Image Mosaics and Panoramas

Lots of slides from Bill Freeman, Alyosha Efros, Steve Seitz and Rick Szeliski. Some slides are from Vaibhav Vaish, Yung-Yu Chuang, Darya Frolova, Denis Simakov, Alex Rav Acha, and from Shmuel Peleg.

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°

Video Compression
  – convert masked images into a background sprite for content-based coding

Video Summarization
• Virtual reality: a sense of being there

http://www.panoramas.dk/

• Video Panorama

The First Panoramas …

Paris, c. 1845-50, photographer unknown

How to create a panorama?

1. Take a sequence of images
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?
Panorama - from Rotated Camera

If the sequence of images are taken from the same position (Rotate the camera about its optical center), then the 3D geometry of the scene is NOT needed.

The Camera


First Panoramic Camera

Al-Vista, 1899 ($20)

A pencil of rays contains all views as long as they have 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

Translation
Affine
Perspective

2 parameters
6 parameters
8 parameters
Trial - aligning images by translation

Translations are not enough to align the images

Pinhole camera model - Perspective Projection

Pinhole point offset

\[(X, Y, Z)^T \mapsto (fX/Z + o_x, fY/Z + o_y)^T\]

principal point

\[
\begin{bmatrix}
X \\
Y \\
Z \\
1
\end{bmatrix}
\mapsto
\begin{bmatrix}
(X + Z o_x) \\
fY + Z o_x
\end{bmatrix}
= 
\begin{bmatrix}
f & o_x & 0 \\
of & o_y & 0 \\
1 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}
\]

Linear projection in homogeneous coordinates!

Perspective warps (Homographies)

\[p_t = K P\]
**Perspective warps (Homographies)**

\[ p_1 \approx K \, P \]
\[ p_2 \approx K \, R \, P \]

Camera Center \((0,0,0)\)

\[ P \ (X, Y, Z) \]

\[ p_1 \]

\[ p_2 \ (x', y') \]

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

\[
\begin{bmatrix}
w'x' \\
w'y' \\
w \\
1
\end{bmatrix} = \begin{bmatrix}
x' \\
y' \\
1
\end{bmatrix} \begin{bmatrix}
\ast & \ast & \ast \\
\ast & \ast & \ast \\
\ast & \ast & \ast
\end{bmatrix} \begin{bmatrix}
w \\
x \\
y \\
p
\end{bmatrix}
\]

**Calculate Homography**

Given corresponding points in 2 images:

\[ p_i \longrightarrow p'_i \]

Find \( H \) satisfying

\[ p'_i = H \, p_i \]

3x3 Homography

Requires 4 matching points.
(8 unknowns since \( H \) up to Scale)
Changing Camera Center

What about mapping between 2 general images (NOT from rotating camera)?

\[ p_1 \approx K P \]
\[ p_2 \approx K (R P+T) \]

\[ p_2 \xrightarrow{\text{NOT Homography}} p_1 \]

Homography

**However:** Mapping between any two arbitrary PPs IS possible if the object is planar.

\[
\begin{bmatrix}
wx' \\
w'y' \\
w
\end{bmatrix}
= \begin{bmatrix}
x \\
y \\
p
\end{bmatrix}
H
\]

3x3 Homography

\[ p_1 = K_1 P \]
\[ p_2 = K_2 P \]

\[ p_1 = K_1 K_2^{-1} p_2 \]
For non planar scene, there is no Homography between two general views, however there is a geometric constraint: The Epipolar Constraint between any two arbitrary PPs.

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

Fundamental Matrix

Epipolar Geometry

Image warping with Homographies

http://users.skynet.be/J.Beever/pave.htm
To unwarp (rectify) an image

- Find the homography \( H \) given a set of \( p \) and \( p' \) pairs
- How many pairs are needed?
- What if more points are given?

**Finding the Homography using Feature Points**

\[
p_2 = K R K^{-1} p_1
\]

Features = e.g. SIFT, Harris etc

**Solving for homographies**

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

- 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:

\[
\min \| Ah - b \|^2
\]
Feature Matching Panorama

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

Homographic Warp - Example

Homographic Warp - Revisited

Homographic Warp - Example
Homographic Warp - Example

Ken Chu, 2004

Full Panoramas

What if you want a 360° field of view?

Cylindrical projection

Map 3D point (X,Y,Z) onto cylinder

\[ (\hat{x}, \hat{y}, \hat{z}) = \frac{1}{\sqrt{X^2 + Z^2}} (X, Y, Z) \]

Convert to cylindrical coordinates

\[ (\sin \theta, \cos \theta) = (\hat{x}, \hat{z}) \]

Convert to cylindrical image coordinates

\[ (\tilde{x}, \tilde{y}) = (\theta t, s h) + (\tilde{z}_c, \tilde{y}_c) \]

Cylindrical Projection

unwrapped cylinder
cylindrical image
In the cylindrical coordinates rotational warping is translated into translational warping.

Assembling the panorama

- Stitch pairs together, blend, then crop

End-to-end alignment and crop

360° Panorama

NOTE: straight lines become conics.
Inverse Cylindrical Projection

\[ \theta = \frac{(\bar{x} - \bar{x}_c)}{s} \]
\[ h = \frac{(\bar{y} - \bar{y}_c)}{s} \]
\[ \hat{x} = \sin \theta \]
\[ \hat{z} = \cos \theta \]
\[ x = \hat{x} / \hat{z} \]
\[ y = h \]

Quicktime VR

Panoramas:
- http://www.panoramas.dk

Demo TimesSquare-Med.mov

Multi-Perspective Panorama

Suppose we do not want a 360° view. We want a view of a "whole street" or even more complicated - a route along city streets.

Using homography to align all images, (i.e. creating a single wide angle view) farthest objects will be distorted to very large regions.

Thus need to simulate a different camera model

Pushbroom Camera Model
The generalized camera model:

- X direction - parallel projection
- Y direction - perspective projection

Pushbroom Camera Model

Implementation through cuts of 3D video array.

- Take images while moving a usual camera
- Stack them into 3D array
- Take a cut along the “time” dimension

Pushbroom Panorama - Example
Gooseneck State Park, Utah

Space-Time Volume Slicing - Video Mosaic
Align the frames of the Space-Time Volume
Stationary Camera
Panning Camera
From: Rav-Acha and Peleg

Space-Time Volume Slicing - Panorama
From: Rav-Acha and Peleg

Space-Time Volume Slicing - Dynamosaicing
Forward Panning camera
From: Rav-Acha and Peleg
Space-Time Volume Slicing - Dynamosaicing

Original - Panning

Iguazu Falls

From: Rav-Acha and Peleg

Space-Time Volume Slicing - Dynamosaicing

Backward - Panning

From: Rav-Acha and Peleg

Space-Time Volume Slicing - Dynamosaicing

Backward Panning camera

From: Rav-Acha and Peleg

Space-Time Volume Slicing - Dynamosaicing

Forward motion

From: Rav-Acha and Peleg

Space-Time Volume Slicing - Dynamosaicing

backward panning forward motion

From: Rav-Acha and Peleg
Space-Time Volume Slicing - Dynamosaicing
Backward Panning - forward motion

From: Rav-Acha and Peleg

Space-Time Volume Slicing - Video Panorama

From: Rav-Acha and Peleg

Another example

From: Rav-Acha and Peleg
From: Rav-Acha and Peleg

Space-Time Volume Slicing - Evolving Time Fronts

Still Camera
Moving objects

$t$

$k$

$k+1$

Mapping each TF to a new frame

$x$

$u$

Rigging a Swimming Competition

From: Rav-Acha and Peleg
Next topic: Stitching and Composition

Stitching = alignment + blending

- geometrical registration
- photometric registration

Panorama Blending

THE END