Image synthesis

Romain Vergne
What is it?

Geometry
BRDF
Light sources
Scene org.
Etc.

Physics
Optics
Vis. attention
Vis. perception

Image

shape
material
lighting
space
For what?

Video games

@Battlefield
For what?

Animations
For what?

Movies / special effects

@Exodus
For what?
Advertising
For what?
Computer aided design

http://www.starjammer-engineering.com/cad.php
For what?

Cultural heritage

https://fredericduuvivier.wordpress.com
For what?

Education

For what?
Scientific visualization

http://datavisualization.ch/showcases/scientific-visualizations-on-ted/
For what?

Medical visualization

[Burns et al. 2007]
For what?

Simulation

http://printf.eu/le-meilleur-simulateur-de-vol/
For what?

Virtual reality

http://www.telegraph.co.uk
More than you would expect...

https://vimeo.com/100095868
Multiple topics

- Modelling
Multiple topics

- Modelling
- Animation
Multiple topics

- Modelling
- Animation
- Rendering
Material: Graphics Processing Unit (GPU), since 1980s
- Specialized for graphics calculations
- Hundreds of operations in parallel
- Programmable!

Controllable via low-level API
- OpenGL, Direct3D

Programmable via specific languages
- GLSL, HSLS, CG
A 3D world

3D Models

Lights

Materials

Cameras
A 3D world

3D Models

Lights

Cameras

Materials

Rendering
3D models

- Meshes (mostly triangles)
- Particle / splats
- Boundary representation
- Constructive Solid Geometry (CSG)
- Billboards
- Implicit functions
- …
3D models

- Scene graph
3D models

- Scene graph
3D models

- Scene graph
3D models

- Scene graph
Materials

- Matte
- Glossy
- Transparent
- Translucent
- Mirror
- ...

18 FREE MATERIALS FOR VRay
Includes Glasses/Metals/Plastics
Great for beginners just starting out or for the experienced user wanting to build up their library

...another great freebie from www.paulw.deviantart.com

Lights

- Point light
- Directional light
- Spot light
- Area light
- ...

http://digital-lighting.150m.com/ch02lev1sec2.html
Cameras

- Orthographic
- Perspective
- Fish eye
- Multi-perspective
- Lens properties
- ...


In 2D

- $E = \mathbb{R}^2$
- $e_1 = (1, 0)^T$ (the x-axis)
- $e_2 = (0, 1)^T$ (the y-axis)
- $B = (e_1, e_2) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
- $x = (-2, 1)^T = -2e_1 + 1e_2$
Mathematical tools

### In 2D
- $E = \mathbb{R}^2$
- $e_1 = (1, 0)^T$ (the x-axis)
- $e_2 = (0, 1)^T$ (the y-axis)
- $B = (e_1, e_2) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
- $\mathbf{x} = (-2, 1)^T = -2e_1 + 1e_2$

### In 3D
- $E = \mathbb{R}^3$
- $e_1 = (1, 0, 0)^T$ (the x-axis)
- $e_2 = (0, 1, 0)^T$ (the y-axis)
- $e_3 = (0, 0, 1)^T$ (the z-axis)
- $B = (e_1, e_2, e_3) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
- $\mathbf{x} = (a_x, a_y, a_z)^T = a_x e_1 + a_y e_2 + a_z e_3$
Mathematical tools

- Vector norm

\[ |x| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} \]
Mathematical tools

- Vector norm
  \[ |\mathbf{x}| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} \]

- Normalization
  \[ \hat{\mathbf{x}} = \frac{\mathbf{x}}{|\mathbf{x}|} = \left( \frac{x_1}{|\mathbf{x}|}, \frac{x_2}{|\mathbf{x}|}, \ldots, \frac{x_n}{|\mathbf{x}|} \right)^T \]
### Mathematical tools

- **Vector norm**
  
  $|\mathbf{x}| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$

- **Normalization**
  
  $\hat{\mathbf{x}} = \frac{\mathbf{x}}{|\mathbf{x}|} = \left( \frac{x_1}{|\mathbf{x}|}, \frac{x_2}{|\mathbf{x}|}, \cdots, \frac{x_n}{|\mathbf{x}|} \right)^T$

- **Scalar product**
  
  \[ a \cdot b = |a||b|\cos(\theta) \]
  
  OR
  
  \[ a \cdot b = a_x b_x + a_y b_y + a_z b_z \]
Mathematical tools

- Vector norm
  \[ |\mathbf{x}| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} \]

- Normalization
  \[ \hat{\mathbf{x}} = \frac{\mathbf{x}}{|\mathbf{x}|} = \left( \frac{x_1}{|\mathbf{x}|}, \frac{x_2}{|\mathbf{x}|}, \cdots, \frac{x_n}{|\mathbf{x}|} \right)^T \]

- Scalar product
  \[ \mathbf{a} \cdot \mathbf{b} = |\mathbf{a}| |\mathbf{b}| \cos(\theta) \]
  \[ \mathbf{a} \cdot \mathbf{b} = a_x b_x + a_y b_y + a_z b_z \]

- Cross product
  \[ \mathbf{a} \times \mathbf{b} = |\mathbf{a}| |\mathbf{b}| \sin(\theta) \mathbf{n}, \text{ with } \mathbf{n}: \text{a unit perpendicular vector to the plane} \]
  \[ \mathbf{a} \times \mathbf{b} = -\mathbf{b} \times \mathbf{a} = \begin{vmatrix} e_1 & e_2 & e_3 \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{vmatrix} = \begin{pmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \end{pmatrix} \]
What is the angle between 2 normalized vectors a and b?

What is the normal of a triangle, given its 3 vertices p1, p2 and p3?

What is the area of this triangle?
2D Transformations

- Scaling

Scaling transformation
Multiply each coordinate by a scaling factor:

\[
\begin{pmatrix}
  x' \\
  y'
\end{pmatrix} =
\begin{pmatrix}
  s_x x \\
  s_y y
\end{pmatrix}
\]

MATRIX FORM: \( P' = SP \)

\[
\begin{pmatrix}
  x' \\
  y'
\end{pmatrix} =
\begin{pmatrix}
  s_x & 0 \\
  0 & s_y
\end{pmatrix}
\begin{pmatrix}
  x \\
  y
\end{pmatrix}
\]
2D Transformations

- Translation

Translation transformation
Translation is a simple vectorial sum

\[
\begin{align*}
(x') &= (x + t_x) \\
(y') &= (y + t_y)
\end{align*}
\]

Matrix form: \( P' = P + T \)

\[
\begin{align*}
(x') &= (x) + (t_x) \\
(y') &= (y) + (t_y)
\end{align*}
\]
2D Transformations

- Rotation

Rotation transformation
Rotate each vector around the origin:

\[
\begin{pmatrix}
  x' \\
  y'
\end{pmatrix} = \begin{pmatrix}
  \cos(\theta)x - \sin(\theta)y \\
  \sin(\theta)x + \cos(\theta)y
\end{pmatrix}
\]

MATRIX FORM: \( P' = R_\theta P \)

\[
\begin{pmatrix}
  x' \\
  y'
\end{pmatrix} = \begin{pmatrix}
  \cos(\theta) & -\sin(\theta) \\
  \sin(\theta) & \cos(\theta)
\end{pmatrix} \begin{pmatrix}
  x \\
  y
\end{pmatrix}
\]
2D Transformations

- Rotation

What is the rotation matrix for 0 rad?
- What is the rotation matrix for \( \pi/2 \) rad?
- What is the rotation matrix for \( \pi \) rad?
- What is the rotation matrix for \( \pi/4 \) rad?
- Draw the axis for each of these angles
- What can we conclude?
Notation issue

- Scaling = matrix multiplication 😊
- Rotation = matrix multiplication 😊
- Translation = matrix addition 😞

We would like a unified representation to concatenate multiple transformations…

How can we modify the translation to become a matrix multiplication?
Homogeneous coordinates

- Add a third coordinate to a 2D point, which become a vector

\[
\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ w \end{pmatrix}
\]

- 2 points are equal if and only if \[x_1/w_1 = x_2/w_2 \text{ and } y_1/w_1 = y_2/w_2\]

- A point is projected a infinity when \(w=0\)
  - Differentiate points and vectors
  - Affine spaces
  - Projection transformations
Translation followed by a rotation:

\[ M = \text{RotationMatrix}(\text{theta}) \times \text{TranslationMatrix}(sx, sy) \]
2D Transformations

- Consider the following matrices
  - Rotation $R$ with theta = $\pi/2$
  - Translation $T$ with $tx$ and $ty$ equal to 2
- Consider the point $p$ with homogeneous coord $(1,0,1)$

- Compute $T^*(R^*p)$ and $R^*(T^*p)$

- What can we conclude?
3D transformations

- Exactly the same… with 4D matrices

**Scaling**

\[ S = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \]

**Translation**

\[ T = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \]

**Rotations** – along the 3 different axis

\[ R_z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \]

\[ R_y = \begin{pmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \]

\[ R_z = \begin{pmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \]

Invariant along the x axis

Invariant along the y axis

Invariant along the z axis
Hierarchical transformations

- Complex objects
Hierarchical transformations

- Complex objects
  - $M$: initial matrix
  - pushmatrix()  
  - $M = MR_\alpha T$
  - draw A
  - $M = MTR_\beta T$
  - draw B
  - pushmatrix()  
  - $M = MTR_\gamma^1 T$
  - draw C1
  - popmatrix()  
  - pushmatrix()  
  - $M = MTR_\gamma^2 T$
  - draw C2
  - popmatrix()  
  - pushmatrix()  
  - $M = MTR_\gamma^3 T$
  - draw C3
  - popmatrix()  
  - popmatrix()
OpenGL

- Low-level library
  - Developed in 1989 (GL) by Silicon Graphics
  - Extended to other architectures since 1993 (OpenGL)
  - Independent of architecture and programming language
  - Often used by other high-level libraries (Qt, Java3D, etc)

- Interface to communicate with the GPU
  - Geometry (arrays of positions and attributes stored in the GPU)
  - Appearance (colors, materials, textures)
  - Transformations & projections (matrices / camera / etc)
OpenGL (syntax)

- Constants
  - GL_CONSTANT

- Types
  - GLtype

- Functions
  - glFunction

- Documentation
  - [www.opengl.org](http://www.opengl.org)
WebGL

- Calls to OpenGL functions via Javascript (without plugins)
- Supported by multiples browsers (chrome, firefox 4.0, etc)
- Specifications by Kronos Group
- Use the html5 element <canvas>

https://www.khronos.org/webgl/
<head>
  <script type="text/javascript">
    /* ... */
  </script>
</head>

<body onload="start()">
  <canvas id="glcanvas" width="640" height="480">
    Your browser doesn't appear to support the HTML5 <code>&lt;canvas&gt;</code> element.
  </canvas>
</body>
<script type="text/javascript">
  var canvas;
  var gl;

  function start() {
    canvas = document.getElementById("glcanvas");
    initWebGL(canvas);
    if (gl) { // initialisation }
  }

  function initWebGL() {
    gl = null;
    try { gl = canvas.getContext("experimental-webgl"); } catch(e) { }
    if (!gl) {alert("Impossible d’initialiser WebGL!"); }
  }
</script>