6.088 Digital and Computational Photography
6.882 Advanced Computational Photography

Panoramas

Frédéric Durand
MIT - EECS

Lots of slides stolen from Alyosha Efros, who stole them from Steve Seitz and Rick Szeliski

Why Mosaic?

• Are you getting the whole picture?
  – Compact Camera FOV = 50 x 35°
  – Human FOV = 200 x 135°

Mosaics: stitching images together

virtual wide-angle camera

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?
A pencil of rays contains all views

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

Aligning images: translation

Translations are not enough to align the images

Questions?

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

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

Image reprojection

- Basic question
  - How to relate 2 images from 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 in 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
Homography

- 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

\[
\begin{bmatrix}
w_x' \\
w_y' \\
w \\
w'
\end{bmatrix} = \begin{bmatrix} * & * & * & * \\
* & * & * & * \\
* & * & * & * \\
* & * & * & * \\
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
w \\
p
\end{bmatrix}
\]

To apply a homography \( H \)
- Compute \( p' = Hp \) (regular matrix multiply)
- Convert \( p' \) from homogeneous to image coordinates

1D homogeneous coordinates

- Add one dimension to make life simpler
- \((x, w)\) represent point \(x/w\)

1D homography

- Reproject to different line

\[
w = 1
\]

1D homography

- Reproject to different line

\[
w = 1
\]

1D homography

- Reproject to different line
- Equivalent to rotating 2D points
  \( \Rightarrow \) reproject is linear in homogeneous coordinates

Same in 2D

- Reprojection = homography
- 3x3 matrix
Image warping with homographies

Questions?

Digression: perspective correction

From Photography, London et al.

From Photography, London et al.
Tilt-shift lens

- 35mm SLR version

Photoshop version (perspective crop)

+ you control reflection and perspective independently

Back to Image rectification

To unwarp (rectify) an image

- Find the homography $H$ given a set of $p$ and $p'$ pairs
- How many correspondences are needed?
- Tricky to write $H$ analytically, but we can solve for it!
  - Find such $H$ that “best” transforms points $p$ into $p'$
  - Use least-squares!

Least Squares Example

- Say we have a set of data points $(X1,X1'), (X2,X2'), (X3,X3')$, etc. (e.g. person’s height vs. weight)
- We want a nice compact formula (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?
  - $X1a + b = X1'$
  - $X2a + b = X2'$
- What if the data is noisy?
  - $\min \|Ax - B\|

Questions?
Solving for homographies

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

- Can set scale factor \( i = 1 \). So, there are 8 unknowns.
- Set up a system of linear equations:
  \[ Ah = b \]
- Note: we do not know \( w \) but we can compute it from \( x \) & \( y \)
  \[ w = gx + hy + 1 \]
- The equations are linear in the unknowns

Solving for homographies

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

- Can set scale factor \( i = 1 \). So, there are 8 unknowns.
- Set up a system of linear equations:
  \[ Ah = b \]
- Need at least 8 eqs, but the more the better…
- Solve for \( h \). If overconstrained, solve using least-squares:
  \[ \min \| Ah - b \| \]
- Can be done in Matlab using \( \text{mldivide} \) command
  - see “help mldivide”

Questions?

Panoramas

1. Pick one image (red)
2. Warp the other images towards it (usually, one by one)
3. Blend

Recap

- Panorama = reprojection
- 3D rotation \( \rightarrow \) homography
  - Homogeneous coordinates are kewl
- Use feature correspondence
- Solve least square problem
  - Sc of linear equations
- Warp all images to a reference one
- Use your favorite blending

Questions?
changing camera center
• Does it still work?

Nodal point
• http://www.reallyrightstuff.com/pano/index.html

Planar mosaic

Questions?

Cool applications of homographies
• Oh, Durand & Dorsey

Limitations of 2D Clone Brushing
• Distortions due to foreshortening and surface orientation
Clone brush (Photoshop)

- Click on a reference pixel (blue)
- Then start painting somewhere else
- Copy pixel color with a translation

Perspective clone brush

Oh, Durand, Dorsey, unpublished

- Correct for perspective
- And other tricks

Questions?

Rotational Mosaics

- Can we say something more about rotational mosaics?
- i.e. can we further constrain our H?

3D $\rightarrow$ 2D Perspective Projection

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} f & 0 & u_c \\ 0 & f & v_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

$$K$$
3D Rotation Model

- Projection equations
  1. Project from image to 3D ray
    \((x_0, y_0, z_0) = (u_0-u_c, v_0-v_c, f)\)
  2. Rotate the ray by camera motion
    \((x_0, y_0, z_0) = R_{01} (x_0, y_0, z_0)\)
  3. Project back into new (source) image
    \((u_1, v_1) = \left(\frac{fx_1}{z_1} + u_c, \frac{fy_1}{z_1} + v_c\right)\)
  Therefore:
    \(H = K_0 R_{01} K_1^{-1}\)
- Our homography has only 3, 4 or 5 DOF, depending if focal length is known, same, or different.
  - This makes image registration much better behaved

Pairwise alignment

- Procrustes Algorithm [Golub & VanLoan]
- Given two sets of matching points, compute \(R\)
  - with 3D rays
    \(p_i' = R p_i\)
  - \(A = \sum_i p_i p_i^T = \sum_i p_i p_i^T R R^T = U S V^T = (U S U^T) R^T\)
  - \(V^T = U^T R^T\)
  - \(R = V U^T\)

Rotation about vertical axis

- What if our camera rotates on a tripod?
- What’s the structure of \(H\)?

Do we have to project onto a plane?

- Mosaic PP

Full Panoramas

- What if you want a 360° field of view?

Cylindrical projection

- Map 3D point \((X,Y,Z)\) onto cylinder
  \((\hat{x}, \hat{y}, \hat{z}) = \frac{1}{\sqrt{X^2 + Y^2}} (X, Y, Z)\)
- Convert to cylindrical coordinates
  \((\sin \theta, h, \cos \theta) = (\hat{x}, \hat{y}, \hat{z})\)
- Convert to cylindrical image coordinates
  \((\tilde{x}, \tilde{y}) = (h \theta, h) + (x_0, y_0)\)
Cylindrical Projection

Inverse Cylindrical projection

\[ \theta = \frac{x_{cyl} - x_c}{f} \]
\[ h = \frac{y_{cyl} - y_c}{f} \]
\[ \hat{x} = \sin \theta \]
\[ \hat{y} = h \]
\[ \hat{z} = \cos \theta \]
\[ x = f\hat{x}/\hat{z} + x_c \]
\[ y = f\hat{y}/\hat{z} + y_c \]

Cylindrical panoramas

• Steps
  – Reproject each image onto a cylinder
  – Blend
  – Output the resulting mosaic
• What are the assumptions here?

Cylindrical image stitching

• What if you don’t know the camera rotation?
  – Solve for the camera rotation
    • Note that a rotation of the camera is a translation of the cylinder!

Assembling the panorama

• Stitch pairs together, blend, then crop

Problem: Drift

• Vertical Error accumulation
  – small (vertical) errors accumulate over time
  – apply correction so that sum = 0 (for 360° pan.)
• Horizontal Error accumulation
  – can reuse first/last image to find the right panorama radius
Full-view (360°) panoramas

Spherical projection
- Map 3D point \((X,Y,Z)\) onto sphere
- Convert to spherical coordinates
  \((\sin \theta \cos \varphi, \sin \varphi, \cos \theta \cos \varphi) = (\hat{x}, \hat{y}, \hat{z})\)
- Convert to spherical image coordinates
  \((\hat{x}, \hat{y}) = (f\theta, f\varphi) + (\hat{x}_c, \hat{y}_c)\)

Spherical Projection

Inverse Spherical projection
\[
\begin{align*}
\theta &= \frac{(x_{sph} - x_c)}{f} \\
\varphi &= \frac{(y_{sph} - y_c)}{f} \\
\hat{x} &= \sin \theta \cos \varphi \\
\hat{y} &= \sin \varphi \\
\hat{z} &= \cos \theta \cos \varphi \\
x &= f\hat{x}/\hat{z} + x_c \\
y &= f\hat{y}/\hat{z} + y_c
\end{align*}
\]

3D rotation
- Rotate image before placing on unrolled sphere
\[
\begin{align*}
\theta &= \frac{(x_{sph} - x_c)}{f} \\
\varphi &= \frac{(y_{sph} - y_c)}{f} \\
\hat{x} &= \sin \theta \cos \varphi \\
\hat{y} &= \sin \varphi \\
\hat{z} &= \cos \theta \cos \varphi \\
x &= f\hat{x}/\hat{z} + x_c \\
y &= f\hat{y}/\hat{z} + y_c
\end{align*}
\]

Full-view Panorama

- Convert to spherical coordinates
- Convert to spherical image coordinates
- Map 3D point \((X,Y,Z)\) onto sphere
- Rotate image before placing on unrolled sphere
- Convert to spherical coordinates
- Convert to spherical image coordinates
- Map 3D point \((X,Y,Z)\) onto sphere
- Rotate image before placing on unrolled sphere
**Polar Projection**

- Extreme “bending” in ultra-wide fields of view

\[ r^2 = x^2 + y^2 \]
\[ (\cos \theta \sin \phi, \sin \theta \sin \phi, \cos \phi) = s (x, y, z) \]

\[ x' = \frac{s \cos \theta}{s \tan^{-1} \frac{y}{r}} \]
\[ y' = \frac{s \sin \theta}{s \tan^{-1} \frac{y}{r}} \]

**Other projections are possible**

- You can stitch on the plane and then warp the resulting panorama
  - What’s the limitation here?
- Or, you can use these as stitching surfaces
  - But there is a catch…

**Cylindrical reprojection**

- Focal length is (highly!) camera dependant
  - Can get a rough estimate by measuring FOV:
  - Can use the EXIF data tag (might not give the right thing)
  - Can use several images together and try to find f that would make them match
  - Can use a known 3D object and its projection to solve for f
  - Etc.
- There are other camera parameters too:
  - Optical center, non-square pixels, lens distortion, etc.

**Distortion**

- Radial distortion of the image
  - Caused by imperfect lenses
  - Deviations are most noticeable for rays that pass through the edge of the lens

**Radial distortion**

- Correct for “bending” in wide field of view lenses

\[ \tilde{r}^2 = \tilde{x}^2 + \tilde{y}^2 \]
\[ \tilde{x}' = \tilde{x} / (1 + \kappa_1 \tilde{r}^2 + \kappa_2 \tilde{r}^4) \]
\[ \tilde{y}' = \tilde{y} / (1 + \kappa_1 \tilde{r}^2 + \kappa_2 \tilde{r}^4) \]
\[ x = f \tilde{x}' / \tilde{z} + x_c \]
\[ y = f \tilde{y}' / \tilde{z} + y_c \]

Use this instead of normal projection
Blending the mosaic

An example of image compositing: the art (and sometime science) of combining images together...

Multi-band Blending

- Burt & Adelson 1983
  - Blend frequency bands over range $\propto \lambda$

Poisson blending

Questions?

Traditional panoramas
19th century panorama

Chinese scroll

Questions?

Magic: ghost removal

**Magic: automatic panos**


**Extensions**

- Video
- Additional objects
- Mok’s panomorph
  - http://www.robots.ox.ac.uk/~vgg/publications/papers/schaffali

**Software**

- http://photocreations.ca/collage/circle.jpg
- http://webuser.fh-furtwangen.de/%7Edersch/
- http://www.ptgui.com/
- http://hugin.sourceforge.net/
- http://epaperpress.com/ptlens/
  - http://www.fdrtools.com/front_e.php

**Refs**

- http://graphics.cs.cmu.edu/courses/15-463/2004_fall/www/Papers/MSR-
  - http://portal.acm.org/citation.cfm?id=218395&dl=ACM&coll=portal
  - http://citeseer.ist.psu.edu/mann94virtual.html
  - http://research.microsoft.com/vision/visionbasedmodeling/publications/Bau

**View morphing**

subject

common view plane

viewpoint 1 viewpoint 2