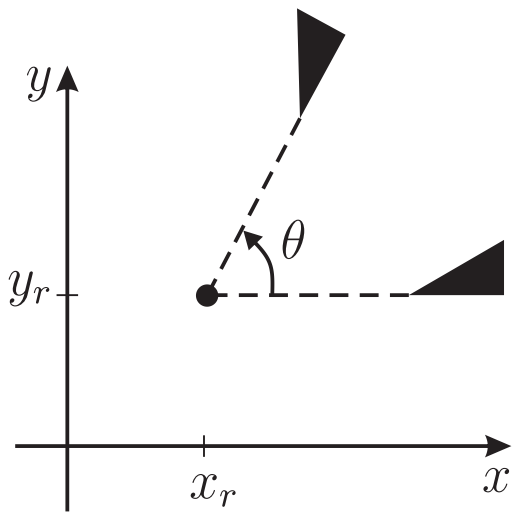
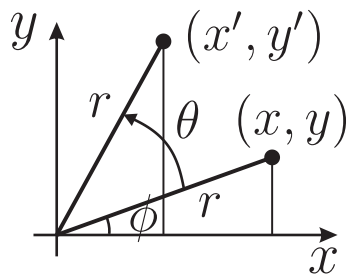


Рис.: Поворот треугольника на угол θ вокруг центра вращения (x_r, y_r)

Поворот



Рассмотрим вначале вращение точки $\mathbf{p}$, когда центром вращения является начало координат. Используя стандартные тригонометрические тождества, преобразованные координаты можно выразить через углы θ и ϕ:

$$\begin{aligned} x' &= r\cos(\phi+\theta) = r\cos(\phi)\cos(\theta) - r\sin(\phi)\sin(\theta); \\ y' &= r\sin(\phi+\theta) = r\cos(\phi)\sin(\theta) + r\sin(\phi)\cos(\theta). \end{aligned} \tag{4}$$

Исходные полярные координаты точки равны: $x = r\cos(\phi)$, $y = r\sin(\phi)$. Подставляя эти выражения в (4), получаем уравнения преобразования для поворота точки с координатами (x, y) на угол θ вокруг начала координат:

$$x' = x\cos(\theta) - y\sin(\theta); \qquad y' = x\sin(\theta) + y\cos(\theta). \tag{5}$$

Используя представления через векторы-столбцы (2), уравнения поворота можно записать в матричной форме: $\mathbf{p}' = \mathbf{R}\cdot\mathbf{p}$, где матрица поворота имеет такой вид:

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \tag{6}$$

Комплексные числа

Сделаем краткий экскурс в комплексную математику.
Напомним, что комплексная пара (a, b) определяет число $a + bi$, где a называется действительной частью, b — мнимой частью, а i это так называемая мнимая единица, которая удовлетворяет соотношению $i^2 = -1$. Любое действительное число k может быть выражено в виде комплексного числа $(k, 0) = k + 0i$. Комплексные числа можно складывать, вычитать, и умножать следующим образом:

$$\begin{aligned}(a + bi) + (c + di) &= (a + c) + (b + d)i,\\ (a + bi) - (c + di) &= (a - c) + (b - d)i,\\ (a + bi)(c + di) &= ac + adi + bci + bdi^2 = ac + (ad + bc)i + bd(-1)\\ &= (ac - bd) + (ad + bc)i.\end{aligned} \tag{7}$$

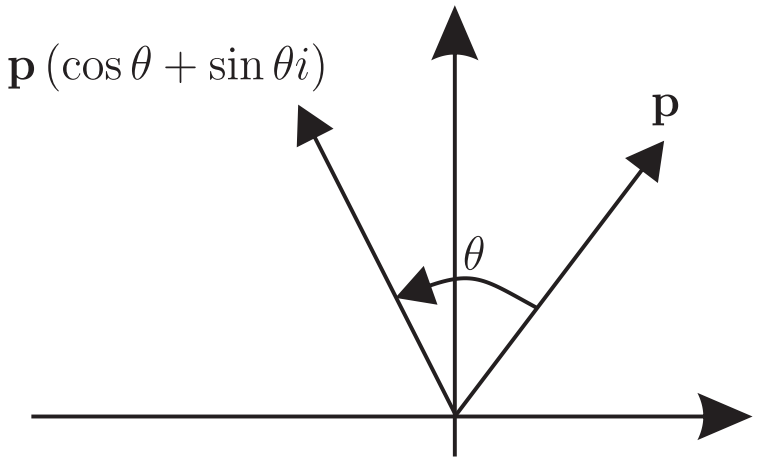
Комплексно-сопряжённое число вычисляется путём умножения мнимой части на -1:

$$\mathbf{p} = a + bi, \qquad \mathbf{p}^* = a - bi. \tag{8}$$

Модуль комплексного числа равен:

$$\begin{aligned}\|\mathbf{p}\| &= \sqrt{\mathbf{p}\mathbf{p}^*},\\ \|(a + bi)\| &= \sqrt{(a + bi)(a - bi)} = \sqrt{a^2 + b^2}.\end{aligned} \tag{9}$$

Повороты с помощью комплексных чисел



Множество комплексных чисел «живет» в 2D плоскости. Мы можем представить себе, что эта плоскость имеет две оси: действительная и мнимая. Т.о., мы интерпретируем комплексное число (x, y) в качестве 2D вектора. Когда комплексные числа интерпретируются таким образом, они могут быть использованы для выполнения поворота в плоскости. Пусть комплексное число $\mathbf{p}$ поворачивается вокруг начала координат на угол θ. Чтобы выполнить это вращение, определим ещё одно комплексное число $\mathbf{r} = (\cos\theta, \sin\theta)$. Повернутый вектор $\mathbf{p}'$ может быть вычислен с помощью комплексного умножения:

$$\begin{aligned}\mathbf{p}' &= \mathbf{p} \cdot \mathbf{r} = (x + yi)(\cos\theta + (\sin\theta)i) \\ &= (x\cos\theta - y\sin\theta) + (x\sin\theta + y\cos\theta)i\end{aligned}$$

Это выражение является аналогом (5).

Поворот

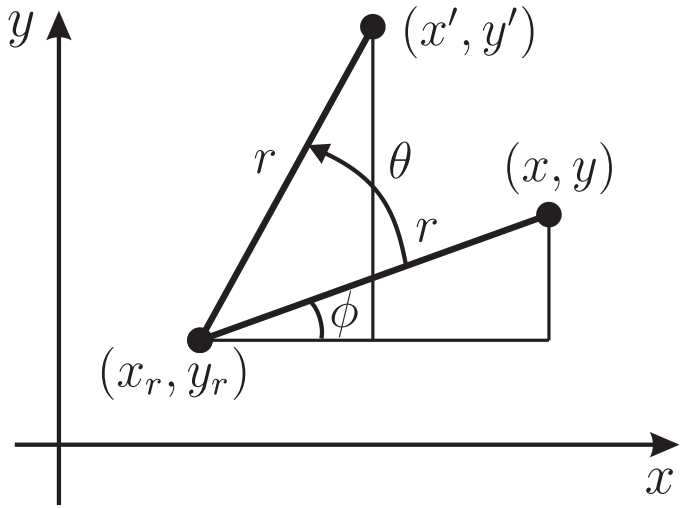


Рис.: Поворот точки (x, y) в точку (x', y') на угол θ вокруг точки (x_r, y_r)

Уравнения (5) можно обобщить и получить формулы преобразования, описывающие поворот точки вокруг любого заданного центра поворота (x_r, y_r):

$$\begin{aligned} x' &= x_r + (x - x_r)\cos(\theta) - (y - y_r)\sin(\theta); \\ y' &= y_r + (x - x_r)\sin(\theta) + (y - y_r)\cos(\theta). \end{aligned} \tag{10}$$

Данные общие уравнения поворота отличаются от уравнений (5) наличием дополнительных членов.

Поворот

Подобно перемещению, поворот — это жёсткое преобразование, не деформирующее объекты. Все точки объекта поворачиваются на одинаковый угол. Чтобы повернуть отрезок прямой, на оба его конца действуют преобразованием вращения (10), после чего между полученными точками проводится новая линия. Поворот многоугольника включает поворот всех его вершин на заданный угол с последующим построением многоугольника. Для поворота кривой сначала поворачиваются её определяющие точки, а затем по ним строится новая кривая. Чтобы, повернуть эллипс вокруг его центра, можно просто вращать большую и малую оси.

Масштабирование

а) б)

Рис.: Превращение квадрата а) в прямоугольник б) с помощью масштабных коэффициентов $s_x = 2$ и $s_y = 1$

Чтобы изменить размер объекта, применяется преобразование *масштабирования*. Простая операция двухмерного масштабирования заключается в умножении точек объекта (x, y) на *масштабные коэффициенты* s_x и s_y, в результате чего получаются преобразованные координаты (x', y'): $x' = x \cdot s_x$, $y' = y \cdot s_y$. Масштабный коэффициент s_x отвечает за изменение размеров объекта по оси x, а коэффициент s_y — по оси y. Стандартные уравнения двухмерного масштабирования также можно записать в следующей матричной форме:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \tag{11}$$

или $\mathbf{p}' = \mathbf{S} \cdot \mathbf{p}$, где $\mathbf{S}$ — матрица масштабирования.

Масштабирование

Масштабными коэффициентами s_x и s_y могут быть любые положительные значения. Значения, которые меньше 1, сокращают размеры объектов; значения, превышающие 1, дают увеличение. Если коэффициентам s_x и s_y присвоены одинаковые значения, получаем *равномерное (uniform) масштабирование*, при котором поддерживаются относительные пропорции объекта. В некоторых системах параметрам масштабирования можно присваивать отрицательные значения. При этом объект не только масштабируется, но и отражается относительно одной или нескольких координатных осей.
Объекты, преобразованные с помощью уравнения (11), масштабируются и переносятся в другое место. При масштабных коэффициентах, которые по модулю меньше 1, объекты приближаются к началу координат, а при масштабных коэффициентах, по модулю превышающих 1, объекты удаляются от начала координат.

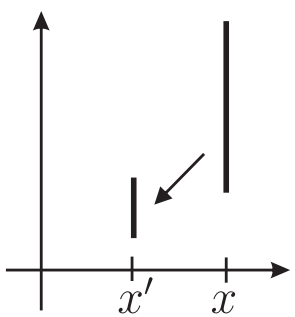
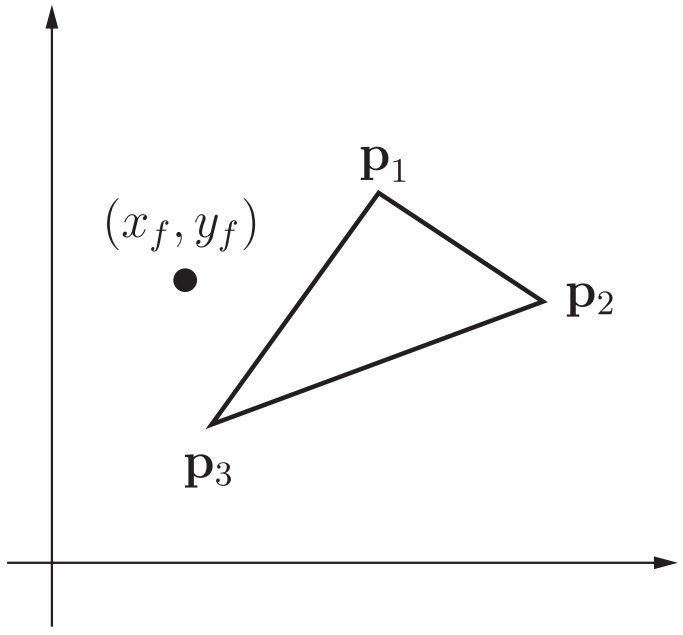


Рис.: Линия, масштабированная с помощью уравнения (11) при $s_x = s_y = 0.5$, сокращается вдвое и приближается к началу координат

Масштабирование



Положение масштабированного объекта можно контролировать, задавая *неподвижную точку*, которая не меняется при преобразовании. Координаты неподвижной точки (x_f, y_f) часто выбираются так, чтобы она принадлежала объекту, зачастую в качестве неподвижной точки выбирается центр масс тела. В таком случае изменение размеров объекта происходит за счёт изменения расстояний между точками объекта и неподвижной точкой. Для точки с координатами (x, y) масштабные коэффициенты (x', y') вычисляются так:

$$x' - x_f = (x - x_f)s_x, \qquad y' - y_f = (y - y_f)s_y. \tag{12}$$

Масштабирование

Уравнения (12) можно переписать, чтобы разделить мультипликативные и аддитивные члены:

$$\begin{aligned} x' &= x \cdot s_x + x_f(1 - s_x), \\ y' &= y \cdot s_y + y_f(1 - s_y), \end{aligned} \tag{13}$$

где аддитивные члены $x_f(1 - s_x)$ и $y_f(1 - s_y)$ являются константами для всех точек объекта.

Координаты неподвижной точки включаются в уравнения масштабирования подобно тому, как в уравнения поворота включаются координаты центра вращения. Вначале можно сформировать вектор-столбец, элементами которого являются постоянные члены в уравнениях (13), затем прибавить этот вектор столбец к произведению $\mathbf{S} \cdot \mathbf{p}$ в уравнении (11).

Многоугольники масштабируются следующим образом: вначале преобразования (13) применяются к каждой вершине, затем по преобразованным вершинам строится новый многоугольник. При обработке других объектов уравнения масштабирования применяются к параметрам, определяющим эти объекты. Чтобы изменить размер круга, можно масштабировать его радиус и рассчитать новые координаты точек окружности. Чтобы изменить размер эллипса, на главные оси воздействуют параметрами масштабирования, а затем строится новый эллипс (неподвижной точкой является центр эллипса).

Матричные представления и однородные координаты

Во многих геометрических приложениях реализованы последовательности геометрических преобразований. В анимации могут потребоваться сложные движения объектов, состоящие, например, из перемещений и поворотов. В графических редакторах и CAD-системах перемещения, повороты и масштабирования выполняются для того, чтобы правильно разместить компоненты изображения в нужных местах. Преобразование наблюдения включает в себя последовательность перемещений и поворотов, необходимых, чтобы перейти от исходной спецификации сцены к изображению на устройстве вывода.
Любое из трёх базовых двухмерных преобразований (перемещение, поворот и масштабирование) можно выразить в общей матричной форме:

$$\mathbf{p}' = \mathbf{M}_1 \cdot \mathbf{p} + \mathbf{v}. \tag{14}$$

Чтобы с помощью этих уравнений получить последовательность преобразований, можно рассчитать преобразованные координаты поэтапно. Вначале точки масштабируются, затем поворачиваются и перемещаются. Однако гораздо эффективнее объединить преобразования, чтобы координаты конечных точек получались непосредственно из координат исходных, без расчёта координат промежуточных точек. Для этого можно переформулировать уравнение (14) и исключить из него операцию матричного сложения.

Однородные координаты

Мультипликативные и аддитивные члены двухмерного геометрического преобразования можно объединить в одну матрицу, если расширить координатные представления с помощью матриц 3×3. При этом в третий столбец матрицы преобразования можно будет ввести аддитивные члены, и тогда все уравнения преобразований можно будет выразить в форме умножения матриц. Чтобы это сделать, нужно также расширить двухэлементное представление точек (x, y) в трехэлементное (x_h, y_h, h), именуемое *представлением в однородных координатах*, где *однородный параметр* h — это такая ненулевая величина, при которой

$$x = \frac{x_h}{h}, \qquad y = \frac{y_h}{h}. \tag{15}$$

Следовательно, произвольное представление в двумерных однородных координатах можно также записать как $(h \cdot x, h \cdot y, h)$. Для геометрических преобразований однородный параметр h можно положить равным любому ненулевому значению. Удобно просто положить $h = 1$. В этом случае любая двухмерная точка будет представляться однородными координатами $(x, y, 1)$.

Выражение точек в однородных координатах позволяет представить все уравнения геометрических преобразований в форме матричного умножения, что является стандартным методом, используемым в графических системах.

Матрица перемещения

Используя концепцию однородных координат, уравнения перемещения точки можно представить с использованием следующего матричного умножения:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{16}$$

Данную операцию перемещения можно записать в сокращённой форме

$$\mathbf{p}' = \mathbf{T} \cdot \mathbf{p}. \tag{17}$$

Матрица поворота

Уравнения двухмерного поворота вокруг начала координат можно выразить в матричной форме следующим образом:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{18}$$

или так:

$$\mathbf{p}' = \mathbf{R} \cdot \mathbf{p}. \tag{19}$$

Поворот вокруг любого другого центра поворота нужно реализовывать как последовательность преобразований.

Матрица масштабирования

Преобразование масштабирования относительно начала координат можно выразить в виде следующего умножения матриц:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{20}$$

или

$$\mathbf{p}' = \mathbf{S} \cdot \mathbf{p}. \tag{21}$$

Преобразование масштабирования относительно другой точки выполняется как последовательность операций преобразования.

Обратные преобразования

В матрице обращения перемещения расстояния, на которые выполнялось перемещение, берутся с противоположными знаками. Следовательно, если двумерное перемещение выполнялось на расстояния t_x и t_y, обратная матрица перемещения имеет такой вид:

$$\mathbf{T}^{-1} = \begin{bmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{22}$$

Эта матрица определяет перемещение в противоположном направлении, причём произведение матрицы перемещения и обратной к ней даёт единичную матрицу.

Обратные преобразования

Двухмерному повороту на угол θ вокруг начала координат соответствует следующая матрица обратного преобразования:

$$\mathbf{R}^{-1} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{23}$$

Отрицательные значения углов поворота дают вращение по часовой стрелке. Поскольку при изменении знака угла поворота меняется значение только функции синус, обратную матрицу можно получить, поменяв местами строки и столбцы прямой матрицы. Следовательно, матрица, обратная к матрице поворота $\mathbf{R}$, равна транспонированной матрице ($\mathbf{R}^{-1} = \mathbf{R}^{T}$).

Матрица обратного преобразования масштабирования получается заменой параметров масштабирования обратными величинами. При двухмерном масштабировании с параметрами s_x и s_y относительно начала координат обратная матрица записывается следующим образом:

$$\mathbf{S}^{-1} = \begin{bmatrix} \frac{1}{s_x} & 0 & 0 \\ 0 & \frac{1}{s_y} & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{24}$$

Двухмерные сложные преобразования

Если есть два преобразования f и g, то композицией этих преобразований называется результат их последовательного применения $(fg)(\mathbf{u}) = f(g(\mathbf{u}))$.
Используя матричные представления, последовательность преобразований можно задать в виде *матрицы сложного преобразования*, вычислив произведения отдельных преобразований. Поскольку обычно одна последовательность преобразований применяется ко многим точкам сцены, эффективнее вначале перемножить матрицы преобразования и получить одну матрицу сложного преобразования. Если применить два преобразования к точке $\mathbf{p}$, новое положение точки будет вычислено следующим образом:

$$\mathbf{p}' = \mathbf{M}_2 \cdot \mathbf{M}_1 \cdot \mathbf{p} = \mathbf{M} \cdot \mathbf{p}. \tag{25}$$

Т.е. точка преобразуется с помощью матрицы $\mathbf{M}$ вместо использования сначала преобразования $\mathbf{M}_1$, а затем преобразования $\mathbf{M}_2$.

Комбинация операций перемещения

Если два вектора перемещения (t_{1x}, t_{1y}) и (t_{2x}, t_{2y}) последовательно применяются к точке $\mathbf{p}$ в двухмерном пространстве, окончательное положение $\mathbf{p}'$ вычисляется следующим образом:

$$\mathbf{p}' = \mathbf{T}(t_{2x}, t_{2y}) \cdot \{\mathbf{T}(t_{1x}, t_{1y}) \cdot \mathbf{p}\} = \{\mathbf{T}(t_{2x}, t_{2y}) \cdot \mathbf{T}(t_{1x}, t_{1y})\} \cdot \mathbf{p}, \qquad (26)$$

где $\mathbf{p}$ и $\mathbf{p}'$ представляются в однородных координатах трёхэлементными векторами-столбцами. Данный результат можно проверить, вычислив матричное произведение двух соответствующих матриц. Кроме того, матрица суммарного преобразования, определённого данной последовательностью перемещений, имеет такой вид:

$$\begin{bmatrix} 1 & 0 & t_{2x} \\ 0 & 1 & t_{2y} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & t_{1x} \\ 0 & 1 & t_{1y} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_{1x} + t_{2x} \\ 0 & 1 & t_{1y} + t_{2y} \\ 0 & 0 & 1 \end{bmatrix} \qquad (27)$$

или

$$\mathbf{T}(t_{2x}, t_{2y}) \cdot \mathbf{T}(t_{1x}, t_{1y}) = \mathbf{T}(t_{1x} + t_{2x}, t_{1y} + t_{2y}), \qquad (28)$$

откуда видно, что два последовательных перемещения являются аддитивными.

Комбинация операций поворота

Два последовательных поворота, применённые к $\mathbf{p}$, дают следующее положение:

$$\mathbf{p}' = \mathbf{R}(\theta_2) \cdot \{\mathbf{R}(\theta_1) \cdot \mathbf{p}\} = \{\mathbf{R}(\theta_2) \cdot \mathbf{R}(\theta_1)\} \cdot \mathbf{p}. \tag{29}$$

Перемножая две матрицы поворота, можно убедиться, что два последовательных поворота являются аддитивными:

$$\mathbf{R}(\theta_2) \cdot \mathbf{R}(\theta_1) = \mathbf{R}(\theta_1 + \theta_2), \tag{30}$$

так что координаты конечной точки можно вычислить с помощью матрицы сложного поворота:

$$\mathbf{p}' = \mathbf{R}(\theta_1 + \theta_2) \cdot \mathbf{p}. \tag{31}$$

Комбинация операций масштабирования

Свертка матриц двух последовательных операций масштабирования даёт следующую сложную матрицу масштабирования:

$$\begin{bmatrix} s_{2x} & 0 & 0 \\ 0 & s_{2y} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_{1x} & 0 & 0 \\ 0 & s_{1y} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_{1x} \cdot s_{2x} & 0 & 0 \\ 0 & s_{1y} \cdot s_{2y} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{32}$$

или

$$\mathbf{S}(s_{2x}, s_{2y}) \cdot \mathbf{S}(s_{1x}, s_{1y}) = \mathbf{S}\left(s_{1x} \cdot s_{2x}, s_{1y} \cdot s_{2y}\right). \tag{33}$$

В этом случае суммарная матрица указывает на то, что последовательные операции масштабирования являются мультипликативными.
Следовательно, если дважды утроить размер объекта, в результате объект увеличится в 9 раз.

Двухмерный поворот вокруг произвольной точки

Двухмерный поворот вокруг любой точки (x_r, y_r) (центра вращения) можно получить, выполнив следующую последовательность операций

1. Переместить объект, чтобы положение центра вращения совместилось с началом координат.
2. Повернуть объект вокруг начала координат.
3. Переместить объект обратно, чтобы центр вращения вернулся в исходное положение.

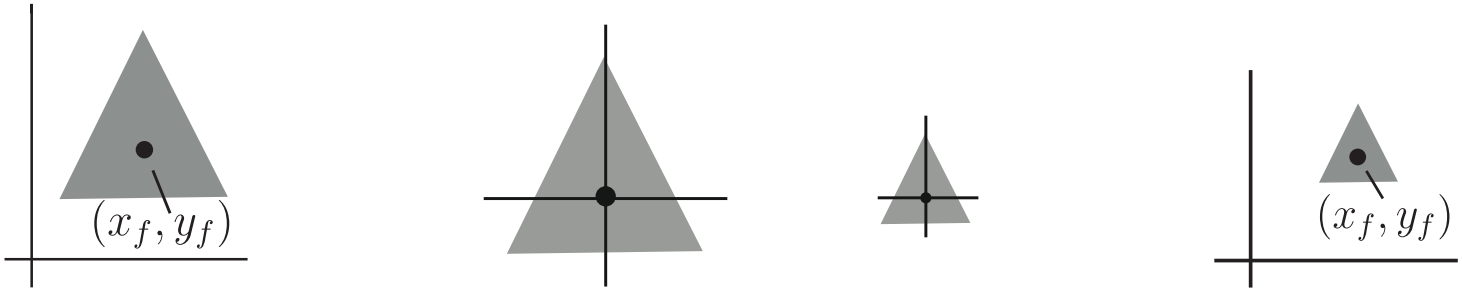
Итоговая матрица для данной последовательности действий имеет вид:

$$\begin{bmatrix} 1 & 0 & x_r \\ 0 & 1 & y_r \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_r \\ 0 & 1 & -y_r \\ 0 & 0 & 1 \end{bmatrix} = \tag{34}$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta & x_r(1-\cos\theta) + y_r \sin\theta \\ \sin\theta & \cos\theta & y_r(1-\cos\theta) - x_r \sin\theta \\ 0 & 0 & 1 \end{bmatrix},$$

или $\mathbf{T}(x_r, y_r) \cdot \mathbf{R}(\theta) \cdot \mathbf{T}(-x_r, -y_r) = \mathbf{R}(x_r, y_r, \theta)$.

Двухмерное масштабирование относительно произвольной точки



Последовательность преобразований, дающая двухмерное масштабирование относительно выбранной точки (x_f, y_f) (центра масштабирования)

1. Переместить объект так, чтобы положение центра масштабирования совместилось с началом координат.
2. Масштабировать объект относительно начала координат.
3. Использовать преобразование, обратное к перемещению в п. 1, чтобы вернуть объект на первоначальное место.

Искомая матрица масштабирования имеет вид:

$$\begin{bmatrix} 1 & 0 & x_f \\ 0 & 1 & y_f \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_f \\ 0 & 1 & -y_f \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & x_f(1-s_x) \\ 0 & s_y & y_f(1-s_y) \\ 0 & 0 & 1 \end{bmatrix} \quad (35)$$

или

$$\mathbf{T}(x_f, y_f) \cdot \mathbf{S}(s_x, s_y) \cdot \mathbf{T}(-x_f, -y_f) = \mathbf{S}(x_f, y_f, s_x, s_y). \quad (36)$$

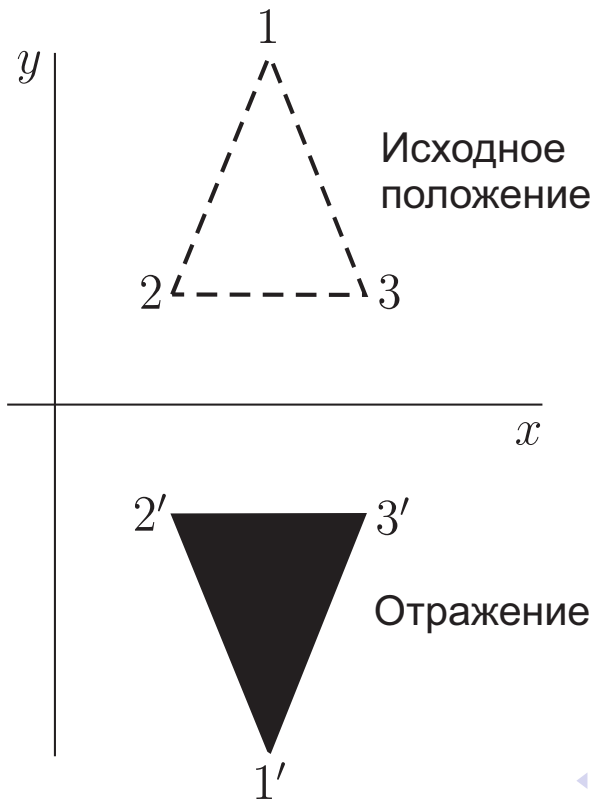
Другие двухмерные преобразования

Такие базовые преобразования, как перемещение, поворот и масштабирование, являются стандартными компонентами графических библиотек. Некоторые пакеты предлагают несколько дополнительных преобразований, полезных в определённых приложениях. Двумя такими преобразованиями являются отражение и сдвиг.

Отражение

Преобразование, дающее зеркальное изображение объекта, называется отражением. При двухмерном отражении это изображение генерируется относительно оси отражения как поворот объекта на 180° относительно оси отражения. Отражение относительно оси Ox выполняется с помощью матрицы преобразования

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{37}$$

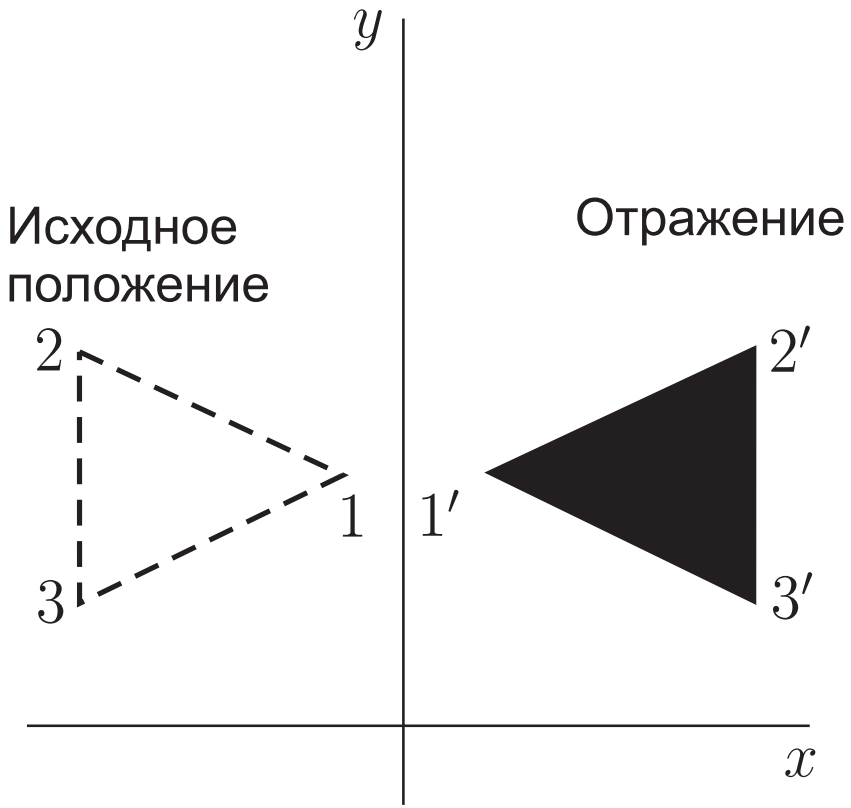


Отражение

Отражение от оси Oy определяется матрицей преобразования

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{38}$$

Данное преобразование эквивалентно повороту на 180° в трёхмерном пространстве вокруг оси Oy.

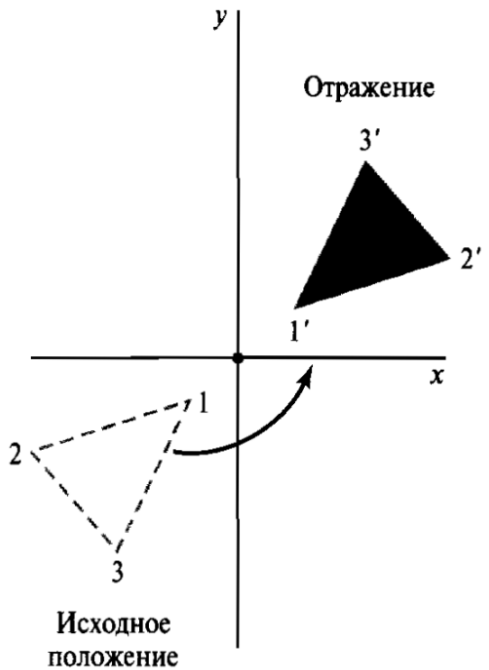


Отражение

Отражение от обеих координатных осей иногда называется отражением относительно начала координат Данное преобразование в матричной форме записывается так:

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{39}$$

Матрица отражения (39) записывается так же, как матрица поворота $R(\theta)$ при $\theta = 180^\circ$. Объект просто поворачивается в плоскости xy на полоборота вокруг начала координат.



Отражение

Отражение (39) можно обобщить на случай произвольного положения центра отражения в плоскости xy. Данное отражение не отличается от поворота на 180° в плоскости xy вокруг центра отражения.

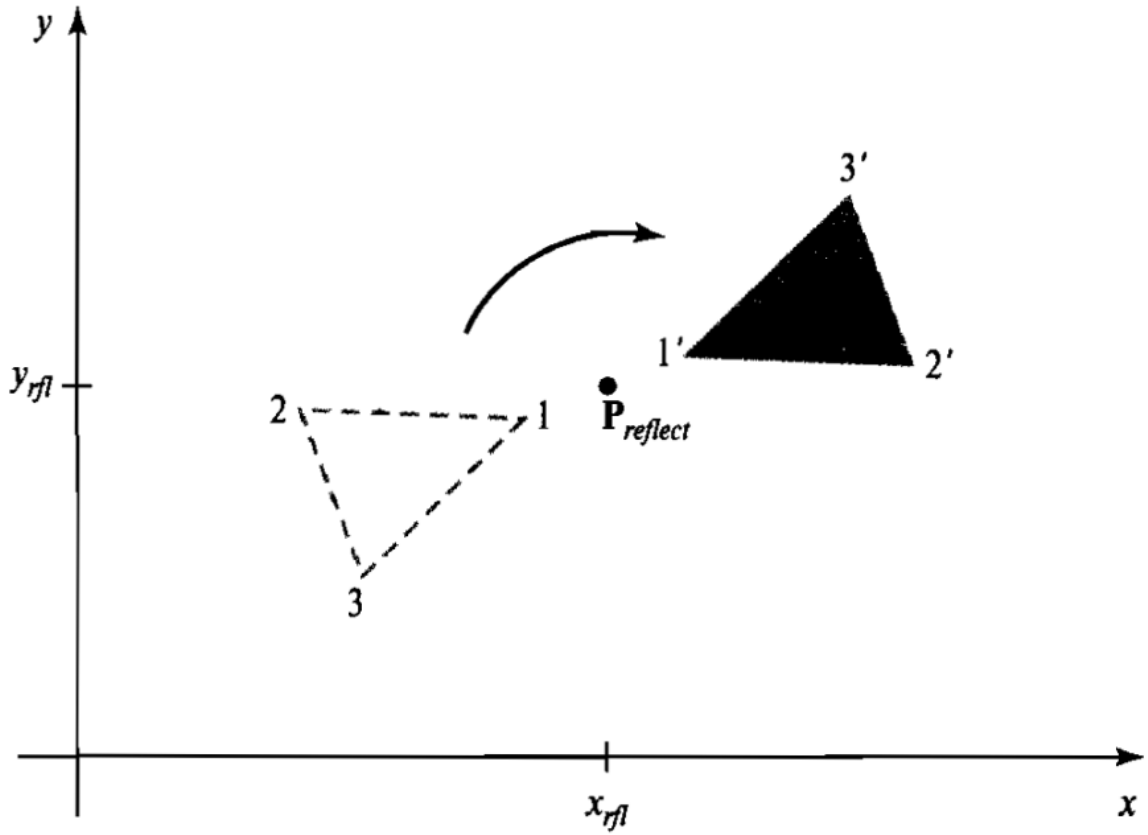


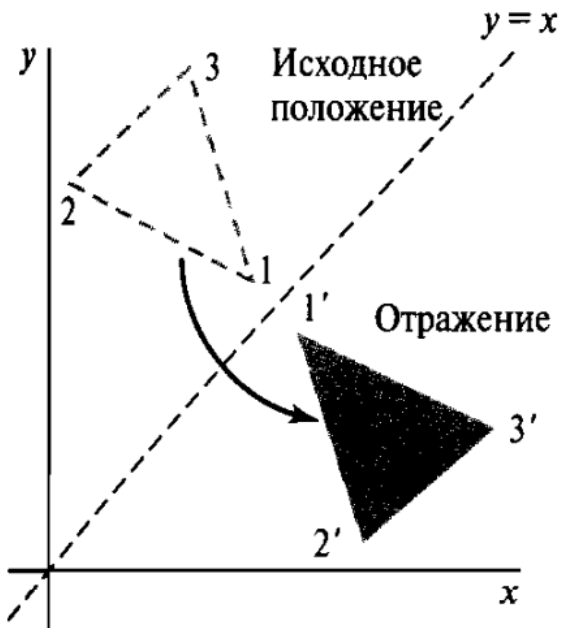
Рис.: Отражение объекта относительно оси, перпендикулярной плоскости xy и проходящей через точку $\mathbf{P}_{\text{reflect}}$

Отражение

Если в качестве оси отражения выбрать диагональ $y = x$, матрица отражения запишется так

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{40}$$

Данную матрицу можно вывести, произведя свертку матриц поворота и отражения от координатной оси. Вначале выполняется поворот по часовой стрелке вокруг начала координат на угол 45°, при этом линия $y = x$ переходит в ось x. Затем выполняется отражение относительно оси x. Последней операцией является возврат линии $y = x$ к исходному положению — поворот против часовой стрелки на угол 45°.



Отражение

Отражение относительно любой прямой $y = mx + b$ в плоскости xy можно выполнить с помощью комбинации преобразований перемещения–поворота–отражения. В общем случае вначале прямая перемещается так, чтобы она проходила через начало координат. Затем прямая поворотом совмещается с одной из координатных осей и отражается относительно этой оси. Наконец, с помощью обратного поворота и перемещения линия возвращается в исходное положение.
Несложно убедиться в обратимости преобразований отражения (определитель соответствующих матриц равен минус единице) и в том, что обратным преобразованием к преобразованию отражения является само преобразование отражения.
Отражение относительно координатных осей или начала координат можно реализовать как масштабирование с отрицательным масштабным коэффициентом. Кроме того, элементам матрицы отражения можно присваивать другие значения, кроме ± 1. Параметр отражения, по модулю превышающий 1, смещает зеркальное изображение точки дальше от оси отражения, а параметр, который по модулю меньше 1, приближает изображение точки к оси отражения. Следовательно, отраженный объект также можно увеличить, уменьшить или исказить.

Сдвиг

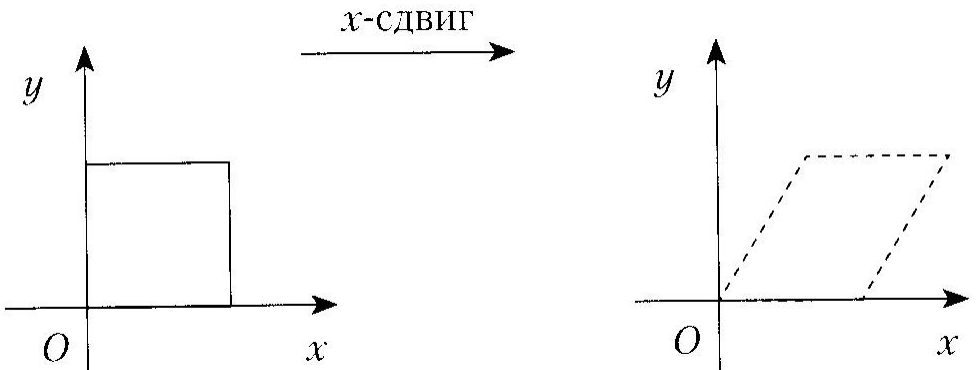
Преобразование, которое так искажает форму объекта, что преобразованная форма выглядит, будто объект составлен из внутренних слоев, которые скользят один по другому, часто называется сдвигом (shear). Двумя распространёнными преобразованиями сдвига являются сдвиги вдоль оси Ox и вдоль оси Oy. Сдвиг вдоль оси Ox получается с помощью матрицы преобразования:

$$\begin{bmatrix} 1 & \lambda & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{41}$$

а координаты при этом меняются следующим образом:

$$x' = x + \lambda y, \qquad y' = y. \tag{42}$$

Параметр сдвига λ может иметь любое действительное значение. Точка (x, y) сдвигается горизонтально на величину, пропорциональную её расстоянию по перпендикуляру до оси Ox (значение y).



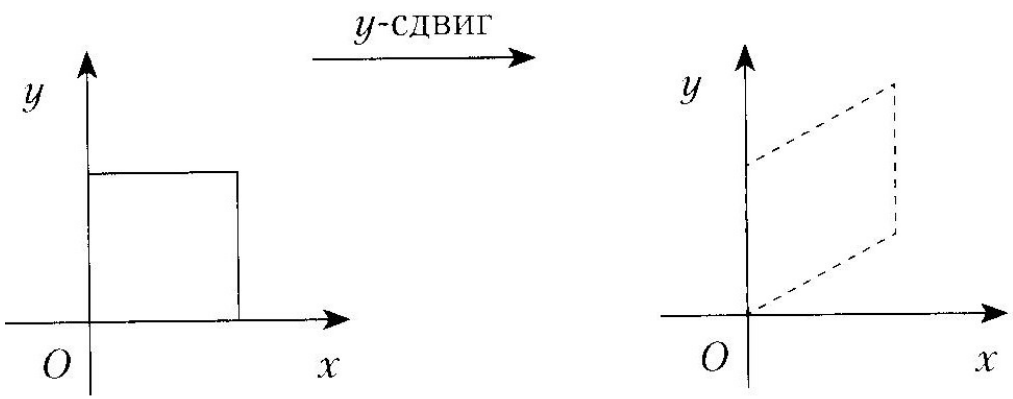
Сдвиг

Сдвиг в направлении оси Oy получается с помощью следующей матрицы преобразования:

$$\begin{bmatrix} 1 & 0 & 0 \\ \lambda & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{43}$$

причем координаты преобразуются так:

$$x' = x, \qquad y' = y + \lambda x. \tag{44}$$



Сдвиг

Определитель матриц сдвига всегда равен единице, поэтому они невырожденны и для них существуют обратные, которые задаются следующими формулами:

$$\begin{bmatrix} 1 & -\lambda & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \begin{bmatrix} 1 & 0 & 0 \\ -\lambda & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Т.о. обратным преобразованием к преобразованию сдвига также является преобразование сдвига.

Работа с векторами и матрицами в JavaScript

JavaScript не имеет встроенной поддержки для работы с матрицами и векторами как с математическими объектами. Библиотека glMatrix (`http://glMatrix.net`) является библиотекой с открытым исходным кодом, которая довольно часто используются в приложениях WebGL. Она была написана Брэндоном Джонсом в основном для WebGL. Большинство библиотек JavaScript, которые помещаются на веб-сервер и используются в реальных веб-приложениях, каким-то образом сжаты, чтобы свести к минимуму время загрузки. Обычно такое сжатие подразумевает удаление пробелов и комментариев. Таким образом, библиотеки JavaScript часто существуют в двух версиях:

- Сжатая версия, которая используется приложением, после его разработки и выгрузки на веб-сервер.
- Несжатая версия, доступная, чтобы можно было легко читать исходный код, отлаживать приложение и понимать, как оно работает.

Для библиотеки glMatrix файл для сжатой версии имеет имя `glMatrix-min.js`, а несжатая версия просто имеет имя `glMatrix.js`. Эту библиотеку необходимо подключить в файле HTML перед использованием объекта. Для этого достаточно просто добавить тег `<script>`:

```
<script src="../lib/gl-matrix.js"></script>
```

Общие функции библиотеки glMatrix

`equals(a, b)`	Проверяет, равны ли числа a и b с погрешностью glMatrix.EPSILON (сравниваются абсолютные значения, если они меньше или равны 1.0, и относительные для больших значений). Возвращает `true` если числа приблизительно равны, иначе `false`.
`setMatrixArrayType(type)`	Устанавливает тип массива (`Float32Array` или `Array`), используемый при создании новых векторов и матриц.
`toRadian(a)`	Преобразует градусы в радианы.

Создание векторов

Чтобы создать вектор с тремя координатами и сразу инициализировать их, используется следующая команда:

```
const u = vec3.fromValues(x, y, z);
```

Для создания трехкомпонентного вектора без инициализации, требуется вызвать функцию:

```
let u = vec3.create();
```

Функция `vec3.str()` преобразует вектор в строку.
В следующем фрагменте кода два вектора `u` и `v` создаются и инициализируются значениями. Затем создаётся третий вектор `r` для сохранения результата операции сложения:

```
const u = vec3.fromValues(1, 2, 3);
const v = vec3.fromValues(4, 5, 6);
let r = vec3.create();
vec3.add(r,u,v);  // r = [5,7,9]
alert(vec3.str(r)); // показать сообщение [5,7,9]
```

Функции для работы с векторами

`angle(a, b)`	Рассчитывает угол между двумя векторами
`clone(a)`	Возвращает копию вектора
`cross(out, a, b)`	Рассчитывает векторное произведение
`distance(a, b)`	Рассчитывает евклидово расстояние
`dot(a, b)`	Рассчитывает скалярное произведение
`equals(a, b)`	Сравнивает два вектора с погрешностью
`exactEquals(a, b)`	Сравнивает два вектора точно
`length(a)`	Возвращает длину вектора
`negate(out, a)`	Домножает компоненты вектора на -1
`normalize(out, a)`	Нормирует вектор
`scale(out, a, b)`	Умножает вектор на число
`set(out, x, y, z)`	Возвращает вектор с заданными компонентами
`subtract(out, a, b)`	Вычитает вектор b из вектора a
`transformMat3(out,a,m)`	Выполняет умножение матрицы на вектор

Создание матриц

Объект mat4 поддерживает две группы методов, связанных с формированием матриц трансформаций: с именами, начинающих с префикса from, и без него. Методы с именами, начинающимися с префикса from, создают матрицу преобразований на основе переданных аргументов и сохраняют её в объекте mat4. Остальные методы выполняют умножение текущей матрицы на матрицу, которая должна быть создана на основе переданных аргументов.
Для создания единичной матрицы 4×4 требуется вызвать функцию:

```
mat4.create();
```

Для создания и инициализации матрицы, используется следующая команда:

```
M = mat4.fromValues(1,0,0,0,   // первый столбец
                    0,1,0,0,   // второй столбец
                    0,0,1,0,   // третий столбец
                    2,3,4,1);  // четвертый столбец
alert(mat4.str(M));     // появится сообщение [1,0,0,0
                        //                     0,1,0,0,
                        //                     0,0,1,0,
                        //                     2,3,4,1]
```

Функция mat4.str() конвертирует матрицу в строку.

Функции для работы с матрицами

`add(out, a, b)`	Складывает две матрицы
`adjoint(out, a)`	Вычисляет присоединённую матрицу
`clone(a)`	Возвращает копию матрицы
`determinant(a)`	Рассчитывает определитель
`equals(a, b)`	Сравнивает две матрицы с погрешностью
`exactEquals(a, b)`	Сравнивает две матрицы точно
`fromRotation(out, rad, axis)`	Создаёт матрицу вращения
`fromScaling(out, v)`	Создаёт матрицу масштабирования
`fromTranslation(out, v))`	Создаёт матрицу перемещения
`identity(out)`	Возвращает единичную матрицу
`invert(out, a)`	Рассчитывает обратную матрицу
`multiply(out, a, b)`	Перемножает две матрицы
`rotate(out, a, rad, axis)`	Домножает матрицу на матрицу вращения
`scale(out, a, v)`	Домножает матрицу на матрицу масштабирования
`subtract(out, a, b)`	Вычитает матрицу b из матрицы a
`translate(out, a, v)`	Домножает матрицу на матрицу перемещения
`transpose(out, a)`	Осуществляет транспонирование