Lecture 7: Image Alignment and Panoramas

What’s inside your fridge?
http://www.cs.washington.edu/education/courses/cse590ss/01wi/
Projection matrix

\[
\Pi = K \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
\end{bmatrix} \begin{bmatrix}
R & 0 \\
0 & 0 \\
0 & 0 & 1 \\
\end{bmatrix} \begin{bmatrix}
I_{3\times3} & -c \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

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

(t in book's notation)
Projection matrix

\[ \mathbf{q} = (x, y, z, 1) \]

(in homogeneous image coordinates)
Questions?
Image alignment

Full screen panoramas (cubic):  http://www.panoramas.dk/
Why Mosaic?

• Are you getting the whole picture?
  – Compact Camera FOV = 50 x 35°
Why Mosaic?

- Are you getting the whole picture?
  - Compact Camera FOV = 50 x 35°
  - Human FOV = 200 x 135°
Why Mosaic?

- Are you getting the whole picture?
  - Compact Camera FOV = 50 x 35°
  - Human FOV = 200 x 135°
  - Panoramic Mosaic = 360 x 180°
Mosaics: stitching images together
Readings

• Szeliski:
  – Chapter 6.1: Feature-based alignment
  – Chapter 9: Panorama stitching
Image alignment

Image taken from same viewpoint, just rotated.

Can we line them up?
Image alignment

Why don’t these image line up exactly?
What is the geometric relationship between these two images?
What is the geometric relationship between these two images?
Is this an affine transformation?
Where do we go from here?

affine transformation

\[
\begin{bmatrix}
  a & b & c \\
  d & e & f \\
  0 & 0 & 1 \\
\end{bmatrix}
\]

what happens when we change this row?
Projective Transformations aka Homographies aka Planar Perspective Maps

\[
H = \begin{bmatrix}
    a & b & c \\
    d & e & f \\
    g & h & 1
\end{bmatrix}
\]

Called a homography (or planar perspective map)

projection of 3D plane can be explained by a (homogeneous) 2D transform
Homographies

\[
\begin{bmatrix}
x' \\
y' \\
w'
\end{bmatrix} = \begin{bmatrix}
a & b & c \\
d & e & f \\
g & h & 1
\end{bmatrix} \begin{bmatrix}
x \\
y \\
1
\end{bmatrix}
\]

What happens when the denominator is 0?

\[
\begin{bmatrix}
\frac{ax+by+c}{gx+hy+1} \\
\frac{dx+ey+f}{gx+hy+1} \\
1
\end{bmatrix}
\]
Homographies

• Example on board
Image warping with homographies

image plane in front

black area where no pixel maps to
Homographies
Homographies

- Homographies ...
  - Affine transformations, and
  - Projective warps

\[
\begin{bmatrix}
x' \\
y' \\
w'
\end{bmatrix} = \begin{bmatrix}
a & b & c \\
d & e & f \\
g & h & 1
\end{bmatrix} \begin{bmatrix}
x \\
y \\
w
\end{bmatrix}
\]

- Properties of projective transformations:
  - Origin does not necessarily map to origin
  - Lines map to lines
  - Parallel lines do not necessarily remain parallel
  - Ratios are not preserved
  - Closed under composition
2D image transformations

These transformations are a nested set of groups

- Closed under composition and inverse is a member

<table>
<thead>
<tr>
<th>Name</th>
<th>Matrix</th>
<th># D.O.F.</th>
<th>Preserves:</th>
<th>Icon</th>
</tr>
</thead>
</table>
| translation     | \[
I \mid t
\]_{2\times3} | 2        | orientation + ⋅ ⋅ ⋅ |   |
| rigid (Euclidean)| \[
R \mid t
\]_{2\times3} | 3        | lengths + ⋅ ⋅ ⋅ |   |
| similarity      | \[
sR \mid t
\]_{2\times3} | 4        | angles + ⋅ ⋅ ⋅  |   |
| affine          | \[
A
\]_{2\times3} | 6        | parallelism + ⋅ ⋅ ⋅ |   |
| projective      | \[
\tilde{H}
\]_{3\times3} | 8        | straight lines  |   |
Questions?
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
Geometric interpretation of mosaics
• 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
Projecting images onto a common plane
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 transform image 2 onto image 1’s projection plane?

\[
\begin{align*}
\text{image 1} & \quad \text{image 2} \\
\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} & = K_2^{-1} \\
\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} \\
\begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} & = R_1 R_2^T K_2^{-1} \\
\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} \\
\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} & \sim K_1 R_1 R_2^T K_2^{-1} \\
\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix}
\end{align*}
\]

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

• What if you want a $360^\circ$ field of view?

mosaic Projection Sphere
Spherical projection

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

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

- 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 \(s = \) camera focal length
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
Unwrapping a sphere

Credit: JHT’s Planetary Pixel Emporium
Spherical panoramas

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

• 3-minute break
Computing transformations

- Given a set of matches between images A and B
  - How can we compute the transform T from A to B?
  - Find transform T that best “agrees” with the matches
Computing transformations
Simple case: translations

How do we solve for $(x_t, y_t)$?
Simple case: translations

Displacement of match $i = (x'_i - x_i, y'_i - y_i)$

$$\begin{pmatrix} x_t, y_t \end{pmatrix} = \begin{pmatrix} \frac{1}{n} \sum_{i=1}^{n} x'_i - x_i, \frac{1}{n} \sum_{i=1}^{n} y'_i - y_i \end{pmatrix}$$
Another view

- System of linear equations
  - What are the knowns? Unknowns?
  - How many unknowns? How many equations (per match)?
Problem: more equations than unknowns

- "Overdetermined" system of equations
- We will find the least squares solution

\[ x_i + x_t = x_i' \]
\[ y_i + y_t = y_i' \]
Least squares formulation

- For each point \((x_i, y_i)\)
  \[
  x_i + x_t = x'_i \\
  y_i + y_t = y'_i
  \]

- we define the *residuals* as
  \[
  r_{x_i}(x_t) = (x_i + x_t) - x'_i \\
  r_{y_i}(y_t) = (y_i + y_t) - y'_i
  \]
Least squares formulation

• Goal: minimize sum of squared residuals

\[ C(x_t, y_t) = \sum_{i=1}^{n} \left( r_{x_i} (x_t)^2 + r_{y_i} (y_t)^2 \right) \]

• “Least squares” solution

• For translations, is equal to mean displacement
Least squares formulation

- Can also write as a matrix equation

\[
\begin{bmatrix}
1 & 0 \\
0 & 1 \\
1 & 0 \\
0 & 1 \\
\vdots \\
1 & 0 \\
0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
x_t \\
y_t \\
\end{bmatrix}
=
\begin{bmatrix}
x'_1 - x_1 \\
y'_1 - y_1 \\
x'_2 - x_2 \\
y'_2 - y_2 \\
\vdots \\
x'_n - x_n \\
y'_n - y_n \\
\end{bmatrix}
\]
Least squares

\[ \mathbf{A} \mathbf{t} = \mathbf{b} \]

- Find \( \mathbf{t} \) that minimizes

\[ \| \mathbf{A} \mathbf{t} - \mathbf{b} \|^2 \]

- To solve, form the *normal equations*

\[ \mathbf{A}^T \mathbf{A} \mathbf{t} = \mathbf{A}^T \mathbf{b} \]

\[ \mathbf{t} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \]
Affine transformations

\[
\begin{bmatrix}
  x' \\
  y' \\
  1
\end{bmatrix} = \begin{bmatrix}
  a & b & c \\
  d & e & f \\
  0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix}
\]

• How many unknowns?
• How many equations per match?
• How many matches do we need?
Affine transformations

• Residuals:
  \[ r_{x_i}(a, b, c, d, e, f) = (ax_i + by_i + c) - x'_i \]
  \[ r_{y_i}(a, b, c, d, e, f) = (dx_i + ey_i + f) - y'_i \]

• Cost function:
  \[ C(a, b, c, d, e, f) = \sum_{i=1}^{n} \left( (r_{x_i}(a, b, c, d, e, f))^2 + (r_{y_i}(a, b, c, d, e, f))^2 \right) \]
Affine transformations

- Matrix form

\[
\begin{bmatrix}
 x_1 & y_1 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & x_1 & y_1 & 1 \\
 x_2 & y_2 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & x_2 & y_2 & 1 \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 x_n & y_n & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & x_n & y_n & 1
\end{bmatrix}
\begin{bmatrix}
 a \\
 b \\
 c \\
 d \\
 e \\
 f
\end{bmatrix}
= \begin{bmatrix}
 x_1' \\
 y_1' \\
 x_2' \\
 y_2' \\
 \vdots \\
 x_n' \\
 y_n'
\end{bmatrix}
\]

\[
A_{2n \times 6} \begin{bmatrix} t_{6 \times 1} \end{bmatrix} = b_{2n \times 1}
\]
Homographies

To unwarp (rectify) an image

• solve for homography $H$ given $p$ and $p'$
• solve equations of the form: $wp' = Hp$
  – linear in unknowns: $w$ and coefficients of $H$
  – $H$ is defined up to an arbitrary scale factor

how many points are necessary to solve for $H$?
Solving for homographies

\[
\begin{bmatrix}
  x'_i \\
  y'_i \\
  1
\end{bmatrix}
= \frac{
\begin{bmatrix}
  h_{00} & h_{01} & h_{02} \\
  h_{10} & h_{11} & h_{12} \\
  h_{20} & h_{21} & h_{22}
\end{bmatrix}
\begin{bmatrix}
  x_i \\
  y_i \\
  1
\end{bmatrix}
}{h_{20} x_i + h_{21} y_i + h_{22}}
\]

\[
x'_i = \frac{h_{00} x_i + h_{01} y_i + h_{02}}{h_{20} x_i + h_{21} y_i + h_{22}}
\]

\[
y'_i = \frac{h_{10} x_i + h_{11} y_i + h_{12}}{h_{20} x_i + h_{21} y_i + h_{22}}
\]

\[
x'_i (h_{20} x_i + h_{21} y_i + h_{22}) = h_{00} x_i + h_{01} y_i + h_{02}
\]

\[
y'_i (h_{20} x_i + h_{21} y_i + h_{22}) = h_{10} x_i + h_{11} y_i + h_{12}
\]
Solving for homographies

\[ x'_i(h_{20}x_i + h_{21}y_i + h_{22}) = h_{00}x_i + h_{01}y_i + h_{02} \]
\[ y'_i(h_{20}x_i + h_{21}y_i + h_{22}) = h_{10}x_i + h_{11}y_i + h_{12} \]
Solving for homographies

Define a least squares problem:

\[
\begin{pmatrix}
  x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1x_1 & -x'_1y_1 & -x'_1 \\
  0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1x_1 & -y'_1y_1 & -y'_1 \\
  \vdots \\
  x_n & y_n & 1 & 0 & 0 & 0 & -x'_nx_n & -x'_ny_n & -x'_n \\
  0 & 0 & 0 & x_n & y_n & 1 & -y'_nx_n & -y'_ny_n & -y'_n \\
\end{pmatrix}
\begin{pmatrix}
  h_{00} \\
  h_{01} \\
  h_{02} \\
  h_{10} \\
  h_{11} \\
  h_{12} \\
  h_{20} \\
  h_{21} \\
  h_{22} \\
\end{pmatrix}
= \begin{pmatrix}
  0 \\
  0 \\
  \vdots \\
  0 \\
  0 \\
\end{pmatrix}
\]

\[
\begin{pmatrix}
  h_0 \\
  h_1 \\
  h_2 \\
\end{pmatrix}
\]

Defines a least squares problem: minimize \( \|Ah - 0\|_2^2 \)

- Since \( h \) is only defined up to scale, solve for unit vector \( \hat{h} \)
- Solution: \( \hat{h} = \text{eigenvector of } A^TA \) with smallest eigenvalue
- Works with 4 or more points
Questions?