Image-based Modeling and Rendering

5. 3D Warping and Relief texture

National Chiao Tung Univ, Taiwan

By: I-Chen Lin, Assistant Professor

Introduction

- Applications of 3D warping
- Relief texture

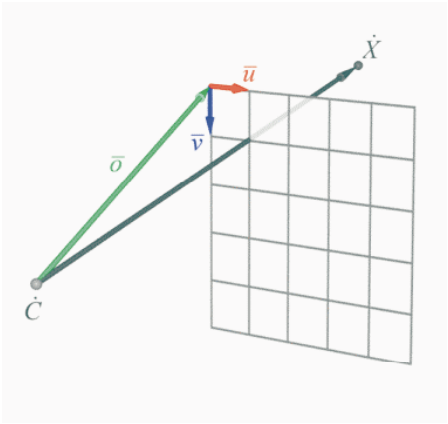
Ref:

• Image-based Modeling and Rendering, SIGGRAPH'99 course notes.

• Manuel M Oliveira, Gary Bishop, David McAllister. Relief Texture Mapping. Proc. of SIGGRAPH'00, pp.359-368.

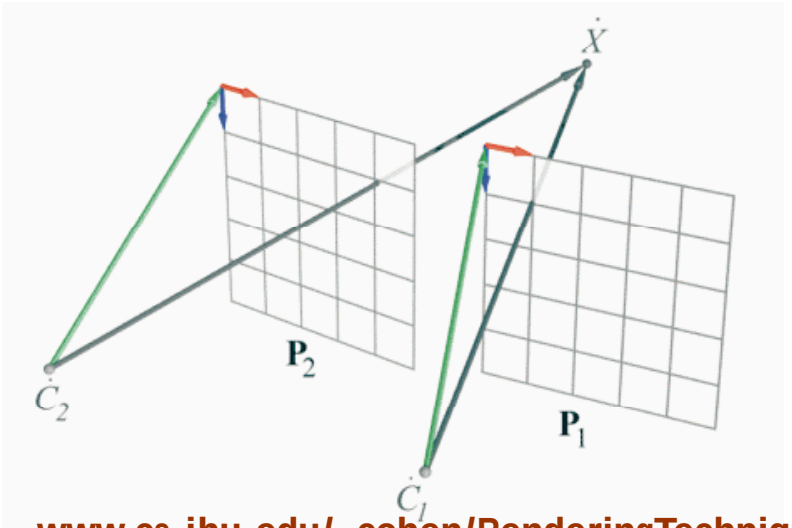
• Fabio Policarpo, Manuel M. Oliveira, João Comba. Real-Time Relief Mapping on Arbitrary Polygonal Surfaces. I3D'05, pp. 155-162.

3D Warping



$$P = \begin{bmatrix} u_x & v_x & o_x \\ u_y & v_y & o_y \\ u_z & v_z & o_z \end{bmatrix}$$

$$\dot{X} = \dot{C} + t\,P\,\vec{x}$$



$$\dot{C}_1 + t_1 P_1 \vec{x}_1 = \dot{C}_2 + st_2 P_2 \vec{x}_2$$

$$t_2 P_2 \vec{x}_2 = \dot{C}_1 - \dot{C}_2 + t_1 P_1 \vec{x}_1$$

$$\frac{t_2}{t_1} P_2 \vec{x}_2 = \frac{1}{t_1}\left(\dot{C}_1 - \dot{C}_2\right) + P_1 \vec{x}_1$$

$$P_2 \vec{x}_2 \doteq \delta(\vec{x}_1)\left(\dot{C}_1 - \dot{C}_2\right) + P_1 \vec{x}_1$$

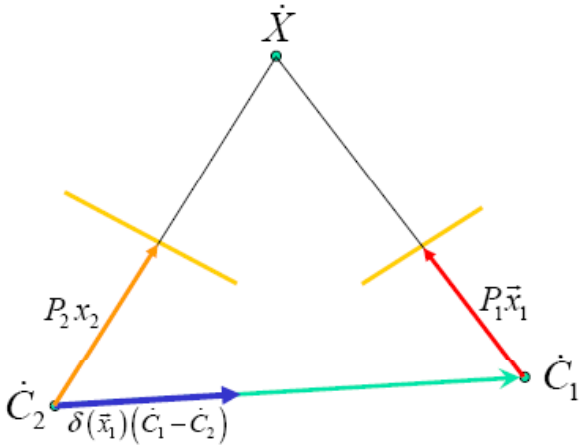
www.cs.jhu.edu/~cohen/RenderingTechniques/Lectures/Image-BasedRendering

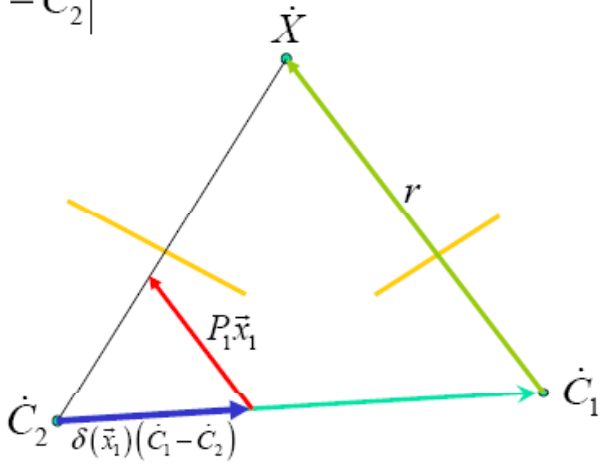
3D Warping (cont.)

$$P_2\vec{x}_2 \doteq \delta(\vec{x}_1)\left(\dot{C}_1 - \dot{C}_2\right) + P_1\vec{x}_1$$

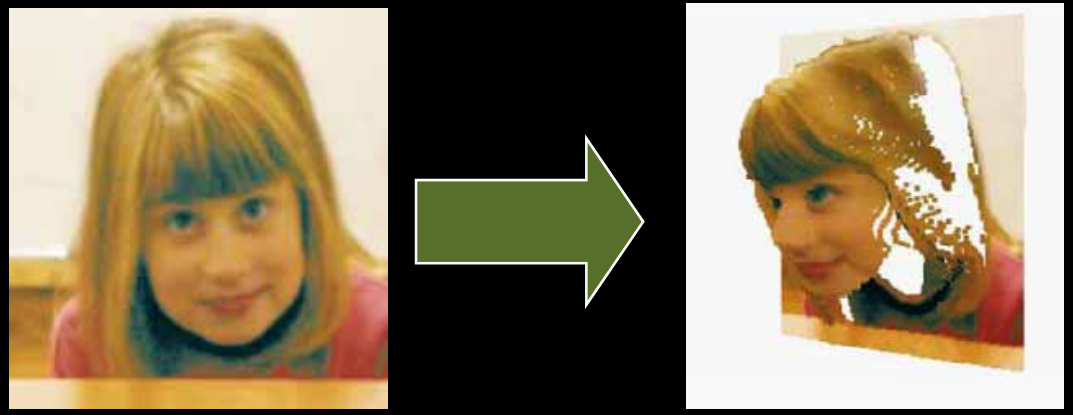
$$\frac{r}{\left|\dot{C}_1 - \dot{C}_2\right|} = \frac{\left|P_1\vec{x}_1\right|}{\delta(\vec{x}_1)\left|\dot{C}_1 - \dot{C}_2\right|}$$

$$\boxed{\delta(\vec{x}_1) = \frac{\left|P_1\vec{x}_1\right|}{r}}$$





3D Warping (cont.)

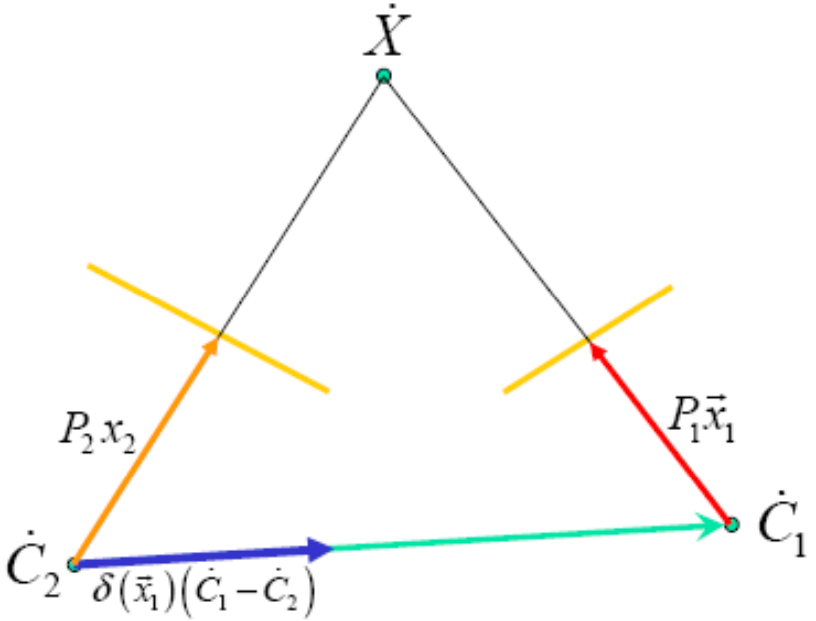


$$P_2\vec{x}_2 \doteq \delta(\vec{x}_1)\left(\dot{C}_1 - \dot{C}_2\right) + P_1\vec{x}_1$$

$$h\vec{x}_2 = P_2^{-1}\left[\delta(\vec{x}_1)\left(\dot{C}_1 - \dot{C}_2\right) + P_1\vec{x}_1\right]$$

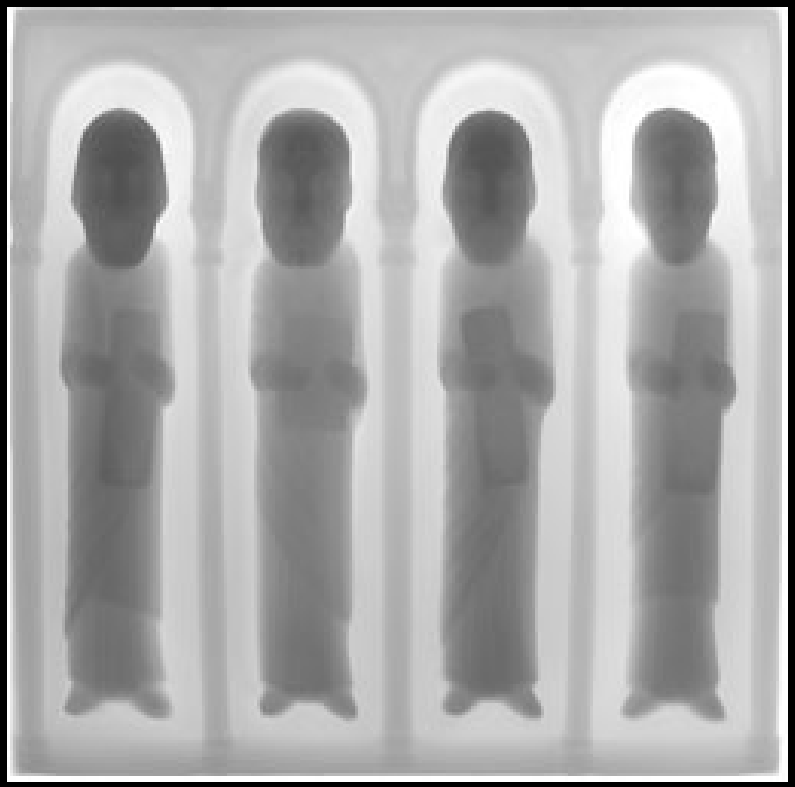
$$h\vec{x}_2 = P_2^{-1}\left[P_1 \quad \left(\dot{C}_1 - \dot{C}_2\right)\right]\begin{bmatrix}\vec{x}_1 \\ \delta(\vec{x}_1)\end{bmatrix}$$

$$h\vec{x}_2 = W\begin{bmatrix}\vec{x}_1 \\ \delta(\vec{x}_1)\end{bmatrix}$$



3D Warping

- Given texture (image) and depth maps,



3D Warping and Relief Texture

conventional texture

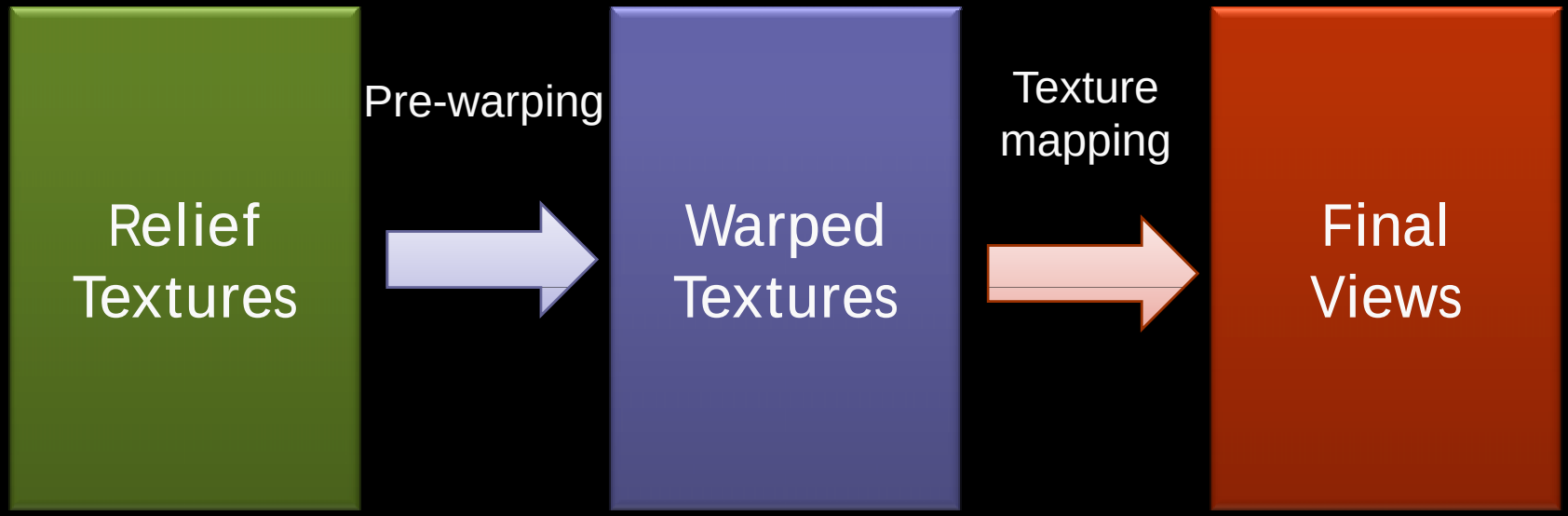
Relief texture mapping

mesh of micro-polygons

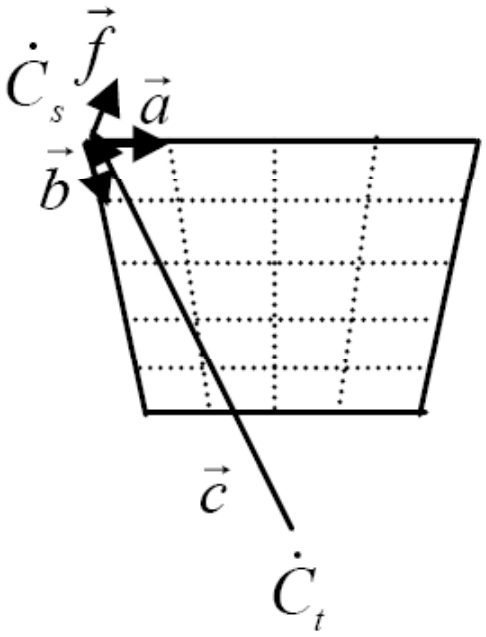
Relief Texture Mapping

- A texture extended with orthogonal displacements per texel.
  - Viewer far way: standard texture mapping.
  - Viewer in a small distance: relief texture. (warping)
  - Viewer close to the surface: micro polygons.

Relief Texture Mapping



S-T Perspective mapping



$$\dot{x} = \dot{C}_s + \begin{bmatrix} a_{si} & b_{si} & f_{si} \\ a_{sj} & b_{sj} & f_{sj} \\ a_{sk} & b_{sk} & f_{sk} \end{bmatrix} \begin{bmatrix} u_s \\ v_s \\ displ(u_s, v_s) \end{bmatrix} = \dot{C}_s + P'_s \vec{x}'_s$$

$$u_t = \frac{Au_s + Bv_s + D + C' displ(u_s, v_s)}{Iu_s + Jv_s + L + K' displ(u_s, v_s)}$$

$$v_t = \frac{Eu_s + Fv_s + H + G' displ(u_s, v_s)}{Iu_s + Jv_s + L + K' displ(u_s, v_s)}$$

where $A = \vec{a}_s \cdot (\vec{b}_t \times \vec{c}_t)$, $B = \vec{b}_s \cdot (\vec{b}_t \times \vec{c}_t)$, $C' = \vec{f}_s \cdot (\vec{b}_t \times \vec{c}_t)$,

$D = (\dot{C}_s - \dot{C}_t) \cdot (\vec{b}_t \times \vec{c}_t)$, $E = \vec{a}_s \cdot (\vec{c}_t \times \vec{a}_t)$, $F = \vec{b}_s \cdot (\vec{c}_t \times \vec{a}_t)$,

$G' = \vec{f}_s \cdot (\vec{c}_t \times \vec{a}_t)$, $H = (\dot{C}_s - \dot{C}_t) \cdot (\vec{c}_t \times \vec{a}_t)$, $I = \vec{a}_s \cdot (\vec{a}_t \times \vec{b}_t)$,

$J = \vec{b}_s \cdot (\vec{a}_t \times \vec{b}_t)$, $K' = \vec{f}_s \cdot (\vec{a}_t \times \vec{b}_t)$, $L = (\dot{C}_s - \dot{C}_t) \cdot (\vec{a}_t \times \vec{b}_t)$

Intermediate Texture

- To utilize pipeline/hardware, warp to standard perspective mapping first.

$$u_i = \frac{u_s + k_1 displ(u_s, v_s)}{1 + k_3 displ(u_s, v_s)}$$

$$v_i = \frac{v_s + k_2 displ(u_s, v_s)}{1 + k_3 displ(u_s, v_s)}$$

$$\frac{Au_i + Bv_i + D}{Iu_i + Jv_i + L} = \frac{Au_s + Bv_s + D + C'displ(u_s, v_s)}{Iu_s + Jv_s + L + K'displ(u_s, v_s)}$$

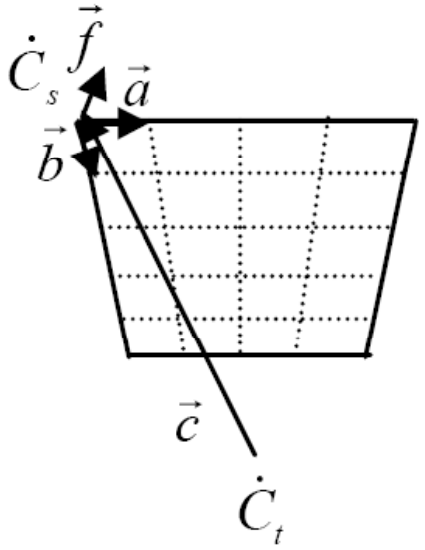
$$\frac{Eu_i + Fv_i + H}{Iu_i + Jv_i + L} = \frac{Eu_s + Fv_s + H + G'displ(u_s, v_s)}{Iu_s + Jv_s + L + K'displ(u_s, v_s)}$$

Intermediate Texture

- Simplified case:

$$u_i = \frac{u_s + k_1 displ(u_s, v_s)}{1 + k_3 displ(u_s, v_s)}$$

$$v_i = \frac{v_s + k_2 displ(u_s, v_s)}{1 + k_3 displ(u_s, v_s)}$$

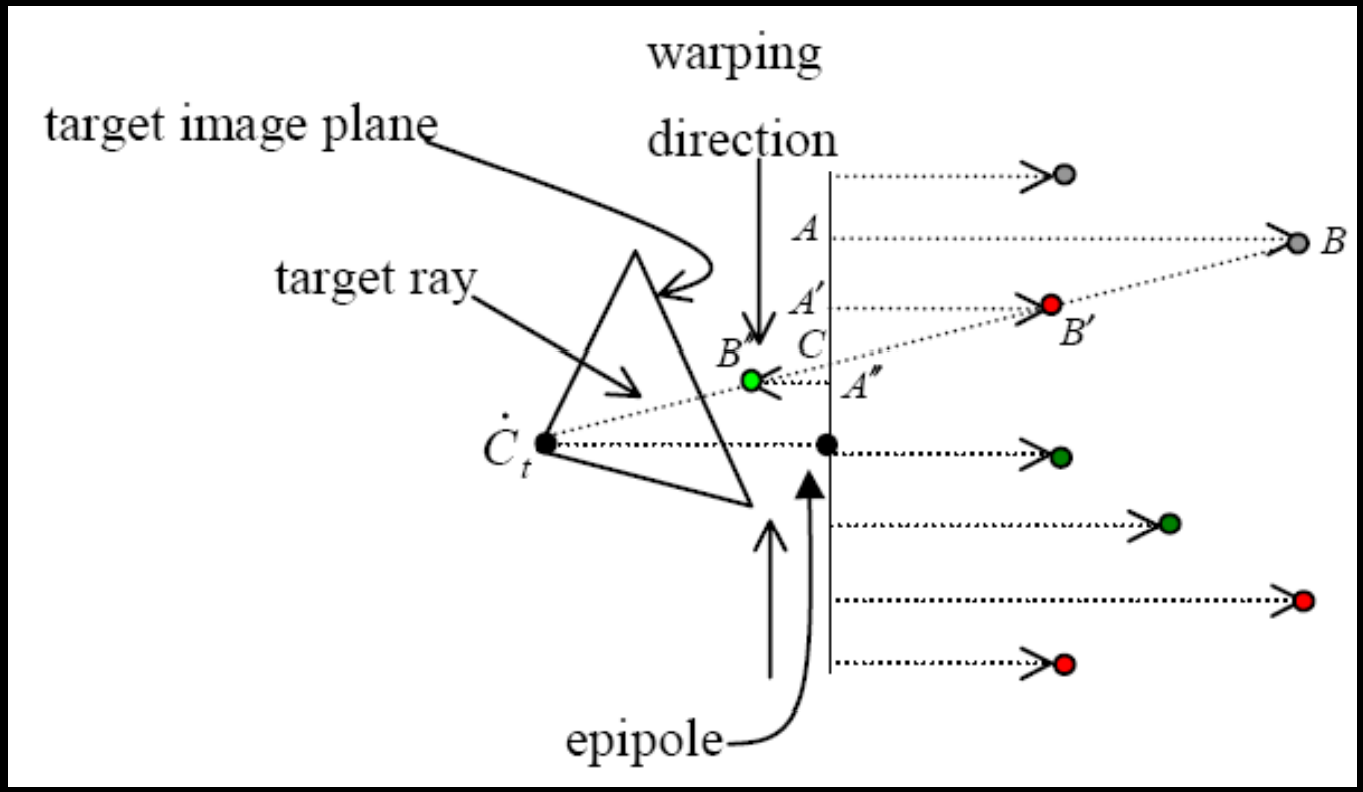


$$\vec{a}_t = \alpha \vec{a}_s\,,\ b_t = \beta b_s \ \text{ and } \ \vec{c}_t = \gamma(C_s - C_t)$$

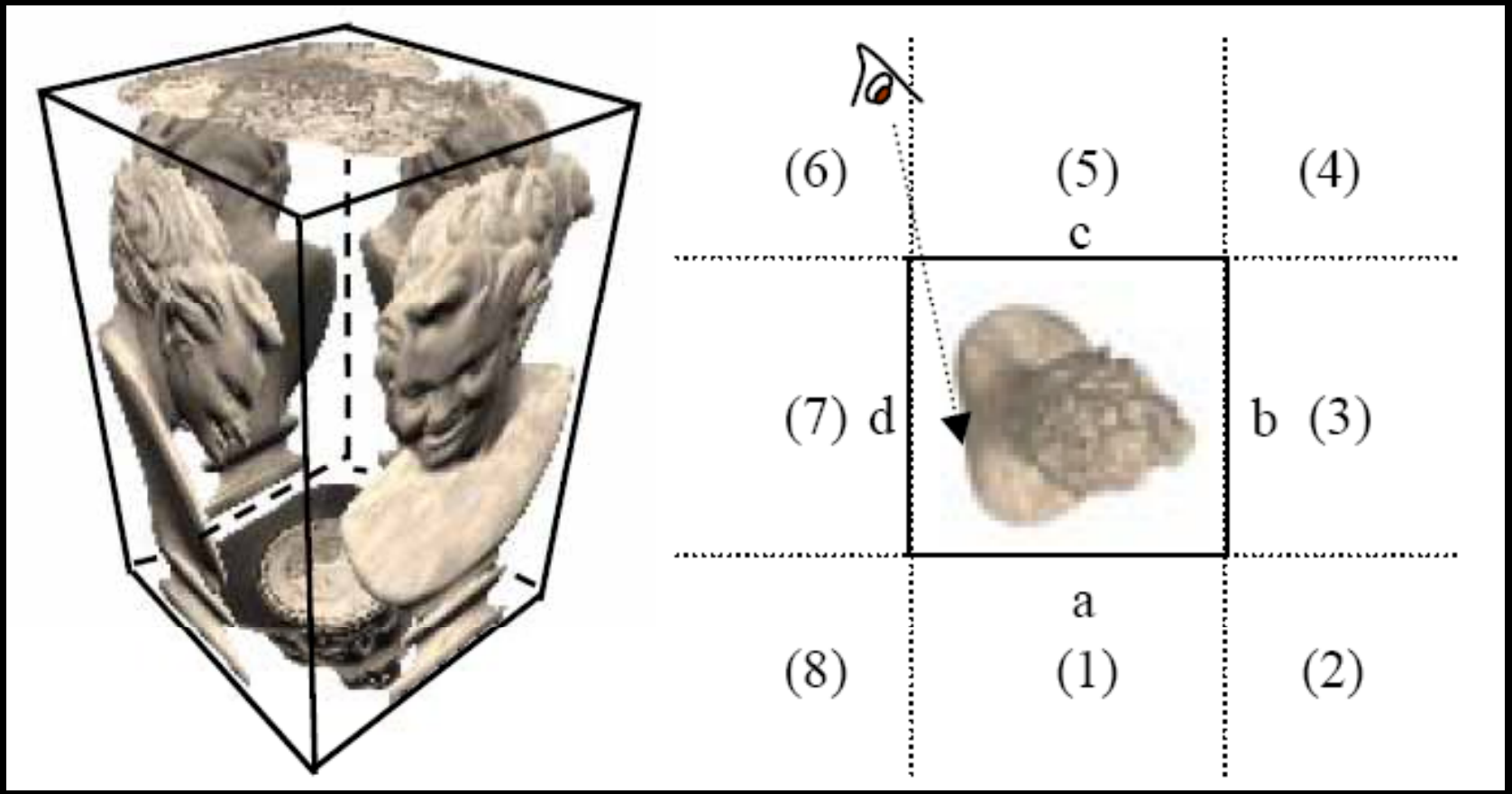
$$k_1 = \frac{\vec{f} \cdot (\vec{b} \times \vec{c})}{\vec{a} \cdot (\vec{b} \times \vec{c})},\ k_2 = \frac{\vec{f} \cdot (\vec{c} \times a)}{\vec{a} \cdot (\vec{b} \times \vec{c})} \text{ and } k_3 = \frac{1}{\vec{c} \cdot \vec{f}}$$

Occlusion

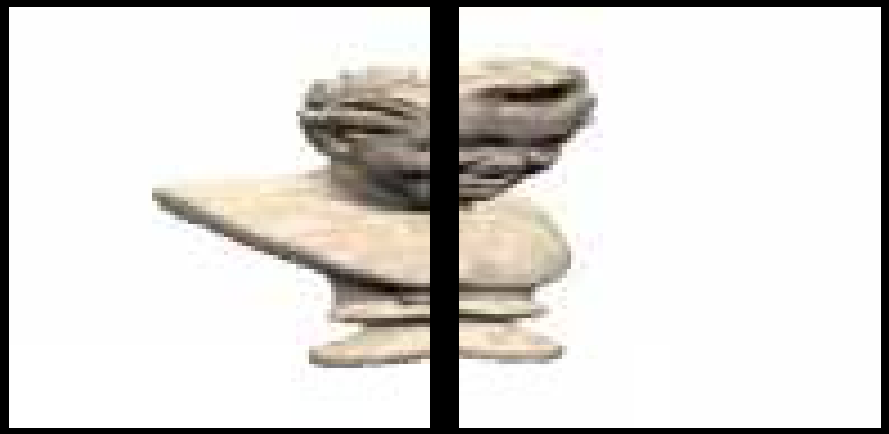
- Similar to the discussion in LDI



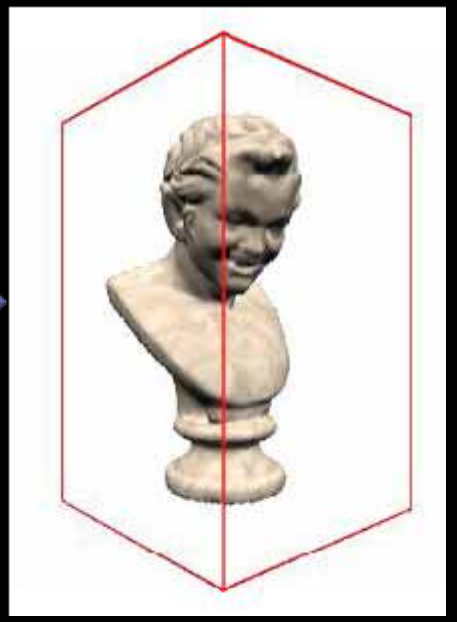
Relief Texture for Objects



Relief Texture for Objects



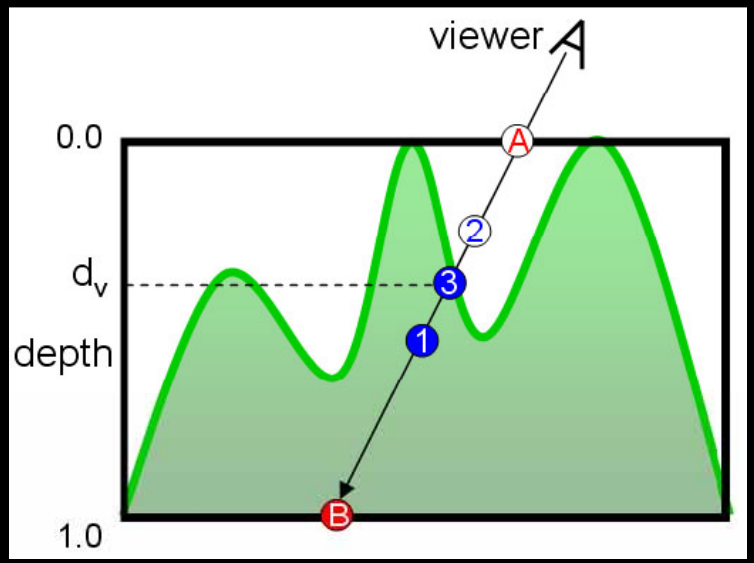
Pre-warped images

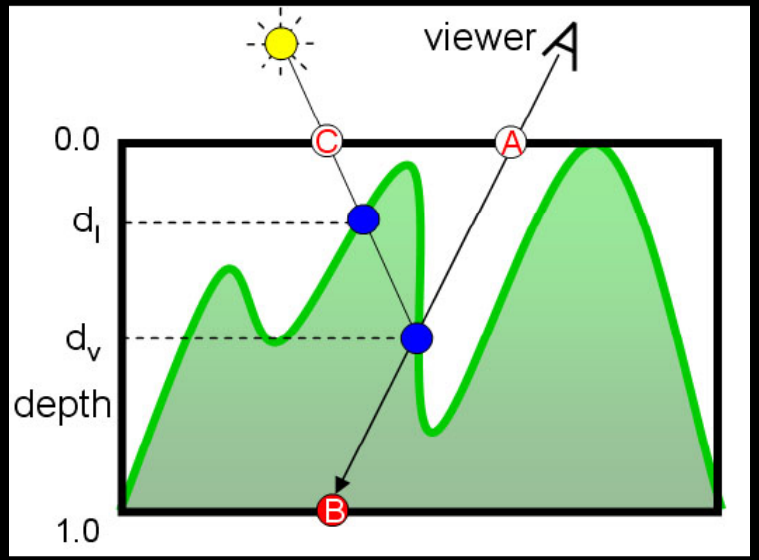


Applying texture mapping

Relief Texture with Shaders

- With GPU (shaders), the computation of relief texture becomes more straight forward.
- With shadow and per-pixel lighting effects.





• F. Policarpo, M. M. Oliveira, J. Comba. Real-Time Relief Mapping on Arbitrary Polygonal Surfaces. I3D'05, pp. 155-162.

Relief Texture with Shaders