Panoramas

Calvin and Hobbes

by Watterson

AHHHH...?

UH-OH, SOMETHING IS SERIOUSLY WRONG HERE.

THE LAWS OF PERSPECTIVE HAVE BEEN REPEALED!

OBJECTS NO LONGER DIMINISH IN SIZE WITH DISTANCE.

LINES DO NOT CONVERGE TOWARD ANY POINT ON THE HORIZON!

ALL SPATIAL RELATIONSHIPS ARE LOST! IT'S IMPOSSIBLE TO JUDGE WHERE ANYTHING IS! OH NO!

CALVIN, QUIET RUNNING AROUND AND CRASHING INTO THINGS, OR I'LL SELL YOU TO THE MONKEY HOUSE!

...AND NOW SHE'S LOST PERSPECTIVE.
6.815 Digital and Computational Photography
6.865 Advanced Computational Photography

Panoramas

Frédo 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°
Why Mosaic?

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

• Are you getting the whole picture?
  – Compact Camera FOV = 50 x 35°
  – Human FOV = 200 x 135°
  – Panoramic Mosaic = 360 x 180°
Traditional panoramas
19th century panorama
Mosaics: stitching images together

virtual wide-angle camera
Big Idea: Multiple-Exposure Photo

• Compensate for limitations of the camera by taking multiples images

• Examples
  – HDR imaging
  – Some color imaging processes
  – Panorama stitching
  – Focus stack to increase depth of field
  – ....
Today

- We assume we know feature correspondences between images
- We seek to align the images into a virtual wider-angle view
- Later lecture: automatic correspondence
Question?
How to do it?

• Basic Procedure
  – Take a sequence of images **from the same position**
    • Rotate the camera about its optical center (entrance pupil)
  – Compute transformation between second image and first
  – Transform (warp) 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**!
Entrance pupil

- Often wrongly called nodal point
- When camera is rotated around nodal point, there is no parallax
  - That is, if two 3D points are superimposed for one orientation, they remain superimposed after rotation
- Finding the entrance pupil is painful
Recap

• When we only rotate the camera (around nodal point) depth does not matter

• It only performs a 2D warp
  – one-to-one mapping of the 2D plane
  – plus of course reveals stuff that was outside the field of view

• Now we just need to figure out this mapping
Other interpretation

- Depth does not matter
- We can pretend that each pixel is at a convenient depth

- Three convenient depth distributions:
  - spherical
  - planar
  - cylindrical

- We focus on planar
  - it makes life more linear

- Still useful for spherical panos
Question?
Aligning images

We have established that pairs of images from the same viewpoint can be aligned through a simple 2D spatial transformation (warp).

What kind of transformation?
Aligning images: translation

Translations are not enough to align the images
Back to Image Warping

Which t-form is the right one for warping Projection Plane PP1 into Projection Plane PP2?

e.g. translation, Euclidean, affine, projective

Translation: 2 unknowns
Affine: 6 unknowns
Perspective: 8 unknowns
Recap

- We are looking for the 2D mapping that corresponds to a 3D rotation of the camera
- We need to understand perspective projection
Simple Perspective Projection

- Project all points to the $z = 1$ plane, viewpoint at the origin:
  - $x' = x/z$
  - $y' = y/z$
Simple Perspective Projection

- Project all points to the $z = 1$ plane, viewpoint at the origin:
  - $x' = x/z$
  - $y' = y/z$

- Can we represent this with a matrix?
  - not directly (division)
  - but we can cheat...
  - add a third coordinate to the result
    - interpret as: we always divide by 3rd coordinate
    - see next slide...
Homogeneous coordinate

- represent 2D points with 3 numbers
- \((x, y, w)\) represents \((x/w, y/w)\)
- Allows us to represent projective transforms
- Nice thing: projecting onto plane \(z=1\) is just the strict interpretation of homogeneous coordinates

- Yes, you can view this as a notation trick
  – But math is all about smart notations
- Homogenous coordinates are central in computer graphics and machine vision
3D rotation

- We observe point $x, y, z$ with camera PP1
- Now we rotate the 3D camera to PP2
- What is the new 2D projection?

\[(x', y', 1)\]
3D rotation

- Now we rotate the 3D camera
- What is the new projection?
- Rotating the camera is the same as rotating the world in the opposite direction
- To project a \((x, y, z)\) wrt rotated camera:
  - Apply rotation \(R\) to \((x, y, z)\)
  - Apply projection division
Recap

• canonical projection = division by z
• Homogeneous coordinates are a notation trick to encapsulate this
• For other direction, just apply 3D rotation first

• But... this applies only when we know z
  – And for panorama stitching, we don’t
  – What are we going to do? Are we in big trouble? Should we give up? By a 3D scanner? Cancel assignment 7?
Camera rotation: unknown z

- Same thing but use \((x', y', 1)\) instead of \((x, y, z)\)
  - Rotate \((x', y', 1)\) by \(R\)
  - Perform division
3D rotation

- Same thing but use \((x', y', 1)\) instead of \((x, y, z)\)
  - Rotate \((x', y', 1)\) by \(R\)
  - Perform division

- Makes sense: depth does not matter, all points along a light ray project to the same point.
  - We arbitrarily (but conveniently) choose the point at depth 1
1D homogeneous coordinates

- Add one dimension to make life simpler
- \((x, w)\) represent point \(x/w\)
Other illustration: 1D homography

- Reproject to different line
1D homography

• Reproject to different line
1D homography

- Reproject to different line
- Equivalent to rotating 2D points
  ➞ reprojection is linear in homogeneous coordinates
Questions?
Wrapping it up: Homography

• Projective – mapping between any two projection planes with the same center of projection
• called Homography
• represented as 3x3 matrix in homogenous coordinates
  – corresponds to the 3D rotation

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

To apply a homography $H$
• Compute $p' = Hp$ (regular matrix multiply)
• Convert $p'$ from homogeneous to image coordinates (divide by $w$)
Wrapping it up: Homography

- rectangles map to arbitrary quadrilateral
- parallel lines aren’t parallel anymore
- but straight lines remain straight

- same as: project, rotate, reproject
Recap

- Reprojection = homography
- 3x3 matrix in homogeneous coordinate
  - (the matrix can be constrained to be a rotation)

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

homography so that image is parallel to floor

homography so that image is parallel to right wall

black area where no pixel maps to
Questions?
Digression: perspective correction

From Photography, London et al.
CONTROLLING CONVERGING LINES: THE KEY:

Standing at street level and shooting straight at a building produces too much street and too little building. Sometimes it is possible to move back far enough to show the entire building while keeping the camera level, but this adds even more foreground and usually something gets in the way.

From Photography, London et al.
CONTROLLING CONVERGING LINES: THE KEYSTONE EFFECT

Standing at street level and shooting straight at a building produces too much street and too little building. Sometimes it is possible to move back far enough to show the entire building while keeping the camera level, but this adds even more foreground and usually something gets in the way.

Tilting the whole camera up shows the entire building but distorts its shape. Since the top is farther from the camera than the bottom, it appears smaller; the vertical lines of the building seem to be coming closer together, or converging, near the top. This is named the keystone effect, after the wedge-shaped stone at the top of an arch. This convergence gives the illusion that the building is falling backward—an effect particularly noticeable when only one side of the building is visible.

From Photography, London et al.
CONTROLLING CONVERGING LINES: THE KEYSTONE EFFECT

Standing at street level and shooting straight at a building produces too much street and too little building. Sometimes it is possible to move back far enough to show the entire building while keeping the camera level, but this adds even more foreground and usually something gets in the way.

Tilting the whole camera up shows the entire building but distorts its shape. Since the top is farther from the camera than the bottom, it appears smaller; the vertical lines of the building seem to be coming closer together, or converging, near the top. This is named the keystone effect, after the wedge-shaped stone at the top of an arch. This convergence gives the illusion that the building is falling backward—an effect particularly noticeable when only one side of the building is visible.

To straighten up the converging vertical lines, keep the camera back parallel to the face of the building. To keep the face of the building in focus, make sure the lens is parallel to the camera back. One way to do this is to level the camera and then use the rising front or falling back movements or both.

Another solution is to point the camera upward toward the top of the building, then use the tilting movements—first to tilt the back to a vertical position (which squares the shape of the building), then to tilt the lens so it is parallel to the camera back (which brings the face of the building into focus). The lens and film will end up in the same positions with both methods.

From Photography, London et al.
Thursday, March 19, 2009
Tilt-shift lens

• 35mm SLR version
Photoshop version (perspective crop) + you control reflection and perspective independently
Question?

- The Ambassadors, Hans Holbein the Younger, 1533
Question?

- The Ambassadors, Hans Holbein the Younger, 1533
To unwarp (rectify) an image

- Find the homography $H$ given a set of $p$ and $p'$ pairs
- How many correspondences are needed?
- We can solve for $H$
  - Find such $H$ that transforms points $p$ into $p'$
Solving for homographies

\[ p' = Hp \]

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

- Can set scale factor \( i = 1 \). So, there are 8 unknowns.
- Set up a system of linear equations:
  - \( AX = B \)
- Where vector of unknowns \( X = [a, b, c, d, e, f, g, h]^T \)

- Note: we do not know \( w \) but we can compute it from \( x \) & \( y \)
  \( w = gx + hy + 1 \)
- The equations are linear in the unknown
Careful: two equations

• Start from $p' = Hp$
• Say that the unknowns are the coefficients of $H$, put them into a vector $h$ of 8 coordinates
• New equation $AX = B$ where $X$ encodes the coefficients of $H$
Solving for homographies

\[ p' = Hp \]

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

- Can set scale factor \( i = 1 \). So, there are 8 unknowns.
- Set up a system of linear equations:
  - \( AX = B \)
- where vector of unknowns \( X = [a,b,c,d,e,f,g,h]^T \)
- Need at least 8 eqs, but the more the better…
- Solve for \( h \). If overconstrained, solve using least-squares:
  \[ \text{argmin} \| AX - B \|^2 \]
- Can be done in Matlab using \“\" command
  - see “help lmdivide”
Questions?

- Julian Beever
- e.g. http://users.skynet.be/J.Beever/
  http://www.crystalinks.com/julian_beever.html
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 \(X\)’s:
  \[ Xa + b = X' \]
- We want to find \(a\) and \(b\)
- How many \((X, X')\) pairs do we need?
- What if the data is noisy?

\[
\begin{bmatrix}
X_1 \\
X_2 \\
X_3 \\
\vdots
\end{bmatrix}
\begin{bmatrix}
a \\
b
\end{bmatrix}
= 
\begin{bmatrix}
X'_1 \\
X'_2 \\
X'_3 \\
\vdots
\end{bmatrix}
\]

\[
\text{overconstrained}
\]

\[
\min ||Ax - B||^2
\]

\[
Ax = B
\]
least square

• $Ax=b$ where $A$ is rectangular (overconstrained)

• Solve $A^TAx = A^Tb$
Questions?
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
Questions?
changing camera center

• Does it still work?
Planar mosaic

<table>
<thead>
<tr>
<th>Image Compositing for Tele-Reality</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Introduction</td>
</tr>
<tr>
<td>2. Previous work</td>
</tr>
<tr>
<td>3. Basic imaging equations</td>
</tr>
<tr>
<td>4. Planar image compositing</td>
</tr>
<tr>
<td>5. Panoramic</td>
</tr>
<tr>
<td>6. Precise-panoramic scenes</td>
</tr>
<tr>
<td>7. Scenes with arbitrary depth</td>
</tr>
<tr>
<td>8. 3-D model recovery</td>
</tr>
<tr>
<td>9. Applications</td>
</tr>
<tr>
<td>10. Discussion &amp; Conclusions</td>
</tr>
</tbody>
</table>

Image Compositing for Tele-Reality

1. Introduction
2. Previous work
3. Basic imaging equations
4. Planar image compositing
5. Panoramic
6. Precise-panoramic scenes
7. Scenes with arbitrary depth
8. 3-D model recovery
9. Applications
10. Discussion & Conclusions
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
Figure 15: The cars and the street furniture have been removed. This example took less than 10 minutes.
Questions?
Other application: View morphing

- We want to morph between two views of the same object
- Standard morphing won’t give a realistic result

Figure 2: A Shape-Distorting Morph. Linearly interpolating two perspective views of a clock (far left and far right) causes a geometric bending effect in the in-between images. The dashed line shows the linear path of one feature during the course of the transformation. This example is indicative of the types of distortions that can arise with image morphing techniques.
View morphing

- Seitz & Dyer
- Interpolation consistent with 3D view interpolation

Figure 1: View morphing between two images of an object taken from two different viewpoints produces the illusion of physically moving a virtual camera.
Main trick - see Panorama lecture

- Prewarp with a homography to "pre-align" images
- So that the two views are parallel
  - Because linear interpolation works when views are parallel

Figure 4: View Morphing in Three Steps. (1) Original images $I_0$ and $I_1$ are prewarped to form parallel views $\hat{I}_0$ and $\hat{I}_1$. (2) $\hat{I}_s$ is produced by morphing (interpolating) the prewarped images. (3) $\hat{I}_s$ is postwarped to form $I_s$. 

Thursday, March 19, 2009
Figure 6: View Morphing Procedure: A set of features (yellow lines) is selected in original images $I_0$ and $I_1$. Using these features, the images are automatically prewarped to produce $\hat{I}_0$ and $\hat{I}_1$. The prewarped images are morphed to create a sequence of in-between images, the middle of which, $\hat{I}_{0.5}$, is shown at top-center. $\hat{I}_{0.5}$ is interactively postwarped by selecting a quadrilateral region (marked red) and specifying its desired configuration, $Q_{0.5}$, in $I_{0.5}$. The postwarps for other in-between images are determined by interpolating the quadrilaterals (bottom).
Figure 10: Image Morphing Versus View Morphing. Top: image morph between two views of a helicopter toy causes the in-between images to contract and bend. Bottom: view morph between the same two views results in a physically consistent morph. In this example the image morph also results in an extraneous hole between the blade and the stick. Holes can appear in view morphs as well.
Figure 7: Facial View Morphs. Top: morph between two views of the same person. Bottom: morph between views of two different people. In each case, view morphing captures the change in facial pose between original images $I_0$ and $I_1$, conveying a natural 3D rotation.
Figure 9: Mona Lisa View Morph. Morphed view (center) is halfway between original image (left) and its reflection (right).
Do we have to project onto a plane?
Full Panoramas

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

mosaic Projection Cylinder
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

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

- Convert to cylindrical image coordinates

\[(\bar{x}, \bar{y}) = (f\theta, fh) + (\bar{x}_c, \bar{y}_c)\]
Cylindrical Projection
Full-view (360°) panoramas
Spherical projection

- Map 3D point \((X, Y, Z)\) onto sphere
  
  \[
  (\hat{x}, \hat{y}, \hat{z}) = \frac{1}{\sqrt{X^2 + Y^2 + Z^2}} (X, Y, Z)
  \]

- Convert to spherical coordinates
  
  \[
  (\sin \theta \cos \phi, \sin \phi, \cos \theta \cos \phi) = (\hat{x}, \hat{y}, \hat{z})
  \]

- Convert to spherical image coordinates
  
  \[
  (\tilde{x}, \tilde{y}) = (f \theta, f h) + (\bar{x}_c, \bar{y}_c)
  \]
Spherical Projection
Full-view Panorama
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

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

Use this instead of normal projection
Blending the mosaic

An example of image compositing: the art (and sometime science) of combining images together…
Questions?
M. Uyttendaele, A. Eden, and R. Szeliski.
Eliminating ghosting and exposure artifacts in image mosaics.
M. Uyttendaele, A. Eden, and R. Szeliski.
Eliminating ghosting and exposure artifacts in image mosaics.
Magic: automatic panos

Extensions

• Video
• Additional objects
• Mok’s panomorph
• http://www.sarnoff.com/products_services/vision/tech_papers/kumarvb.pdf
• http://www.cs.huji.ac.il/~peleg/papers/pami00-manifold.pdf
• http://www.cs.huji.ac.il/~peleg/papers/cvpr00-rectified.pdf
• http://www.cs.huji.ac.il/~peleg/papers/cvpr05-dynmos.pdf
• http://www.robots.ox.ac.uk/~vgg/publications/papers/schaffalitzky02.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/

http://www.fdrtools.com/front_e.php
• http://www.cs.washington.edu/education/courses/csep576/05wi/readings/szeliskiShum97.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://grail.cs.washington.edu/projects/panovidtex/
• http://research.microsoft.com/vision/visionbasedmodeling/publications/Baudisch-OZCHI05.pdf
• http://www.vision.caltech.edu/lihi/Demos/SquarePanorama.html
• http://graphics.stanford.edu/papers/multi-cross-slits/
View morphing

subject

common view plane

viewpoint 1

viewpoint 2