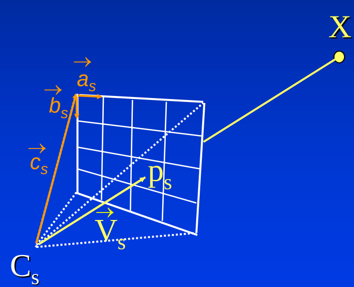
3-D Image Warping

3-D Image Warping [McMillan - Bishop 95]

- Maps images with depth onto arbitrary image planes
- Scene geometry implicitly represented by a pinhole camera and per pixel depth

Demo

3-D Image Warping [McMillan - Bishop 95]



$$\dot{C}_s + t_s \vec{V}_s = X$$

$$\vec{V}_s = \vec{c}_s + p_{sx}\vec{a}_s + p_{sy}\vec{b}_s$$

$$\dot{C}_s + t_s(\vec{c}_s + p_{sx}\vec{a}_s + p_{sy}\vec{b}_s) = X$$

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

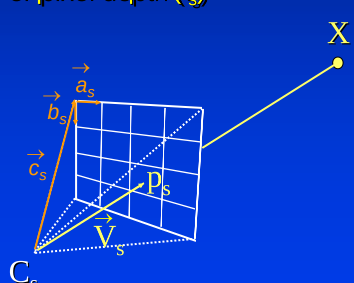
$$M_s p_s = \begin{bmatrix} a_{si} & b_{si} & c_{si} \\ a_{sj} & b_{sj} & c_{sj} \\ a_{sk} & b_{sk} & c_{sk} \end{bmatrix} \begin{bmatrix} p_{sx} \\ p_{sy} \\ 1 \end{bmatrix}$$

$$t_s = \frac{\|X - C_s\|}{\|\vec{V}_s\|}$$

IBMRL

3-D Image Warping (Cont.)

- What we need to know
  - Camera parameters
    - Position (C_s)
    - Orientation, focal distance, horizontal and vertical FOVs (M_s)
  - Per pixel depth (t_s)



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

$$\|\vec{a}\| = \frac{width\ of\ the\ projection\ plane\ in\ 3D}{number\ of\ columns\ in\ the\ image}$$

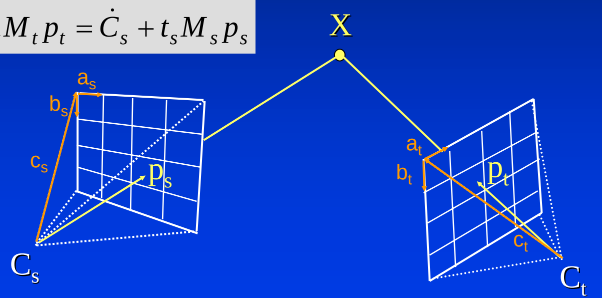
$$\|\vec{b}\| = \frac{height\ of\ the\ projection\ plane\ in\ 3D}{number\ of\ rows\ in\ the\ image}$$

IBMRL

3-D Image Warping (Cont.)

- Produces correct 3-D reprojections from arbitrary views
- Compute the projections onto arbitrary image planes by solving

$$\dot{C}_t + t_t M_t p_t = \dot{C}_s + t_s M_s p_s$$



source image   target image

3-D Warping Equation

$$\dot{C}_t + t_t M_t p_t = \dot{C}_s + t_s M_s p_s$$

$$p_t \doteq M_t^{-1}((\dot{C}_s - \dot{C}_t) + t_s M_s p_s)$$

$$p_t \doteq \underbrace{M_t^{-1}(\dot{C}_s - \dot{C}_t)}_{epipole}\delta_s + \underbrace{M_t^{-1} M_s p_s}_{planar\ \ pespectiveproj}$$

$\delta_s = 1/t_s$ is called the *generalized disparity* of pixel p_s

- 3-D image warping is equivalent to a texture mapping operation followed by a per pixel shift proportional to δ_s in the direction of the epipole
- Generalizes conventional texture mapping

3-D Warping Equation (Cont.)

$$p_t \doteq \underbrace{M_t^{-1}(\dot{C}_s - \dot{C}_t)}_{epipole}\delta_s + \underbrace{M_t^{-1}M_s p_s}_{planar\ \ pespectiveproj}$$

$$\alpha\begin{bmatrix}u_t\\v_t\\1\end{bmatrix} = \begin{bmatrix}\vec{a}_t & \vec{b}_t & \vec{c}_t\end{bmatrix}^{-1}\left(\dot{C}_s - \dot{C}_t\right)\delta_s(u_s, v_s) + \begin{bmatrix}\vec{a}_t & \vec{b}_t & \vec{c}_t\end{bmatrix}^{-1}\begin{bmatrix}\vec{a}_s & \vec{b}_s & \vec{c}_s\end{bmatrix}\begin{bmatrix}u_s\\v_s\\1\end{bmatrix}$$

$$\alpha\begin{bmatrix}u_t\\v_t\\1\end{bmatrix} = \begin{bmatrix}\vec{a}_t & \vec{b}_t & \vec{c}_t\end{bmatrix}^{-1}\begin{bmatrix}\vec{a}_s & \vec{b}_s & \vec{c}_s & \left(\dot{C}_s - \dot{C}_t\right)\end{bmatrix}\begin{bmatrix}u_s\\v_s\\1\\\delta(u_s, v_s)\end{bmatrix}$$

3-D Warping Equation (Cont.)

$$u_t = \frac{\vec{a}_s\cdot(\vec{b}_t\times\vec{c}_t)u_s + \vec{b}_s\cdot(\vec{b}_t\times\vec{c}_t)v_s + \vec{c}_s\cdot(\vec{b}_t\times\vec{c}_t) + (\dot{C}_s - \dot{C}_t)\cdot(\vec{b}_t\times\vec{c}_t)\delta(u_s, v_s)}{\vec{a}_s\cdot(\vec{a}_t\times\vec{b}_t)u_s + \vec{b}_s\cdot(\vec{a}_t\times\vec{b}_t)v_s + \vec{c}_s\cdot(\vec{a}_t\times\vec{b}_t) + (\dot{C}_s - \dot{C}_t)\cdot(\vec{a}_t\times\vec{b}_t)\delta(u_s, v_s)}$$

$$v_t = \frac{\vec{a}_s\cdot(\vec{c}_t\times\vec{a}_t)u_s + \vec{b}_s\cdot(\vec{c}_t\times\vec{a}_t)v_s + \vec{c}_s\cdot(\vec{c}_t\times\vec{a}_t) + (\dot{C}_s - \dot{C}_t)\cdot(\vec{c}_t\times\vec{a}_t)\delta(u_s, v_s)}{\vec{a}_s\cdot(\vec{a}_t\times\vec{b}_t)u_s + \vec{b}_s\cdot(\vec{a}_t\times\vec{b}_t)v_s + \vec{c}_s\cdot(\vec{a}_t\times\vec{b}_t) + (\dot{C}_s - \dot{C}_t)\cdot(\vec{a}_t\times\vec{b}_t)\delta(u_s, v_s)}$$

$$r = w_{11}u_s + w_{12}v_s + w_{13} + w_{14}\delta(u_s, v_s)$$

$$s = w_{21}u_s + w_{22}v_s + w_{23} + w_{24}\delta(u_s, v_s)$$

$$t = w_{31}u_s + w_{32}v_s + w_{33} + w_{34}\delta(u_s, v_s)$$

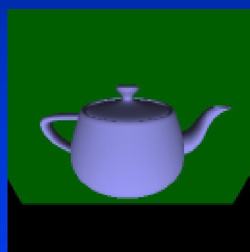
w_{ij}: recomputed as the target camera parameters change

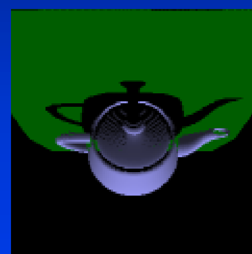
Sub-expressions of *r*, *s* and *t* can be updated incrementally

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

Disocclusion Artifacts

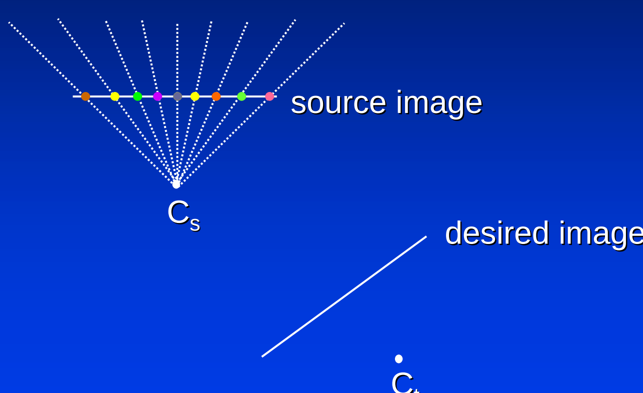
- Exposures of areas not visible in the original image introduce artifacts
- Need samples from other images to fill the holes





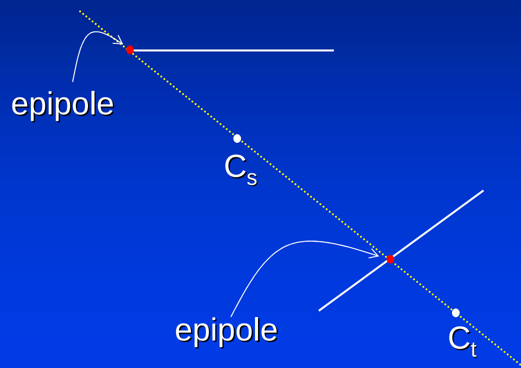
Occlusion-Compatible Order

- List priority algorithm for solving visibility

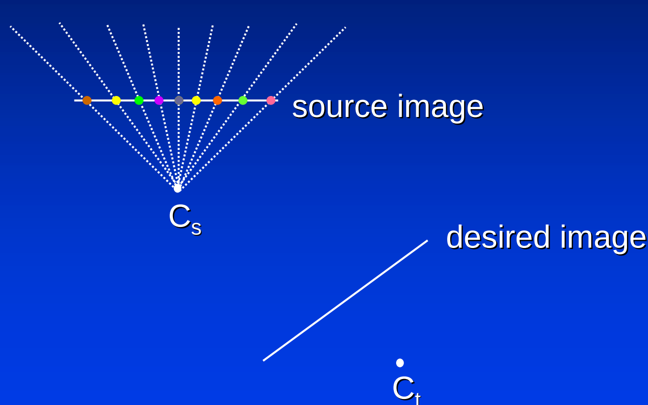


Occlusion-Compatible Order (Cont.)

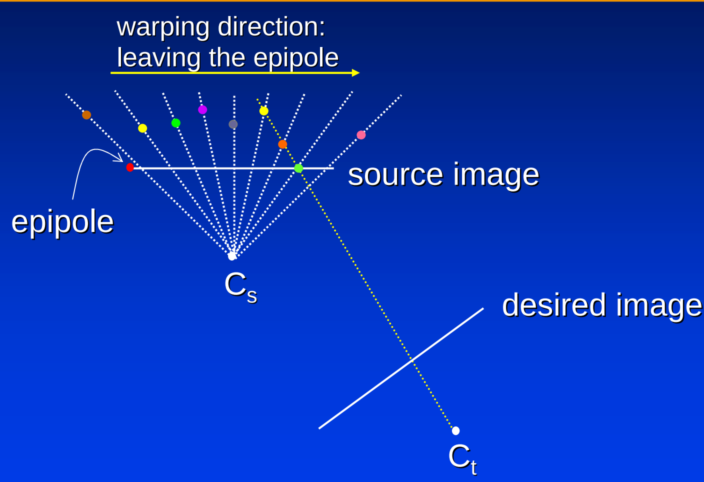
Epipole: projection one center of projection onto the image plane of the other camera



Occlusion-Compatible Order



Occlusion-Compatible Order (Cont.)



Occlusion-Compatible Order (Cont.)

- Algorithm summary

Find the project. of target COP onto the source image plane (epipole)

Divide source image plane into at most 4 sheets based on the epipole

If *target COP* is behind *source COP*

    for each sheet

        warp samples from the epipole towards the borders of the sheet

else

    for each sheet

        warp samples from the borders of the sheet towards the epipole

Computing the Epipole

- Projecting C_t onto the source image plane

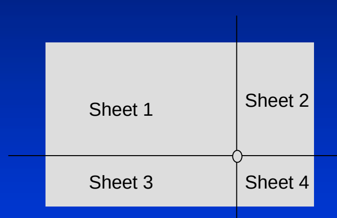
$$\begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix} = M_s^{-1}\left(\dot{C}_t - \dot{C}_s\right)$$

- The coordinates of the projected point are given by

$$\bar{e} = \begin{bmatrix} e_x / e_z \\ e_y / e_z \end{bmatrix}$$

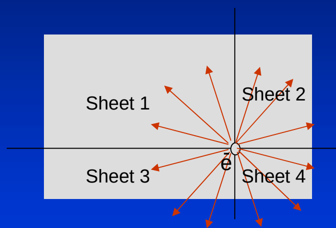
Splitting the Source Image

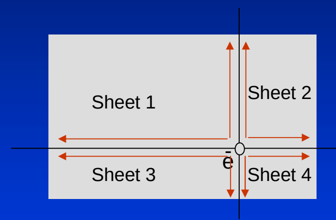
- The epipole splits the source image in at most 4 sheets



Warping Direction

- e_z negative: target COP behind source COP

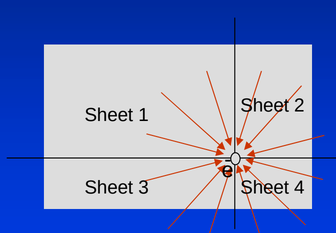


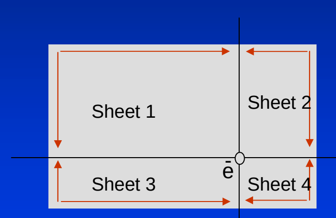


IBMRL

Warping Direction

- e_z non-negative: target COP aligned with or in front of the source COP





IBMRL