Panoramas

What’s inside your fridge?
Announcements

• Project 3 (Autostitch) is out, due next Thursday, March 28 by 11:59pm
  – Artifact due Friday, March 29 by 11:59pm
  – Project to be done in pairs

• Take-home midterm
  – To be distributed at the end of class
  – Due at the beginning of class in one week, Wednesday, March 20
Camera projection matrix: recap

Let $q = (x, y, z, 1)$ be a point in homogeneous image coordinates.

The projection matrix $\Pi$ maps $q$ to $\Pi q$ in homogeneous coordinates.
The $K$ matrix converts 3D rays in the camera’s coordinate system to 2D image points in image (pixel) coordinates.

The part converts 3D points in world coordinates to 3D rays in the camera’s coordinate system. There are 6 parameters represented (3 for position/translation, 3 for rotation).
\[ \Pi = K \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & 0 \\ 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I_{3 \times 3} & -c \\ 0 & 0 & 0 & 1 \end{bmatrix} \]

Projection matrix

\[ \Pi = K \begin{bmatrix} R & -Rc \end{bmatrix} \]

(t in book’s notation)
Typical intrinsics matrix

\[ K = \begin{bmatrix}
-f & 0 & x_c \\
0 & -f & y_c \\
0 & 0 & 1
\end{bmatrix} \]

- **2D affine transform** corresponding to a scale by \( f \) (focal length) and a translation by \((x_c, y_c)\) (principal point)
Questions?
Perspective distortion

• Problem for architectural photography: converging verticals
Perspective distortion

• Problem for architectural photography: converging verticals

Tilting the camera upwards results in converging verticals

Keeping the camera level, with an ordinary lens, captures only the bottom portion of the building

Shifting the lens upwards results in a picture of the entire subject

• Solution: view camera (lens shifted w.r.t. film)

(Corresponds to shifting the principal point)

Source: F. Durand
Perspective distortion

• Problem for architectural photography: converging verticals

• Result:
Perspective distortion

• What does a sphere project to?

Image source: F. Durand
Perspective distortion

- The exterior columns appear bigger
- The distortion is not due to lens flaws
- Problem pointed out by Da Vinci
Perspective distortion: People
Lens distortion

• **Radial distortion** of the image
  – Caused by imperfect lenses
  – Points are distorted along radial lines
  – Deviations are most noticeable for rays that pass through the edge of the lens
Radial distortion

(a) Orthoscopic

(b) Barrel

(c) Pin-cushion
Radial distortion

- Arrows show motion of projected points relative to an ideal (distortion-free lens)

[Image credit: J. Bouguet http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example.html]
Correcting radial distortion

from Helmut Dersch
Modeling distortion

To model lens distortion

- Use the above conversion of rays to pixels, rather than simply multiplying by the intrinsics matrix.
Other types of projection

- Lots of intriguing variants...
- (I’ll just mention a few fun ones)
360 degree field of view...

• Basic approach
  – Take a photo of a parabolic mirror with an orthographic lens (Nayar)
  – Or buy one a lens from a variety of omnicam manufacturers...
    • See [http://www.cis.upenn.edu/~kostas/omni.html](http://www.cis.upenn.edu/~kostas/omni.html)
Tilt-shift

http://www.northlight-images.co.uk/article_pages/tilt_and_shift_ts-e.html

Tilt-shift images from Olivo Barbieri and Photoshop imitations
Rotating sensor (or object)

Rollout Photographs © Justin Kerr
http://research.famsi.org/kerrmaya.html

Also known as “cyclographs”, “peripheral images”
Can we use homographies to create a 360 panorama?
Idea: projecting images onto a common plane

Each image is warped with a homography $H$.

We’ll see what this homography means next.

Can’t create a 360 panorama this way... we’ll fix this shortly.
Creating a panorama

• Basic Procedure
  – Take a sequence of images from the same position
    • Rotate the camera about its optical center
  – Compute transformation between second image and first
  – Transform the second image to overlap with the first
  – Blend the two together to create a mosaic
  – If there are more images, repeat
If we capture all 360° of rays, we can create a 360° panorama.

The basic operation is *projecting* an image from one plane to another.

The projective transformation is scene-INDEPENDENT.

- This depends on all the images having the same optical center.
Image reprojection

• Basic question
  – How to relate two images from the same camera center?
    • how to map a pixel from PP1 to PP2

Answer

• Cast a ray through each pixel in PP1
• Draw the pixel where that ray intersects PP2
What is the transformation?

How do we map points in image 2 into image 1?

intrinsics

intrinsics

extrinsics (rotation only)

Step 1: Convert pixels in image 2 to rays in camera 2’s coordinate system.

\[
\begin{bmatrix}
X_2 \\
Y_2 \\
Z_2
\end{bmatrix} = K_2^{-1}
\begin{bmatrix}
x_2 \\
y_2 \\
1
\end{bmatrix}
\]

Step 2: Convert rays in camera 2’s coordinates to rays in camera 1’s coordinates.

\[
\begin{bmatrix}
X_1 \\
Y_1 \\
Z_1
\end{bmatrix} = R_2^T K_2^{-1}
\begin{bmatrix}
x_2 \\
y_2 \\
1
\end{bmatrix}
\]

Step 3: Convert rays in camera 1’s coordinates to pixels in image 1’s coordinates.

\[
\begin{bmatrix}
x_1 \\
y_1 \\
1
\end{bmatrix} \sim K_1 R_2^T K_2^{-1}
\begin{bmatrix}
x_2 \\
y_2 \\
1
\end{bmatrix}
\]

3x3 homography
Can we use homography to create a 360 panorama?
Panoramas

- What if you want a $360^\circ$ field of view?
Spherical projection

- Map 3D point \((X, Y, Z)\) onto sphere

\[
(x, y, z) = \frac{1}{\sqrt{X^2 + Y^2 + Z^2}}(X, Y, Z)
\]

- Convert to spherical coordinates

\[
(sin \theta cos \phi, sin \phi, cos \theta cos \phi) = (\tilde{x}, \tilde{y}, \tilde{z})
\]

- Convert to spherical image coordinates

\[
(\tilde{x}, \tilde{y}) = (s \theta, s \phi) + (\tilde{x}_c, \tilde{y}_c)
\]

- \(s\) defines size of the final image
  - often convenient to set \(s = \text{camera focal length}\)
Unwrapping a sphere

Credit: JHT’s Planetary Pixel Emporium
Spherical reprojection

- Map image to spherical coordinates
  - need to know the focal length
Aligning spherical images

• Suppose we rotate the camera by $\theta$ about the vertical axis
  – How does this change the spherical image?
Aligning spherical images

- Suppose we rotate the camera by $\theta$ about the vertical axis
  - How does this change the spherical image?
    - Translation by $\theta$
  - This means that we can align spherical images by translation
Assembling the panorama

• Stitch pairs together, blend, then crop
Problem: Drift

• Error accumulation
  – small errors accumulate over time
Problem: Drift

Solution

- add another copy of first image at the end
- this gives a constraint: \( y_n = y_1 \)
- there are a bunch of ways to solve this problem
  - add displacement of \((y_1 - y_n)/(n - 1)\) to each image after the first
  - apply an affine warp: \( y' = y + ax \) [you will implement this for P3]
  - run a big optimization problem, incorporating this constraint
    - best solution, but more complicated
    - known as “bundle adjustment”
Project 3

• Take pictures on a tripod (or handheld)
• Warp to spherical coordinates (optional if using homographies to align images)
• Extract features
• Align neighboring pairs using RANSAC
• Write out list of neighboring translations
• Correct for drift
• Read in warped images and blend them
• Crop the result and import into a viewer

• Roughly based on Autostitch
  – By Matthew Brown and David Lowe
Spherical panoramas

Microsoft Lobby: http://www.acm.org/pubs/citations/proceedings/graph/258734/p251-szeliski
Different projections are possible

Cube-map
Blending

- We’ve aligned the images – now what?
Blending

• Want to seamlessly blend them together
Image Blending
Effect of window size
Effect of window size
Good window size

“Optimal” window: smooth but not ghosted

- Doesn’t always work...
Create a Laplacian pyramid, blend each level

The Laplacian Pyramid

Gaussian Pyramid

\[ G_i = L_i + \text{expand}(G_{i+1}) \]

Laplacian Pyramid

\[ L_i = G_i - \text{expand}(G_{i+1}) \]

\[ G_n = L_n \]

\[ G_{n-1} = L_{n-1} \]

\[ G_{n-2} = L_{n-2} \]

\[ G_0 = L_0 \]
Encoding blend weights:  \( I(x,y) = (\alpha R, \alpha G, \alpha B, \alpha) \)  

color at \( p = \) \[
\frac{(\alpha_1 R_1, \alpha_1 G_1, \alpha_1 B_1) + (\alpha_2 R_2, \alpha_2 G_2, \alpha_2 B_2) + (\alpha_3 R_3, \alpha_3 G_3, \alpha_3 B_3)}{\alpha_1 + \alpha_2 + \alpha_3}
\]

Implement this in two steps:

1. accumulate:  add up the (\( \alpha \) premultiplied) RGB\( \alpha \) values at each pixel  
2. normalize:  divide each pixel’s accumulated RGB by its \( \alpha \) value  

Q: what if \( \alpha = 0 \)?

Optional:  see Blinn (CGA, 1994) for details:  
http://ieeexplore.ieee.org/iel1/38/7531/00310740.pdf?isNumber=7531&prod=JNL&arnumber=310740&arSt=83&ared=87&author=Blinn%2C+J.F.
Poisson Image Editing

For more info: Perez et al, SIGGRAPH 2003
  - http://research.microsoft.com/vision/cambridge/papers/perez_siggraph03.pdf
Some panorama examples

“Before SIGGRAPH Deadline” Photo credit: Doug Zongker
Some panorama examples

• Every image on Google Streetview
M. Uyttendaele, A. Eden, and R. Szeliski.

Eliminating ghosting and exposure artifacts in image mosaics.

M. Uyttendaele, A. Eden, and R. Szeliski.

Eliminating ghosting and exposure artifacts in image mosaics.

Other types of mosaics

- Can mosaic onto *any* surface if you know the geometry
  - See NASA’s [Visible Earth project](http://earthobservatory.nasa.gov/Newsroom/BlueMarble/) for some stunning earth mosaics
    - Click for images...
Questions?
Alternative to feathering

• **Cut** and **fuse**