CS4495/6495
Introduction to Computer Vision

3D-L2 Homographies and mosaics
Projective Transformsations

Projective transformations: for 2D images it’s a 3x3 matrix applied to homogenous coordinates

$$\begin{bmatrix} w' & x' \end{bmatrix} = \begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} x \end{bmatrix}$$

$$\begin{bmatrix} w' & y' \end{bmatrix} = \begin{bmatrix} d & e & f \end{bmatrix} \begin{bmatrix} y \end{bmatrix}$$

$$\begin{bmatrix} w' \end{bmatrix} = \begin{bmatrix} g & h & i \end{bmatrix} \begin{bmatrix} w \end{bmatrix}$$
The projective plane

What is the geometric intuition of using homogenous coordinates?

- A point in the image is a ray in projective space
The projective plane

Each point \((x,y)\) on the plane (at \(z=1\)) is represented by a ray \((sx,sy,s)\)

All points on the ray are equivalent:
\[(x, y, 1) \cong (sx, sy, s)\]
Basic question:

How to relate two images from the same camera center?

How to map a pixel from projective plane PP1 to PP2?

Source: Alyosha Efros
Answer

- Cast a ray through each pixel in PP1
- Draw the pixel where that ray intersects PP2
Image reprojection

Observation:
• Rather than thinking of this as a 3D reprojection, think of it as a 2D **image warp** from one image (plane) to another (plane).
Application: Simple mosaics
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?
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.

*Source: Steve Seitz*
Warning: This model only holds for angular views up to 180°. Beyond that need to use sequence that “bends the rays” or map onto a different surface, say, a cylinder.
Obtain a wider angle view by combining multiple images *all of which are taken from the same camera center.*
Image reprojection: Homography

A projective transform is a mapping between any two PPs with the same center of projection.

- Lines map to lines
- So rectangle maps to arbitrary quadrilateral

Called Homography

\[
\begin{bmatrix}
wx' \\
w'y' \\
w \\
w
\end{bmatrix}
= \begin{bmatrix}
* & * & * \\
* & * & * \\
* & * & * \\
* & * & * \\
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
l \\
1
\end{bmatrix}
\]

\[
p' \quad H \quad p
\]

Source: Alyosha Efros
Homography

\[
\begin{align*}
(x_1, y_1) & \\
(x_2, y_2) & \\
& \quad \vdots \\
(x_N, y_N) & \\
\end{align*}
\]

\[
\begin{align*}
(x'_1, y'_1) & \\
(x'_2, y'_2) & \\
& \quad \vdots \\
(x'_N, y'_N) & \\
\end{align*}
\]
Solving for homographies

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

\[ p' = Hp \]
Solving for homographies – non-homogeneous

\[ p' = H\mathbf{p} \]

Since 8 unknowns, can set scale factor \( i = 1 \).
Set up a system of linear equations \( A\mathbf{h} = \mathbf{b} \) where vector of unknowns

\[ \mathbf{h} = [a, b, c, d, e, f, g, h]^T \]

Need at least 4 points for 8 eqs, but the more the better...
Solve for \( \mathbf{h} \) by \( \min \| A\mathbf{h} - \mathbf{b} \|^2 \) using least-squares
Solving for homographies – homogeneous

\[ \mathbf{p}' = \mathbf{Hp} \]

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

Just like we did for the extrinsics, multiply through, and divide out by \( w \). Gives two homogeneous equations per point.
Solve using SVD just like before. This is the cool way.
Apply the Homography

\[ p' = H p \]

\[ (w x'/w, w y'/w) = (x', y') \]
Mosaics
Mosaics for Video Coding

- Convert masked images into a background sprite for “content-based coding”
Quiz

We said that the transformation between two images taken from the same center of projection is a homography $H$. How many pairs of corresponding points do I need to compute $H$?

a) 6  
b) 4  
c) 2  
d) 8
We said that the transformation between two images taken from the same center of projection is a *homography* $H$. How many pairs of corresponding points do I need to compute $H$?

a) 6  
b) 4  
c) 2  
d) 8
Homographies and 3D planes

Remember this:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \approx \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$
Homographies and 3D planes

• Suppose the 3D points are on a plane:

\[ aX + bY + cZ + d = 0 \]
Homographies and 3D planes

• On the plane $[a \ b \ c \ d]$ can replace Z:

$$
\begin{bmatrix}
    u \\
    v \\
    1
\end{bmatrix}
\overset{\sim}{\rightarrow}
\begin{bmatrix}
    m_{00} & m_{01} & m_{02} & m_{03} \\
    m_{10} & m_{11} & m_{12} & m_{13} \\
    m_{20} & m_{21} & m_{22} & m_{23}
\end{bmatrix}
\begin{bmatrix}
    X \\
    Y \\
    (aX + bY + d) / (-c) \\
    1
\end{bmatrix}
$$
Homographies and 3D planes

• So, can put the Z coefficients into the others:

\[
\begin{bmatrix}
    u \\
    v \\
    1
\end{bmatrix} \approx \begin{bmatrix}
    m'_{00} & m'_{01} & 0 & m'_{03} \\
    m'_{10} & m'_{11} & 0 & m'_{13} \\
    m'_{20} & m'_{21} & 0 & m'_{23}
\end{bmatrix} \begin{bmatrix}
    X \\
    Y \\
    (aX + bY + d) / (-c) \\
    1
\end{bmatrix}
\]

3x3 Homography!
Image reprojection

• Mapping between planes is a homography.

• Whether a plane in the world to the image or between image planes.
Rectifying slanted views
Rectifying slanted views

Corrected image (front-to-parallel)
Measuring distances
Measurements on planes

Approach: unwarp then measure

What kind of warp is this?

*Homography*...
Image rectification

If there is a planar rectangular grid in the scene you can map it into a rectangular grid in the image...
Some other images of rectangular grids...
Same pixels – via a homography
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?
Forward warping

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

\[
f(x, y) = (1 - a)(1 - b) \ f[i, j] \\
+ a(1 - b) \ f[i + 1, j] \\
+ ab \ f[i + 1, j + 1] \\
+ (1 - a)b \ f[i, j + 1]
\]

See Matlab (Octave) function \texttt{interp2}
Review: How to make a panorama (or 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)