Olivier Gondry
- Director of music video and commercial
- Special effect specialist (Morphing, rotoscoping)
- Today at 5:40pm in 32-141

Why Mosaic?
- Are you getting the whole picture?
  - Compact Camera FOV = 50 x 35°
  - Human FOV = 200 x 135°
  - Panoramic Mosaic = 360 x 180°

Mosaics: stitching images together.
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

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

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

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
- 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

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

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

1D homography
- Reproject to different line

Same in 2D
- Reprojection = homography
- \( 3 \times 3 \) matrix

\[
\begin{bmatrix}
w'x' \\
w'y' \\
w
\end{bmatrix} = \begin{bmatrix}
* & * & * \\
* & * & * \\
* & * & *
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
p
\end{bmatrix}
\]
Image warping with homographies

Digression: perspective correction

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 \((X_1,X_1'), (X_2,X_2'), (X_3,X_3')\), etc. (e.g. person’s height vs. weight)
• We want a nice compact formula (line) to predict \( X' \)’s from \( X \): \( X_a + b = X' \)
• We want to find \( a \) and \( b \)
• How many \((X,X')\) pairs do we need?
• What if the data is noisy?

\[
\begin{bmatrix}
X_1 & 1 & a \\
X_2 & 1 & b \\
X_3 & 1 & c \\
\vdots & \vdots & \vdots \\
X_n & 1 & d
\end{bmatrix}
\begin{bmatrix}
a \\
b \\
c \\
\vdots \\
d
\end{bmatrix}
= 
\begin{bmatrix}
X_1' \\
X_2' \\
X_3' \\
\vdots \\
X_n'
\end{bmatrix}
\]

overconstrained

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:
  \( A h = b \)
• where vector of unknowns \( h = [a,b,c,d,e,f,g,h]^T \)
• Note: we do not know \( w \) but we can compute it from \( x \) & \( y \):
  \( w = g x + h y + 1 \)
• The equations are linear in the unknown

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
  - Se of linear equations
- Warp all images to a reference one
- Use your favorite blending

changing camera center

- Does it still work?

Nodal point


Planar mosaic

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

Rotational Mosaics

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

3D → 2D Perspective Projection

3D Rotation Model

• Projection equations
  1. Project from image to 3D ray
  \((x_Uy_Uz_U) = (u_xu_yu_zf)\)
  2. Rotate the ray by camera motion
  \((x_{U'}y_{U'}z_{U'}) = R_{U_0} (x_{U_0}y_{U_0}z_{U_0})\)
  3. Project back into new (source) image
  \((u_{U'}v_{U'}) = (x_{U'}y_{U'}z_{U'}f + v)\)
  Therefore:
  \(H = K_uR_uK_u^{-1}\)
  • 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 $R$
• $p_i' = R p_i$ with 3D rays
• $p_i = N(x_i, y_i, z_i) = N(u_i - u_c, v_i - v_c, f)$
• $A = \sum_i p_i p_i' = \sum_i p_i p_i' R^T = U S V^T = (U S U^T) R^T$
• $V^T = U^T R^T$
• $R = V U^T$

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?

Full Panoramas

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

Cylindrical projection

• Map 3D point $(X, Y, Z)$ onto cylinder
  $(\xi, \eta, \zeta) = \frac{(X, Y, Z)}{\sqrt{X^2 + Y^2}}$
• Convert to cylindrical coordinates
  $(\sin \theta, \cos \theta) = (\xi, \eta)$
• Convert to cylindrical image coordinates
  $(\theta, \phi) = (f \phi, f h) + (\theta_c, \phi_c)$

Cylindrical Projection
Inverse Cylindrical projection

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

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 rotations
  - 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
  \( (\tilde{x}, \tilde{y}, \tilde{z}) = \frac{(x, y, z)}{\sqrt{x^2 + y^2 + z^2}} \)
- Convert to spherical coordinates
  \( (\theta, \phi, \tilde{z}) = (\tilde{y}, \tilde{z}, \tilde{x}) \)
- Convert to spherical image coordinates
  \( (\tilde{x}', \tilde{y}') = (f \theta, f \phi) + (x_c, y_c) \)

Inverse Spherical projection

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

3D rotation

- Rotate image before placing on unrolled sphere

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

Full-view Panorama

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

\[
\begin{align*}
\tilde{r}^2 &= \tilde{x}^2 + \tilde{y}^2 \\
\cos \theta \sin \phi, \sin \theta \sin \phi, \cos \phi &= s (x, y, z) \\
\end{align*}
\]

Polar Projection

\[
\begin{align*}
x' &= s_0 \cos \theta = \frac{x}{r} \tan^{-1} \frac{y}{z} \\
y' &= s_0 \sin \theta = \frac{y}{r} \tan^{-1} \frac{z}{x} \\
\end{align*}
\]
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…

What’s your focal length, buddy?

• 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.

Cylindrical reprojection

Focal length – the dirty secret...

Image 384x300 f = 180 (pixels) f = 280 f = 380

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

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

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 $\propto \lambda$.

Traditional panoramas

19th century panorama

Chinese scroll
Magic: automatic panos


Magic: ghost removal

• See also HDR lecture

M. Uyttendaele, A. Eden, and R. Szeliski.
Eliminating ghosting and exposure artifacts in image mosaics.

Extensions

• Video
• Additional objects
• Mok’s panomorph
  http://www.sarnoff.com/products_services/vision/tech_papers/kumar95.pdf
  http://www.cs.huji.ac.il/~peleg/papers/pami00-manifold.pdf
  http://www.cs.huji.ac.il/~peleg/papers/cvpr00-rectified.pdf
  http://citeseer.ist.psu.edu/cache/papers/cs/20590/http:zSzzSzw
  www.sarnoff.comzSzcareer_movezSztech_paperszSzpdfzSzvisrep
  p95.pdf/kumar95representation.pdf
  http://www.robots.ox.ac.uk/~vgg/publications/papers/schaffaltz
  zky97.pdf

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/
  Tutorials
  http://www.fdrtools.com/front_c.php

Refs

• http://graphics.cs.cmu.edu/courses/15-463/2004_fall/www/Papers/MSR-
• http://www.cs.washington.edu/education/courses/cse576/05w/readings/xr
  leIsShum97.pdf
• http://portal.acm.org/citation.cfm?id=218395&dl=ACM&coll=portal
• http://research.microsoft.com/~brown/papers/cvpr05.pdf
• http://citeseer.ist.psu.edu/mann94virtual.html
• http://grafl.cs.washington.edu/projects/panovidex/
• http://research.microsoft.com/vision/ivisionbasedmodeling/publications/Bau
  disch-OZCHI05.pdf
• http://www.vision.caltech.edu/llb/Demos/SquarePanorama.html
• http://graphics.stanford.edu/papers/multi-cross-alts/