Review of Thursday

• Level-Of-Detail Methods
  • Reduce or increase geometric complexity when needed
  • Exploit properties of perspective projection

• Procedural Methods
  • Details on-demand
  • Language-Based Systems
  • Trees, Plants

• Perlin Noise
  • Nature is “random”
  • White noise is not optimal; Perlin noise better
Level-Of-Detail Techniques

• Applying mesh operations can be tricky
  • Suitable mesh format needed
  • Requires good simplification heuristics
  • Special cases need to be handled to avoid producing invalid meshes

• Is there an easier way to change visual and rendering complexity?
  • Answer: Yes! Use the fragment processor!
Alpha Mapping

- We typically modeled objects by defining their geometry

bay

bay with (untextured) ship geometry

- Can we do better with less triangles?
Alpha Mapping

- We can use a detailed texture and mask out irrelevant regions
Alpha Mapping

- This is called **alpha mapping** or **alpha masking**
- In 2D this can be used to give an object a highly detailed silhouette

- Can we also ‘fake’ geometry details in 3D?
Normal Mapping

- Fine mesh details are perceived through lighting
- Details are often not prevalent in object shape/silhouette

Stanford bunny with lighting

Stanford bunny without lighting
Normal Mapping

- **Idea:** Use a low resolution mesh whose shape is very similar to the original and add some extra fine-detail information for the lighting

- This can be realized with the help of normal maps
Normal Mapping

- **Normal maps** allow us to specify normal vectors at positions other than vertices.
- A normal map is a texture, whose three components are interpreted as normal vectors rather than colors.
Normal Mapping

- Normal mapping best suited for simulating small surface details (bumps), since object silhouette does not change

- Normal mapping is a recent version of **bump mapping**
  - Geometry of the object remains unaltered

- Computationally efficient:
  - Normals are given in tangent space
  - Normals in object space are produced by rotation
Normal Mapping

- Converting normals from tangent space to object space

- Normals are defined in the coordinate frame of the texture

- In object space, the texture plane is rotated to match the plane of the triangle

- To compute the rotation from tangent space to object space, we save normals AND tangents per vertex

\[
\begin{pmatrix}
T_x & B_x & N_x \\
T_y & B_y & N_y \\
T_z & B_z & N_z 
\end{pmatrix}
\]

Tangent in object space

Bitangent in object space

Normal in object space
Parallax Mapping

• Normal mapping is only really suitable for bumps.
• We know that larger objects would lead to parallax effects and occlusion if modeled properly.

• **Parallax mapping** additionally simulates this parallax effect.
Parallax Mapping

- **Parallax mapping** shifts texture coordinates of pixels, based on a **height-map texture** and current viewing parameters.
Parallax Mapping

- Texture coordinate shift approximation:

\[ \Delta T = \text{depth}(T_0) \cdot \tan(\theta) \]

- Heights defined w.r.t. normal (using tangent space)
- Approximation becomes worse for large angles
Parallax Mapping

- Improved forms are more accurate and allow self-shadowing:
  - **Parallax Occlusion Mapping**
    - Samples the height field along ray profile
  - **Relief Mapping**
    - Performs local ray-tracing against height-field
    - Combines linear and binary search to find ray-height-field intersection

- Creating ‘real’ silhouettes takes additional effort
From local to global effects

• All effects we have talked about were local
  • Geometry simplification/refinement
  • Normal shading/Normal Mapping
  • Parallax Mapping

• How can we include global effects in an efficient way?
Reflections

- **Environment mapping** is a technique that approximates scene reflections on highly specular objects.
Environment Mapping

• Reflections are approximated by two-step process
  • Render the scene as ‘seen’ by the reflective object
  • Draw rendered scene on object by performing appropriate texture mapping (coordinates are view dependent)
Environment Mapping

• **Sphere mapping**
  • Sphere map represents scene as reflected in chrome sphere

Dynamic texture coordinates:

\[ r = v - 2(n \cdot v)n \]

\[ p = \sqrt{r_x^2 + r_y^2 + (r_z + 1)^2} \]

\[ s = \frac{r_x}{2p} + \frac{1}{2} \]

\[ t = \frac{r_y}{2p} + \frac{1}{2} \]
Environment Mapping

- **Cube Mapping**: scene represented as ‘seen’ from the six sides of a cube
Environment Mapping

- Example

CC-SA Seb Przd
Environment Mapping

- Example

From: Wikimedia
Environment Mapping

- Environment mapping can also be used for **refraction**

- This is especially powerful when combined with normal mapping
From local to global effects

- **Shadows** are another important non-local effect (includes self-shadowing)

  ![Diagram showing shadows]

- We will talk about two approaches:
  - Shadow mapping
  - Shadow volumes (stencil shadows)
Shadows

- **Shadow mapping**
  - Render scene from point-of-view of light
  - Save depth values in texture
  - Render scene from POV of camera
    - Compare distance of fragment to light to its value in ‘light’ depth buffer
    - If distance > depth, then fragment lies in shadow
Shadows

• **Shadow volumes** create shadow by clipping geometry against volumes cast by light
Shadows

- In OpenGL shadow volumes can be implemented efficiently using the **stencil buffer**
  - Draw objects that can receive shadows
  - Disable write to depth and color buffer
  - Draw front-faces of all shadow volumes (increase stencil value of all fragments that pass depth test)
  - Draw back-faces of all shadow volumes (decrease stencil value of all fragments that pass depth test)
  - Enable write to color buffer (and depth, if required)
  - Draw shadow color everywhere, where stencil buffer > 0
Shadows

- **Shadow volumes vs. Shadow maps**
  - **Shadow volumes**
    - Require the extraction of silhouettes/volumes
    - Exact per-pixel shadows
    - Shadows generally have sharp edges
  - **Shadow maps**
    - Inherently a two/multi-pass method
    - Resolution of depth map governs resolution of shadows
    - Shadow maps can be blurred