Image Mosaics and Panoramas

Lots of slides stolen from Bill Freeman, who stole them from Alyosha Efros, who stole them from Steve Seitz and Rick Szeliski. Some slides are from Vaibhav Vaish, Yung-Yu Chuang, Darya Frolova, Denis Simakov.

Links:
http://groups.csail.mit.edu/graphics/classes/CompPhoto06/
http://robotics.stanford.edu/a33305/

Building a Panorama


Why Panoramas?

• Cartography: stitching aerial images to make maps
- Virtual wide-angle camera
  - Consumer camera: 50° x 35°

- Virtual wide-angle camera
  - Consumer camera: 50° x 35°
  - Human Vision: 176° x 135°

- Virtual wide-angle camera
  - Consumer camera: 50° x 35°
  - Human Vision: 176° x 135°
  - Panoramic mosaics: up to 360° x 180°

Mosaics: stitching images together
• Video Coding:
  – convert masked images into a background sprite for content-based coding

• Virtual reality: a sense of being there

How to make a panorama?
1. Take a sequence of images from the same position (Rotate the camera about its optical center)
2. Compute transformation between second image and first
3. Transform the second image to overlap with the first
4. Blend the two together to create a mosaic
5. If there are more images, repeat

• Don’t we need the 3D geometry of the scene?

The Camera
• http://www.reallyrightstuff.com/pano/index.html
A pencil of rays contains all views

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

Increasing the Field of View

Camera Center

Example

Aligning Images

What’s the relation between corresponding points? e.g. translation, Euclidean, affine, projective
Trial - aligning images by translation

Translations are not enough to align the images

Perspective warps (Homographies)

$P_1 = KP$

$p_1 = K P$

$p_2 = K R P$

Camera Center (0,0,0)
**Perspective warps (Homographies)**

- Projective – mapping between any two projection planes (PPs) with the same center of projection
  - rectangle should map to arbitrary quadrilateral
  - parallelism is not preserved
  - straight lines are preserved

- called Homography

\[
\begin{bmatrix}
wx' \\
w' \\
w
\end{bmatrix}
= \begin{bmatrix}
x' \\
y' \\
y
\end{bmatrix}^T \begin{bmatrix}
0 & -y' & y \\
x' & 0 & -x \\
-1 & 0 & 0
\end{bmatrix} \begin{bmatrix}
x \\
y \\
1
\end{bmatrix}
\]

**Homography**

- Mapping between any two arbitrary PPs if the object is planar.

\[
\begin{bmatrix}
wx' \\
w' \\
w
\end{bmatrix}
= \begin{bmatrix}
x' \\
y' \\
y
\end{bmatrix}^T \begin{bmatrix}
0 & -y' & y \\
x' & 0 & -x \\
-1 & 0 & 0
\end{bmatrix} \begin{bmatrix}
x \\
y \\
1
\end{bmatrix}
\]

**Epipolar Geometry**

- The epipolar geometry introduces constraints between any two arbitrary PPs:

\[
0 = \begin{bmatrix}
x' \\
y'
\end{bmatrix} \begin{bmatrix}
0 & -y' & y \\
x' & 0 & -x \\
-1 & 0 & 0
\end{bmatrix} \begin{bmatrix}
x \\
y \\
1
\end{bmatrix}
\]

\*Fundamental Matrix*
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!

Finding the Homography by Matching Features

$$p_2 \approx KRK^{-1}p_1$$

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

$$x' = \frac{ax + by + c}{gx + hy + 1}$$

$$y' = \frac{dx + ey + f}{gx + hy + 1}$$

Two linear equations per matching feature

Homographies by Feature Matching
Solving for homographies

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

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

- Can set scale factor \( i = 1 \). So, there are 8 unknowns.
- Set up a system of linear equations:
  \[ Ah = b \]
- 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:
  \[ \min \| Ah - b \| \]

Feature Matching Panorama

- Reference image is marked red.
- For each blue image the homography was calculated using matched features.

Full Panoramas

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

Cylindrical projection

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

Assembling the panorama

• Stitch pairs together, blend, then crop

End-to-end alignment and crop

Inverse Cylindrical projection

\[
\begin{align*}
\theta &= \frac{(x_{cyl} - x_c)}{f} \\
h &= \frac{(y_{cyl} - y_c)}{f} \\
\hat{x} &= \sin \theta \\
\hat{y} &= h \\
\hat{z} &= \cos \theta \\
x &= f\frac{\hat{x}}{\hat{z}} + x_c \\
y &= f\frac{\hat{y}}{\hat{z}} + y_c
\end{align*}
\]

Matching with Features

- Detect feature points in both images
- Find corresponding pairs
- Use these pairs to align images

Problem 1: Detect the same point in both images (invariant to possible deformations)

Problem 2: For each point correctly recognize the corresponding one

We need a reliable and distinctive feature descriptor
SIFT – scale invariant feature transform

- Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters.


SIFT = Scale Invariant Feature Transform

Descriptor overview:
- Determine scale (by maximizing DoG in scale and in space), local orientation as the dominant gradient direction. Use this scale and orientation to make all further computations invariant to scale and rotation.
- Compute gradient orientation histograms of several small windows (128 values for each point)
- Normalize the descriptor to make it invariant to intensity change

Select canonical orientation

- Create histogram of local gradient directions computed at selected scale
- Assign canonical orientation at peak of smoothed histogram
- Each key specifies stable 2D coordinates (x, y, scale, orientation)

SIFT descriptor vector

- Thresholded image gradients are sampled over 16x16 array of locations in scale space
- Create array of orientation histograms
- 8 orientations x 4x4 histogram array = 128 dimensions
RANSAC

- RANSAC = Random Sample Consensus
- an algorithm for robust fitting of models in the presence of many data outliers
- Compare to robust statistics

Given \( N \) data points \( x_i \), assume that majority of them are generated from a model with parameters \( \Theta \), try to recover \( \Theta \).

RANSAC algorithm

Run \( k \) times:

1. draw \( n \) samples randomly
2. fit parameters \( \Theta \) with these \( n \) samples
3. for each of other \( N-n \) points, calculate its distance to the fitted model, count the number of inlier points, \( c \)

Output \( \Theta \) with the largest \( c \)

How to determine \( k \)

\( p \): probability of real inliers
\( P \): probability of success after \( k \) trials

\[ P = 1 - (1 - p^n)^k \]

For \( P=0.99 \)

\[ k = \frac{\log(1-P)}{\log(1-p^n)} \]

<table>
<thead>
<tr>
<th>( n )</th>
<th>( p )</th>
<th>( k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.5</td>
<td>35</td>
</tr>
<tr>
<td>6</td>
<td>0.6</td>
<td>97</td>
</tr>
<tr>
<td>6</td>
<td>0.5</td>
<td>293</td>
</tr>
</tbody>
</table>
Example: line fitting

Model fitting

Measure distances
Count inliers

Another trial

The best model

Finding SIFT features
Rejecting outliers using RANSAC

Finding the Homography and stitch images

Image Stitching
- Naïve stitching might seem unnatural due to parallax, lens distortion, scene motion, exposure difference
- Solution: Multiband blending

Multi-band Blending
- Burt & Adelson 1983
Image Blending

Blending

The effect of blending window
Multibands
Lowpass Images

Multiband Pyramid

Gaussian Pyramid

Laplacian Pyramid

Bandpass Images

Gaussian pyramid construction

What does blurring take away?

- Repeat
  - Filter
  - Subsample
- Until minimum resolution reached
  - can specify desired number of levels (e.g., 3-level pyramid)
- The whole pyramid is only 4/3 the size of the original image!
What does blurring take away?

smoothed (5x5 Gaussian)

High-Pass filter

smoothed – original

Multiband Blending

Pyramid Blending
Pyramid Blending

Blending Regions

The colored version

© prof. dmartin
**Blending in Gradient Domain**

- In Pyramid Blending, we decomposed our image into 2nd derivatives (Laplacian) and a low-res image
- Let us now look at 1st derivatives (gradients):
  - No need for low-res image
    - captures everything (up to a constant)
  - Idea:
    - Differentiate
    - Blend
    - Reintegrate

**Gradient Domain Blending – 2D**

*Perez et. al. 2003*

- Given vector field \( v \) (pasted gradient), find the value

\[
\min_f \int_\Omega |\nabla f - v|^2 \quad \text{with} \quad f|_{\partial \Omega} = f^+|_{\partial \Omega}
\]

**Example**

*From Perez et al. 2003*
Comparisons: Levin et al, 2004

Pyramid blending
Feathering

Pyramid blending on the gradients
GIST1

Don’t blend, CUT!

Moving objects become ghosts

• So far we only tried to blend between two images. What about finding an optimal seam?

Davis, 1998

• Segment the mosaic
  – Single source image per segment
  – Avoid artifacts along boundaries
    • Dijkstra’s algorithm

Efros & Freeman, 2001

Random placement of blocks
Neighboring blocks constrained by overlap
Minimal error boundary cut

Input texture

block
Minimal error boundary

overlapping blocks → vertical boundary

\[ \begin{array}{c}
\text{overlap error} \\
\hline
\text{min. error boundary}
\end{array} \]

Video Mosaics

- So far we make panoramas from still images.
- Is it possible to make panorama from video?
- Dynamosaics (A. Rav-Acha, Y. Pritch, D. Liscinski, S. Peleg, CVPR 2005)
- Slides: courtesy of A. Rav Acha

DynaMosaics

Dynamosaics: A. Rav-Acha, Y. Pritch, D. Liscinski, S. Peleg, CVPR 2005

Slides: courtesy of A. Rav Acha

Dynamic Mosaics

- Cancel Camera Motion
- Preserve Scene Motion
Constructing an **aligned** Space-Time Volume

Motion Parallax: Align by straightening lines in SP Volume

**Aligned ST Volume: View from Top**

Stationary Camera  Panning Camera

**Evolving Time Front**

Mapping each TF to a new frame

**Temporal Video Editing**

Rigging a Swimming Competition
Reversing Scan Direction

Iguazu Falls