

# Relief Texture Mapping

IBMRL

Copyright © Manuel M. Oliveira

## Relief Texture Mapping [Oliveira 00]

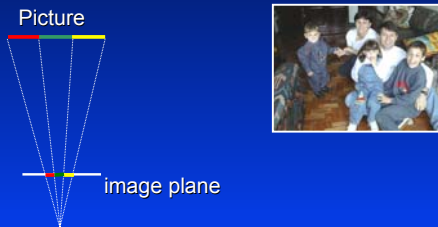
- Extension to conventional texture mapping that supports:
  - 3-D Surface detail
  - True motion parallax
  - Correct silhouettes
- Integration of IBR and polygonal rendering

IBMRL

Copyright © Manuel M. Oliveira

## Conventional Texture Mapping

- Very successful in high-quality image synthesis

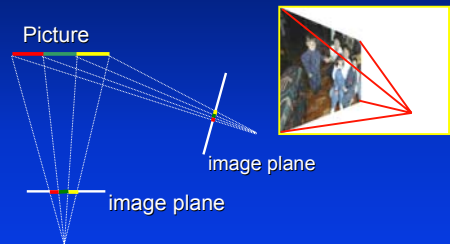


IBMRL

Copyright © Manuel M. Oliveira

## Conventional Texture Mapping

- No changes in visibility



IBMRL

Copyright © Manuel M. Oliveira

## Relief Texture Mapping

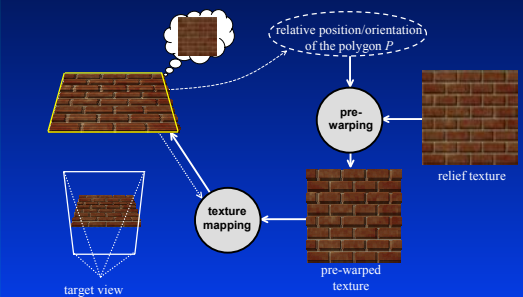
- Overview



IBMRL

Copyright © Manuel M. Oliveira

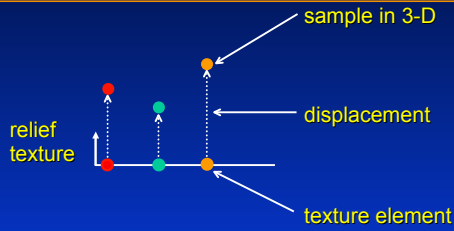
## Relief Texture Mapping Pipeline



IBMRL

Copyright © Manuel M. Oliveira

## Relief Textures

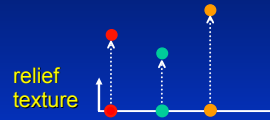


Relief Texture = color + displacement

IBMRL

Copyright © Manuel M. Oliveira

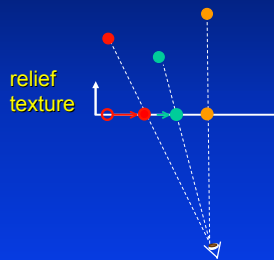
## Pre-Warping



IBMRL

Copyright © Manuel M. Oliveira

## Pre-Warping (Cont.)

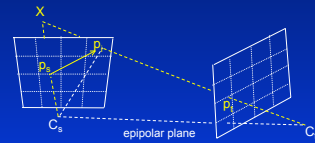


IBMRL

Copyright © Manuel M. Oliveira

## Pre-warping

- Obtaining the image to be texture mapped
- Epipolar geometry
  - Perspective source camera case



IBMRL

Copyright © Manuel M. Oliveira

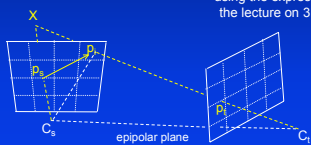
## Warping

- Mapping the source image to the target view
- $w_i$  only depend on the two cameras parameters

$$\begin{aligned} r &= w_{11}u_s + w_{12}v_s + w_{13} + w_{14}\delta(u_s, v_s) \\ s &= w_{21}u_s + w_{22}v_s + w_{23} + w_{24}\delta(u_s, v_s) \\ t &= w_{31}u_s + w_{32}v_s + w_{33} + w_{34}\delta(u_s, v_s) \end{aligned}$$

$$u_t = \frac{r}{t} \quad v_t = \frac{s}{t}$$

The values of  $w_i$  are computed using the expressions presented in the lecture on 3D Image Warping



IBMRL

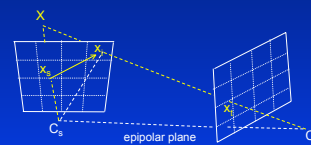
Copyright © Manuel M. Oliveira

## Pre-warping

- For the pre-warped image, let  $\delta(u_i, v_i) = 1$  for all pixels

$$\begin{aligned} r' &= w_{11}u_i + w_{12}v_i + w_{13} + w_{14} \\ s' &= w_{21}u_i + w_{22}v_i + w_{23} + w_{24} \\ t' &= w_{31}u_i + w_{32}v_i + w_{33} + w_{34} \end{aligned}$$

$$u_t = \frac{r'}{t'} \quad v_t = \frac{s'}{t'}$$



IBMRL

Copyright © Manuel M. Oliveira

## Pre-warping

- Therefore

$$u_t = \frac{w_{11}u_s + w_{12}v_s + w_{13} + w_{14}}{w_{31}u_s + w_{32}v_s + w_{33} + w_{34}} = \frac{w_{11}u_s + w_{12}v_s + w_{13} + w_{14}\delta(u_s, v_s)}{w_{31}u_s + w_{32}v_s + w_{33} + w_{34}\delta(u_s, v_s)}$$

$$v_t = \frac{w_{21}u_s + w_{22}v_s + w_{23} + w_{24}}{w_{31}u_s + w_{32}v_s + w_{33} + w_{34}} = \frac{w_{21}u_s + w_{22}v_s + w_{23} + w_{24}\delta(u_s, v_s)}{w_{31}u_s + w_{32}v_s + w_{33} + w_{34}\delta(u_s, v_s)}$$

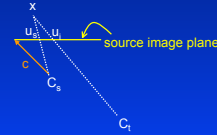
- All we need to do now is solve for  $u_t$  and  $v_t$

IBMRL

Copyright © Manuel M. Oliveira

## Simplifying the Expressions

- The amount of shift  $(\Delta u, \Delta v)$  that maps  $(u_s, v_s)$  into  $(u_t, v_t)$  only depends on:
  - the target COP (no other target camera parameters)
  - the position and orientation of the source image plane
  - the disparity values

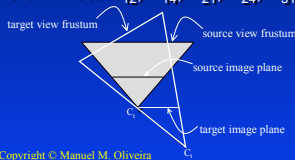


IBMRL

Copyright © Manuel M. Oliveira

## Simplifying the Expressions

- We have the freedom to specify  $a_t, b_t, c_t$
- Define a temporary camera for the pre-warping
- Force many  $w_{ij}$  to have the form  $v \cdot (v \times w)$  or  $w \cdot (v \times w)$
- By making  $a_t = \alpha a_s$ ,  $b_t = \beta b_s$ , and  $c_t = \gamma(C_s - C_t)$  for non-zero  $\alpha, \beta, \gamma \in \mathbb{R}$ , we eliminate  $w_{12}, w_{14}, w_{21}, w_{24}, w_{31}, w_{32}$



IBMRL

Copyright © Manuel M. Oliveira

## Pre-warping equations

$$u_t = \frac{w_{11}u_s + w_{13}}{w_{33} + w_{34}} = \frac{w_{11}u_s + w_{13}}{w_{33} + w_{34}\delta(u_s, v_s)}$$

$$v_t = \frac{w_{22}v_s + w_{23}}{w_{33} + w_{34}} = \frac{w_{22}v_s + w_{23}}{w_{33} + w_{34}\delta(u_s, v_s)}$$

$$u_t = \frac{u_s - \frac{w_{13}w_{34}}{w_{11}w_{33}}\delta(u_s, v_s)}{1 + \frac{w_{34}}{w_{33}}\delta(u_s, v_s)}$$

$$v_t = \frac{v_s - \frac{w_{23}w_{34}}{w_{22}w_{33}}\delta(u_s, v_s)}{1 + \frac{w_{34}}{w_{33}}\delta(u_s, v_s)}$$

IBMRL

Copyright © Manuel M. Oliveira

## Relief Textures

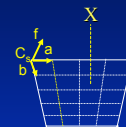
- Parallel projection images with depth
- Simpler pre-warping
- Constant sampling density
- Rendering advantages



IBMRL

Copyright © Manuel M. Oliveira

## Orthographic camera model



$$\mathcal{X} = \mathcal{C}_s + \begin{bmatrix} a_{si} & b_{si} & f_{si} \\ a_{sj} & b_{sj} & f_{sj} \\ a_{sk} & b_{sk} & f_{sk} \end{bmatrix} \begin{bmatrix} u_s \\ v_s \\ \text{displ}(u_s, v_s) \end{bmatrix}$$

$$\mathcal{X} = \mathcal{C}_s + M'_s p_s$$

Remember that for the perspective case, we had

$$\mathcal{X} = \mathcal{C}_s + t_s M_s p_s$$

$$\mathcal{X} = \mathcal{C}_s + t_s \begin{bmatrix} a_{si} & b_{si} & c_{si} \\ a_{sj} & b_{sj} & c_{sj} \\ a_{sk} & b_{sk} & c_{sk} \end{bmatrix} \begin{bmatrix} u_s \\ v_s \\ 1 \end{bmatrix}$$

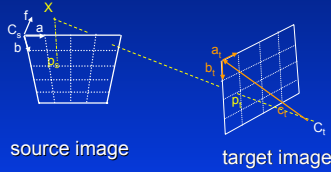
IBMRL

Copyright © Manuel M. Oliveira

## Warping from Ortho to Perspective

$$\mathcal{C}_t + t_t M_t p_t = \mathcal{X} = \mathcal{C}_s + M'_s p_s$$

$$p_t \approx M_t^{-1} (M'_s p_s + (\mathcal{C}_s - \mathcal{C}_t))$$



IBMRL

Copyright © Manuel M. Oliveira

## Comparing Ortho & Perspective Eqs.

- Warping equation using an orthographic source image

$$p_t \approx M_t^{-1} (M'_s p_s + (\mathcal{C}_s - \mathcal{C}_t)) \quad (1)$$

- Warping equation using a perspective source image

$$p_t \approx M_t^{-1} (M_s p_s + (\mathcal{C}_s - \mathcal{C}_t) \delta(u_s, v_s)) \quad (2)$$

- In (1),  $w_{24}$  is multiplied by 1, whereas in (2) it is multiplied by  $\delta(u_s, v_s)$
- In (2),  $w_{23}$  is multiplied by 1, whereas in (1), a new term  $w'_{23}$  is multiplied by  $\text{disp}(u_s, v_s)$

IBMRL

Copyright © Manuel M. Oliveira

## Warping from Ortho to Perspective

$$\begin{aligned} r &= w_{11}u_s + w_{12}v_s + w_{14} + C' \text{disp}(u_s, v_s) \\ s &= w_{21}u_s + w_{22}v_s + w_{24} + G' \text{disp}(u_s, v_s) \\ t &= w_{31}u_s + w_{32}v_s + w_{34} + K' \text{disp}(u_s, v_s) \end{aligned}$$

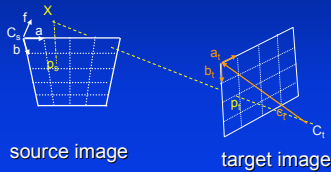
$$u_t = \frac{r}{t} \quad v_t = \frac{s}{t}$$

where

$$C' = f \cdot \left( \frac{p}{b_t} \times \frac{p}{b_t} \right)$$

$$G' = f \cdot \left( \frac{p}{b_t} \times \frac{p}{b_t} \right)$$

$$K' = f \cdot \left( \frac{p}{b_t} \times \frac{p}{b_t} \right)$$



IBMRL

Copyright © Manuel M. Oliveira

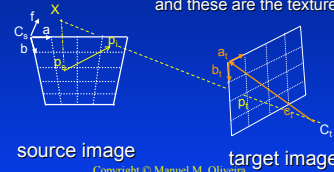
## Warping from Ortho to Perspective

- For the pre-warped image, let  $\text{disp}(u_i, v_i) = 0$  for all pixels

$$\begin{aligned} r &= w_{11}u_i + w_{12}v_i + w_{14} \\ s &= w_{21}u_i + w_{22}v_i + w_{24} \\ t &= w_{31}u_i + w_{32}v_i + w_{34} \end{aligned}$$

$$u_t = \frac{r}{t} \quad v_t = \frac{s}{t}$$

This is the same as texture mapping and these are the texture mapping eqs.!



IBMRL

Copyright © Manuel M. Oliveira

## Pre-warping

- Therefore, for relief textures we have

$$\begin{aligned} u_t &= \frac{w_{11}u_i + w_{12}v_i + w_{14}}{w_{31}u_i + w_{32}v_i + w_{34}} = \frac{w_{11}u_s + w_{12}v_s + w_{14} + C' \text{disp}(u_s, v_s)}{w_{31}u_s + w_{32}v_s + w_{34} + K' \text{disp}(u_s, v_s)} \\ v_t &= \frac{w_{21}u_i + w_{22}v_i + w_{24}}{w_{31}u_i + w_{32}v_i + w_{34}} = \frac{w_{21}u_s + w_{22}v_s + w_{24} + G' \text{disp}(u_s, v_s)}{w_{31}u_s + w_{32}v_s + w_{34} + K' \text{disp}(u_s, v_s)} \end{aligned}$$

- where

$$C' = f \cdot \left( \frac{p}{b_t} \times \frac{p}{b_t} \right)$$

$$G' = f \cdot \left( \frac{p}{b_t} \times \frac{p}{b_t} \right)$$

$$K' = f \cdot \left( \frac{p}{b_t} \times \frac{p}{b_t} \right)$$

IBMRL

Copyright © Manuel M. Oliveira

## Yes, We Can Simplify

- $a_t = \alpha a_s$ ,  $b_t = \beta b_s$ , and  $c_t = \gamma(C_s - C_t)$  is trivially satisfied by letting the two image planes coincide, including their image planes and basis vectors
- Such a configuration defines the temporary camera for the pre-warp



IBMRL

Copyright © Manuel M. Oliveira

## Pre-Warping Equations for Relief Textures

- Horizontal shift independent of the vertical position
- Vertical shift independent of the horizontal position

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

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

- where

$$k_1 = \frac{p}{f} \cdot \frac{(b \times d)}{d \cdot (b \times c)}$$

$$k_2 = \frac{p}{b} \cdot \frac{(c \times d)}{c \cdot (d \times a)}$$

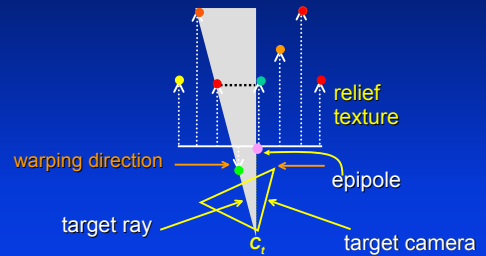
$$k_3 = \frac{1}{c} \cdot \frac{p}{f}$$

IBMRL

Copyright © Manuel M. Oliveira

## Handling Visibility

- Occlusion-compatible order

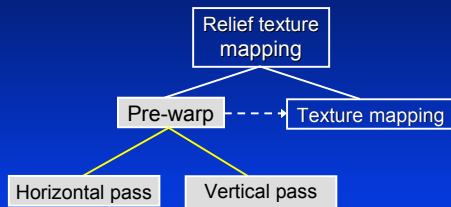


IBMRL

Copyright © Manuel M. Oliveira

## Image Resampling

- Two-pass image resampling



IBMRL

Copyright © Manuel M. Oliveira

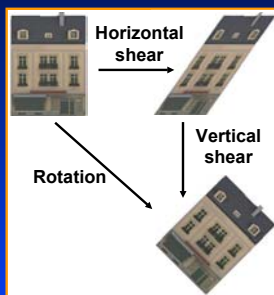
## Serial Warps

- Series of 1-D operations equivalent to a 2-D or to a 3-D transformation

IBMRL

Copyright © Manuel M. Oliveira

## Serial Warps (Example)

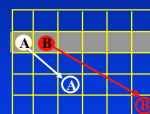


IBMRL

Copyright © Manuel M. Oliveira

## Two-Pass 1-D Resampling

- Original and warped coordinates of two texels

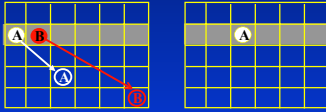


IBMRL

Copyright © Manuel M. Oliveira

## Horizontal Pass (step 1)

- Move first texel to its final column

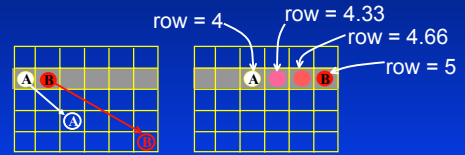


IBMRL

Copyright © Manuel M. Oliveira

## Horizontal Pass (step 2)

- Warp next texel
- Interpolate color and row coordinates

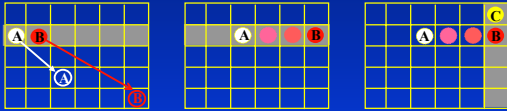


IBMRL

Copyright © Manuel M. Oliveira

## Horizontal Pass

- After all rows have been warped

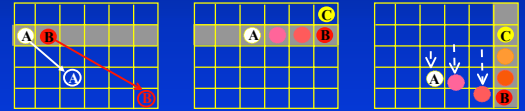


IBMRL

Copyright © Manuel M. Oliveira

## Vertical Pass

- Move samples to their final rows and interpolate colors



IBMRL

Copyright © Manuel M. Oliveira

Relief Texture



Intermediate Image



Horizontal Pass

Vertical Pass



Texture Mapping



Final View

Pre-warped Texture

IBMRL

Copyright © Manuel M. Oliveira

## Object Modeling

- Six relief textures acquired from the faces of the object's bounding box
- Example: statue modeled with 35,280 polygons



IBMRL

Copyright © Manuel M. Oliveira

## Object Rendering

- Statue rendered by relief texture mapping two polygons



Pre-warped textures

IBMRL

Copyright © Manuel M. Oliveira

## Correct Occlusions

- $\Delta Z$  values interpolated along rows and columns
- Depth buffer modulation



IBMRL

Copyright © Manuel M. Oliveira

## Multiresolution Representations

- Relief Texture Pyramids
  - Similar to Mip map pyramids
  - Filter both color and displacement data



IBMRL

Copyright © Manuel M. Oliveira

## Multiresolution (Cont.)

- Textured LODs
  - Rendered using the algorithms described
  - Reduce rendering cost and aliasing artifacts



256x256

128x128

64x64

32x32

IBMRL

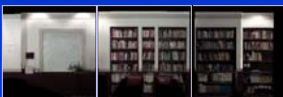
Copyright © Manuel M. Oliveira

## Immersive Environments

- Texture Acquisition



- Relief Textures

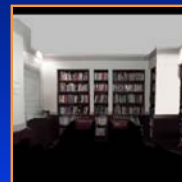


IBMRL

Copyright © Manuel M. Oliveira

## Immersive Environments

- Modeling and Rendering
  - Textures instantiated at the same relative positions used for acquisition



IBMRL

Copyright © Manuel M. Oliveira