Last time

• Feature-based alignment
  – 2D transformations
  – Affine fit
  – RANSAC

Alignment problem

• In alignment, we will fit the parameters of some transformation according to a set of matching feature pairs ("correspondences").
Main questions

Alignment: Given two images, what is the transformation between them?

Warping: Given a source image and a transformation, what does the transformed output look like?

Motivation for feature-based alignment:
Recognition

Figures from David Lowe

Motivation for feature-based alignment:
Medical image registration

Slide credit: Kristen Grauman
Motivation for feature-based alignment: Image mosaics

Parametric (global) warping

Examples of parametric warps:
- translation
- rotation
- aspect
- affine
- perspective

Transformation $T$ is a coordinate-changing machine:

$p = (x, y)$
$p' = (x', y')$

What does it mean that $T$ is global?
- Is the same for any point $p$
- Can be described by just a few numbers (parameters)

Let’s represent $T$ as a matrix:

$$
\begin{bmatrix}
    x' \\
    y'
\end{bmatrix} = \begin{bmatrix}
    M
\end{bmatrix}
\begin{bmatrix}
    x \\
    y
\end{bmatrix}
$$
Homogeneous coordinates

To convert to homogeneous coordinates:

\[(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}\]

homogeneous image coordinates

Converting from homogeneous coordinates

\[\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)\]

Slide credit: Kristen Grauman

2D Affine Transformations

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

Affine transformations are combinations of …
- Linear transformations, and
- Translations

Parallel lines remain parallel

Slide credit: Kristen Grauman

Projective Transformations

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

Projective transformations:
- Affine transformations, and
- Projective warps

Parallel lines do not necessarily remain parallel

Slide credit: Kristen Grauman
Fitting an affine transformation

- Assuming we know the correspondences, how do we get the transformation?

\[
\begin{bmatrix}
  x'_i \\
  y'_i
\end{bmatrix} = \begin{bmatrix}
  m_1 & m_2 & x_i & t_x \\
  m_3 & m_4 & y_i & t_y
\end{bmatrix} \begin{bmatrix}
  x_i \\
  y_i \\
  1
\end{bmatrix}
\]

RANSAC: General form

- **RANSAC loop:**
  1. Randomly select a seed group of points on which to base transformation estimate (e.g., a group of matches)
  2. Compute transformation from seed group
  3. Find inliers to this transformation
  4. If the number of inliers is sufficiently large, re-compute estimate of transformation on all of the inliers

- Keep the transformation with the largest number of inliers

RANSAC example: Translation

- Select one match, count inliers
RANSAC example: Translation

Select one match, count inliers

RANSAC example: Translation

Find “average” translation vector

RANSAC pros and cons

- **Pros**
  - Simple and general
  - Applicable to many different problems
  - Often works well in practice
- **Cons**
  - Lots of parameters to tune
  - Doesn’t work well for low inlier ratios (too many iterations, or can fail completely)
Today

- Image mosaics
  - Fitting a 2D transformation
    - Homography
  - 2D image warping
  - Computing an image mosaic

HP frames commercial

- [http://www.youtube.com/watch?v=2RPl5vPExQk](http://www.youtube.com/watch?v=2RPl5vPExQk)

Mosaics

Obtain a wider angle view by combining multiple images.

Slide credit: Kristen Grauman
Panoramic Photos are old

- Sydney, 1875
- Beirut, late 1800's

How to stitch together a panorama (a.k.a. mosaic)?

- 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)

  ...but wait, why should this work at all?
  - What about the 3D geometry of the scene?
  - Why aren’t we using it?

Pinhole camera

- Pinhole camera is a simple model to approximate imaging process, perspective projection.

If we treat pinhole as a point, only one ray from any given point can enter the camera.
Mosaics: generating synthetic views

Can generate any synthetic camera view as long as it has the same center of projection!

Source: Alyosha Efros

Mosaics

Obtain a wider angle view by combining multiple images.

Image reprojection

The mosaic has a natural interpretation in 3D
- The images are reprojected onto a common plane
- The mosaic is formed on this plane
- Mosaic is a synthetic wide-angle camera

Source: Steve Seitz
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

Observation:
Rather than thinking of this as a 3D reprojection, think of it as a 2D image warp from one image to another.

Image reprojection: Homography

A projective transform is a mapping between any two PPs with the same center of projection
• Rectangle should map to arbitrary quadrilateral
• Parallel lines aren’t preserved
• But must preserve straight lines

called Homography

The projective plane

Why do we need homogeneous coordinates?
• Represent points at infinity, homographies, perspective projection, multi-view relationships

What is the geometric intuition?
• A point in the image is a ray in projective space

• Each point (x, y) on the plane is represented by a ray \((sx, sy, s)\)
  – All points on the ray are equivalent: \((x, y, 1) \equiv (sx, sy, s)\)
To compute the homography given pairs of corresponding points in the images, we need to set up an equation where the parameters of $H$ are the unknowns...
Solving for homographies

\[
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
-x_1c_1 \\
-x_1c_2 \\
-x_1c_3 \\
-x_1c_4
\end{pmatrix}
= \begin{pmatrix}
A_{10} \\
A_{20} \\
A_{30} \\
A_{40}
\end{pmatrix}
\]

\[
\begin{pmatrix}
A_{11} \\
A_{21} \\
A_{31} \\
A_{41}
\end{pmatrix}
\begin{pmatrix}
h_1 \\
h_2 \\
h_3 \\
h_4
\end{pmatrix}
= \begin{pmatrix}
0 \\
0 \\
0 \\
0
\end{pmatrix}
\]

\[
A = \begin{pmatrix}
2n \times 9 \\
9 \times 2n
\end{pmatrix}
\]

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

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

Homography

\[
(x, y) \rightarrow \left( \begin{array}{c}
w_{x'} \\
w_{y'} \\
w
\end{array} \right)
= \left( \begin{array}{c}
x' \\
y'
\end{array} \right)
\]

To apply a given homography \( H \)

- Compute \( p' = Hp \) (regular matrix multiply)
- Convert \( p' \) from homogeneous to image coordinates

Today

- RANSAC for robust fitting
  - Lines, translation
- Image mosaics
  - Fitting a 2D transformation
    - Homography
  - 2D image warping
  - Computing an image mosaic
Image warping

Given a coordinate transform and a source image \( f(x,y) \), how do we compute a transformed image \( g(x',y') = f(T(x,y)) \)?

Forward warping

Send each pixel \( f(x,y) \) to its corresponding location \( (x',y') = T(x,y) \) in the second image.

Q: what if pixel lands “between” two pixels?

A: distribute color among neighboring pixels \( (x',y') \) — known as “splatting”
Inverse warping

Get each pixel \( g(x',y') \) from its corresponding location \( (x,y) = T^{-1}(x',y') \) in the first image.

Q: what if pixel comes from “between” two pixels?

Inverse warping

Get each pixel \( g(x',y') \) from its corresponding location \( (x,y) = T^{-1}(x',y') \) in the first image.

Q: what if pixel comes from “between” two pixels?

A: Interpolate color value from neighbors
- nearest neighbor, bilinear...

Bilinear interpolation

Sampling at \( f(x,y) \):

\[
f(x, y) = (1 - a)(1 - b) f[i,j] + a(1 - b) f[i + 1,j] + a b f[i + 1, j + 1] + (1 - a)b f[i, j + 1]
\]
Recap: How to stitch together a panorama (a.k.a. mosaic)?

- **Basic Procedure**
  - Take a sequence of images from the same position
    - Rotate the camera about its optical center
  - Compute transformation (homography) between second image and first using corresponding points.
  - Transform the second image to overlap with the first.
  - Blend the two together to create a mosaic.
  - (If there are more images, repeat)

---

Image warping with homographies

Image rectification

---

Slide credit: Kristen Grauman
Analysing patterns and shapes

What is the shape of the b/w floor pattern?

The floor (enlarged)

Automatically rectified floor

Slide from Antonio Criminisi

From Martin Kemp. The Science of Art (manual reconstruction)

Analysing patterns and shapes

Automatic rectification

St. Lucy Altarpiece, D. Veneziano

Automatically rectified floor

Slide from Criminisi
Analysing patterns and shapes

Automatic rectification

From Martin Kemp, *The Science of Art* (manual reconstruction)

Julian Beever: Manual Homographies

[Image of Julian Beever's street art examples]

http://users.skynet.be/J.Beever/pave.htm

Changing camera center

Does it still work?

[Diagram of camera center and perspective changes]

Source: Alyosha Efros
Recall: same camera center

Can generate synthetic camera view as long as it has the same center of projection.

Source: Alyosha Efros

…Or: Planar scene (or far away)

PP3 is a projection plane of both centers of projection, so we are OK!
This is how big aerial photographs are made.

Source: Alyosha Efros
RANSAC for estimating homography
• RANSAC loop:
  1. Select four feature pairs (at random)
  2. Compute homography $H$ (exact)
  3. Compute inliers where $SSD(p_i', Hp_i) < \varepsilon$
  4. Keep largest set of inliers
  5. Re-compute least-squares $H$ estimate on all of the inliers

Robust feature-based alignment

• Extract features
Robust feature-based alignment

• Extract features
• Compute putative matches

Source: L. Lazebnik

Robust feature-based alignment

• Extract features
• Compute putative matches
• Loop:
  – Hypothesize transformation $T$ (small group of putative matches that are related by $T$)

Source: L. Lazebnik

Robust feature-based alignment

• Extract features
• Compute putative matches
• Loop:
  – Hypothesize transformation $T$ (small group of putative matches that are related by $T$)
  – Verify transformation (search for other matches consistent with $T$)

Source: L. Lazebnik
Robust feature-based alignment

• Extract features
• Compute putative matches
• Loop:
  – Hypothesize transformation $T$ (small group of putative matches that are related by $T$)
  – Verify transformation (search for other matches consistent with $T$)

Summary: alignment & warping

• Write 2d transformations as matrix-vector multiplication (including translation when we use homogeneous coordinates)
• Fitting transformations: solve for unknown parameters given corresponding points from two views (affine, projective (homography)).
• Perform image warping (inverse)
• Mosaics: uses homography and image warping to merge views taken from same center of projection.

Next time: which features should we match?
Questions?