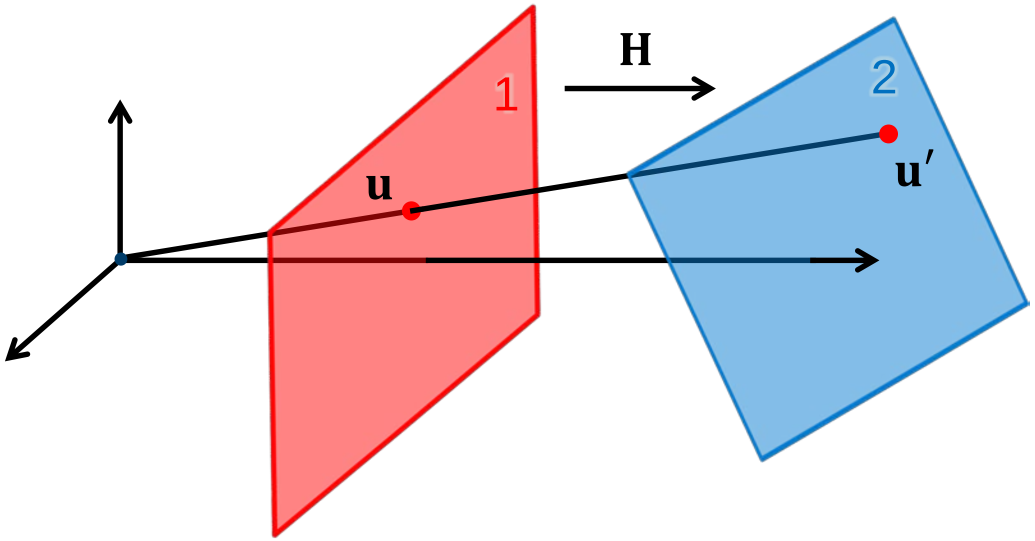
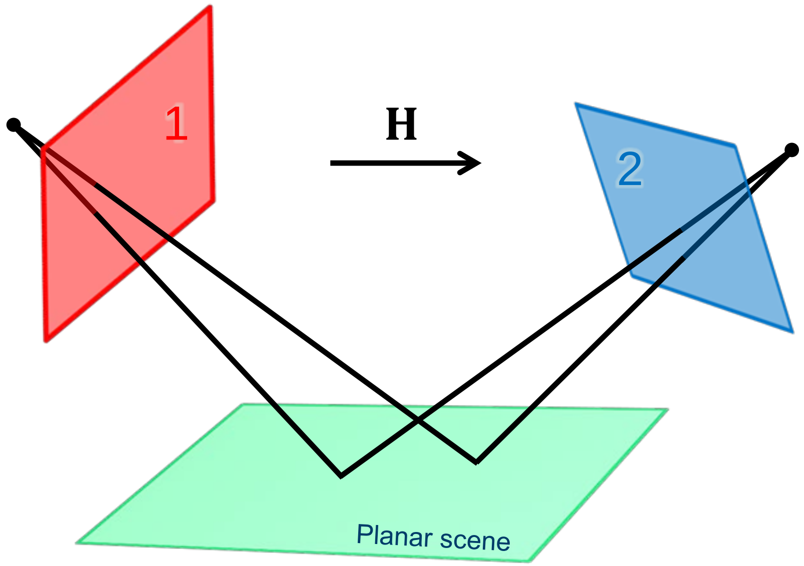
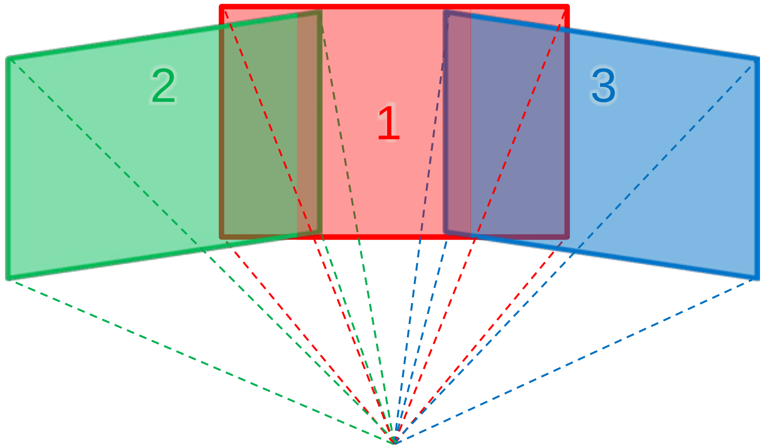
Lecture 4.3
Estimating homographies from feature correspondences

Thomas Opsahl

Homographies induced by central projection





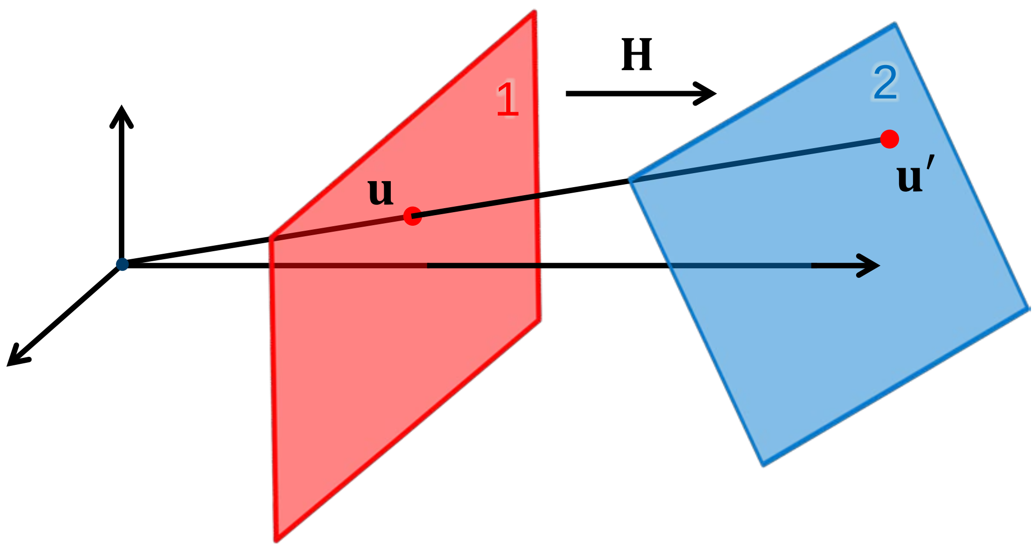


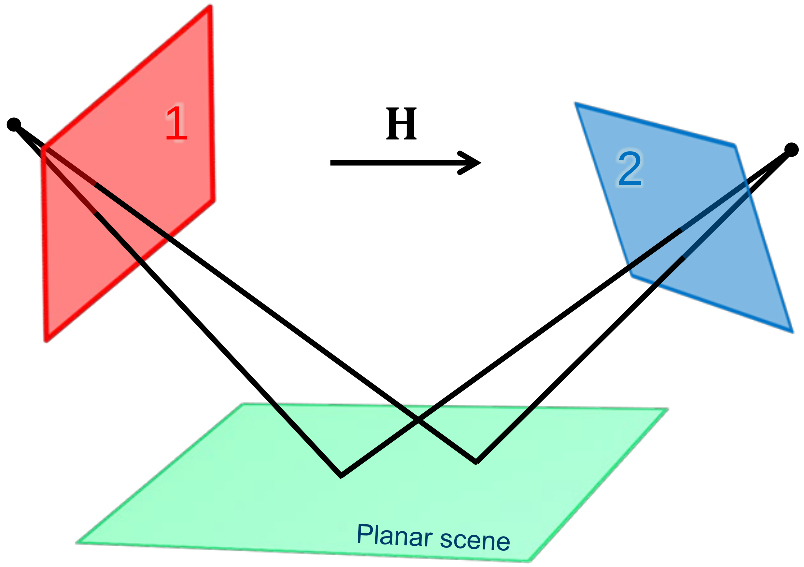
- Homography $\mathbf{H}\tilde{\mathbf{u}} = \tilde{\mathbf{u}}'$

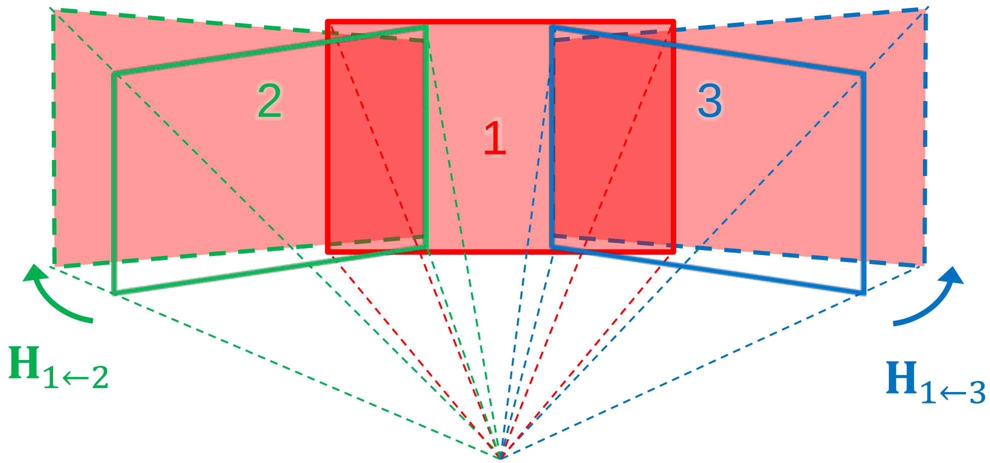
$$\mathbf{H} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$$

- Point-correspondences can be determined automatically
- Erroneous correspondences are common
- Robust estimation is required to find $\mathbf{H}$

Homographies induced by central projection







- Homography $\mathbf{H}\tilde{\mathbf{u}} = \tilde{\mathbf{u}}'$

$$\mathbf{H} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$$

- Point-correspondences can be determined automatically
- Erroneous correspondences are common
- Robust estimation is required to find $\mathbf{H}$

Estimating the homography between overlapping images

- Establish point correspondences $\mathbf{u}_i \leftrightarrow \mathbf{u'}_i$
  - Find key points $\{\mathbf{u}_i \in \mathrm{Img1}\}$ and $\{\mathbf{u'}_i \in \mathrm{Img2}\}$
  - Represent key points by suitable descriptors
  - Determine correspondences $\mathbf{u}_i \leftrightarrow \mathbf{u'}_i$ by matching descriptors
  - Some wrong correspondences are to be expected

- Estimate the homography $\mathbf{H}$ such that $\mathbf{u'}_i = \mathbf{H}\mathbf{u}_i \; \forall i$
  - Robust estimation with RANSAC
  - Improved estimation based on RANSAC inliers

- This homography enables us to compose the images into a larger image
  - Image mosaicing
  - Panorama

Adaptive RANSAC

Objective

To robustly fit a model $\mathbf{y} = f(\mathbf{x}; \boldsymbol{\alpha})$ to a data set S containing outliers

Algorithm

1. Let $N = \infty$, $S_{IN} = \emptyset$ and $\#iterations = 0$
2. while $N > \#iterations$ repeat 3-5
3. Estimate parameters $\boldsymbol{\alpha}_{tst}$ from a random n-tuple from S
4. Determine inlier set S_{tst}, i.e. data points within a distance t of the model $\mathbf{y} = f(\mathbf{x}; \boldsymbol{\alpha}_{tst})$
5. If $|S_{tst}| > |S_{IN}|$, set $S_{IN} = S_{tst}$, $\boldsymbol{\alpha} = \boldsymbol{\alpha}_{tst}$, $\omega = \frac{|S_{IN}|}{|S|}$ and $N = \frac{log(1-p)}{log(1-\omega^n)}$ with $p = 0.99$
   Increase $\#iterations$ by 1

Estimating the homography

- Estimating the homography in a RANSAC scheme requires
  1. A basic homography estimation method for n point-correspondences
  2. A way to determine the inlier set of point-correspondences for a given homography

Estimating the homography

- Estimating the homography in a RANSAC scheme requires
  1. **A basic homography estimation method for n point-correspondences**
  2. A way to determine the inlier set of point-correspondences for a given homography

- The homography has 8 degrees of freedom, but it is custom to treat all 9 entries of the matrix as unknowns instead of setting one of the entries to 1 which excludes all potential solutions where this entry is 0

- Let us solve the equation $\mathbf{H}\tilde{\mathbf{u}} = \tilde{\mathbf{u}}'$ for the entries of the homography matrix

$$\mathbf{H}\tilde{\mathbf{u}} = \tilde{\mathbf{u}}'$$

$$\begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}$$

Basic homography estimation

$$\begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \Leftrightarrow \begin{Bmatrix} uh_1 + vh_2 + h_3 = u' \\ uh_4 + vh_5 + h_6 = v' \\ uh_7 + vh_8 + h_9 = 1 \end{Bmatrix} \Leftrightarrow \begin{bmatrix} 0 & 0 & 0 & -u & -v & -1 & v'u & v'v & v' \\ u & v & 1 & 0 & 0 & 0 & -u'u & -u'v & -u' \\ -v'u & -v'v & -v' & u'u & u'v & u' & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \Leftrightarrow \mathbf{Ah} = \mathbf{0}$$

Basic homography estimation

$$\begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \Leftrightarrow \begin{Bmatrix} uh_1 + vh_2 + h_3 = u' \\ uh_4 + vh_5 + h_6 = v' \\ uh_7 + vh_8 + h_9 = 1 \end{Bmatrix} \Leftrightarrow \begin{bmatrix} 0 & 0 & 0 & -u & -v & -1 & v'u & v'v & v' \\ u & v & 1 & 0 & 0 & 0 & -u'u & -u'v & -u' \\ -v'u & -v'v & -v' & u'u & u'v & u' & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \Leftrightarrow \mathbf{A}\mathbf{h} = \mathbf{0}$$

Observe that the third row in $\mathbf{A}$ is a linear combination of the first and second row

$$row_3 = -u' \cdot row_1 - v' \cdot row_2$$

Hence every correspondence $\mathbf{u}_i \leftrightarrow \mathbf{u}'_i$ contribute with 2 equations in the 9 unknown entries

Basic homography estimation

- Since $\mathbf{H}$ (and thus $\mathbf{h}$) is homogeneous, we only need the matrix $\mathbf{A}$ to have rank 8 in order to determine $\mathbf{h}$ up to scale

- It is sufficient with 4 point correspondences where no 3 points are collinear

- We can calculate the non-trivial solution to the equation $\mathbf{Ah} = \mathbf{0}$ by SVD
$$\mathrm{svd}(\mathbf{A}) = \mathbf{USV}^T$$

- The solution is given by the right singular vector without a singular value which is the last column of $\mathbf{V}$, i.e. $\mathbf{h} = \mathbf{v}_9$

$$\begin{bmatrix} 0 & 0 & 0 & -u_1 & -v_1 & -1 & v_1'u_1 & v_1'v_1 & v_1' \\ u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1'u_1 & -u_1'v_1 & -u_1' \\ 0 & 0 & 0 & -u_2 & -v_2 & -1 & v_2'u_2 & v_2'v_2 & v_2' \\ u_2 & v_2 & 1 & 0 & 0 & 0 & -u_2'u_2 & -u_2'v_2 & -u_2' \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}$$

$$\mathbf{Ah} = \mathbf{0}$$

Basic homography estimation

- Estimating the homography in a RANSAC scheme requires
  1. **A basic homography estimation method for n point-correspondences**
  2. A way to determine which of the point correspondences that are inliers for a given homography

Direct Linear Transform

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & -u_1 & -v_1 & -1 & v_1'u_1 & v_1'v_1 & v_1' \\ u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1'u_1 & -u_1'v_1 & -u_1' \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

1. Build the matrix $\mathbf{A}$ from at least 4 point-correspondences $(u_i, v_i) \leftrightarrow (u'_i, v'_i)$
2. Obtain the SVD of $\mathbf{A}$: $\mathbf{A} = \mathbf{USV}^T$
3. If $\mathbf{S}$ is diagonal with positive values in descending order along the main diagonal, then $\mathbf{h}$ equals the last column of $\mathbf{V}$
4. Reconstruct $\mathbf{H}$ from $\mathbf{h}$

Basic homography estimation

- The basic DLT algorithm is never used with more than 4 point-correspondences
- This is because the algorithm performs better when all the terms of A has a similar scale
  - Note that some of the terms will always be of scale 1
- To achieve this, it is common to extend the algorithm with a normalization and a denormalization step

Normalized Direct Linear Transform

1. Normalize the set of points $\mathbf{u}_i = [u_i, v_i]^T$ by computing a similarity transform T that translates the centroid to the origin and scales such that the average distance from the origin is $\sqrt{2}$
2. In the same way normalize the set of points $\mathbf{u'}_i = [u'_i, v'_i]^T$ by computing a similarity transform $\mathbf{T}'$
3. Apply the basic DLT algorithm on the normalized points to obtain a homography $\widehat{\mathbf{H}}$
4. Denormalize to get the homography: $\mathbf{H} = \mathbf{T}'^{-1}\widehat{\mathbf{H}}\mathbf{T}$

Basic homography estimation

- Estimating the homography in a RANSAC scheme requires
  1. A basic homography estimation method for n point-correspondences
  2. **A way to determine the inlier set of point-correspondences for a given homography**

- For a point correspondence $(u_i, v_i) \leftrightarrow (u'_i, v'_i)$ and homography $\mathbf{H}$, we can choose from several errors
  - Algebraic error: $\varepsilon_i = \|\mathbf{A}_i \mathbf{h}\|$ where

$$\mathbf{A}_i = \begin{bmatrix} 0 & 0 & 0 & -u_i & -v_i & -1 & v'_i u_i & v'_i v_i & v'_i \\ u_i & v_i & 1 & 0 & 0 & 0 & -u'_i u_i & -u'_i v_i & -u'_i \end{bmatrix}$$

  - Geometric errors:
    1. $\varepsilon_i = d(\mathbf{H}\mathbf{u}_i, \mathbf{u}'_i) + d(\mathbf{u}_i, \mathbf{H}^{-1}\mathbf{u}'_i)$ **(Reprojection error)**
    2. $\varepsilon_i = d(\mathbf{u}_i, \mathbf{H}^{-1}\mathbf{u}'_i)$
    3. $\varepsilon_i = d(\mathbf{H}\mathbf{u}_i, \mathbf{u}'_i)$

Notation	
Euclidean distance	$d(\cdot,\cdot)$
Inhomogenous $\mathbf{H}\tilde{\mathbf{u}}_i$	$\mathbf{H}\mathbf{u}_i$
Inhomogeneous $\mathbf{H}^{-1}\tilde{\mathbf{u}}'_i$	$\mathbf{H}^{-1}\mathbf{u}'_i$

Robust homography estimation

RANSAC estimation of homography
For a set of point-correspondences $S = \{\mathbf{u_i} \mapsto \mathbf{u'}_i\}$, perform N iterations, where N is determined adaptively

1. Estimate $\mathbf{H}_{tst}$ from 4 random correspondences $\mathbf{u_i} \mapsto \mathbf{u'}_i$ using the basic DLT algorithm
2. Determine the set of inlier-correspondences $S_{tst} = \{\mathbf{u_i} \mapsto \mathbf{u'}_i \; such\ that\ \epsilon_i < t\}$
   Here one can choose $\epsilon_i = d(\mathbf{H}\mathbf{u}_i, \mathbf{u'}_i) + d(\mathbf{u}_i, \mathbf{H}^{-1}\mathbf{u'}_i)$ and $t = \sqrt{5.99}\sigma$ where σ is the expected uncertainty in key-point positions
3. If $|S_{tst}| > |S_{IN}|$ update N, homography and inlier set: $\mathbf{H} = \mathbf{H}_{tst}$, $S_{IN} = S_{tst}$

- Finally we would typically re-estimate $\mathbf{H}$ from all correspondences in S_{IN}
  - Normalized DLT
  - Minimize $\epsilon = \sum \epsilon_i$ in an iterative optimization method like Levenberg Marquardt

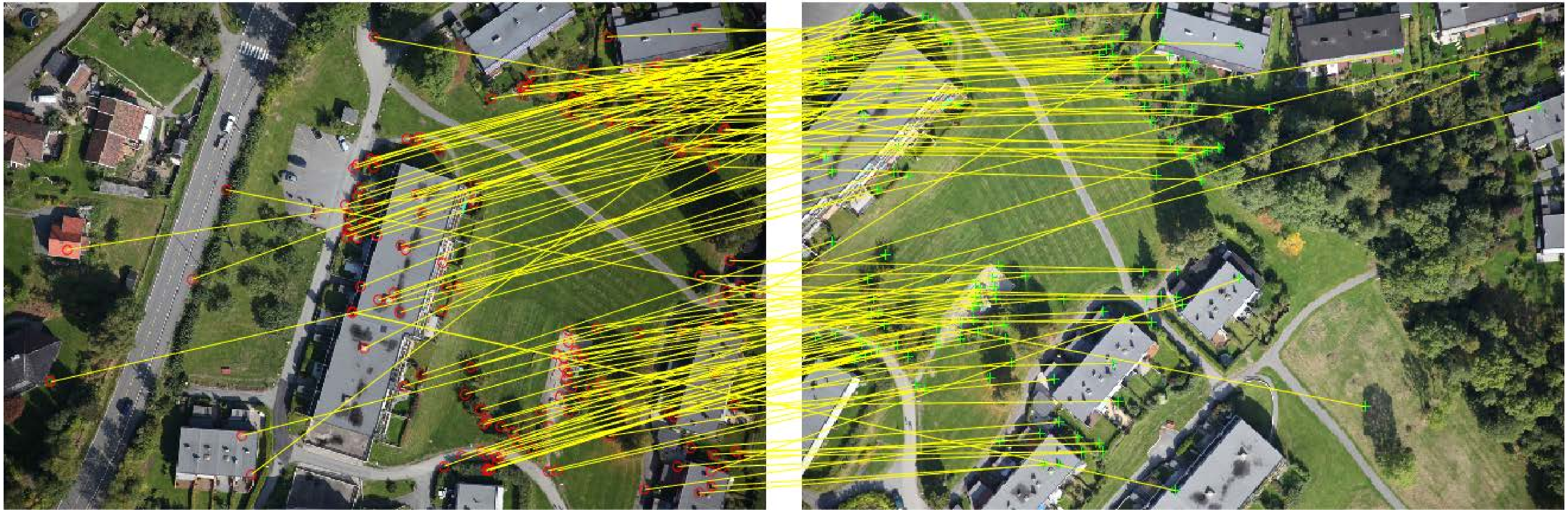
Image mosaicing

- Let us compose these two images into a larger image

Image mosaicing

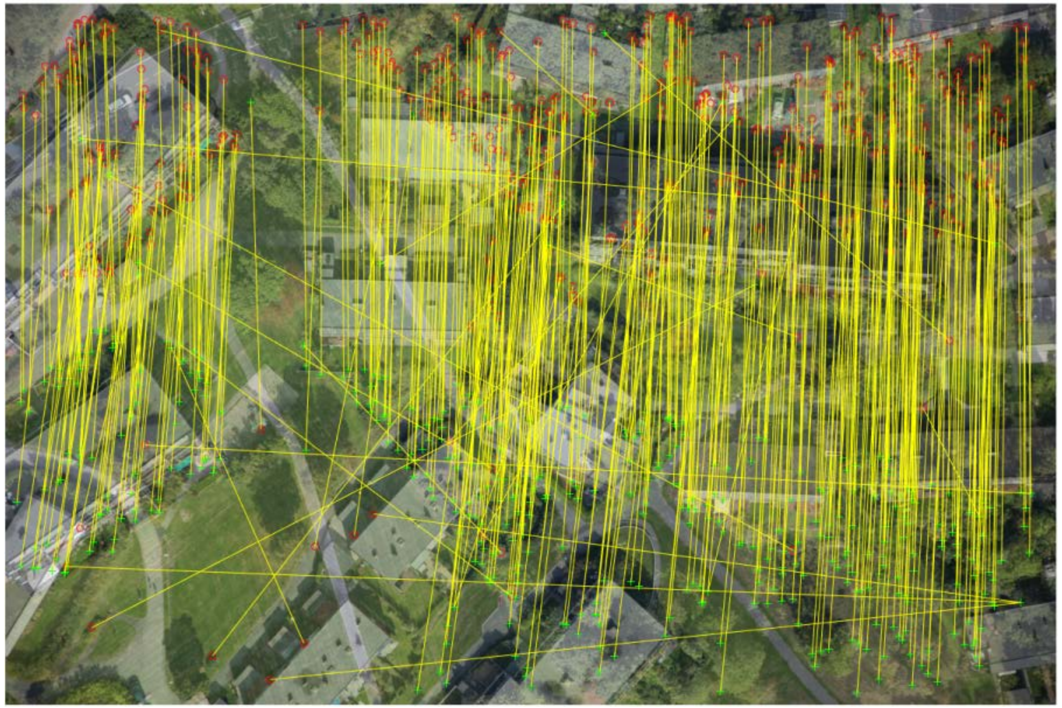
- Find key points and represent by descriptors

Image mosaicing



- Establish point-correspondences by matching descriptors
- Several wrong correspondences

Image mosaicing



- Establish point-correspondences by matching descriptors
- Several wrong correspondences

Image mosaicing

$\xrightarrow{\mathbf{H}}$

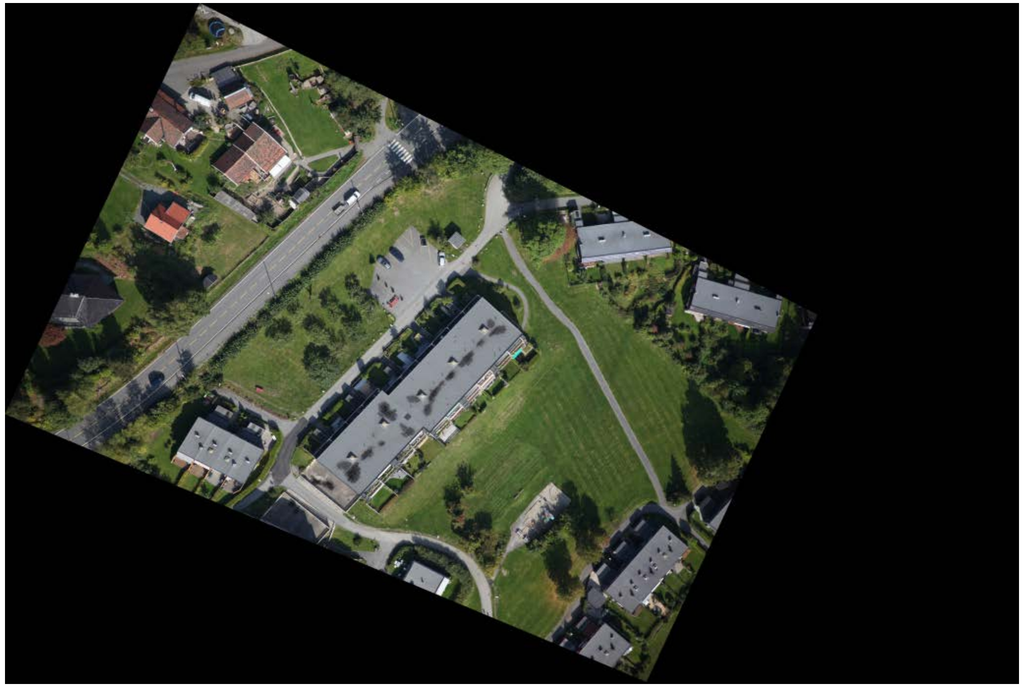
- Estimate homography $\mathbf{H}\tilde{\mathbf{u}} = \tilde{\mathbf{u}}'$
  - OpenCV

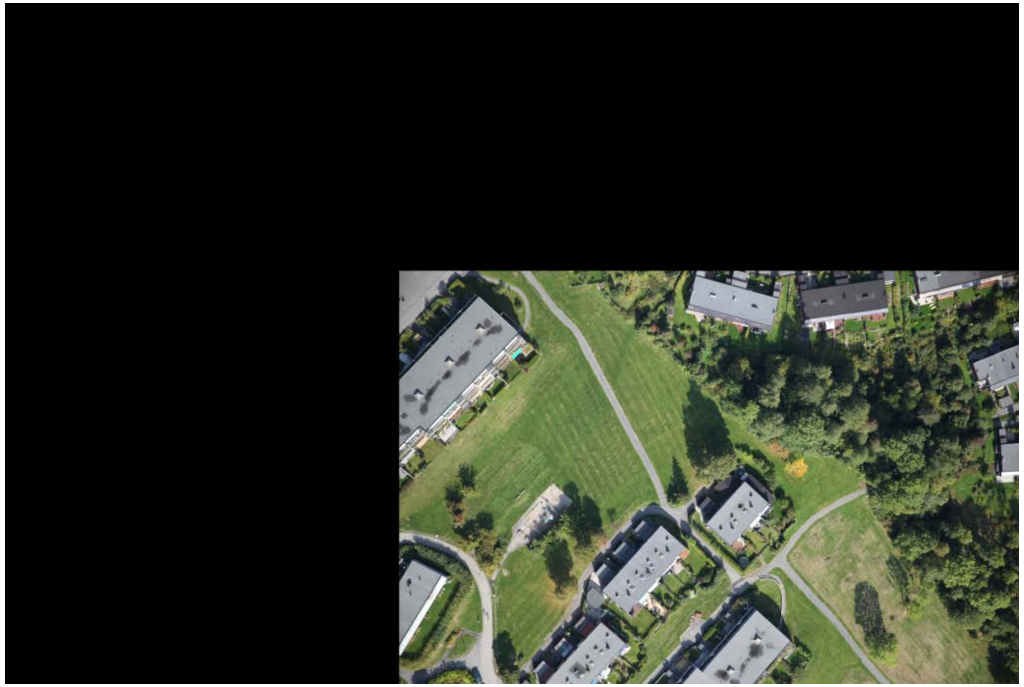
```
#include "opencv2/calib3d.hpp"
cv::findHomography(srcPoints, dstPoints, CV_RANSAC);
```

  - Matlab

```
tform = estimateGeometricTransform(srcPoints,dstPoints,’projective’);
```

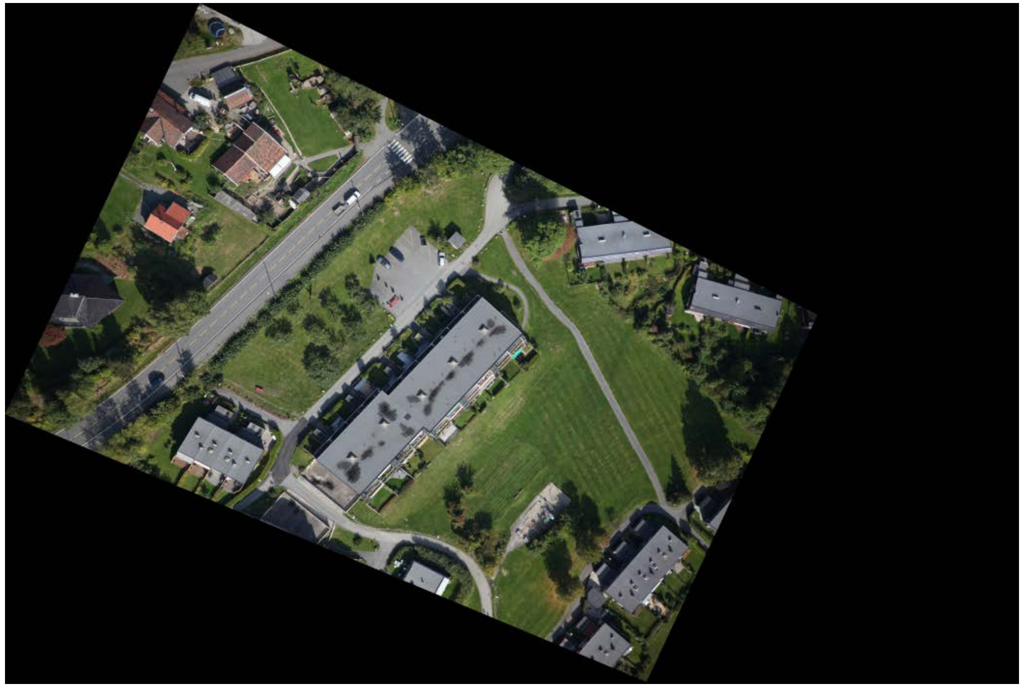
Image mosaicing

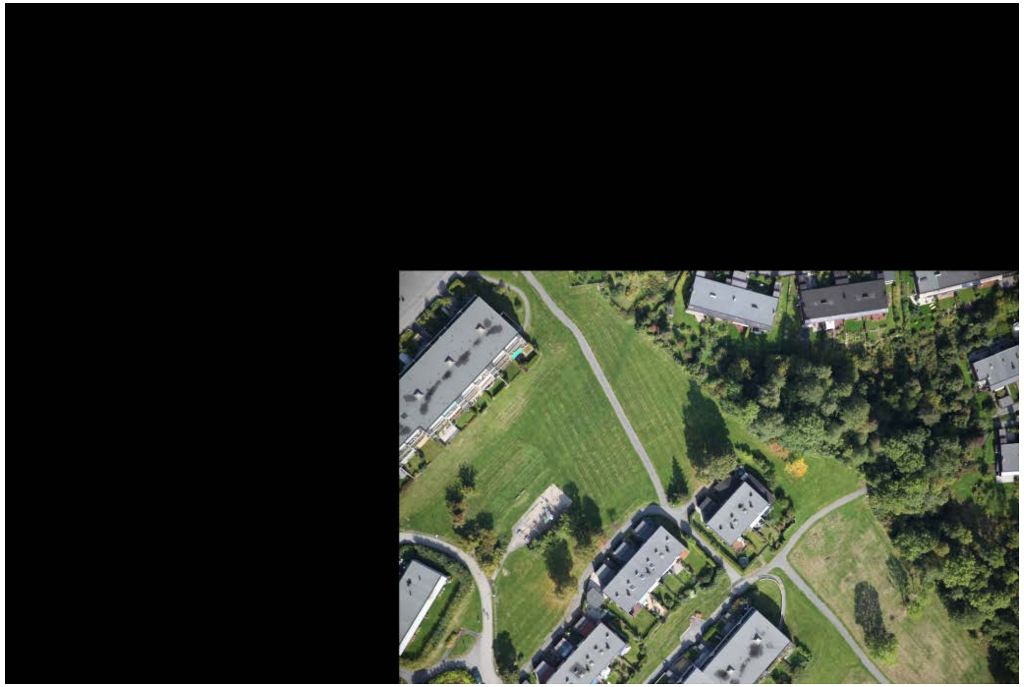




- Represent the images in common coordinates (Note the additional translation!)
  - OpenCV
```
#include "opencv2/calib3d.hpp"
cv::warpPerspective(img1, img2, H, output_size);
```
  - Matlab
```
img2 = imwarp(img1,tform);
```

Image mosaicing





- Now we can compose the images

Overwriting

Blending with a ramp

Blending with a ramp + histogram equalization

SVD

Singular Value Decomposition

The singular value decomposition of a real $m \times n$ matrix $\mathbf{A}$ is a factorization $\mathbf{A} = \mathbf{USV}^T$

Here $\mathbf{U}$ is a orthogonal $m \times m$ matrix, $\mathbf{V}$ is a orthogonal $n \times n$ matrix and $\mathbf{S}$ is a real positive diagonal $m \times n$ matrix

The diagonal entries of $\mathbf{S} = \mathrm{diag}\left(s_1, \dots, s_{\min(m,n)}\right)$ are known as the singular values of $\mathbf{A}$ and the columns of $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_m]$ and $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_n]$are known as the left and right singular vectors of $\mathbf{A}$ respectively

The nullspace of $\mathbf{A}$ is the span of the right singular vectors $\mathbf{v}_i$ that corresponds to a zero singular value s_i (or does not have a corresponding singular value)

How to use

- Matlab
  ```
  [U,S,V] = svd(A);
  ```
  Right singular vectors are columns in V

- OpenCV
  ```
  cv::SVD::compute(A, S, U, Vtranspose, cv::SVD::FULL_UV);
  ```
  Right singular vectors are rows in `Vtranspose`

- Eigen
  ```
  Eigen::JacobiSVD<Eigen::MatrixXd> svd(A, Eigen::ComputeFullU | Eigen::ComputeFullV);
  ```
  Right singular vectors are columns in `svd.matrixV()`

SVD

Singular Value Decomposition

The singular value decomposition of a real $m \times n$ matrix $\mathbf{A}$ is a factorization $\mathbf{A} = \mathbf{USV}^T$

Here $\mathbf{U}$ is a orthogonal $m \times m$ matrix, $\mathbf{V}$ is a orthogonal $n \times n$ matrix and $\mathbf{S}$ is a real positive diagonal $m \times n$ matrix

The diagonal entries of $\mathbf{S} = \mathrm{diag}\left(s_1, \dots, s_{\min(m,n)}\right)$ are known as the singular values of $\mathbf{A}$ and the columns of $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_m]$ and $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_n]$are known as the left and right singular vectors of $\mathbf{A}$ respectively

The nullspace of $\mathbf{A}$ is the span of the right singular vectors $\mathbf{v}_i$ that corresponds to a zero singular value s_i (or does not have a corresponding singular value)

Applications of SVD

Solving homogeneous linear equations like

$$\mathbf{Ah} = \mathbf{0}$$

Method

For theoretical problems, $\mathbf{h} \in \mathrm{null}(\mathbf{A})$ so $\mathbf{h}$ is a linear combination of the right singular vectors $\mathbf{v}_i$ that correspond to a zero singular value s_i

$$\mathbf{h} = \sum k_i \mathbf{v}_i \,; \quad k_i \in \mathbb{R}, s_i = 0 \text{ (or missing)}$$

For practical problems, the presence of noise force us to expand the solution by including those right singular vectors that correspond to small singular values $s_i \approx 0$

$$\mathbf{h} = \sum k_i \mathbf{v}_i \,; \quad k_i \in \mathbb{R}, s_i \approx 0 \text{ (or missing)}$$

SVD

Example

$$\mathbf{Ax} = \mathbf{0}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

From $[\mathbf{U}, \mathbf{S}, \mathbf{V}] = svd(\mathbf{A})$ we get

$$\mathbf{U} = \begin{bmatrix} -0.3863 & -0.9224 \\ -0.9224 & 0.3863 \end{bmatrix} \quad \mathbf{S} = \begin{bmatrix} 9.5080 & 0 & 0 \\ 0 & 0.7729 & 0 \end{bmatrix}$$

$$\mathbf{V} = \begin{bmatrix} -0.4287 & 0.8060 & 0.4082 \\ -0.5663 & 0.1124 & -0.8165 \\ -0.7039 & -0.5812 & 0.4082 \end{bmatrix}$$

From this we see that $\mathbf{A}$ has:

- 2 left singular vectors

$$\mathbf{u}_1 = \begin{bmatrix} -0.3863 \\ -0.9224 \end{bmatrix} \quad \mathbf{u}_2 = \begin{bmatrix} -0.9224 \\ 0.3863 \end{bmatrix}$$

- 2 nonzero singular values

$$s_1 = 9.5080 \quad s_2 = 0.7729$$

- 3 right singular vectors

$$\mathbf{v}_1 = \begin{bmatrix} -0.4287 \\ -0.5663 \\ -0.7039 \end{bmatrix} \quad \mathbf{v}_2 = \begin{bmatrix} 0.8060 \\ 0.1124 \\ -0.5812 \end{bmatrix} \quad \mathbf{v}_3 = \begin{bmatrix} 0.4082 \\ -0.8165 \\ 0.4082 \end{bmatrix}$$

Since $\mathbf{v}_3$ does not have a corresponding singular value, $\mathbf{x} = \mathbf{v}_3$ is a non-trivial solution to $\mathbf{Ax} = \mathbf{0}$ and $\mathbf{x} = k \cdot \mathbf{v}_3;\ k \in \mathbb{R}\backslash\{0\}$ is the family of all non-trivial solutions

SVD

Example

$$\mathbf{Ax} = \mathbf{0}$$

$$\begin{bmatrix} 1.0792 & 2.0656 & 3.0849 \\ 4.0959 & 5.0036 & 6.0934 \\ 1.0679 & 2.0743 & 3.0655 \\ 4.0758 & 5.0392 & 6.0171 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

This time singular value decomposition give us the following singular values and right singular vectors:

$$s_1 = 13.6295 \quad s_2 = 1.0849 \quad s_3 = 0.0506$$

$$\mathbf{v}_1 = \begin{bmatrix} -0.4336 \\ -0.5635 \\ -0.7032 \end{bmatrix} \quad \mathbf{v}_2 = \begin{bmatrix} -0.8103 \\ -0.0975 \\ 0.5778 \end{bmatrix} \quad \mathbf{v}_3 = \begin{bmatrix} 0.3942 \\ -0.8203 \\ 0.4143 \end{bmatrix}$$

This time all right singular vectors correspond to a non-zero singular value, so the equation does not have any non-trivial solutions!

If this equation came from a practical problem, instead of looking for solutions to $\mathbf{Ax} = \mathbf{0}$, we might be looking for the $\mathbf{x}$ that minimize $||\mathbf{Ax}||$

Since $s_1 \not\approx 0, s_2 \not\approx 0, s_3 \approx 0$, we would conclude that $\mathbf{x} = \mathbf{v}_3$ solves the equation in a least-squares sense

Check:

$$\mathbf{Av}_3 = \begin{bmatrix} 0.0091 \\ 0.4288 \\ -0.0105 \\ -0.0341 \end{bmatrix}$$

Summary

- Homography $\mathbf{H}\tilde{\mathbf{u}} = \tilde{\mathbf{u}}'$
$$\mathbf{H} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$$

- Automatic point-correspondences

- Wrong correspondences are common

- RANSAC estimation
  - Basic DLT (Direct Linear Transform) on 4 random correspondences
  - Inliers determined from the reprojection error $\epsilon_i = d(\mathbf{H}\mathbf{u}_i, \mathbf{u'}_i) + d(\mathbf{u}_i, \mathbf{H}^{-1}\mathbf{u'}_i)$

- Improve estimate by normalized DLT on inliers or iterative methods for an even better estimate

- Additional reading
  - Szeliski: 6.1.1 – 6.1.3