libgeometry

Rodrigo G. López

rgl@antares-labs.eu

Antares Telecom Laboratories
Albatera, Alicante

1. Data Structures

1.1. Point2

```
struct Point2 {
        double x, y, w;
};
```

Point2 represents a point in two-dimensional projective space, which itself is an extension of the two-dimensional euclidean space that allows us to work with vectors and compose affine transformations in a friendly manner. A point

$$(x,y,w)$$

made out of homogenous coordinates *x*, *y*, and *w*, yields a point with euclidean coordinates

$$(x/w,y/w).$$

1.2. Point3

```
struct Point3 {
        double x, y, z, w;
};
```

Point3 is a point in three-dimensional projective space.

1.3. Matrix

```
typedef double Matrix[3][3];
```

Matrix represents a 3x3 matrix, thought to compose affine transformations to apply to homogeneous 2D points.

1.4. Matrix3

```
typedef double Matrix3[4][4];
```

Matrix3 represents a 4x4 matrix, thought to compose affine transformations to apply to homogeneous 3D points.

ACHTUNG! this is a **WORK IN PROGRESS**

1.5. Quaternion

```
struct Quaternion {
        double r, i, j, k;
};
```

Quaternions are a numbering system that extends the complex numbers up to four-dimensional space, and are used to apply rotations and model mechanical interactions in 3D space. Their main advantages with respect to their matrix relatives are increased computational and storage performance and gimbal lock avoidance.

2. Algorithms

2.1. Point2

Addition

```
Point2 addpt2(Point2 a, Point2 b)
```

$$a+b = \left[x_a+x_b,\ y_a+y_b,\ w_a+w_b\right]$$

Substraction

```
Point2 subpt2(Point2 a, Point2 b)
```

$$a-b = \left[x_a-x_b,\ y_a-y_b,\ w_a-w_b\right]$$

Multiplication

```
Point2 mulpt2(Point2 p, double s)
```

$$p*s = \left[xs,\ ys,\ ws\right]$$

Division

```
Point2 divpt2(Point2 p, double s)
```

$$p/s = \left[\frac{x}{s},\ \frac{y}{s},\ \frac{w}{s}\right]$$

Vector Dot Product

```
double dotvec2(Point2 a, Point2 b)
```

$$\vec{a} \bullet \vec{b} = x_a x_b + y_a y_b$$

Vector Magnitude/Length

```
double vec2len(Point2 v)
```

$$|\vec{v}| = \sqrt{x^2+y^2}$$

Vector Normalization

```
Point2 normvec2(Point2 v)
```

$$\vec{n} = \left[\frac{x}{|\vec{v}|},\ \frac{y}{|\vec{v}|}\right]$$

2.2. Point3

Addition

```
Point3 addpt3(Point3 a, Point3 b)
```

$$a+b = \left[x_a+x_b,\ y_a+y_b,\ z_a+z_b,\ w_a+w_b\right]$$

Substraction

```
Point3 subpt3(Point3 a, Point3 b)
```

$$a-b = \left[x_a-x_b,\ y_a-y_b,\ z_a-z_b,\ w_a-w_b\right]$$

Multiplication

```
Point3 mulpt3(Point3 p, double s)
```

$$p*s = \left[xs,\ ys,\ zs,\ ws\right]$$

Division

```
Point3 divpt3(Point3 p, double s)
```

$$p/s = \left[\frac{x}{s},\ \frac{y}{s},\ \frac{z}{s},\ \frac{w}{s}\right]$$

Vector Dot Product

```
double dotvec3(Point3 a, Point3 b)
```

$$\vec{a} \bullet \vec{b} = x_a x_b + y_a y_b + z_a z_b$$

Vector Cross Product

```
double crossvec3(Point3 a, Point3 b)
```

$$\vec{a} \text{ X } \vec{b} = \left[y_a z_b - z_a y_b,\ z_a x_b - x_a z_b,\ x_a y_b - y_a x_b\right]$$

Vector Magnitude/Length

```
double vec3len(Point3 v)
```

$$|\vec{v}| = \sqrt{x^2+y^2+z^2}$$

Vector Normalization

```
Point3 normvec3(Point3 v)
```

$$\vec{n} = \left[\frac{x}{|\vec{v}|}, \frac{y}{|\vec{v}|}, \frac{z}{|\vec{v}|} \right]$$

2.3. Matrix

Addition

```
void addm(Matrix A, Matrix B)
```

$$(\mathbf{A}+\mathbf{B})_{i,j} = \mathbf{A}_{i,j}+\mathbf{B}_{i,j}$$

Substraction

```
void subm(Matrix A, Matrix B)
```

$$(\mathbf{A}-\mathbf{B})_{i,j} = \mathbf{A}_{i,j}-\mathbf{B}_{i,j}$$

Multiplication

```
void mulm(Matrix A, Matrix B)
```

$$\left[\mathbf{AB}\right]_{i,j} = \sum_{k=0}^{3-1} \mathbf{A}_{i,k}\mathbf{B}_{k,j}$$

Transpose

```
void transposem(Matrix M)
```

$$(\mathbf{M}^{T})_{i,j} = \mathbf{A}_{j,i}$$

Identity

```
void identity(Matrix M)
```

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Determinant

```
double detm(Matrix M)
```

$$\det(\mathbf{M}) = \begin{array}{l} m_{00}(m_{11}m_{22}-m_{12}m_{21})+ \\ m_{01}(m_{12}m_{20}-m_{10}m_{22})+ \\ m_{02}(m_{10}m_{21}-m_{11}m_{20}) \end{array}$$

2.4. Matrix3

Addition

```
void addm3(Matrix3 A, Matrix3 B)
```

$$(\mathbf{A}+\mathbf{B})_{i,j} = \mathbf{A}_{i,j}+\mathbf{B}_{i,j}$$

Substraction

```
void subm3(Matrix3 A, Matrix3 B)
```

$$(\mathbf{A}-\mathbf{B})_{i,j} = \mathbf{A}_{i,j}-\mathbf{B}_{i,j}$$

Multiplication

```
void mulm3(Matrix3 A, Matrix3 B)
```

$$\left[\mathbf{AB}\right]_{i,j} = \sum_{k=0}^{4-1} \mathbf{A}_{i,k}\mathbf{B}_{k,j}$$

Transpose

```
void transposem3(Matrix3 M)
```

$$(\mathbf{M}^{T})_{i,j} = \mathbf{A}_{j,i}$$

Identity

```
void identity3(Matrix3 M)
```

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Determinant

```
double detm3(Matrix3 M)
```

$$\det(\mathbf{M}) = \begin{array}{l} m_{00}(m_{11}(m_{22}m_{33}-m_{23}m_{32})+m_{12}(m_{23}m_{31}-m_{21}m_{33})+m_{13}(m_{21}m_{32}-m_{22}m_{31})) \\ -m_{01}(m_{10}(m_{22}m_{33}-m_{23}m_{32})+m_{12}(m_{23}m_{30}-m_{20}m_{33})+m_{13}(m_{20}m_{32}-m_{22}m_{30})) \\ +m_{02}(m_{10}(m_{21}m_{33}-m_{23}m_{31})+m_{11}(m_{23}m_{30}-m_{20}m_{33})+m_{13}(m_{20}m_{31}-m_{21}m_{30})) \\ -m_{03}(m_{10}(m_{21}m_{32}-m_{22}m_{31})+m_{11}(m_{22}m_{30}-m_{20}m_{32})+m_{12}(m_{20}m_{31}-m_{21}m_{30})) \end{array}$$

2.5. Quaternion

Addition

```
Quaternion addq(Quaternion q, Quaternion r)
```

$$q+r = (r_q+r_r,\ i_q+i_r,\ j_q+j_r,\ k_q+k_r)$$

Substraction

```
Quaternion subq(Quaternion q, Quaternion r)
```

$$q - r = (r_q - r_r,\ i_q - i_r,\ j_q - j_r,\ k_q - k_r)$$

Multiplication

```
Quaternion mulq(Quaternion q, Quaternion r)
```

$$q = [r_q,\ \vec{v}_q]\, r = [r_r,\ \vec{v}_r]\, qr = [r_q r_r - \vec{v}_q \bullet \vec{v}_r,\ \vec{v}_r r_q + \vec{v}_q r_r + \vec{v}_q X \vec{v}_r]$$

Inverse

```
Quaternion invq(Quaternion q)
```

$$q^{-1} = \left[\frac{r}{|q|^2},\ \frac{-i}{|q|^2},\ \frac{-j}{|q|^2},\ \frac{-k}{|q|^2}\right]$$

Magnitude/Length

```
double qlen(Quaternion q)
```

$$|q| = \sqrt{r^2 + i^2 + j^2 + k^2}$$