Lecture 30

Robust Homography Estimation

(Hartley/Zisserman)

i) Compute Interest points (e.g. HCD)

ii) Compute putative correspondences (e.g. NCC)

iii) Apply RANSAC (Random Sample Consensus) $\rightarrow H$

iv) Re-calculate Optimal estimation of $H$ for all inliers

v) Re-compute Interest points using Guided matching

Step iv) usually involves minimization with LM, but the initialization step of Zhang's algorithm may suffice (see below)

Step v) allows for more and better interest points to be found since a search window can restrict more elaborate (complex) searches
RANSAC

Robust line estimation: 1) select two (s) random points to define a line
2) support for this line is measured by # of points that lie within a distance threshold
3) repeat 1) "many" times (N) and the line w/ most support is the robust fit
4) points within threshold are outliers

Questions:
- how many points make it robust?
- how many outliers can it detect/eliminate?
Number of samples: Unnecessary to use all samples

We need $N$ samples to guarantee with probability $p$ that one of the random samples with $s$ points is free of outliers.

Typically $p = 0.99$

If $w$ is the probability $x$ is an inlier

$\Rightarrow 1 - w = \epsilon$ \Rightarrow $1 - \epsilon$ outlier

$N$ selections (each of $s$ points) are required, where

$$(1 - w^s)^N = 1 - p \quad \text{or} \quad N = \log(1 - p)/\log(1 - (1 - \epsilon)^s)$$

<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>3</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>11</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>19</td>
<td>35</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>5</td>
<td>9</td>
<td>13</td>
<td>17</td>
<td>34</td>
<td>72</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>6</td>
<td>12</td>
<td>17</td>
<td>26</td>
<td>57</td>
<td>146</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>7</td>
<td>16</td>
<td>24</td>
<td>37</td>
<td>97</td>
<td>293</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>8</td>
<td>20</td>
<td>33</td>
<td>54</td>
<td>163</td>
<td>588</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>9</td>
<td>26</td>
<td>44</td>
<td>78</td>
<td>272</td>
<td>1177</td>
</tr>
</tbody>
</table>

Table 4.3. The number $N$ of samples required to ensure, with a probability $p = 0.99$, that at least one sample has no outliers for a given size of sample, $s$, and proportion of outliers, $\epsilon$. 
**Size N: Algorithm:**

\[ N = \infty \; ; \; \text{sample-count} = 0 \]

\[
\text{while} \; N > \text{sample-count} \; \{
\]

- choose a sample & count # inliers
- set \[ e = 1 - \left( \frac{\text{# inliers}}{\text{total number points}} \right) \]
- set \[ N \] as above
- increment sample-count

\[
\}
\]

---

**RANSAC: Algorithm: (Step iii) of Homography estimation**

Given the algorithm for \( N \) above

1) select a random sample of 4 correspondences and compute \( N \) (homography) (\( s = 4 \))

2) calculate the distance \( d_i \) for all other putative correspondences

3) compute the # of inliers consistent w/ \( N \) (i.e. \( d_i < \sqrt{5.99} \frac{1}{\text{inliers}} \))

Choose \( N \) w/ largest # of inliers
What is a good $d_1$?

Symmetric Transfer error

$$d_1^2 = d_{\text{transfer}}^2 = d(x, H^{-1}x')^2 + d(Hx, x')^2$$

Homography $H$:

**Normalization**: Given 4 or more corresponding points $(x, x')$ where $x = \begin{bmatrix} x \\ 1 \end{bmatrix}$, $x' = \begin{bmatrix} x' \\ 1 \end{bmatrix}$

i) Normalize $x$: compute a similarity transform $T$ (translation & scale) such that $x_i \rightarrow x_i$, w/ centroid $(0, 0)$ and average distance to origin is $\sqrt{2}$

ii) Normalize $x'$: same as above

iii) DLT (Direct Linear Transform) $N = \begin{bmatrix} h_1' \\ h_2' \\ h_3' \end{bmatrix}$

$$\begin{bmatrix} x^T \\ 0^T \\ -u'x^T \end{bmatrix} \begin{bmatrix} h_1' \\ h_2' \\ h_3' \end{bmatrix} = 0$$

*Note: We can derive as we did in class $(x' = Hx)$ or using the Homography constraint:

$$x'^T Hx = 0$$

$$x'^T Hx' = \begin{bmatrix} u'^T x' - h'^T x \\ h'^T x' - u'^T x' \\ u'^T x' - v'^T x' \end{bmatrix}$$

$$\begin{bmatrix} 0^T & -x^T & u'x^T \\ x^T & 0^T & -u'x^T \\ -u'x^T & u'x^T & 0^T \end{bmatrix} \begin{bmatrix} h_1' \\ h_2' \\ h_3' \end{bmatrix} = 0$$

*Only two rows are linearly independent*
(iii) cont. To solve for $U$ — each correspondence gives 2 rows. If we have 4 correspondences $\Rightarrow 8 \times 9$ system of equations.

$A$ is $8 \times 9$, which is still ok for using SVD & the (column) null vector of $A = UDV^T$:

\[
\begin{bmatrix}
\beta_1^2 \\
\beta_2^2 \\
\beta_3^2
\end{bmatrix}
\]

is the last column of $V$.

*Note: 4 pts are sufficient if you can guarantee they are not collinear (i.e. 3 of them)
**Putting it ALL TOGETHER**

**RANSAC Homography Estimation (CORRESP_POINTS, INLIERs, N)**

```plaintext
N = \infty; p = 0.39; \varepsilon = 0.5; \text{current\_max} = 1; \text{sample\_count} = 0;
while \( N > \text{sample\_count} \) {
    \text{randomly pick 4 pairs of points from CORRESP_POINTS;}
    if random 4 pairs are "good" \( \leftarrow \) "optional" test
        \begin{align*}
            H_{\text{tmp}} &= \text{Comp\_Homog(4 pairs)}; \\
            \text{inliers\_tmp} &= \text{Calculate\_distance}(H_{\text{tmp}}, \text{CORRESP_POINTS}); \\
            \text{if} (\text{length(inliers\_tmp)} > \text{current\_max}) \{
                \text{current\_max} = \text{length(inliers\_tmp)}; \\
                \text{inliers} = \text{inliers\_tmp}; \\
                H = H_{\text{tmp}}; \\
                e_{\text{tmp}} = 1 - \text{current\_max}/\text{length(CORRESP_POINTS)}; \\
                \text{if} (e > e_{\text{tmp}}) e = e_{\text{tmp}}; \\
            \}
        \end{align*}
    \text{sample\_count} += 1; \\
}\}
```

(see notes #2 and #3 below)
Notes: 1) the 4 pairs are "good" if they are not co-linear \( \Rightarrow \) co-linear points lead to singular \( H \).
There are "smart" ways to check co-linearity in all 3 out of 4 points. You could use:
1) \(|(a \times b) \cdot c| \approx 0 \) or 
2) \((a-b), (a-c) \approx 0 \)
or 
3) \( \text{det} \begin{vmatrix} a_x & a_y & 1 \\ b_x & b_y & 1 \\ c_x & c_y & 1 \end{vmatrix} \approx 0 \) and so on...
Be "smart" when applying this to "all" 3 out of 4 pts
This step is "optional" if you opt to use 5 pts.
2) comp_homog() - this function computes the homography \( H \) using DLT above (slides 5 & 6)
and \( n \) pairs of corresponding points, where:
\( n=4 \) in 
\( \text{RansacHomographyEstimation}() \)
\( n=\# \) of inliers afterwards (@ the end for final \( H \))
3) **Calculate distance**

be **very** careful here! You do **not** want to update $\tau$ in this function. Instead, you want to calculate the # of outliers that would guarantee an arbitrary $\tau$ for the distribution of inliers vs. outliers. In other words, that would guarantee an arbitrary probability $p$ that all inliers are within $\tau = \sqrt{5.99} \tau$

$$p = \int_{-\infty}^{\infty} N(0, \tau)$$

4) **Remember:** $x' = Hx \text{ or } [x'] = H[x']$

   it is **not**: $x' = Hx$

So, instead: $x' = \frac{Hx}{x}$
5) In case someone wants to synthesize some data to test their algorithm, a homography can be decomposed as:

\[ H = \lambda \left( R + \frac{1}{d} t n^T \right) \]

where:

- \( R, t \) = rotation & translation between cameras \( C \) & \( C' \)
- \( n \) = normal vector of plane \( \Pi \) w.r.t. \( C \)
- \( d \) = distance of \( \Pi \) to \( C \)