



LOBACHEVSKY STATE UNIVERSITY
of NIZHNI NOVGOROD
National Research University

Computing Mathematics and Cybernetics faculty
Software department

CS255. Computer Graphics Introduction Course

Координатный метод в КГ

- Преобразование объектов сцены
- Проецирование сцены
- Стереографика
- Захват движения

Турлапов, Вадим Евгеньевич
проф. каф. МО ЭВМ, ВМК, ННГУ

Векторные полигональные модели 3D объектов

Неявное уравнение:

$$F(x,y,z)=0$$

Параметрическая форма

Сфера: $x = R \cdot \sin \varphi \cdot \cos \psi, y = R \cdot \sin \varphi \cdot \sin \psi, z = R \cdot \cos \varphi$

Сплайны (*) и **патчи** (**) Безье:

$$P(t) = \sum_{i=0}^m C_m^i t^i (1-t)^{m-i} \cdot P_i \quad (*)$$

$$P(s,t) = \sum_{i=0}^m \sum_{j=0}^n C_m^i s^i (1-s)^{m-i} \cdot C_n^j t^j (1-t)^{n-j} \cdot P_{ij} \quad (**)$$

$$C_m^i = \frac{m!}{i!(m-i)!}, \quad 0 \leq s, t \leq 1$$

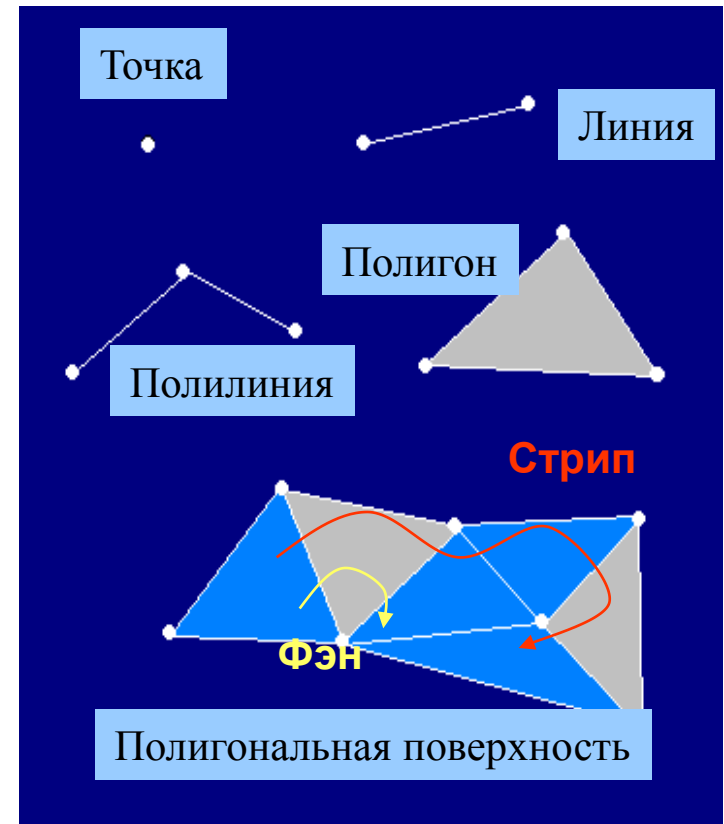
Структуры данных. Линейно-узловая модель

Способ 1: все грани отдельно и для каждой хранятся координаты вершин. Объем памяти: $M1 = N_{\text{Face}} \cdot n_{\text{Vert}} \cdot 3 \cdot \text{Byte}_{\text{Coor}}$

Способ 2: вершины пронумерованы, хранятся в отдельной таблице; каждая грань задана списком индексов вершин. Объем памяти: $M2 = n_{\text{Vert}} \cdot 3 \cdot \text{Byte}_{\text{Coor}} + N_{\text{Face}} \cdot N_{\text{FaceVert}} \cdot \text{Byte}_{\text{Index}}$

Способ 3: линейно-узловая модель (иерархия: вершины, ребра, грани). Объем памяти: $M3 = n_{\text{Vert}} \cdot 3 \cdot \text{Byte}_{\text{Coor}} + k_{\text{Edge}} \cdot 2 \cdot \text{Byte}_{\text{VertIndex}} + N_{\text{Face}} \cdot N_{\text{FaceVert}} \cdot \text{Byte}_{\text{EdgeIndex}}$

Базовые элементы ВПМ:



Св-ва: 1 избыточна; 1и2 дважды рисуют ребра; 2и3 сохраняют топологию.

Векторные полигональные модели трехмерных объектов

Стрипы и фэны (экономия памяти)

Strip – полоса, лента из треугольников или квадратов; Fan – веер из треугольников.

Объем памяти для треугольников: $M_{\text{Triang}} = n_{\text{Vert}} \cdot 3 \cdot \text{Byte}_{\text{Coor}} [+ (n_{\text{Vert}} - 2) \cdot \text{Byte}_{\text{Index}}]$

Объем памяти для квадратов: $M_{\text{Quad}} = n_{\text{Vert}} \cdot 3 \cdot \text{Byte}_{\text{Coor}} + (n_{\text{Vert}} - 2) / 2 \cdot \text{Byte}_{\text{Index}}$

Режимы OpenGL: glBegin(GL_TRIANGLE_STRIP), а также GL_QUAD_STRIP; GL_TRIANGLE_FAN.

Равномерная сетка

Не надо хранить x,y. Для описания сложных поверхностей необходимо большое количество узлов, которое может быть ограничено объемом памяти компьютера

Неравномерная сетка. (TIN-Triangulated Irregular Network)

Изолинии. Преобразование изолиний в полигональную модель выполняется методами триангуляции

Воксельная модель. Voxel-Volume Element

Однородные координаты

Однородные координаты [homogeneous - однородный]

точки $P(x,y)$: $(x,y) \rightarrow (xh, yh, h)$ после нормировки на $h \rightarrow (x,y,1)$

Однородные координаты вектора $\mathbf{v}(x,y)=P_2-P_1$:

$(x,y) \rightarrow (x,y,0)$ - вектор эквивалентен бесконечно удаленной (несобственной) точке.

Линейные комбинации однородных координат. Аффинная комбинация

1) $\mathbf{v}+\mathbf{v} \sim \mathbf{v}$, $a\mathbf{v}+b\mathbf{v} \sim \mathbf{v}$

2) $P+\mathbf{v} \sim P$

3) P_1+P_2 не P (пока не отнормируем на 2)

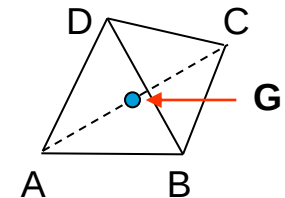
4) $p \cdot P_1 + q \cdot P_2 \sim P$, если $p+q=1$ – аффинная комбинация точек, как и 5)

5) $P_1+t \cdot (P_2-P_1)=P_1+t \cdot \mathbf{v}=t \cdot P_2+(1-t) \cdot P_1 \sim P$ - линейная интерполяция

Центр треугольника как аффинная комбинация точек

$C=(A+B+D)/3$ - точка пересечения медиан треугольника

Центр тетраэдра как аффинная комбинация точек: $G=(A+B+C+D)/4$



Для точки $P(x,y,z)$: $(x,y,z) \rightarrow (xh, yh, zh, h)$ после нормировки на $h \rightarrow (x,y,z,1)$

Однородные координаты вектора $\mathbf{v}(x,y,z)=P_2-P_1$: $(x,y,z) \rightarrow (x,y,z,0)$

Твин-преобразование фигур с одинаковым количеством точек (движение, морфинг): $P(t)=\text{tween}(A,B,t)$.

Преобразование объектов сцены

Варианты моделирования преобразований точки P:

$$[P][T] = [x \ y \ z \ 1] \cdot [T] = [P^*] \quad (1)$$

или

$$[T'] [P'] = [T'] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x^* \\ y^* \\ z^* \\ 1 \end{bmatrix} = [P'^*] \quad (2)$$

$$([P][T])^T = [T]^T [P]^T = [T'] [P']$$

Далее по умолчанию будем использовать вариант (1).

Элементы матрицы T и типы преобразований

$$[T] = \begin{bmatrix} M_{xx} & S_x^y & S_x^z & l \\ S_y^x & M_{yy} & S_y^z & m \\ S_z^x & S_z^y & M_{zz} & n \\ p & q & r & M_h \end{bmatrix}$$

Аффинные (в т.ч. ортогональные) преобразования

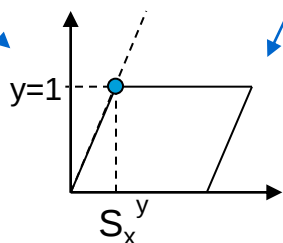
$M_{xx} = -1$ – отражение по x , $|M_{xx}| \neq 1$ – масштаб по x

S_x^y – сдвиг по x пропорциональный y

l, m, n – вектор параллельного переноса

p, q, r – перспективные преобразования

M_h – общий масштаб



Повороты и параллельный перенос в 3D

Поворот вокруг оси OX
осуществляется с помощью матрицы
следующего вида:
Остается неподвижной вся ось X:
[x,0,0,1] и вектор [1,0,0,0].

$$[R_x] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Поворот вокруг оси OY
осуществляется с помощью матрицы
следующего вида:
Остается неподвижной вся ось Y:
[0,y,0,1] и орт [0,1,0,0].

$$[R_y] = \begin{bmatrix} \cos \varphi & 0 & -\sin \varphi & 0 \\ 0 & 1 & 0 & 0 \\ \sin \varphi & 0 & \cos \varphi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Поворот вокруг оси OZ
осуществляется с помощью матрицы
следующего вида:
Остается неподвижной вся ось Z:
[0,0,z,1] и орт [0,0,1,0].

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

Параллельный перенос начала
координат в точку $O'=[l,m,n]$
осуществляется матрицей вида:

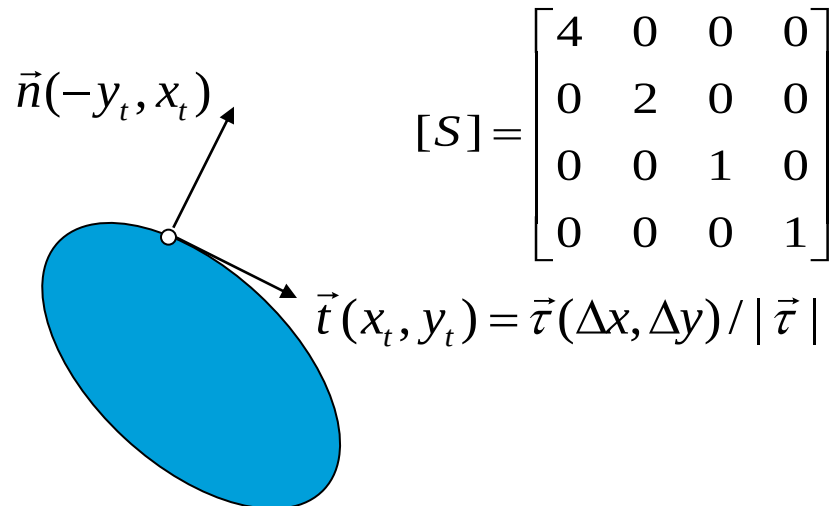
$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ l & m & n & 1 \end{bmatrix}$$

Преобразование нормалей при деформации поверхности

Правила преобразования точек поверхности объекта и нормалей к поверхности при неоднородной деформации:

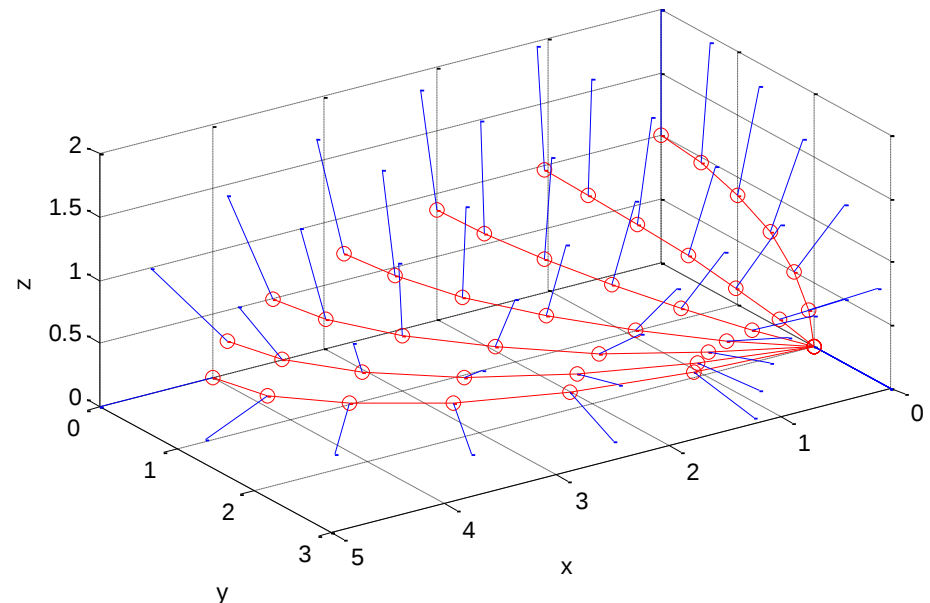
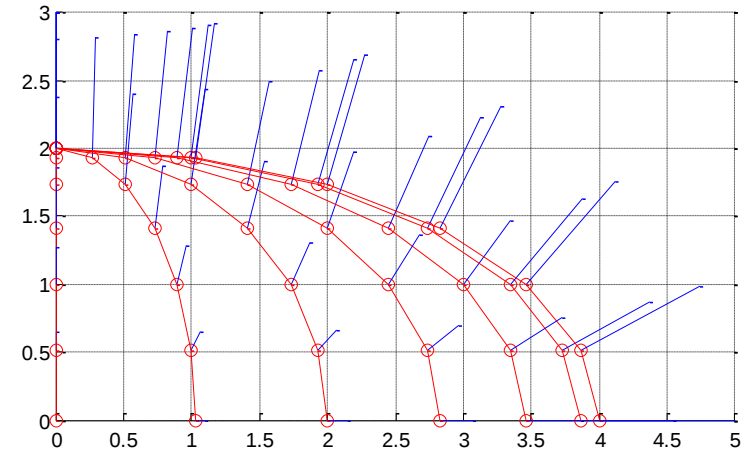
$$\vec{r} = \vec{r} \cdot [S]; \quad \vec{n} = \vec{n} \cdot [S^{-1}]^T; \quad (3)$$

Пример преобразования нормалей при неоднородной деформации сегмента единичной сферы матрицей S:

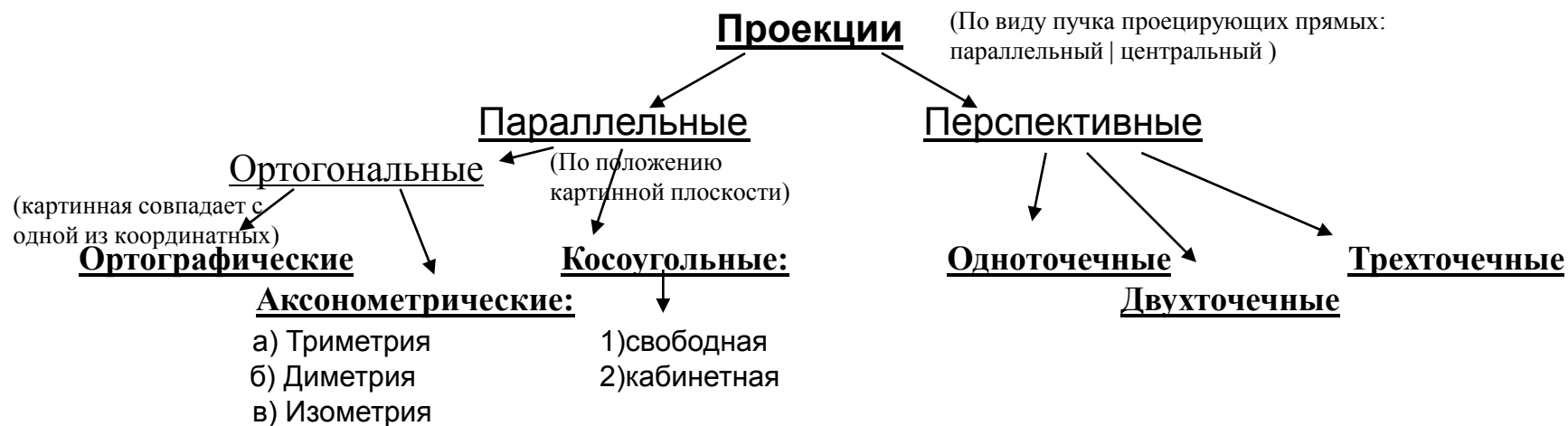


$$[S] = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

→



Иерархическая систематизация проекций



Параллельные проекции

1.Ортографические: Многовидовая ортографическая проекция – изображение объекта сверху, снизу, с боков, т.е. проекции на основные плоскости ($z=0$, $x=0$, $y=0$). Не искажают размеров. Широко применяется в инженерной графике. Для сложных объектов используют также сечения и прозрачность.

2.Косоугольные: картинная плоскость находится под углом к направлению проецирования.

Свободная (cavalier): угол=45 градусов для передачи размеров по оси **Oz** без искажения, **x** и **y** также без искажений и сохраняют ортогональность.

Кабинетная (cabinet): угол=63.435 градуса, её назначение - наглядность изображения при удобном для вычислений коэффициенте искажения размеров по **Z** (он равен 1/2).

АксонOMETрические проекции

В общем случае получаются поворотом вокруг 2 осей на произвольные углы и проецированием на плоскость $Z=0$ (чтобы были видны 3 соседние грани). Определим коэффициенты искажения единичных отрезков осей координат для аксонометрической проекции:

$$[E'] = [E][T] = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} [T] = \begin{bmatrix} x'_x & y'_x & 0 & 1 \\ x'_y & y'_y & 0 & 1 \\ x'_z & y'_z & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos \varphi & \sin \varphi \cdot \sin \theta & 0 & 1 \\ 0 & \cos \theta & 0 & 1 \\ \sin \varphi & -\cos \varphi \sin \theta & 0 & 1 \end{bmatrix}, \quad [П_z] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

где $[T] = [R_y][R_x][П_z]$, $[R_x]$ – матрица поворота на угол θ вокруг оси Ox , $[R_y]$ – матрица поворота на угол φ вокруг оси Oy , а $[П_z]$ – матрица проецирования на плоскость $z=0$:

Коэффициенты искажения вдоль осей определяются как:

$$f_x = \sqrt{(x'_x)^2 + (y'_x)^2} \quad - \text{ по оси } Ox, \text{ и аналогично по другим осям.}$$

В зависимости от соотношения этих параметров аксонометрические проекции делятся на:

- 1.триметрические - при $f_x \neq f_y \neq f_z$
- 2.диметрические - при $f_x = f_y \neq f_z$, по 4 проекции для f_z : $\theta = \arcsin(\pm f_z / \sqrt{2})$; $\varphi = \arcsin(\pm f_z / \sqrt{2 - f_z^2})$
- 3.изометрические - при $f_x = f_y = f_z = (2/3)^{1/2} = 0.8165$ (в технических чертежах принимают $f_x = f_y = f_z = 1$), 4 проекции: $\{\varphi = \pm 45^\circ, \theta = \pm 35.26439^\circ\}$.

Ортогональное проецирование в OpenGL и канонический видимый объем (CVV)

```
void glOrtho( GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble near, GLdouble far );
```

Параметры

left, right - координаты левой и правой отсекающих плоскостей;

bottom, top - координаты нижней и верхней отсекающих плоскостей;

near, far - расстояния (глубины) до ближней и дальней отсекающих плоскостей в сторону отрицательных Z.

Замечания

glOrtho формирует матрицу параллельного проецирования. Глаз наблюдателя принимается расположенным в точке (0,0,0). Точки (*left, bottom, -near*) и (*right, top, -far*) преобразуются матрицей в (-1,-1,-1) и (1,1,1) соответственно канонического видимого объема (Canonical View Volume). Вид матрицы:

$$[O] = \begin{bmatrix} 2/(right - left) & 0 & 0 & -(right + left)/(right - left) \\ 0 & 2/(top - bottom) & 0 & -(top + bottom)/(top - bottom) \\ 0 & 0 & -2/(far - near) & -(far + near)/(far - near) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Текущая матрица M умножается на матрицу параллельного проецирования O и замещается на результат: $M \bullet O$. Методы **glPushMatrix()** и **glPopMatrix()** сохраняют и восстанавливают текущую матрицу из стека.

The following functions retrieve information related to **glOrtho**:

glGet(GL_MATRIX_MODE) – возвращает моду матрицы; **glGet(GL_MODELVIEW_MATRIX)** или

glGet(GL_PROJECTION_MATRIX), **glGet(GL_TEXTURE_MATRIX)** – возвращают соответствующие матрицы.

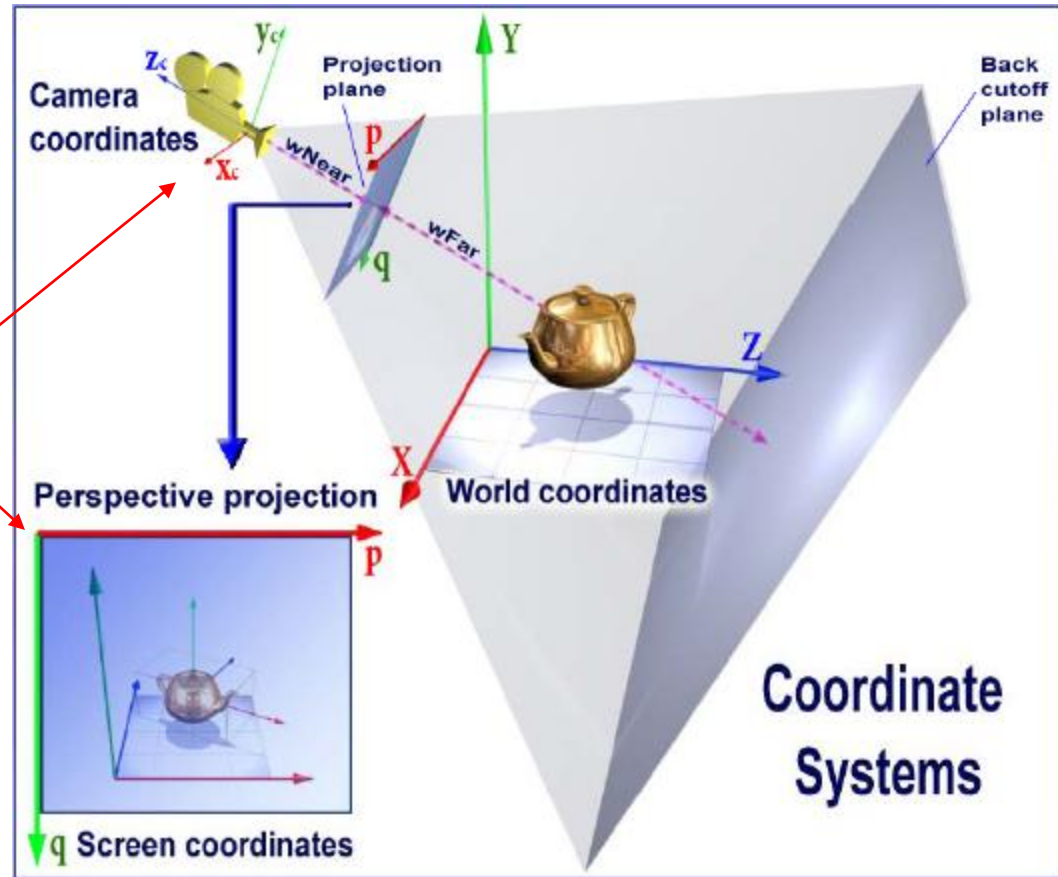
Перспективные проекции

Способ проецирования 3D-сцены на проекционную плоскость камеры определяется **Матрицей проецирования:** (device.Transform.Projection)

Канонический видимый объем (CVV) в перспективных проекциях.

В перспективных преобразованиях возникает понятие центра проецирования.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Перспективные проекции

Перспективные преобразования имеют место, если не равен нулю хотя бы один из трех элементов последнего столбца матрицы преобразования однородных координат.

1. Одноточечная перспектива.

Когда не равен нулю только один коэффициент последнего столбца матрицы преобразования.

То есть, например

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. Двухточечные перспектива.

Когда не равны нулю два коэффициента последнего столбца матрицы преобразования.

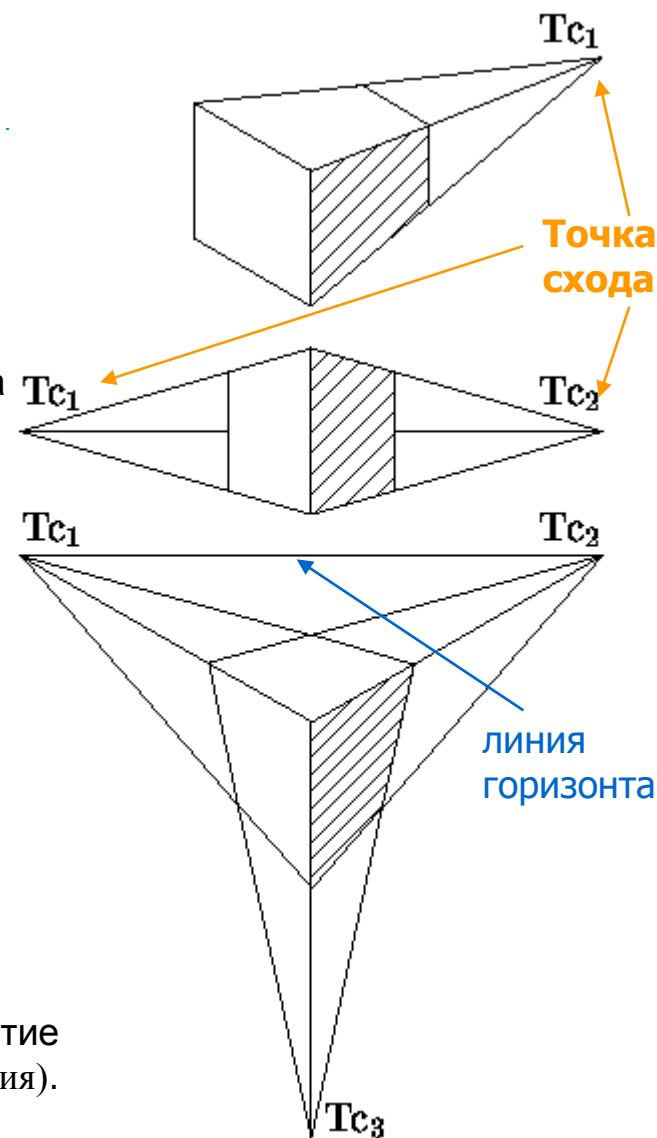
3. Трехточечные перспектива.

Когда не равны нулю все три коэффициента последнего столбца матрицы преобразования.

То есть, например

$$[T] = \begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

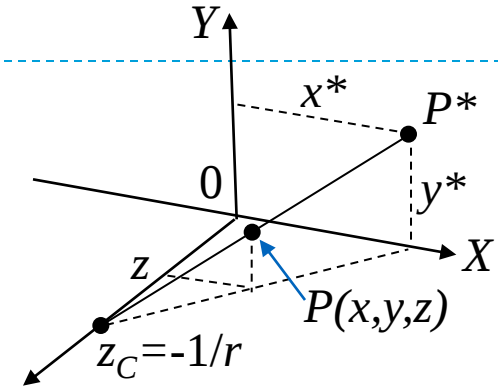
При рассмотрении перспективных преобразований возникает понятие точек схода (Тс), линий горизонта и центров проекции (проецирования).



Геометрическая интерпретация одноточечной перспективы

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & rz+1 \end{bmatrix}$$

$$\begin{bmatrix} x^* & y^* & z^* & 1 \end{bmatrix} = \begin{bmatrix} \frac{x}{1+rz} & \frac{y}{1+rz} & \frac{z}{1+rz} & 1 \end{bmatrix}$$



Если объединить с проецированием на плоскость $Z=0$, то 3 элемент будет $=0$

Чему соответствует r ? Расположим центр проецирования на оси Z при $z=z_C$ и построим проекцию P^* точки $P(x,y,z)$ на плоскость $Z=0$ из этого центра.

Справедлива пропорция: $x^*/x = z_C/(z_C - z)$. Откуда: $x^* = x/(1 - z/z_C)$, т. е.

$$r = -1/z_C$$

Перспективное проецирование точки, находящейся в бесконечности на оси Z дает точку схода:

$$\begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & r \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 1/r & 1 \end{bmatrix}$$

Методы создания перспективных видов

Для того чтобы создавать визуально объемную сцену на плоскости экрана существует несколько методов.

1. Перенос в сочетании с одноточечной перспективой

Комбинируем два преобразования $[T]=[To][\Pi z]$, где $[To]$ – матрица переноса начала координат в точку O' , а $[\Pi z]$ – матрица одноточечной перспективы по z с проецированием на плоскость $Z=0$:

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & r \\ l & m & 0 & 1+rn \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/z_C \\ l & m & 0 & 1-n/z_C \end{bmatrix}$$

Если $n=0$, то размеры объекта не изменяются, так как масштабный коэффициент $s=1+r \cdot n = 1+0 \cdot r = 1$.
Если же $n \neq 0$, то размеры изменятся. При этом если $n > z_C$, то изображение перевернется.

Методы создания перспективных видов

Вращение в сочетании с одноточечной перспективой

Так же, как в аксонометрической проекции, комбинируются 2 преобразования поворота, но с одноточечной перспективой. Одновременно м.б. внесен и перенос:

$$[T]=[R_y][R_x][T_o][\Pi_z],$$

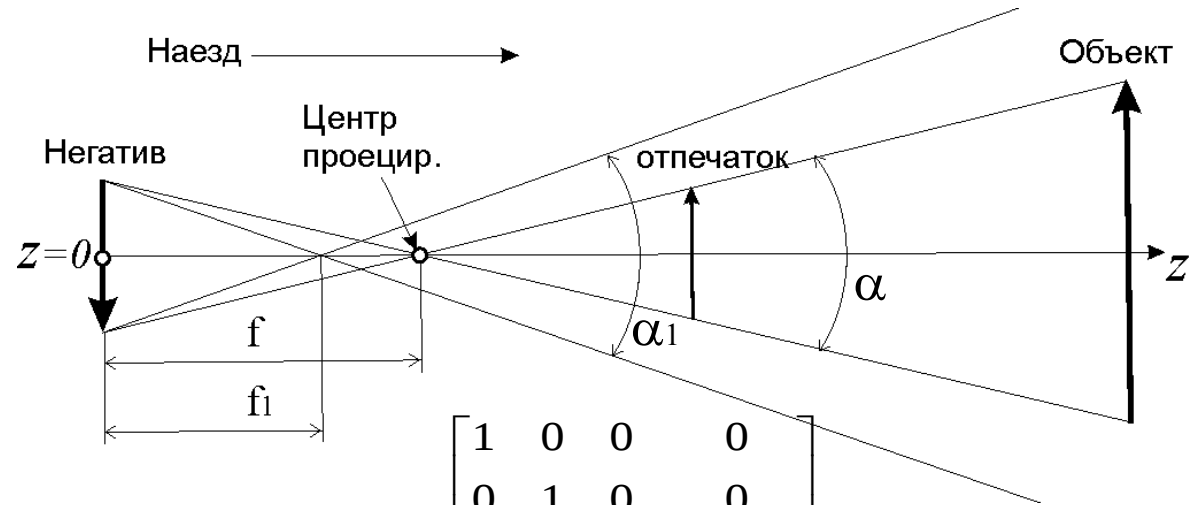
где $[R_y]$ – поворот вокруг оси Oy на угол φ , $[R_x]$ – поворот вокруг оси Ox на угол θ , $[T_o]$ – параллельный перенос на вектор $\{l,m,n\}$, а $[\Pi_z]$ – одноточечная перспектива на плоскость $z=0$.

$$[T] = \begin{bmatrix} \cos \varphi & \sin \varphi \sin \theta & 0 & -r \sin \varphi \cos \theta \\ 0 & \cos \theta & 0 & r \sin \theta \\ \sin \varphi & -\cos \varphi \sin \theta & 0 & r \cos \varphi \cos \theta \\ l & m & 0 & 1 + r \cdot n \end{bmatrix}$$

Получили трехточечную перспективу.

Фотография и перспективные преобразования

Фотография и видео являются
одноточечными перспективными
преобразованиями:



При этом матрица преобразования имеет вид: $[T_\phi] =$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/f \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Применим преобразование к точке: $[x, y, z, 1][T_\phi] = [x, y, 0, 1 - z/f]$

В итоге точка имеет координаты:

$$x^* = \frac{x}{1 - z/f} \quad y^* = \frac{y}{1 - z/f}$$

Т.е. чем ближе z к фокусному расстоянию или/и чем меньше фокусное расстояние, тем больше искажения.

Что лучше для увеличения изображения: наезд камерой при неизменном α (FOV, field of view) или уменьшение α ?

Стереοграфические проекции и виртуальная реальность

Человеческий глаз видит плоскую картину, но мозг преобразует полученное от глаз, так что создается ощущение объемного изображения. Для создания возможности стереовидения необходимо смоделировать проекции для правого и левого глаза. Это так называемые стереοграфические проекции, которые применяются для создания зрительного ощущения в виртуальной реальности.

Преобразование левой картинки: $[S_L] = [T_r][\Pi_z] =$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/f \\ e & 0 & 0 & 1 \end{bmatrix}$$

, где e - это половина расстояния между глазами.

Преобразование правой картинки:

$$[S_R] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/f \\ -e & 0 & 0 & 1 \end{bmatrix}$$

Для просмотра стереοграфических проекций двумя глазами применяют: пассивные очки (цветовой анаглиф, поляризационные); активные затворные очки - пространственно-временное, оно же черестрочно-временное (вертикальная пара) и временное разделение воспроизведения проекций на экране (попеременная пара); шлемы (пространственное разделение); моно и мультиоконные стереοэкраны высокого разрешения (<http://scv.bu.edu/>) с передней подсветкой и задней подсветкой (Stewart Filmscreen), металлизированные стереοэкраны для сохранения поляризации отраженного света).

Стереографика. Элементы технологии



The Scientific Computing and Visualization (**SCV**) group и **Стена-дисплей** (университет Бостона, USA, <http://scv.bu.edu/>, Stewart Filmscreen):

<http://www.bu.edu/tech/support/research/visualization/computergraphicslab/displaywall/>

- [SCV Display Wall Technology](#) (local)
- [OpenRM / Chromium Demo](#) (local)

Стереографические технологии в России:

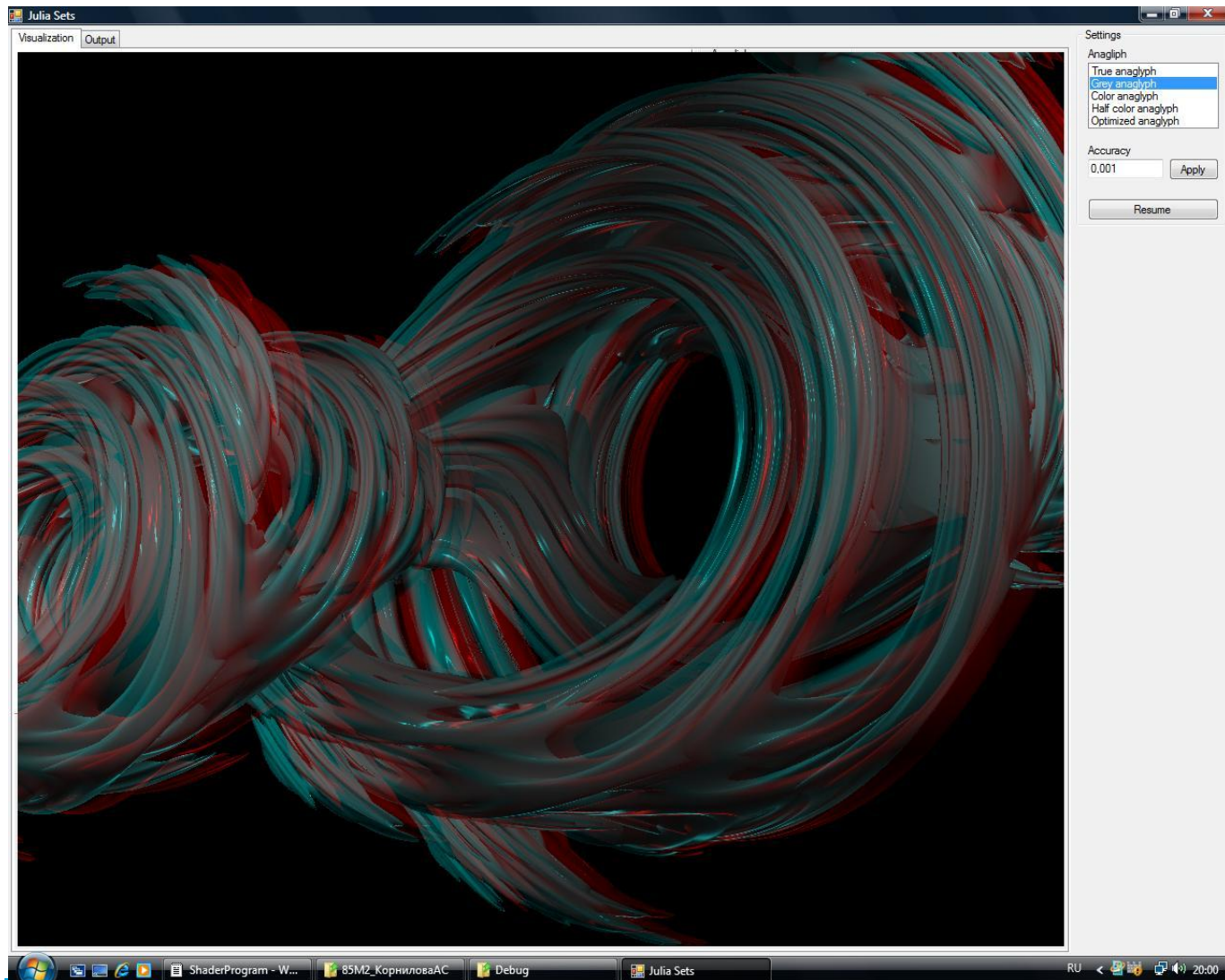
- Варианты построения стереовидео проекции. СТЭЛ ([local](#))
- Компоненты для стереовидеопроекции.
- Активные стереоочки с инфракрасным контроллером



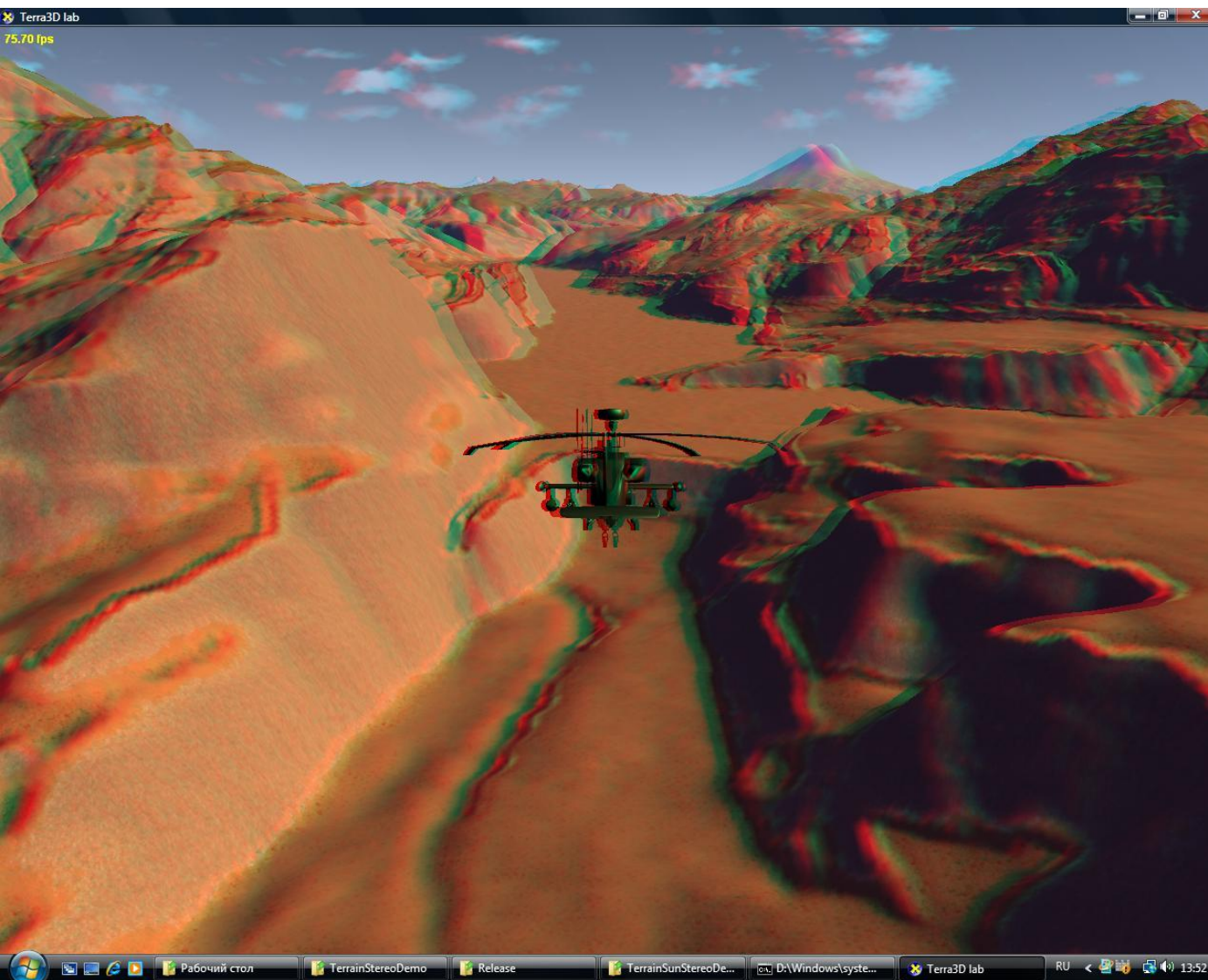
Стереографика.

Серый анаглиф (GrayAnaglyphs)

$$\begin{pmatrix} r_a \\ g_a \\ b_a \end{pmatrix} = \begin{pmatrix} 0,299 & 0,587 & 0,114 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} r_1 \\ g_1 \\ b_1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0,299 & 0,587 & 0,114 \\ 0,299 & 0,587 & 0,114 \end{pmatrix} \cdot \begin{pmatrix} r_2 \\ g_2 \\ b_2 \end{pmatrix}$$



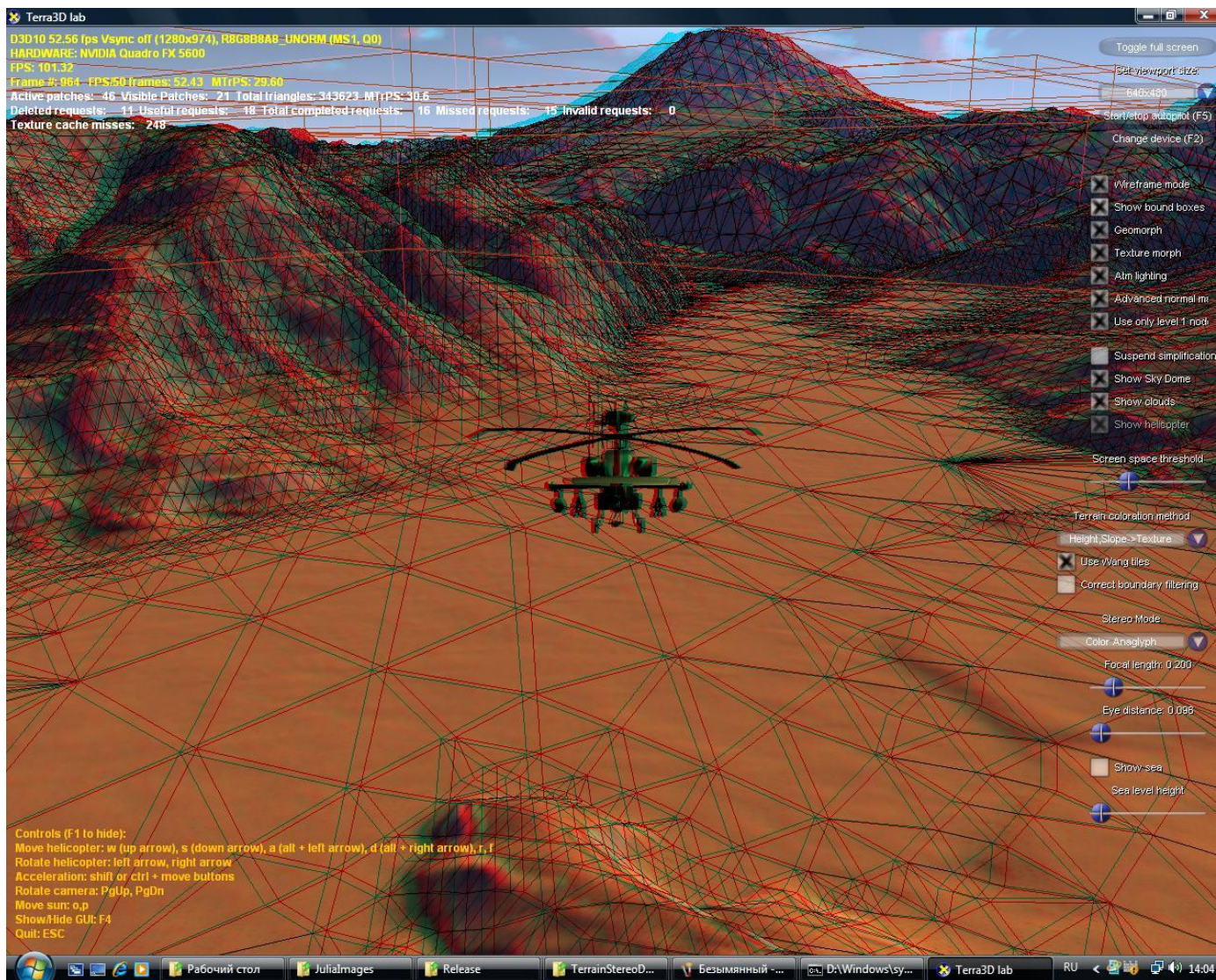
Стереографика. Цветной анаглиф (Color Anaglyphs)



$$\begin{pmatrix} r_a \\ g_a \\ b_a \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} r_1 \\ g_1 \\ b_1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} r_2 \\ g_2 \\ b_2 \end{pmatrix}$$

Частичная цветопередача.
Вызывает усталость глаз,
избытком красных и синих
тонов

Стереографика. Цветной анаглиф (Color Anaglyphs)



$$\begin{pmatrix} r_a \\ g_a \\ b_a \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} r_1 \\ g_1 \\ b_1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} r_2 \\ g_2 \\ b_2 \end{pmatrix}$$

Частичная цветопередача.
Вызывает усталость глаз,
избытком красных и синих
ТОНОВ

Стереографика. Полуцветной анаглиф (Half Color Anaglyphs)

Полуцветной анаглиф

$$\begin{pmatrix} r_a \\ g_a \\ b_a \end{pmatrix} = \begin{pmatrix} 0,299 & 0,587 & 0,114 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} r_1 \\ g_1 \\ b_1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} r_2 \\ g_2 \\ b_2 \end{pmatrix}$$

- Частичная цветопередача (немного хуже, чем в цветном анаглифе)
- Более благоприятен для сетчатки глаз, чем цветной



Motion capture.



Xsens is the leading innovator in 3D motion tracking technology <https://www.xsens.com/>



Motion capture & analysis. Solutions. Samples

the **motion** 504 X 338; 70 КБ - jpg
cdc.gov
[Найти похожие картинки](#)



Test subject 444 X 316; 23 КБ - jpg
cdc.gov
[Найти похожие картинки](#)



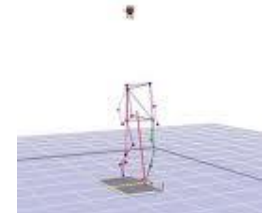
Home > **Motion** 499 X 418; 31 КБ - jpg
zflomotion.com



Motion Analysis 1024 X 768; 60 КБ - jpg
joshuadavidbrown.com
[Найти похожие картинки](#)



Biomechanics and 500 X 461; 36 КБ - jpg
klab.surgery.duke.edu
[Найти похожие картинки](#)



Motion Analysis 400 X 344; 66 КБ - png
binarybutoh.blogspot.com
[Найти похожие картинки](#)



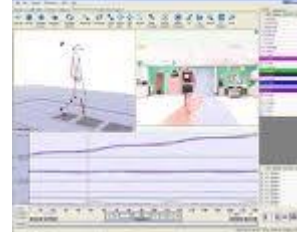
dirty **motion** 300 X 288; 49 КБ - jpg
hpcsport.com



and **Motion** 300 X 259; 11 КБ - jpg
tmsportsperformance.com
[Найти похожие картинки](#)



Motion Analysis 1000 X 799; 216 КБ - jpg
motionanalysis.com



Quantitative Movement 237 X 300; 24 КБ - jpg
hpcsport.com



Motion analysis 631 X 447; 34 КБ - jpg
bsamliq.uku.fi



to further **motion** 500 X 387; 53 КБ - jpg
contemplas.com



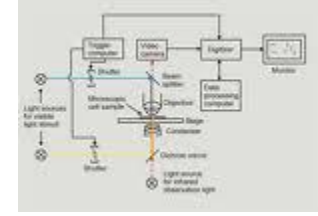
Motion Analysis 600 X 450; 101 КБ - jpg
tech-em.com
[Найти похожие картинки](#)



Motion Analysis 636 X 480; 97 КБ - png
gembarantarel.com
[Найти похожие картинки](#)



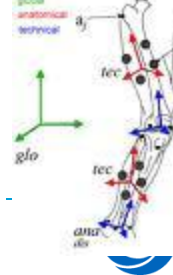
motion analysis 1056 X 745; 50 КБ - jpg
blochem.mpg.de



Taylor Made **Motion** 475 X 475; 32 КБ - jpg
golfacademy.ie



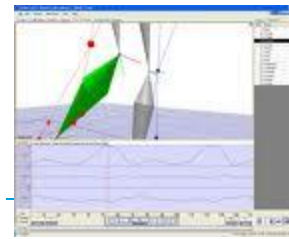
SOUTAS-LITTLE: **Motion** 412 X 550; 36 КБ - jpg
laboratorium.dist...



Sport and **Motion** 300 X 285; 14 КБ - jpg
tmsportsperformance.com
[Найти похожие картинки](#)



Motion Analysis 864 X 689; 103 КБ - jpg
motionanalysis.com



BODY MOTION 320 X 480; 29 КБ - jpg
wldingolf.com



Восстановление трехмерных объектов по перспективным проекциям. Захват движения

Изображения с фотокамеры, видеокамеры, web-камеры являются перспективными проекциями. Тем не менее именно они наиболее часто являются источником данных, по которым восстанавливаются трехмерные координаты реальных объектов, происходит захват движения реальных объектов (примеры [MotionAnalysis www.mastudios.com](http://www.mastudios.com), [Robot Training](http://www.robottraining.com), CMU Graphics Lab Motion Capture Database <http://mocap.cs.cmu.edu/>)

Для восстановления трехмерных координат (x, y, z) какой-либо ключевой точки P используются обычно 2 перспективных проекции, на каждой из которых детектированы координаты (x^*, y^*) ее перспективных проекций $P1^*$ и $P2^*$. Рассмотрим перспективное проецирование искомой точки и уравнения связывающие при этом координаты P и P^* :

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} T_{11} & T_{12} & 0 & T_{14} \\ T_{21} & T_{22} & 0 & T_{24} \\ T_{31} & T_{32} & 0 & T_{34} \\ T_{41} & T_{42} & 0 & T_{44} \end{bmatrix} = h \cdot \begin{bmatrix} x^* & y^* & 0 & 1 \end{bmatrix} \Rightarrow \begin{cases} T_{11}x + T_{21}y + T_{31}z + T_{41} = h \cdot x^* \\ T_{12}x + T_{22}y + T_{32}z + T_{42} = h \cdot y^* \\ T_{14}x + T_{24}y + T_{34}z + T_{44} = h \end{cases} \quad \begin{array}{l} \text{Подставив } h \\ \text{в 2 первых} \\ \text{уравнения} \rightarrow \end{array}$$

$$\begin{cases} (T_{11} - T_{14}x^*)x + (T_{21} - T_{24}x^*)y + (T_{31} - T_{34}x^*)z = -(T_{41} - T_{44}x^*) \\ (T_{12} - T_{14}y^*)x + (T_{22} - T_{24}y^*)y + (T_{32} - T_{34}y^*)z = -(T_{42} - T_{44}y^*) \end{cases} \Rightarrow \text{Для двух проекций получим:} \quad [A1][P] = [b1] \rightarrow \begin{bmatrix} A1 \\ A2 \end{bmatrix} [P] = \begin{bmatrix} b1 \\ b2 \end{bmatrix} \rightarrow [A][P] = [b]$$

4 уравнения для 3 неизвестных, но можно найти решение методом наименьших квадратов (домножив слева на A^T) $\rightarrow$

$$[P] = \left[[A]^T [A] \right]^{-1} \cdot \left[[A]^T [b] \right] \Rightarrow \text{Единственное решение}$$

Размер матриц: $\begin{matrix} 3 \times 4 & 4 \times 3 & 3 \times 4 & 4 \times 1 \end{matrix}$

Источники

1. М. В. Михайлюк. Основы компьютерной графики. Изд-во: ИТЦ `МАТИ`, 2001. -194с.
2. Порев В.Н. Компьютерная графика. –СПб.: БХВ-Петербург, 2002. –432с.
3. Роджерс Д. Алгоритмические основы машинной графики: Пер. с англ. -М.: Мир, 1989. -512 с.
4. Никулин Е.А. Компьютерная геометрия и алгоритмы машинной графики. – С.Пб : БХВ-Петербург, 2003. –560с.
5. Хилл Ф. OpenGL. Программирование компьютерной графики. – С.Пб: Питер, 2002. 1088с.