# CGRA

Telmo Baptista

March 25, 2020

# 1 Geometric Transformations

## 1.1 2D Geometric Transformations

Transformations:

- Translation
- Scaling
- Rotation

### 1.1.1 Translation

$$\begin{cases} x_T = x + T_x \\ y_T = y + T_y \end{cases} \tag{1}$$

Matricial form:

$$\begin{bmatrix} x_T \\ y_T \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2}$$

### 1.1.2 Scaling

Regarding the origin:

$$\begin{cases} x_S = x \times S_x \\ y_S = y \times S_y \end{cases} \tag{3}$$

Matricial form:

$$\begin{bmatrix} x_S \\ y_S \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{4}$$

Scaling factor:

- $> 1$ - increases the object
- $< 1$ - decreases the object
- $S_x = S_y$ - uniform scaling factor $\rightarrow$ doesn't distort object

### 1.1.3 Rotation

Around the origin:

$$\begin{cases} x = R.cos(\alpha) \\ y = R.sin(\alpha) \end{cases} \tag{5}$$

$$\begin{cases} x_R = R.cos(\alpha + \beta) = R.cos(\alpha).cos(\beta) - R.sin(\alpha).sin(\beta) = x.cos(\beta) - y.sin(\beta) \\ y_R = R.sin(\alpha + \beta) = R.cos(\alpha).sin(\beta) - R.sin(\alpha).cos(\beta) = x.sin(\beta) + y.cos(\beta) \end{cases} \tag{6}$$

Matricial form:

$$\begin{bmatrix} x_R \\ y_R \end{bmatrix} = \begin{bmatrix} cos(\beta) & -sin(\beta) \\ sin(\beta) & cos(\beta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{7}$$

## 1.2 3D Geometric Transformations

Transformations:

- Translation
- Scaling
- Rotation

| Rotation Axis | Positive Rotation Direction |
|---|---|
| $x$ | $y \rightarrow z$ |
| $y$ | $z \rightarrow x$ |
| $z$ | $x \rightarrow y$ |

### 1.2.1 Translation

$$\begin{cases} x_T = x + T_x \\ y_T = y + T_y \\ z_T = z + T_z \end{cases} \tag{8}$$

Matricial form:

$$\begin{bmatrix} x_T \\ y_T \\ z_T \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{9}$$

### 1.2.2 Scaling

Regarding the origin:

$$\begin{cases} x_S = x \times S_x \\ y_S = y \times S_y \\ z_S = z \times S_z \end{cases} \tag{10}$$

Matricial form:

$$\begin{bmatrix} x_S \\ y_S \\ z_S \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{11}$$

Regarding an arbitrary point:

$$\begin{cases} x_S = x \times S_x \\ y_S = y \times S_y \\ z_S = z \times S_z \end{cases} \tag{12}$$

Matricial form:

$$T(x_F, y_F, z_F).S(S_x, S_y, S_z).T(-x_F, -y_F, -z_F) = \begin{bmatrix} S_x & 0 & 0 & (1 - S_x).x_F \\ 0 & S_y & 0 & (1 - S_y).y_F \\ 0 & 0 & S_z & (1 - S_z).z_F \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{13}$$

### 1.2.3 Rotation

Around the $z$ axis $\rightarrow$ $z$ constant

$$\begin{cases} x_{Rz} = x.cos(\alpha) - y.sin(\alpha) \\ y_{Rz} = x.sin(\alpha) + y.cos(\alpha) \\ z_{Rz} = z \end{cases} \tag{14}$$

Matricial form:

$$\begin{bmatrix} x_{Rz} \\ y_{Rz} \\ z_{Rz} \\ 1 \end{bmatrix} = \begin{bmatrix} cos(\alpha) & -sin(\alpha) & 0 & 0 \\ sin(\alpha) & cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{15}$$

Around the $x$ axis $\rightarrow$ $x$ constant

$$\begin{cases} x_{Rz} = x \\ y_{Rz} = y.cos(\alpha) - z.sin(\alpha) \\ z_{Rz} = y.sin(\alpha) + z.cos(\alpha) \end{cases} \tag{16}$$

Matricial form:

$$\begin{bmatrix} x_{Rz} \\ y_{Rz} \\ z_{Rz} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\alpha) & -sin(\alpha) & 0 \\ 0 & sin(\alpha) & cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{17}$$

Around the $y$ axis $\rightarrow$ $y$ constant

$$\begin{cases} x_{Rz} = x.cos(\alpha) + y.sin(\alpha) \\ y_{Rz} = y \\ z_{Rz} = -x.sin(\alpha) + z.cos(\alpha) \end{cases} \tag{18}$$

Matricial form:

$$\begin{bmatrix} x_{Rz} \\ y_{Rz} \\ z_{Rz} \\ 1 \end{bmatrix} = \begin{bmatrix} cos(\alpha) & 0 & sin(\alpha) & 0 \\ 0 & 1 & 0 & 0 \\ -sin(\alpha) & 0 & cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{19}$$

# 2 Illumination Model

The lighting models express the lighting components that define the intensity of light reflected by a given surface, allowing the calculation of the color for each surface point of the objects contained in the image
The incident light on the face is reflected in two ways:

- Diffuse Reflection - light reflects in all directions, with an equal intensity value, due to the roughness of the reflecting surface
- Specular Reflection - point sources of light produce over-lit areas on the reflecting surface

## 2.1 Elementary Illumination Model

### 2.1.1 Ambient Lighting

Corresponds to the diffuse lighting, whose light from numerous reflections

$$I = k_a.I_a \tag{20}$$

- $k_a$ - ambient (diffuse) reflection coefficient of the face, varies between 0 and 1
- $I$ - observed intensity

The intensity $I_a$ is constant in all direction. If we only considered this component to define the light reflected by the object, then all faces would have the same luminous intensity
The reflected light is uniform across the face and independent of the observer's position
The edges are not distinguishable

### 2.1.2 Diffuse Reflection

The diffuse reflection due to a **point light source** is calculated according to Lambert's Law: the reflected light intensity depends on the angle of illumination
The intensity observed on the object varies, depending on the orientation of the surface and the distance to the light source
**Note**: The intensity of reflected light doesn't depend on the position of the observer. It depends on the angle of incidence of the light

$$I = \frac{k_d.I_p}{d + d_0} cos(\theta) \tag{21}$$

$$cos(\theta) = N.L \tag{22}$$

Vectors are unitary:

- $\theta$ - angle of incidence of the light source
- $N$ - normal to the source (unit vector)
- $L$ - direction of the illumination beam (incident radius)
- $I_p$ - light source intensity
- $K_d$ - diffuse reflection coefficient

**Adding the two components**:

$$I = k_a.I_a + \frac{k_d.I_p}{d + d_0} N.L \tag{23}$$

### 2.1.3 Specular Reflection/Phong's Model

Observed reflection on polished surfaces

- $R$ - maximum reflection direction
- $\alpha$ - angle between $R$ and the direction of the observer $V$

$$I_s = \frac{k_s.I_p}{d+d_0} cos^n(\alpha) \tag{24}$$

The specular reflection depends on the position of the observer
$k_s$ is a constant that depends on the material, as well as the exponent $n$ (strictly speaking, one should use a function $W(\theta)$ instead of $k_s$)
On an ideal surface (ideal mirror), the light reflected only in the $R$ direction
On a non-ideal surface, the $R$ direction will have the greatest reflection intensity, the other directions will have lesser intensities
The intensity of the specular reflection is proportional to $cos^n(\alpha)$, where $n$ depends on the characteristics of the surface (value 1 for unpolished surfaces and 200 for perfectly polished faces)
If $V$ and $R$ are unitary vectors:

$$I_s = \frac{k_s.I_p}{d+d_0} cos^n(\alpha) = \frac{k_s.I_p}{d+d_0}(V.R)^n \tag{25}$$

### 2.1.4 Elementary Model

The expression for the intensity results on:

$$I = k_a.I_a + I_p[\frac{k_d}{d+d_0} N.L + \frac{k_s}{d+d_0}(R.V)^n] \tag{26}$$

Reflection coefficients:

- $k_a$ and $k_d$ are commonly the same

Can be broken into components colored (RGB or other):

- $I$, $I_a$, $I_p$
- $k_a$, $k_d$
- $k_s$
- $n$

### 2.1.5 Refraction (for modeling transparent objects)

When the object is transparent, it is necessary to predict the light that passes through a face, it is called transmitted light or refracted light
Because the speed of light is different in different materials, the angle of refraction is different materials, the angle of refraction is different from the angle of incidence

- $\eta_i$ - refractive index of air
- $\eta_r$ - material refractive index
- $\eta$ - is obtained for a given material as the ratio between the speed of light in the void and the speed in the material

**Snell's Law**:

$$sin(\theta_r) = \frac{\eta_i}{\eta_r} sin(\theta_i) \tag{27}$$

### 2.1.6 Calculation of Vector R is complex...

$$\vec{L} + \vec{R} = \vec{N}.2.|\vec{R}|.cos(\theta) \tag{28}$$

$$\vec{R} = 2.\vec{N}.(\vec{N}.\vec{L}) - \vec{L} \tag{29}$$

# 3 Shading and Textures

## 3.1 Shading & Smooth Shading

**Objective**: Calculate the color of each point of the visible surface
Solution **brute-force**: calculate the normal at each point and apply the desired illumination model
**Different methods**:

- Constant Shading
- Interpolated Shading = Smooth Shading
  - Gouraud Method
  - Phong Method