CS 395/495-26: Spring 2002

IBMR: Week 7A

New Direction: Applying Projections

Jack Tumblin
jet@cs.northwestern.edu

Reminders

No midterm, no final, but …
Alternating homework / project

- Project 2 Due today (C^*_∞ part optional)
- Homework 1 due Thurs May 16
- Project 3 Due Thurs May 24
- Revised Syllabus…

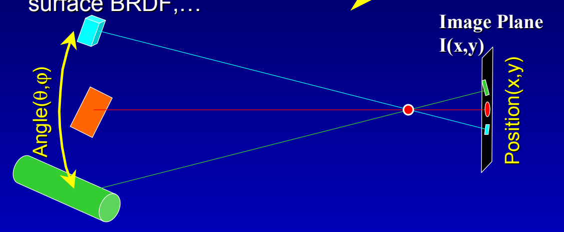
Recutting the Course Content

- 4 weeks left: More Fun, More Imaging, Less math
- More Project / CG work, Less Book Deciphering…
- Revised Topics:
  - 2D Warps of all kinds
    - non-planar projections
    - panorama building, camera error correction (spherical)
    - Light Probes
  - 3D/2D:Camera Matrix (Ch 5)
  - 3D: Epipolar Geom. (Ch 8)

An Image Is…

Light + 3D Scene: → **2D Image:**

Illumination, shape, movement, surface BRDF,… → Collection of rays through a point



'Image' : Angle→Position Map

- So far:
  - planar perspective projection
  - P^2, P^3 projective transformations
  - how to find these transformations from images.

- But Planar Perspective Projection is just ONE KIND of image, and
- P^2 and P^3 linear projective transformation **H** is just ONE KIND of image warping

$$H = H_S H_A H_P$$

'Image' : Angle→Position Map

- All Cameras make mistakes (geometry,shading/vignetting) book corrects them; we'll ignore them.

- Many kinds of camera projections

- 'Funhouse Mirrors'
  - Non-planar perspective projections:
    - Spherical, Conical, cylindrical, hyperbolic

- Applications: Panoramas, 'Light Probes' …

Image Warping: General Idea

2D→2D continuous coordinate map, a 'rubber sheet'

- Notation: input(x,y)→output(u,v)

- Demo: http://www.angelfire.com/biz/beamersandblasters/PicWarp.html.
- Usually done patch-by-patch, with polynomials

  $u = u(x,y) = \begin{bmatrix} 1 & y & y^2 \end{bmatrix} \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}$

  $v = v(x,y) = \ldots$

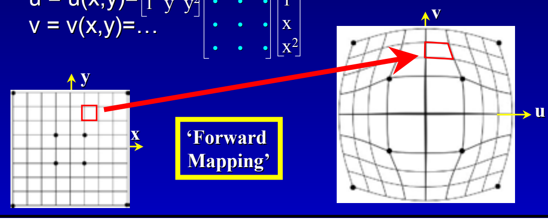


Image Warping: General Idea

2D→2D continuous coordinate map, a 'rubber sheet'

- PROBLEM: How would you undo such a warp?
- **Answer**: 'inverse mapping'; interate in (u,v)
- $x = x(u,v) = \begin{bmatrix} 1 & v & v^2 \end{bmatrix} \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} 1 \\ u \\ u^2 \end{bmatrix}$

  $y = y(u,v) = \ldots$

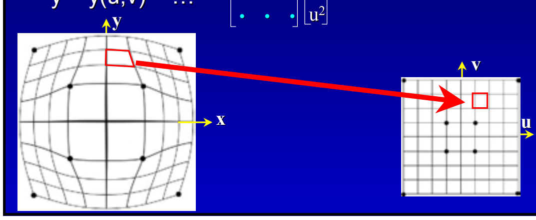


Image Warping: General Idea

2D→2D continuous coordinate map, a 'rubber sheet'

- PROBLEM: How would you undo such a warp?
- **2nd Answer**: 'parameter mapping'; interate in (s,t)
- $x = x(s,t) = \begin{bmatrix} 1 & t & t^2 \end{bmatrix} \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} 1 \\ s \\ s^2 \end{bmatrix}$

  $y = y(s,t) = \ldots$

  $u = u(s,t)$

  $v = v(s,t)$

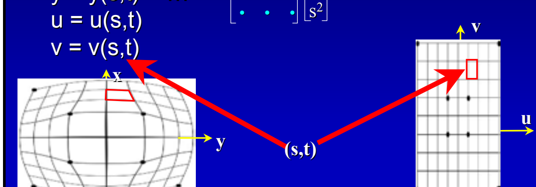


Image Warping: General Idea

2D→2D continuous coordinate map, a 'rubber sheet'

- PROBLEM: How would you undo such a warp?
- **Answer**: 'inverse mapping'; interate in (u,v)
- $x = x(u,v) = \begin{bmatrix} 1 & v & v^2 \end{bmatrix} \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} 1 \\ u \\ u^2 \end{bmatrix}$
  $y = y(u,v) = \ldots$

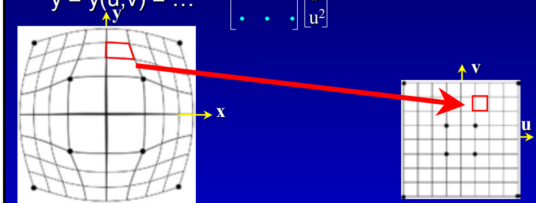


Image Warping: General Idea

2D→2D continuous coordinate map: 'rubber sheet'

PROBLEM: inverse **required** sometimes (mess)

PROBLEM: pixels aren't continuous; sampling errors

- aliasing (output pixels skip some input pixels)
- holes (input pixels skip some output pixels)

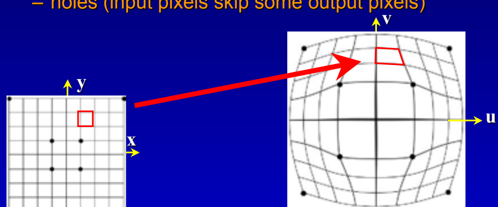


Image Warping: IBMR Form

2D→2D continuous coordinate map: 'rubber sheet'

PROBLEM: inverse **required** sometimes (mess)

IBMR Answer: H matrix is invertible

PROBLEM: pixels aren't continuous; sampling errors

IBMR Answer: Use vertices, not pixels:
let OpenGL texture mapping keep the image "continuous"

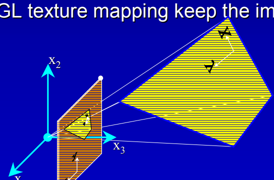
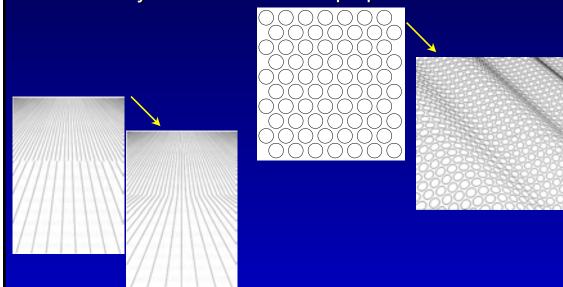


Image Warping: IBMR Form

- Projective transformation H in P2 is a 'warp'
- But many more kinds of warps possible!



Panoramas: Planar 'Bow-Tie'

- – For limited-size mosaics only (angle limits)
- – Find all **H** from correspondence in overlapped regions
- – Choose a (central) reference image (book pg. 196)
- – Reproject, cross-dissolve in reference image plane

Panoramas: Sphere / Cylinder

- Assemble from P^2 Correspondences:
  - – Find H to link each image only to its neighbors
- Spherical/Cylindrical? Do this **last**
  - – CAN'T convert planar-spherical & then easily align in 2D because…
  - – Spherical images behave poorly near poles
  - – e.g. 'Can't comb the hairs on a tennis ball'

  (no uniform 2D rectangular sampling grid exists)

Panorama Making

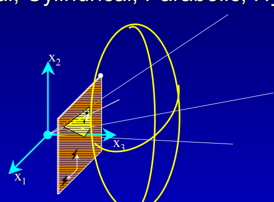
Planar:
- Start with overlapped planar proj. images
- Do 4-point corresp. (or better) for alignment
- Merge images by cross-dissolve

Early IBR: QuickTime VR (Chen, Williams '93)

1) Four Planar Images → 1 Cylindrical Panorama:

Image Warping: IBMR Form

- Especially useful: 3D-like warps in P2:
- Plane is just ONE of many shapes formed by varying x_2 …
- Spherical, Cylindrical, Parabolic, Hyperbolic…



Early IBR: QuickTime VR (Chen, Williams '93)

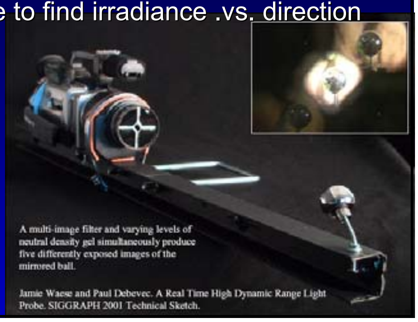
2) Windowing, Horizontal-only Reprojection:

Light Probes: What?

- Photograph a mirrored sphere
- warp image to find irradiance .vs. direction

1 picture==
half-sphere

High contrast?
Full sphere?
More Pictures!



END