Homographies and Panoramas

Slides from Steve Seitz, Rick Szeliski, Alexei Efros, Fredo Durand, and Kristin Grauman

cs129: Computational Photography
James Hays, Brown, Fall 2012
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°
Panoramic Photos are old

Sydney, 1875

Beirut, late 1800’s
Panorama Capture
Mosaics: stitching images together
Big idea

As with high dynamic range imaging, we are compensating for the shortcomings of traditional cameras by capturing and fusing multiple images.
Today’s lecture

Today we will assume that correspondence between photos is known and we focus on finding and applying homographies.

Wednesday and Friday we will discuss automatic correspondence.
How to do it?

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?
Aligning images

Translations are not enough to align the images
A pencil of rays contains all views

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

But don’t we need to know the geometry of the two planes with respect to the eye?

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

Which t-form is the right one for warping PP1 into PP2?
e.g. translation, Euclidean, affine, projective

- Translation: 2 unknowns
- Affine: 6 unknowns
- Perspective: 8 unknowns
Homography

A: Projective – mapping between any two PPs with the same center of projection

- rectangle should map to arbitrary quadrilateral
- parallel lines aren’t
- but must preserve straight lines
- same as: project, rotate, reproject

called Homography

\[
\begin{bmatrix}
wx' \\
wy' \\
w \\
p'
\end{bmatrix} = \begin{bmatrix}
* & * & * \\
* & * & * \\
* & * & * \\
* & * & *
\end{bmatrix} \begin{bmatrix}
x \\
y \\
l
\end{bmatrix}
\]

To apply a homography \( H \)

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

image plane in front

black area where no pixel maps to
Mosaics: main steps

- Collect correspondences (manually)
- Solve for homography matrix $H$

- Warp content from one image frame to the other to combine: say $im_1$ into $im_2$ reference frame

- Overlay $im_2$ content onto the warped $im_1$ content.
Homography

To apply a given homography $H$
- Compute $p' = Hp$ (regular matrix multiply)
- Convert $p'$ from homogeneous to image coordinates
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...
Least Squares Example

Say we have a set of data points \((X_1, X_1'), (X_2, X_2'), (X_3, X_3'), \ldots\) (e.g. person’s height vs. weight)

We want a nice compact formula (a line) to predict \(X\)'s from \(X\)s: \(Xa + b = X'\)

We want to find \(a\) and \(b\)

How many \((X,X')\) pairs do we need?

\[
\begin{align*}
X_1a + b &= X'_1 \\
X_2a + b &= X'_2 \\
X_3a + b &= X'_3 \\
&\vdots
\end{align*}
\]

What if the data is noisy?

\[
\begin{bmatrix}
X_1 \\
X_2 \\
X_3 \\
\vdots
\end{bmatrix}
\begin{bmatrix}
a \\
b
\end{bmatrix} =
\begin{bmatrix}
X'_1 \\
X'_2 \\
X'_3 \\
\vdots
\end{bmatrix}
\]

\[
\min \|Ax - B\|^2
\]

overconstrained
Solving for homographies

\[ p' = Hp \]

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

Can set scale factor \( i=1 \). So, there are 8 unknowns. Set up a system of linear equations:

\[ Ah = b \]

where vector of unknowns \( h = [a,b,c,d,e,f,g,h]^T \)

Need at least 8 eqs, but the more the better…

Solve for \( h \). If overconstrained, solve using least-squares:

\[ \min \| Ah - b \|^2 \]

>> help lmdivide
Example system for finding homography...

destination coordinates of the four corners of a quadrilateral. Let the correspondence map \((u_k, v_k)^T\) to \((x_k, y_k)^T\) for vertices numbered cyclically \(k = 0, 1, 2, 3\). All coordinates are assumed to be real (finite). To compute the forward mapping matrix \(M_{sd}\), assuming that \(i = 1\), we have eight equations in the eight unknowns \(a-h\):

\[
x_k = \frac{au_k + bv_k + c}{gu_k + hv_k + 1} \Rightarrow u_k a + v_k b + c - u_k x_k g - v_k x_k h = x_k
\]

\[
y_k = \frac{du_k + ev_k + f}{gu_k + hv_k + 1} \Rightarrow u_k d + v_k e + f - u_k y_k g - v_k y_k h = y_k
\]

for \(k = 0, 1, 2, 3\). This can be rewritten as an \(8 \times 8\) system:

\[
\begin{pmatrix}
    u_0 & v_0 & 1 & 0 & 0 & 0 & -u_0 x_0 & -v_0 x_0 \\
    u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1 x_1 & -v_1 x_1 \\
    u_2 & v_2 & 1 & 0 & 0 & 0 & -u_2 x_2 & -v_2 x_2 \\
    u_3 & v_3 & 1 & 0 & 0 & 0 & -u_3 x_3 & -v_3 x_3 \\
    0 & 0 & 0 & u_0 & v_0 & 1 & -u_0 y_0 & -v_0 y_0 \\
    0 & 0 & 0 & u_1 & v_1 & 1 & -u_1 y_1 & -v_1 y_1 \\
    0 & 0 & 0 & u_2 & v_2 & 1 & -u_2 y_2 & -v_2 y_2 \\
    0 & 0 & 0 & u_3 & v_3 & 1 & -u_3 y_3 & -v_3 y_3
\end{pmatrix}
\begin{pmatrix}
    a \\
    b \\
    c \\
    d \\
    e \\
    f \\
    g \\
    h
\end{pmatrix}
= 
\begin{pmatrix}
    x_0 \\
    x_1 \\
    x_2 \\
    x_3 \\
    y_0 \\
    y_1 \\
    y_2 \\
    y_3
\end{pmatrix}

Source: Paul Heckbert, *Fundamentals of Texture Mapping and Image Warping*
Mosaics: main steps

• Collect correspondences (manually)
• Solve for homography matrix $H$
  – Least squares solution
• Warp content from one image frame to the other to combine: say $\text{im1}$ into $\text{im2}$ reference frame
  – Determine bounds of the new combined image
    • Where will the corners of $\text{im1}$ fall in $\text{im2}$’s coordinate frame?
    • We will attempt to lookup colors for any of these positions we can get from $\text{im1}$. \texttt{:meshgrid}
  – Compute coordinates in $\text{im1}$’s reference frame (via homography) for all points in that range: $H^{-1}$
  – Lookup all colors for all these positions from $\text{im1}$
    • Inverse warp: \texttt{interp2} (watch for nans: \texttt{isnan})
• Overlay $\text{im2}$ content onto the warped $\text{im1}$ content.
  – Careful about new bounds of the output image: $\text{minx}$, $\text{miny}$
Mosaics: main steps

• Collect correspondences (manually)
• Solve for homography matrix H
  – Least squares solution
• Warp content from one image frame to the other to combine: say im1 into im2 reference frame
  – Determine bounds of the new combined image
    • Where will the corners of im1 fall in im2’s coordinate frame?
  • We will attempt to lookup colors for any of these positions we can get from im1. : meshgrid
  – Compute coordinates in im1’s reference frame (via homography) for all points in that range: \( H^{-1} \)
  – Lookup all colors for all these positions from im1
    • Inverse warp : interp2 (watch for nans : isnan)
• Overlay im2 content onto the warped im1 content.
  – Careful about new bounds of the output image: minx, miny
Mosaics: main steps

• Collect correspondences (manually)
• Solve for homography matrix H
  – Least squares solution
• Warp content from one image frame to the other to combine: say \textit{im1} into \textit{im2} reference frame
  – Determine bounds of the new combined image
    • Where will the corners of \textit{im1} fall in \textit{im2}’s coordinate frame?
    • We will attempt to lookup colors for any of these positions we can get from \textit{im1}.
      \texttt{meshgrid}
  – Compute coordinates in \textit{im1}’s reference frame (via homography) for all points in that range: \texttt{H}^{-1}
  – Lookup all colors for all these positions from \textit{im1}
    • Inverse warp: \texttt{interp2} (watch for \texttt{nans} : \texttt{isnan})
• Overlay \textit{im2} content onto the warped \textit{im1} content.
  – Careful about new bounds of the output image: \texttt{minx}, \texttt{miny}
Use interp2 to ask for the colors (possibly interpolated) from im1 at all the positions needed in im2’s reference frame.
Mosaics: main steps

• Collect correspondences (manually)
• Solve for homography matrix \( H \)
  – Least squares solution
• Warp content from one image frame to the other to combine: say \( \text{im1} \) into \( \text{im2} \) reference frame
  – Determine bounds of the new combined image
    • Where will the corners of \( \text{im1} \) fall in \( \text{im2} \)’s coordinate frame?
    • We will attempt to lookup colors for any of these positions we can get from \( \text{im1} \).
      : \text{meshgrid}
  – Compute coordinates in \( \text{im1} \)’s reference frame (via homography) for all points in that range: \( H^{-1} \)
  – Lookup all colors for all these positions from \( \text{im1} \)
    • Inverse warp: \text{interp2} (watch for nans: \text{isnan})
• Overlay \( \text{im2} \) content onto the warped \( \text{im1} \) content.
  – Careful about new bounds of the output image: \text{minx}, \text{miny}
Fun with homographies

Original image

Virtual camera rotations

St. Petersburg photo by A. Tikhonov
Analyzing patterns and shapes

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

The floor (enlarged)

Homography

Automatically rectified floor

Slide from Criminisi
Analyzing patterns and shapes

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

2 patterns have been discovered!
Analyzing patterns and shapes

What is the (complicated) shape of the floor pattern?

Autically rectified floor

*St. Lucy Altarpiece, D. Veneziano*

Slide from Criminisi
Analyzing patterns and shapes

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

Slide from Criminisi
Analyzing patterns and shapes

The Ambassadors by Hans Holbein the Younger, 1533
changing camera center

Does it still work?

PP1

PP2

synthetic PP
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
Planar mosaic

Image Compositing for Tele-Reality

1. Introduction
2. Previous work
3. Basic imaging equations
4. Planar image compositing
5. Panoramic
6. Piecewise-planar scenes
7. Scenes with arbitrary depth
8. 3-D model recovery
9. Applications
10. Discussion & Exercises