Движение геометрических объектов

Никулин К геом. и А. М. Граф

## Геометрические преобразования (ГП)

Цель: Овладеть мат. языком описания динамики и визуализации

**Геометрическое преобразование** – отображение $p' = f(p)$ т. $p \in R^n$ n-мерного пр-ва образа в т. $p' \in R^{n'}$ n'-мерного пространства прообраза

Виды

1) Линейное

$$p' = pA + B \Rightarrow \begin{bmatrix} p_1' \\ p_2' \\ p_3' \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} A + \begin{bmatrix} B \\ \vdots \\ B \end{bmatrix}$$

где A и B матрицы преобразования $\in R^{n \times n'}$, $\in R^{1 \times n'}$

$|A| = 0$ вырожденные (проективные)

восстановление исходного изобр. невозможно

$|A| \neq 0$ невырожденные (аффинные)

св-ва: 1) $n = n'$

2) $rang(A) = n$ → квадратная невырожденная

восстановление

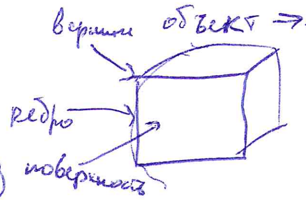
$$p = (p' - B) A^{-1}$$

2) Нелинейное

отражение в кривом зеркале

процесс

объект → (проволочная) каркасная модель



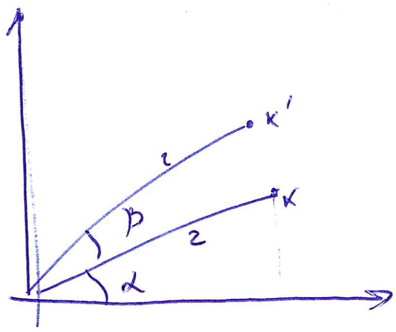

## Аффинные преобразования

родство = affinity

С целью упрощения и унификации записи ГП вводится понятие расширенного пространства (РП) $R^{n+1}$, в к-ом (n+1)-коор. равна 1 или 0

$$x' = a_{xx} x + a_{xy} y + a_{xz} z + b_x$$
$$y' = a_{yx} x + a_{yy} y + a_{yz} z + b_y$$
$$z' = a_{zx} x + a_{zy} y + a_{zz} z + b_z$$

замена с-мы к-мат




$$\begin{matrix} \cos\beta & \sin\beta \\ -\sin\beta & \cos\beta \end{matrix}$$

| | |
|---|---|
| $x = r\cos\alpha$ | $x' = r\cos(\alpha+\beta)$ |
| $y = r\sin\alpha$ | $y' = r\sin(\alpha+\beta)$ |

$$x' = r(\cos\beta\cos\alpha - \sin\beta\sin\alpha) = x\cos\beta - y\sin\alpha$$

$$y' = r(\sin\alpha\cos\beta + \cos\alpha\sin\beta) = x\sin\beta + y\cos\beta$$

---

## Кватернионы 1843 Гамильтон

Решает проблему gimbal lock (карданов подвес)

Расширение комплексных чисел (упорядоч. пара)
гиперкомплексное число.

$q = (s, v)$ ~~s — действит.~~ $s \in R$

$v \to v(x, y, z)$ $x, y, z \in R$

Если нужно повернуть на угол $\theta$ относит. вектора $\bar{u}$

$s = \cos\frac{\theta}{2}$ $\bar{u}$ -> должен быть нормализован

$v = \bar{u}\sin\frac{\theta}{2}$ $q = \left(\cos\frac{\theta}{2}, \bar{u}\sin\frac{\theta}{2}\right)$

$$q \cdot q' = (s, v)\cdot(s', v') = (ss' - \vec{v}\cdot\vec{v}',\ s\vec{v}' + s'\vec{v} + \vec{v}\times\vec{v}')$$

~~s, ix, jy, kz~~ $(s + ix + jy + kz)\cdot(s' + ix' + jy' + kz') =$

$$= ss' + \underline{six' + sjy' + skz'} + s'(ix + jy + kz) + \underline{i^2xx' + ixjy' + ix\cdot kz'}$$

### Вращение

$q' = (s; -v)$

$P = (0, p)$ $p(x, y, z)$ точку, которую для вр. $P' = (0, p')$ $p' = (x', y', z')$

$P' = qPq^{-1}$ $p' = s^2p + v(p\cdot v) + 2s(v\times p) + v\times(v\times p)$

$i^2 = j^2 = k^2 = ijk = -1$

| $1 \backslash 2$ | i | j | k |
|---|---|---|---|
| i | −1 | k | −j |
| j | −k | −1 | i |
| k | j | −i | −1 |

**Кватернионы** (1843 г) → Гамильтон | использ. 1985 г. |  Херн – КГ и стандарт OpenGL

расширение ~~двух~~ комплексных чисел (упорядоченная пара)  gimbal lock
гиперкомплексное

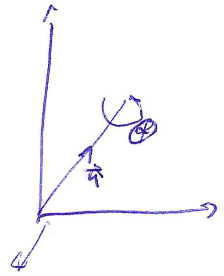
можно также | действ. | скаляр компонента
3 компл. чис. (вектор)

$$q = (s; v) \qquad s = \cos\frac{\theta}{2} \quad ; \quad v = \vec{u}\sin\frac{\theta}{2}$$

$$q^{-1} = (s; v)$$

где $\vec{u}$ – (ось вращения) ед. вектор

$\theta$ → угол поворота



$$P = (0, p) \qquad p = (x, y, z)$$

$$P' = q P q^{-1} \qquad \cancel{q = (s, v)}$$

$$P' = (0, p') \qquad p' = s^2 p + v(p \cdot v) + 2s(v \times p) + v \times (v \times p)$$

$$R_x^{-1}(\alpha) \cdot R_y^{-1}(\beta) \cdot R_z(\theta) \cdot R_y(\beta) \cdot R_x(\alpha)$$

$$v = (a, b, c) \qquad q(s, v)$$

$$M_R(\theta) = \begin{bmatrix} 1 - 2b^2 - 2c^2 & 2ab - 2sc & 2ac + 2sb \\ 2ab + 2sc & 1 - 2a^2 - 2c^2 & 2bc - 2sa \\ 2ac - 2sb & 2bc + 2sa & 1 - 2a^2 - 2b^2 \end{bmatrix}$$

$$\cos^2\frac{\theta}{2} - \sin^2\frac{\theta}{2} = 1 - 2\sin^2\frac{\theta}{2} = \cos\theta$$

$$2\cos\frac{\theta}{2}\sin\frac{\theta}{2} = \sin\theta$$

$$M_R(\theta) = \begin{bmatrix} u_x^2(1-\cos\theta) + \cos\theta & u_x u_y(1-\cos\theta) - u_z\sin\theta \\ u_y u_x(1-\cos\theta) + u_z\sin\theta & u_y^2(1-\cos\theta) + \cos\theta \\ u_z u_x(1-\cos\theta) - u_y\sin\theta & u_z u_y(1-\cos\theta) + u_x\sin\theta \end{bmatrix}$$

$$u_x u_z(1-\cos\theta) + u_y\sin\theta$$
$$u_y u_z(1-\cos\theta) - u_x\sin\theta$$
$$u_z^2(1-\cos\theta) + \cos\theta$$

$$R(\theta) = T^{-1} \cdot M_R \cdot T$$

Кватернион (23)

Боресков. Г 3-х мерн. компьют. игры на основе OpenGL

про проектирование тоже здесь

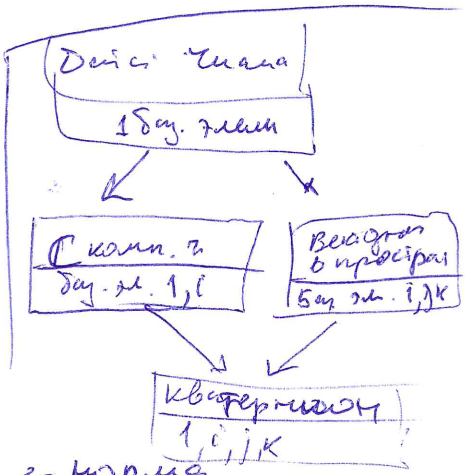
$$q = w + xi + yj + zk \qquad i, j, k \text{ — мнимые единицы}$$

$$ij = -ji = k$$

$$i^2 = -1$$

$$q_1 + q_2 = (w_1 + x_1 i + y_1 j + z_1 k) + (w_2 + x_2 i + y_2 j + z_2 k) =$$

$$q_2 q_1 \neq q_1 \cdot q_2 = w_1 w_2 + i w_1 x_2 + j w_1 y_2$$




$$q^* = w - xi - yj - zk \leftarrow \text{сопряженный кватернион}$$

св-ва:

$$(q^*)^* = q \qquad N(q) = w^2 + x^2 + y^2 + z^2 \leftarrow \text{норма}$$

$$(pq)^* = q^* p^*$$

$$(p+q)^* = p^* + q^*$$

$$q \cdot q^{-1} = q^{-1} \cdot q = 1$$

$$q \cdot q^* = q^* \cdot q = N(q) \quad \Rightarrow \quad q^{-1} = \frac{q^*}{N(q)}$$

---

$$q = [w, v] \qquad q' = [w', v']$$

$$q + q' = [w + w', v + v']$$

$$qq' = [w w' - \overset{\text{скал.}}{v v'}, wv' + v \times v' + vw']$$

$$ww' + vw' + vv' + wv'$$

---

$$(s; v)(0; p)(s; -v) = (-(v, p), v \times p + sp)(s; -v) =$$

$$= (-vps + vsp + (v \times p)v, \; vp + sps + s(v \times p) + v \times p \times v) - sp \times v$$

$$(s, v)(s, p)(s; -v) =$$

$$v' = q \cdot [0, v] \cdot q^{-1}$$

$p' = q p q^{-1}$

$s, \sigma)$

$(s,\sigma)(0,p)(s,-\sigma)$

$\cancel{(s\cdot 0 - \sigma p,\ s\cdot p + \sigma\cdot 0 + \sigma\times p)}$

$\cancel{(-\sigma p,\ s\cdot p + \sigma\times p)(s,-\sigma)}$

$\cancel{(-\sigma p s - \sigma\cdot p\cdot\sigma,\ \sigma p \sigma + s\cdot s p + s\,\sigma\times p + -\sigma\times(\sigma\times p) - s\cdot p\cdot\sigma}$

$(s,\vec{\sigma})(0,\vec{p})$

$(s\cdot 0 - \vec{\sigma}\cdot\vec{p},\ s\cdot\vec{p} + 0\bar{\sigma} + [\bar{\sigma}\times\bar{p}]) = (-\bar{\sigma}\cdot\bar{p},\ s\bar{p} + [\bar{\sigma}\times\bar{p}])$

$(-\bar{\sigma}\cdot\bar{p},\ s\bar{p} + [\bar{\sigma}\times\bar{p}])(s,-\bar{\sigma}) =$

$= -\bar{\sigma}\bar{p}\cdot s + (s\bar{p} + [\bar{\sigma}\times\bar{p}])\cdot\bar{\sigma},$

$-\bar{\sigma}\bar{p}\,(-\bar{\sigma}) + s(s\bar{p} + [\bar{\sigma}\times\bar{p}]) + (s\bar{p} + [\bar{\sigma}\times\bar{p}])\times(-\bar{\sigma}) =$

$[\bar{\sigma}^2\times p]$ $= -\bar{\sigma}\bar{p}(\bar{\sigma}) + s^2\bar{p} + s[\bar{\sigma}\times\bar{p}] + (s\bar{p})\times(-\bar{\sigma}) + \bar{\sigma}\times[\bar{\sigma}\times\bar{p}]$

$\bar{\sigma}[\bar{\sigma}\times p],$ $\overset{\text{скал. произв}}{\sigma(p\cdot\sigma)} + s^2 p + 2s[\bar{\sigma}\times\bar{p}] + \sigma\times[\bar{\sigma}\times p]$

$q p q^{-1}$ $\left(\cos\frac{\Theta}{2} + \bar{u}\sin\frac{\Theta}{2}\right)\bar{\sigma}\left(\cos\frac{\Theta}{2} - \bar{u}\sin\frac{\Theta}{2}\right)$

$\sigma' = \sigma\cos^2\frac{\alpha}{2} + \bar{\sigma}\bar{u}\sin\frac{\Theta}{2}\cdot\cos - \bar{\sigma}\bar{u}\sin\cos$

$\sigma\cos^2\Theta + \bar{u}^2\bar{\sigma}\sin^2$

Кватернион $q = (s, v)$

$s \in R$

$v \in R^3$ $\quad v = v(x, y, z)$

$$q_1 + q_2 = (s_1 + s_2, v_1 + v_2)$$

$$q_1 \cdot q_2 = (s_1 \cdot s_2 - v_1 v_2, s_1 \cdot v_1 + s_2 v_2 + (v_1 \times v_2))$$

$$p' = q p q^{-1} \qquad q^{-1} = (s, -v)$$

---

Трехмерные преобразования

Роджерс
Мат. основы машинной графики

Точки изображаются в виде вектора-столбца
координатный вектор $\begin{bmatrix} x \\ y \\ z \end{bmatrix}$

Преобразования и матрицы.

$A \cdot T = B \qquad T = A^{-1} B$

для чего нужны
1) вращать
2) передвижение
3) масштабировать
4) строить различные виды проекций

чтобы можно было
выполнить операцию
нужно использовать
однородные координаты

$p' = Ap + B$ как $p^{*\prime} = A p^{*}$

$p = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \Rightarrow p^* \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$

Но иногда удобнее записывать

Array $p^*$ Array $p$

$$\begin{bmatrix} x & y & z & 1 \\ & & & \\ & & & \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \\ & & & \\ & & & \end{bmatrix} \begin{bmatrix} a & b & c & p \\ d & e & f & q \\ g & i & j & r \\ l & m & n & s \end{bmatrix}$$

$A \Rightarrow 4 \times 4$

однородная

Билинейное преобразование | 2 линейные преобразован

масштабирование
сдвиг
отражение
вращение

перемещение
перспективное преобразование
общее масштабирование

после операции м.б. $h \neq 1$
тогда $\left[\frac{x}{h} \ \frac{y}{h} \ \frac{z}{h} \ \frac{h}{h}\right]$

① Масштабирование

$$XT = [x\,y\,z\,1] \begin{bmatrix} a & & & \\ & e & & 0 \\ & 0 & j & \\ & & & s \end{bmatrix} = [ax \quad ey \quad jz \quad s]$$

если $|a| = |e| = |z| = |s| = 1$ отражение или ничего ④
однородное расширение / сжатие

② Сдвиг

$$XT = [x\,y\,z\,1] \begin{array}{c} \begin{array}{ccc} x & y & z \end{array} \\ \begin{bmatrix} 1 & b & c & 0 \\ d & 1 & f & 0 \\ g & i & 1 & 0 \\ l & m & n & 1 \end{bmatrix} \end{array} = \begin{bmatrix} x + yd + gz + l \\ bx + y + iz + m \\ cx + fy + z + n \\ 1 \end{bmatrix}$$

Перемещение ⑤

③ Вращение | это вокруг осей, а если нужно вокруг оси вокруг произвольной линии.

С отражением относит. оси тоже самое

$$XT = [x\,y\,z\,1] \begin{bmatrix} \cos\alpha & \sin\alpha & 0 & 0 \\ -\sin\alpha & \cos\alpha & 0 & 0 \\ & 0 & 1 & 0 \\ & & & 1 \end{bmatrix}$$

вращение относительно оси $z$

⑥ Перспективное преобразование

$$\begin{bmatrix} x \\ y \\ z \\ xp + yq + zr + s \end{bmatrix}$$

композиция преобразований
поскольку $AB \neq BA$

проецирование ⑦ на какую либо прямую

геометрические преобразования

Хери _ КГ и стандарт OpenGL (есть примеры)

Борисов - Г 3-хмерных комп. игр на основе OpenGL

Трансляция (сдвиг)
Поворот) вращение
масштабирование

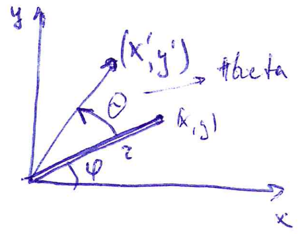
$M_t$ 1) Трансляция → жесткое преобразование · без деформации

трансляция кривой

$p' = p + t$

$x' = x + t_x$
$y' = y + t_y$
$z' = z + t_z$

$M_R$ 2) поворот (вращение) → жесткое преобразование   углы Эйлера



$$\begin{cases} x = r\cos\varphi \\ y = r\sin\varphi \end{cases} \Rightarrow \quad x' = r\cos(\varphi+\theta) = r(\cos\varphi\cos\theta - \sin\varphi\sin\theta)$$
$$y' = r\sin(\varphi+\theta) = r(\cos\varphi\sin\theta + \sin\varphi\cos\theta)$$

$$\begin{cases} x' = x\cos\theta - y\sin\theta \\ y' = x\sin\theta + y\cos\theta \end{cases}$$

$p' = R \cdot p$

$(x\ y) \cdot \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$

матрица транспонирована нужно иметь в виду

замечание: поворот относительно любой оси

Важно: необходимо сдвинуть точки в начало координат

$$p' = T^{-1} R_x(\theta) \cdot T \cdot p$$

$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ относительно диагон. линии

3) масштабирование → изм. разм. объекта

$M_S$

$x' = x \cdot s_x$
$y' = y \cdot s_y$
$z' = z \cdot s_z$

$p' = S \cdot p$   $\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix}$ обращается

$s_x < 1$ → сокращ.
$s_x > 1$ → увелич.
$s_x = 1$ → ~~=~~ относит. и

4) отражение
$s_x = -1$

5) сдвиг (зсув) translate
Shear

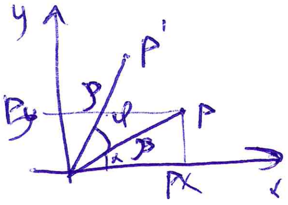
используя однородную матрицу

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & t_x \\ 0 & s_y & 0 & t_y \\ 0 & 0 & s_z & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

углы Эйлера

$$T(x_f, y_f, z_f) \cdot S(s_x, s_y, s_z) \cdot T(-x_f, -y_f, -z_f) =$$
$$= \begin{bmatrix} s_x & 0 & 0 & (1-s_x)x_f \\ 0 & s_y & 0 & (1-s_y)y_f \\ 0 & 0 & s_z & (1-s_z)z_f \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

T → translate → glTranslate (x, y, z)
$R_x$ → rotate → glRotate (α, x, y, z)
S → scale → glScale (x, y, z)




$$\frac{P_x}{P} = \cos\alpha$$

$$\frac{P_y}{P} = \sin\alpha$$

$$\begin{cases} P_x = P\cos\alpha \\ P_y = P\sin\alpha \end{cases}$$

$$\begin{cases} P'_x = P\cos(\alpha+\beta) \\ P'_y = P\sin(\alpha+\beta) \end{cases} \Rightarrow \begin{cases} P'_x = P(\cos\alpha\cos\beta - \sin\alpha\sin\beta) \\ P'_y = P(\cos\alpha\sin\beta + \sin\alpha\cos\beta) \end{cases} \Rightarrow$$

$$\Rightarrow \begin{cases} P'_x = \underline{P\cos\alpha}\cos\beta - \underline{P\sin\alpha}\sin\beta \\ P'_y = \underline{P\cos\alpha}\sin\beta + P\sin\alpha\cos\beta \end{cases} \Rightarrow \begin{cases} P'_x = P_x\cos\beta - P_y\sin\beta \\ P'_y = P_x\sin\beta + P_y\cos\beta \end{cases} =$$

$$= \left(P'\right) = \begin{pmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{pmatrix} \left(P\right)$$

---

Матрица полного преобразования

$$P' = T^{-1} R_x(\alpha)\, T\, P$$

GL_COLOR → переход из одной цвет. ~~режим~~ схемы в другую
GL_TEXTURE → отображения текстурных узлов на поверхности
GL_MODELVIEW → геом. преобразований
GL_PROJECTION → проецирования

---

glMatrixMode - формирует текущую матрицу преобраз. - геом. наблюдения, проецирования, наложения текст. и изменения цвета

glLoadIdentity → текущая матрица становится единичной

glMultMatrix → умножение матриц

glGetIntegerv → значение глубины стека или текущее число матриц в стеке для заданного режима

glPushMatrix | матрицы записываются по столбцам
glPopMatrix | GL_MAX_+//_STACK_DEPTH

---

## 29. Знакомство с OpenGL

Основная библиотека – 1) спецификации граф. примитивов

2) атрибутов

3) геометрич. преобраз.

4) преобраз. наблюд. и т.д.

Граф. библ. независима от аппаратной платформы.

~~Вспомогат.~~ Дополн. ф-ии доступны во вспомогат. библиотеках (I/O, звук и т.д.)

Перед именами ф-ии <u>основной библиотеки</u> OpenGL ставится префикс gl, GL_

GLU → OpenGL Utility ф-ии для работы с матрицами

приставка glu — рисовать сплайны, би-сплайны и т.д.

закрашивать поверхности

Создает область в которой ~~рисуются~~ происходит отрисовка

Расширение OpenGL для с-мы X-Windows (GLX) glx

AppleGL (AGL) agl — PGL и т.д.

Windows-to-OpenGL (WGL) wgl

Open GL Utility Toolkit (GLUT) предлагает библиотеку ф-ий для раб. с окн.

glut – описывать и закрашивать к-вы 2-го порядка

#include <GL/glut.h> подгруж. <GL/gl.h> и <GL/glu.h>

Управление окнами изображения

```
glutInit(&argc, argv);
glutCreateWindow("Hello")
glutDisplayFunc(function)
glutMainLoop();
glutInitWindowPosition
glutWindowSize
glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
```

glutMainLoop(); – завершить операцию по обработке

GLUT_SINGLE — один буфер генерации; GLUT_RGB — цветовой режим

```
glClearColor(1.0, 1.0, 1.0, 0.0);
```

0.0 — альфа-фактор, <u>параметр смешивания.</u> прозрачность

```
glClear(GL_COLOR_BUFFER_BIT)
```

glClear — очистить окно изображения; GL_COLOR_BUFFER_BIT — буфер, параметр из glClearColor

```
glMatrixMode(GL_PROJECTION)
gluOrtho2D(    );
```

параметры проекции → ортогон. проекция.

Матрица преобразований

Модельно видовое преобразование

void glMatrixMode({GL_MODELVIEW, GL_PROJECTION});

void glLoadIdentity(); E

glMultMatrixd(GLdouble c[16]);

glTranslated

glScaled

glRotated

gluPerspective

~~матрица преобр~~

Видовое преобразование

gluLookAt

Кватернионы 1843г. Гамильтон. | Проблема вращения gimbal lock

$$\searrow N, Z, Q, R, \overbrace{C, H}^{\text{гиперкомплекс. ч.}}$$

~~$q_1 + q_2 = (s$~~ $q = (s, v) = s + \bar{i} v_x + \bar{j} v_y + \bar{k} v_z$, где $s, x, y, z \in R$

$$q_1 + q_2 = (s_1, v_1) + (s_2, v_2) = (s_1 + s_1, v_1 + v_2)$$

$v_1 \times v_2$ или $[v_1, v_2]$

$$q_1 \cdot q_2 = (s_1, v_1) \cdot (s_2, v_2) = (s_1 s_2 - \underbrace{(v_1, v_2)}_{\text{скаляр}}, s_1 v_2 + s_2 v_1 + [v_1, v_2])$$

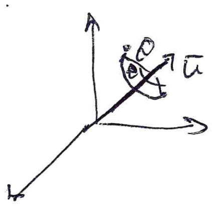
ijk

| 1\2 | i | j | k |
|---|---|---|---|
| i | -1 | k | -j |
| j | -k | -1 | i |
| k | j | -i | -1 |

$q^{-1}$ компл. сопр. число $q^{-1} = (s, -v)$

Как выполнить вращение вокруг произвольной оси $\bar{u}$ на угол $\theta$ | $p' = q\, p\, q^{-1}$

Для этого нужно сделать следующее



1. Координаты точки $p$: $q_p = (0, p)$, где $p(x, y, z)$
2. Ось вращения $q_R = (\cos\frac{\theta}{2}, \bar{u} \sin\frac{\theta}{2})$, где $\bar{u} = (u_x, u_y, u_z)$ → нормир. вектор ед.

   $u = \left(\frac{u_x}{|u|}; \frac{u_y}{|u|}; \frac{u_z}{|u|}\right)$ ← как нормировать
3. $q'_p = q_R\, q_p\, q^{-1}$, ~~где~~ тогда $q'_p = (s', p')$, где $p'$ — резуль. $(x', y', z')$