Image Stitching

Computer Vision
Jia-Bin Huang, Virginia Tech

Many slides from S. Seitz and D. Hoiem
Administrative stuffs

- HW 3 is out due 11:59 PM Oct 17

- Please start early. Deadlines are firm.
  - No emails requesting extensions

- Getting help?
  - *Five* free late days without penalty
  - Piazza
  - Office hours

- No free late dates for final projects
Review: Camera Projection Matrix

\[
\begin{align*}
\mathbf{x} &= \mathbf{K}[\mathbf{R} \quad \mathbf{t}] \mathbf{X} \\
\begin{bmatrix}
\mathbf{u} \\
\mathbf{v} \\
\mathbf{1}
\end{bmatrix} &= \\
\begin{bmatrix}
\mathbf{f} & \mathbf{s} & \mathbf{u}_0 \\
0 & \alpha \mathbf{f} & \mathbf{v}_0 \\
0 & 0 & 1
\end{bmatrix} \\
\begin{bmatrix}
\mathbf{r}_{11} & \mathbf{r}_{12} & \mathbf{r}_{13} & \mathbf{t}_x \\
\mathbf{r}_{21} & \mathbf{r}_{22} & \mathbf{r}_{23} & \mathbf{t}_y \\
\mathbf{r}_{31} & \mathbf{r}_{32} & \mathbf{r}_{33} & \mathbf{t}_z \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1}
\end{bmatrix} \\
\begin{bmatrix}
\mathbf{X} \\
\mathbf{Y} \\
\mathbf{Z} \\
\mathbf{1}
\end{bmatrix}
\end{align*}
\]
Review: Camera Calibration

Method 1: Use an object (calibration grid) with known geometry

- Correspond image points to 3d points
- Get least squares solution (or non-linear solution)

\[
\begin{bmatrix}
    w_1 & m_{11} & m_{12} & m_{13} & m_{14} \\
    w_2 & m_{21} & m_{22} & m_{23} & m_{24} \\
    w_3 & m_{31} & m_{32} & m_{33} & m_{34} \\
    1 & 1 & 1 & 1 & 1
\end{bmatrix}
\]

Known 2d image coordinates

Known 3d locations

Unknown Camera Parameters
Unknown Camera Parameters

Known 2d image coords

$$\begin{bmatrix} s \ u \\ s \ v \\ s \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Known 3d locations

$$0 = m_{11} X + m_{12} Y + m_{13} Z + m_{14} - m_{31} u X - m_{32} u Y - m_{33} u Z - m_{34} u$$

$$0 = m_{21} X + m_{22} Y + m_{23} Z + m_{24} - m_{31} v X - m_{32} v Y - m_{33} v Z - m_{34} v$$

- Homogeneous linear system.
  Solve for m's entries using linear least squares

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1 X_1 & -u_1 Y_1 & -u_1 Z_1 & -u_1 \\ 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1 X_1 & -v_1 Y_1 & -v_1 Z_1 & -v_1 \\ \vdots \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_n X_n & -u_n Y_n & -u_n Z_n & -u_n \\ 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_n X_n & -v_n Y_n & -v_n Z_n & -v_n \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$[U, S, V] = \text{svd}(A);$$

$$M = V(:, \text{end});$$

$$M = \text{reshape}(M, [], 3)';$$
Review: Calibration by vanishing points

Orthogonality constraints $X_i^T X_j = 0$

Unknown camera parameters $f, u_0, v_0$

$$K = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$K^{-1} = \begin{bmatrix} 1 & 0 & -u_0 \\ f & 0 & -v_0 \\ 0 & 1 & -f \end{bmatrix}$$

Constraints for $p_1, p_2, p_3$

$$p_1^T(K^{-1})^T(K^{-1})p_2 = 0 \quad (x_1 - u_0)(x_2 - u_0) + (y_1 - v_0)(y_2 - v_0) + f^2 = 0 \quad \text{Eqn (1)}$$

$$p_1^T(K^{-1})^T(K^{-1})p_3 = 0 \quad (x_1 - u_0)(x_3 - u_0) + (y_1 - v_0)(y_3 - v_0) + f^2 = 0 \quad \text{Eqn (2)}$$

$$p_2^T(K^{-1})^T(K^{-1})p_3 = 0 \quad (x_2 - u_0)(x_3 - u_0) + (y_2 - v_0)(y_3 - v_0) + f^2 = 0 \quad \text{Eqn (3)}$$

Eqn (1) – Eqn (2) $\Rightarrow (x_1 - u_0)(x_2 - x_3) + (y_1 - v_0)(y_2 - y_3) = 0$

Eqn (2) – Eqn (3) $\Rightarrow (x_3 - u_0)(x_1 - x_2) + (y_3 - v_0)(y_1 - y_2) = 0$

Solve for $u_0, v_0$

$$f = \sqrt{-(x_1 - u_0)(x_2 - u_0) - (y_1 - v_0)(y_2 - v_0)}$$
Review: Calibration by vanishing points

Rotation matrix \( R = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix} \)

Unknown camera parameters \( R \)

\( p_i = KRX_i \)

Set directions of vanishing points

\( X_1 = [1, 0, 0]^T \)
\( X_2 = [0, 1, 0]^T \)
\( X_3 = [0, 0, 1]^T \)

Special properties of \( R \)

- \( \text{inv}(R) = R^T \)
- Each row and column of \( R \) has unit length

\( p_1 = Kr_1 \quad r_1 = K^{-1}p_1 \)
\( p_2 = Kr_2 \quad r_2 = K^{-1}p_2 \)
\( p_3 = Kr_3 \quad r_3 = K^{-1}p_3 \)
Measuring height

vanishing line (horizon)

\[ v \approx (b \times b_0) \times (v_x \times v_y) \]

image cross ratio

\[ \frac{\|t - b\|}{\|b - t\|} \cdot \frac{\|v_z - r\|}{\|r - b\|} = \frac{H}{R} \]
This class: Image Stitching

• Combine two or more overlapping images to make one larger image

Slide credit: Vaibhav Vaish
Concepts introduced/reviewed in today’s lecture

- Camera model
- Homographies
- Solving homogeneous systems of linear equations
- Keypoint-based alignment
- RANSAC
- Blending
- How the iphone stitcher works
Illustration

Camera Center
Problem set-up

• $x = K [R \ t] X$
• $x' = K' [R' \ t'] X$
• $t = t' = 0$

• $x' = Hx$  where  $H = K' R' R^{-1} K^{-1}$

• Typically only $R$ and $f$ will change (4 parameters), but, in general, $H$ has 8 parameters
Homography

• Definition
  • General mathematics:
    \( \text{homography} = \text{projective linear transformation} \)
  • Vision (most common usage):
    \( \text{homography} = \text{linear transformation between two image planes} \)

• Examples
  • Project 3D surface into frontal view
  • Relate two views that differ only by rotation
Homography example: Image rectification

To unwarp (rectify) an image solve for homography $H$ given $p$ and $p'$: $wp' = Hp$
Homography example: Planar mapping
1. Detect keypoints (e.g., SIFT)
2. Match keypoints (e.g., 1st/2nd NN < thresh)
3. Estimate homography with four matched keypoints (using RANSAC)
4. Combine images
Computing homography

Assume we have four matched points: How do we compute homography $H$?

Direct Linear Transformation (DLT)

$$x' = Hx \quad x' = \begin{bmatrix} w'u' \\ w'v' \\ w' \end{bmatrix} \quad H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$$

$$\begin{bmatrix} -u & -v & -1 & 0 & 0 & 0 & 0 & uu' & vv' & u' \\ 0 & 0 & 0 & -u & -v & -1 & uv' & vv' & v' \end{bmatrix}h = 0$$
Computing homography

Direct Linear Transform

\[
\begin{bmatrix}
-u_1 & -v_1 & -1 & 0 & 0 & 0 & u_1u'_1 & v_1u'_1 & u'_1 \\
0 & 0 & 0 & -u_1 & -v_1 & -1 & u_1v'_1 & v_1v'_1 & v'_1 \\
\vdots \\
0 & 0 & 0 & -u_n & -v_n & -1 & u_nv'_n & v_nv'_n & v'_n
\end{bmatrix}
\]

\[h = 0 \implies Ah = 0\]

- Apply SVD: \(UDV^T = A\)
- \(h = V_{\text{smallest}}\) (column of \(V\) corr. to smallest singular value)

\[
h = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_9 \end{bmatrix}
\]

Matlab

\[
[U, S, V] = \text{svd}(A);
h = V(:, \text{end});
\]

Explanations of SVD and solving homogeneous linear systems
Computing homography

• Assume we have four matched points: How do we compute homography $\mathbf{H}$?

Normalized DLT

1. Normalize coordinates for each image
   a) Translate for zero mean
   b) Scale so that average distance to origin is $\sim\sqrt{2}$
      
      $\tilde{x} = Tx \quad \tilde{x}' = T'x'$
      
      This makes problem better behaved numerically (see HZ p. 107-108)

2. Compute $\tilde{\mathbf{H}}$ using DLT in normalized coordinates

3. Unnormalize: $\mathbf{H} = T'^{-1}\tilde{\mathbf{H}}T$
   
   $x'_i = \mathbf{H}x_i$
Computing homography

• Assume we have matched points with outliers: How do we compute homography $H$?

Automatic Homography Estimation with RANSAC

1. Choose number of samples $N$

For probability $p$ of no outliers:

$$N = \frac{\log(1 - p)}{\log(1 - (1 - \epsilon)^s)}$$

- $N$, number of samples
- $s$, size of sample set
- $\epsilon$, proportion of outliers

<table>
<thead>
<tr>
<th>Sample size</th>
<th>5%</th>
<th>10%</th>
<th>20%</th>
<th>25%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>13</td>
<td>23</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>8</td>
<td>11</td>
<td>22</td>
<td>47</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>4</td>
<td>8</td>
<td>12</td>
<td>17</td>
<td>38</td>
<td>95</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>4</td>
<td>10</td>
<td>16</td>
<td>24</td>
<td>63</td>
<td>191</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>5</td>
<td>13</td>
<td>21</td>
<td>35</td>
<td>106</td>
<td>382</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>6</td>
<td>17</td>
<td>29</td>
<td>51</td>
<td>177</td>
<td>766</td>
</tr>
</tbody>
</table>

e.g. for $p = 0.95$
Computing homography

• Assume we have matched points with outliers: How do we compute homography $H$?

Automatic Homography Estimation with RANSAC
1. Choose number of samples $N$
2. Choose 4 random potential matches
3. Compute $H$ using normalized DLT
4. Project points from $x$ to $x'$ for each potentially matching pair: $x'_i = Hx_i$
5. Count points with projected distance < $t$
   • E.g., $t = 3$ pixels
6. Repeat steps 2-5 $N$ times
   • Choose $H$ with most inliers
Automatic Image Stitching

1. Compute interest points on each image
2. Find candidate matches
3. Estimate homography $H$ using matched points and RANSAC with normalized DLT
4. Project each image onto the same surface and blend
   - Matlab: maketform, imtransform
RANSAC for Homography

Initial Matched Points
RANSAC for Homography

Final Matched Points
RANSAC for Homography
Choosing a Projection Surface

Many to choose: planar, cylindrical, spherical, cubic, etc.
Planar Mapping

1) For red image: pixels are already on the planar surface
2) For green image: map to first image plane
Planar Projection

Photos by Russ Hewett
Planar Projection
Cylindrical Mapping

1) For red image: compute $h$, theta on cylindrical surface from $(u, v)$
2) For green image: map to first image plane, then map to cylindrical surface
Cylindrical Projection

Cylindrical
Cylindrical Projection

Cylindrical
Recognizing Panoramas

Some of following material from Brown and Lowe 2003 talk

Recognizing Panoramas

Input: N images
1. Extract SIFT points, descriptors from all images
2. Find K-nearest neighbors for each point (K=4)
3. For each image
   a) Select M candidate matching images by counting matched keypoints (m=6)
   b) Solve homography $H_{ij}$ for each matched image
Recognizing Panoramas

Input: N images
1. Extract SIFT points, descriptors from all images
2. Find K-nearest neighbors for each point (K=4)
3. For each image
   a) Select M candidate matching images by counting matched keypoints (m=6)
   b) Solve homography $H_{ij}$ for each matched image
   c) Decide if match is valid ($n_i > 8 + 0.3 \ n_f$)
Recognizing Panoramas (cont.)

(now we have matched pairs of images)

4. Find connected components
Finding the panoramas
Finding the panoramas
Recognizing Panoramas (cont.)

(now we have matched pairs of images)

4. Find connected components

5. For each connected component
   a) Perform bundle adjustment to solve for rotation \((\theta_1, \theta_2, \theta_3)\) and focal length \(f\) of all cameras
   b) Project to a surface (plane, cylinder, or sphere)
   c) Render with multiband blending
Bundle adjustment for stitching

• Non-linear minimization of re-projection error

\[ R_i = e^{[\theta_i]_\times}, \quad [\theta_i]_\times = \begin{bmatrix} 0 & -\theta_i 3 & \theta_i 2 \\ \theta_i 3 & 0 & -\theta_i 1 \\ -\theta_i 2 & \theta_i 1 & 0 \end{bmatrix} \]

\[ \hat{x}' = Hx \quad \text{where} \quad H = K' R' R^{-1} K^{-1} \]

error = \sum_{i=1}^{N} \sum_{j}^{M_i} \sum_{k} \text{dist}(x', \hat{x}')

• Solve non-linear least squares (Levenberg-Marquardt algorithm)
  
  • See paper for details
Bundle Adjustment

- New images initialised with rotation, focal length of best matching image
Bundle Adjustment

• New images initialised with rotation, focal length of best matching image
Details to make it look good

- Choosing seams
- Blending
Choosing seams

• Easy method
  • Assign each pixel to image with nearest center
Choosing seams

• Easy method
  • Assign each pixel to image with nearest center
  • Create a mask:
    • $\text{mask}(y, x) = 1$ iff pixel should come from im1
  • Smooth boundaries (called “feathering”):
    • $\text{mask}\_sm = \text{imfilter}(\text{mask}, \text{gausfil})$;
• Composite
  • $\text{imblend} = \text{im1}\_c.*\text{mask} + \text{im2}\_c.*(1-\text{mask})$;
Choosing seams

• Better method: dynamic program to find seam along well-matched regions

Gain compensation

• Simple gain adjustment
  • Compute average RGB intensity of each image in overlapping region
  • Normalize intensities by ratio of averages
Multi-band Blending

• Burt & Adelson 1983
  • Blend frequency bands over range $\propto \lambda$
Multiband Blending with Laplacian Pyramid

- At low frequencies, blend slowly
- At high frequencies, blend quickly
Multiband blending

1. Compute Laplacian pyramid of images and mask

2. Create blended image at each level of pyramid

3. Reconstruct complete image
Blending comparison (IJCV 2007)

(a) Linear blending

(b) Multi-band blending
Blending Comparison

(b) Without gain compensation

(c) With gain compensation

(d) With gain compensation and multi-band blending
Further reading

• DLT algorithm: HZ p. 91 (alg 4.2), p. 585
• Normalization: HZ p. 107-109 (alg 4.2)
• RANSAC: HZ Sec 4.7, p. 123, alg 4.6

• [link] Rick Szeliski’s alignment/stitching tutorial
• [link] Recognising Panoramas: Brown and Lowe, IJCV 2007 (also bundle adjustment)
How does iphone panoramic stitching work?

• Capture images at 30 fps

• Stitch the central 1/8 of a selection of images
  • Select which images to stitch using the accelerometer and frame-to-frame matching
  • Faster and avoids radial distortion that often occurs towards corners of images

• Alignment
  • Initially, perform cross-correlation of small patches aided by accelerometer to find good regions for matching
  • Register by matching points (KLT tracking or RANSAC with FAST (similar to SIFT) points) or correlational matching

• Blending
  • Linear (or similar) blending, using a face detector to avoid blurring face regions and choose good face shots (not blinking, etc)

Things to remember

- Homography relates rotating cameras
- Recover homography using RANSAC and normalized DLT
- Bundle adjustment minimizes reprojection error for set of related images
- Details to make it look nice (e.g., blending)
See you on Thursday

- Next class: Epipolar Geometry and Stereo Vision