Homography Computation

\[
\lambda \begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}
\]

\[
v_x = \frac{h_{11}u_x + h_{12}u_y + h_{13}}{h_{31}u_x + h_{32}u_y + h_{33}}
\]

\[
v_y = \frac{h_{21}u_x + h_{22}u_y + h_{23}}{h_{31}u_x + h_{32}u_y + h_{33}}
\]
Homography Computation

\[
\lambda \begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}
\]

\[
v_x = \frac{h_{11}u_x + h_{12}u_y + h_{13}}{h_{31}u_x + h_{32}u_y + h_{33}}
\]

\[
v_y = \frac{h_{21}u_x + h_{22}u_y + h_{23}}{h_{31}u_x + h_{32}u_y + h_{33}}
\]

\[
\begin{align*}
h_{11}u_x + h_{12}u_y + h_{13} - h_{31}u_xv_x - h_{32}u_yv_x - h_{33}v_x &= 0 \\
h_{21}u_x + h_{22}u_y + h_{23} - h_{31}u_xv_y - h_{32}u_yv_y - h_{33}v_y &= 0
\end{align*}
\]
Homography Computation

\[
\lambda \begin{bmatrix}
  v_x \\
v_y \\
1
\end{bmatrix} =
\begin{bmatrix}
h_{11} & h_{12} & h_{13} \\
h_{21} & h_{22} & h_{23} \\
h_{31} & h_{32} & h_{33}
\end{bmatrix}
\begin{bmatrix}
u_x \\
u_y \\
1
\end{bmatrix}
\]

\[
v_x = \frac{h_{11}u_x + h_{12}u_y + h_{13}}{h_{31}u_x + h_{32}u_y + h_{33}}
\]

\[
v_y = \frac{h_{21}u_x + h_{22}u_y + h_{23}}{h_{31}u_x + h_{32}u_y + h_{33}}
\]

\[
\begin{align*}
h_{11}u_x + h_{12}u_y + h_{13} - h_{31}u_xv_x - h_{32}u_yv_x - h_{33}v_x &= 0 \\
h_{21}u_x + h_{22}u_y + h_{23} - h_{31}u_xv_y - h_{32}u_yv_y - h_{33}v_y &= 0
\end{align*}
\]

\[
\begin{bmatrix}
u_x & u_y & 1
\end{bmatrix}
\begin{bmatrix}
-u_xv_x & -u_yv_x & -v_x \\
u_x & -u_yv_y & -v_y
\end{bmatrix}
\begin{bmatrix}
X
\end{bmatrix} = 0
\]

\[2\times9\]
Linear System for Homography Matrix

\[
\begin{bmatrix}
  u_x & u_y & 1 \\
  u_x & u_y & -v_x & -v_y & -v_x & -v_y \\
\end{bmatrix}
\begin{bmatrix}
  h_{11} \\
  h_{12} \\
  h_{13} \\
  h_{21} \\
  h_{22} \\
  h_{23} \\
\end{bmatrix} = \begin{bmatrix} x \end{bmatrix}
\]
How Many Correspondences?

What is minimum $m$?
\[ \mathcal{I}_1 \]

\[
\begin{align*}
\{ \mathbf{v}_1 \leftrightarrow \mathbf{u}_1 \\ \mathbf{v}_2 \leftrightarrow \mathbf{u}_2 \\ \mathbf{v}_3 \leftrightarrow \mathbf{u}_3 \\ \mathbf{v}_4 \leftrightarrow \mathbf{u}_4 \} \rightarrow H
\end{align*}
\]

Homography computation

\[ \mathcal{I}_2 \]

\[
\begin{bmatrix}
\mathbf{u}_x' & \mathbf{u}_y' & 1 \\
\mathbf{u}_x^2 & \mathbf{u}_y^2 & 1 \\
\mathbf{u}_x^3 & \mathbf{u}_y^3 & 1 \\
\mathbf{u}_x^4 & \mathbf{u}_y^4 & 1 \\
\end{bmatrix}
\begin{bmatrix}
\mathbf{h}_1 \\
\mathbf{h}_2 \\
\mathbf{h}_3 \\
0
\end{bmatrix}
= 
\begin{bmatrix}
\mathbf{0} \\
\mathbf{0} \\
\mathbf{0} \\
\mathbf{0}
\end{bmatrix}
\]
\[ \begin{align*} I_1 & \quad \begin{cases} v_1 \leftrightarrow u_1 \ \\ v_2 \leftrightarrow u_2 \ \\ v_3 \leftrightarrow u_3 \ \\ v_4 \leftrightarrow u_4 \end{cases} \rightarrow H \end{align*} \]

Homography computation
$\mathcal{I}_1$

$$\begin{align*}
\begin{cases}
\text{Homography computation} \\
\v_1 &\leftrightarrow \mathbf{u}_1 \\
\v_2 &\leftrightarrow \mathbf{u}_2 \\
\v_3 &\leftrightarrow \mathbf{u}_3 \\
\v_4 &\leftrightarrow \mathbf{u}_4
\end{cases} \rightarrow \mathbf{H}
\end{align*}$$

$\mathcal{I}_2$

$$\begin{align*}
[u, d, v] &= \text{svd}(A) \\
X &= v(:, \text{end})/v(\text{end}, \text{end}) \\
H &= \text{reshape}(X, 3, 3)^\top
\end{align*}$$
Fun with Homography

The image can be rectified as if it is seen from top view.
Fun with Homography

RectificationViaHomography.m

\[ u = [u_1'; u_2'; u_3'; u_4']; \]

\[ v = [v_1'; v_2'; v_3'; v_4']; \]

\% Need at least non-colinear four points

\[ H = \text{ComputeHomography}(v, u); \]

\[ \text{im}_\text{warped} = \text{ImageWarping}(\text{im}, \text{inv}(H)); \]
Fun with Homography

RectificationViaHomography.m

\[
\begin{bmatrix}
1
\end{bmatrix}
\begin{bmatrix}
\lambda

\end{bmatrix} = H^{-1}
\]

\[ u = [u_1'; u_2'; u_3'; u_4']; \]
\[ v = [v_1'; v_2'; v_3'; v_4']; \]

% Need at least non-colinear four points
H = ComputeHomography(v, u);

im_warped = ImageWarping(im, inv(H));

ImageWarping.m

\[
\begin{align*}
\begin{bmatrix}
\lambda
\end{bmatrix} &= H^{-1} \\
u_x &= H(1,1)*v_x + H(1,2)*v_y + H(1,3) \\
u_y &= H(2,1)*v_x + H(2,2)*v_y + H(2,3) \\
u_z &= H(3,1)*v_x + H(3,2)*v_y + H(3,3); \\
\end{align*}
\]

u_x = u_x./u_z;

u_y = u_y./u_z;

im_warped(:,:,1) = reshape(interp2(im(:,:,1), u_x(:), u_y(:)), [h, w]);
im_warped(:,:,2) = reshape(interp2(im(:,:,2), u_x(:), u_y(:)), [h, w]);
im_warped(:,:,3) = reshape(interp2(im(:,:,3), u_x(:), u_y(:)), [h, w]);
im_warped = uint8(im_warped);

Cf) ImageWarpingEuclidean.m

\[
\begin{align*}
\begin{bmatrix}
u_x \\
v_y \\
1
\end{bmatrix} &= H^{-1} \\
u_x &= H(1,1)*v_x + H(1,2)*v_y + H(1,3) \\
u_y &= H(2,1)*v_x + H(2,2)*v_y + H(2,3); \\
\end{align*}
\]

u_x = u_x./u_z;

u_y = u_y./u_z;

im_warped(:,:,1) = reshape(interp2(im(:,:,1), u_x(:), u_y(:)), [h, w]);
im_warped(:,:,2) = reshape(interp2(im(:,:,2), u_x(:), u_y(:)), [h, w]);
im_warped(:,:,3) = reshape(interp2(im(:,:,3), u_x(:), u_y(:)), [h, w]);
im_warped = uint8(im_warped);
Fun with Homography
Fun with Homography
Local Patch
Local Patch (Orientation)
Local Visual Descriptor

Desired properties:
- Repeatability: the same point is repeatedly detected.
- Discriminativity: the point is unique.
Local Visual Descriptor

Desired properties:
- Repeatability: the same point is repeatedly detected.
- Discriminativity: the point is unique.
- Orientation aware
Local Scale Invariant Feature Transform (SIFT)

SIFT automatically finds the optimal scale of feature point and its orientation.

**Desired properties:**
- Repeatability: the same point is repeatedly detected.
- Discriminativity: the point is unique.
- Orientation aware
Local Scale Invariant Feature Transform (SIFT)
Local Scale Invariant Feature Transform (SIFT)
Local Scale Invariant Feature Transform (SIFT)
Local Scale Invariant Feature Transform (SIFT)
Nearest Neighbor Search

Feature match candidates

d_1, d_2, d_3, d_4, d_5

descriptor1 - descriptor2
Nearest Neighbor Search

**Discriminativity**: how is the feature point unique?
Nearest Neighbor Search w/ Ratio Test

Discriminativity: how is the feature point unique?

\[
\frac{d_1}{d_2} < 0.7
\]

Feature match candidates
Nearest Neighbor Search w/o Ratio Test

Left image ➔ right image
Nearest Neighbor Search w/ Ratio Test

Left image ➔ right image
Nearest Neighbor Search w/o Ratio Test

Left image <-> right image
Nearest Neighbor Search w/ Ratio Test

Left image ← right image
Bi-directional Consistency Check

Consistency: would a feature match correspond to each other?

Feature match candidates
Bi-directional Consistency Check
RANSAC: Random Sample Consensus: Linear Least Squares
Recall: Line Fitting ($Ax=b$)
Line fitting error:

\[ E = \left( ex_1 - fy_1 - g \right)^2 + \cdots + \left( ex_N - fy_N - g \right)^2 \]

\[ = \sum_{i=1}^{N} \left( ex_i - fy_i - g \right)^2 \]

Perpendicular distance

Quadratic magnification of error of outliers

\[ E_i = |ex_i + fy_i + g| \]
Outlier rejection strategy:

To find the best line that explains the maximum number of points.
Outlier rejection strategy:

To find the best line that explains the maximum number of points.

Assumptions:
1. Majority of good samples agree with the underlying model (good apples are same and simple).
2. Bad samples does not consistently agree with a single model (all bad apples are different and complicated).
RANSAC: Random Sample Consensus
RANSAC: Random Sample Consensus

1. Random sampling
RANSAC: Random Sample Consensus

1. Random sampling
2. Model building
RANSAC: Random Sample Consensus

1. Random sampling
2. Model building
3. Thresholding
1. Random sampling
2. Model building
3. Thresholding
4. Inlier counting

RANSAC: Random Sample Consensus

# of inliers: 7
RANSAC: Random Sample Consensus

1. Random sampling
2. Model building
3. Thresholding
4. Inlier counting
RANSAC: Random Sample Consensus

1. Random sampling
2. Model building
3. Thresholding
4. Inlier counting
RANSAC: Random Sample Consensus

1. Random sampling
2. Model building
3. Thresholding
4. Inlier counting
RANSAC: Random Sample Consensus

1. Random sampling
2. Model building
3. Thresholding
4. Inlier counting

# of inliers: 10
RANSAC: Random Sample Consensus

1. Random sampling
2. Model building
3. Thresholding
4. Inlier counting
RANSAC: Random Sample Consensus

1. Random sampling
2. Model building
3. Thresholding
4. Inlier counting
RANSAC: Random Sample Consensus

1. Random sampling
2. Model building
3. Thresholding
4. Inlier counting
1. Random sampling

2. Model building

3. Thresholding

4. Inlier counting

# of inliers: 23
Maximum number of inliers

RANSAC: Random Sample Consensus
Required number of iterations with \( p \) success rate:
Required number of iterations with $p$ success rate:

Probability of choosing an inlier:

$$w = \frac{\# \text{ of inliers}}{\# \text{ of samples}}$$
Required number of iterations with $p$ success rate:

Probability of choosing an inlier: $w = \frac{\# \text{of inliers}}{\# \text{of samples}}$

Probability of building a correct model: $w^n$, where $n$ is the number of samples to build a model.
Required number of iterations with $p$ success rate:

Probability of choosing an inlier: $w = \frac{\text{# of inliers}}{\text{# of samples}}$

Probability of building a correct model: $w^n$ where $n$ is the number of samples to build a model.

Probability of not building a correct model during $k$ iterations: $\left(1 - w^n\right)^k$
Probability of choosing an inlier: \[ w = \frac{\text{# of inliers}}{\text{# of samples}} \]

Probability of building a correct model: \[ w^n \] where \( n \) is the number of samples to build a model.

Probability of not building a correct model during \( k \) iterations: \[ (1 - w^n)^k = 1 - p \] where \( p \) is desired RANSAC success rate.

Required number of iterations with \( p \) success rate:

\[ k = \frac{\log(1 - p)}{\log(1 - w^n)} \]
Required number of iterations with $p$ success rate:

$$k = \frac{\log(1-p)}{\log(1-w^n)}$$

where $w = \frac{\# \text{ of inliers}}{\# \text{ of samples}}$

Probability of choosing an inlier:

$$w = \frac{\# \text{ of inliers}}{\# \text{ of samples}}$$

Probability of building a correct model: $w^n$ where $n$ is the number of samples to build a model.

Probability of not building a correct model during $k$ iterations:

$$\left(1-w^n\right)^k = 1-p \quad \text{where } p \text{ is desired RANSAC success rate.}$$

$$k = \frac{\log(1-p)}{\log(1-w^n)}$$
\[ I_1 \rightarrow I_2 \]

\[
\begin{align*}
\{v_1 & \leftrightarrow u_1 \\
v_2 & \leftrightarrow u_2 \\
v_3 & \leftrightarrow u_3 \\
v_4 & \leftrightarrow u_4
\}
\rightarrow H
\end{align*}
\]

Homography computation
\[ I_1 \quad \sim \quad I_2 \]

\[
\begin{align*}
\{ v_1 \leftrightarrow u_1 \} \\
\{ v_2 \leftrightarrow u_2 \} \\
\{ v_3 \leftrightarrow u_3 \} \\
\{ v_4 \leftrightarrow u_4 \}
\end{align*}
\]

\[ \rightarrow H \rightarrow \]

\[ v = Hu \]

Inlier evaluation

Homography computation
\[ \mathcal{I}_2(v) = \mathcal{I}_1(Hu) \]

where

\[ v = Hu \]
If the correspondence is bad, the computed homography will fit the four points still perfectly, but how do we know it is wrong?
If the correspondence is bad, it has no prediction power!
# of inliers: 16 out of 1865
# of inliers: 16 out of 1865
# of inliers: 36 out of 1865
# of inliers: 36 out of 1865
# of inliers: 57 out of 1865
# of inliers: 216 out of 1865
Image Panorama (Cylindrical Projection)

First camera:
Point on cylindrical surface: \([h, \theta]\)
\[\leftrightarrow\] Point in 3D space: \([f \cos(\theta), h, f \sin(\theta)]\)
\[\leftrightarrow\] Point in image coordinate: \(K[f \cos(\theta), h, f \sin(\theta)]^T\)
Image Panorama (Cylindrical Projection)
Image Panorama (Cylindrical Projection)

Second camera:
Point on cylindrical surface: \([h, \theta]\)
\[\leftrightarrow\] Point in 3D space: \([f \cos(\theta), h, f \sin(\theta)]\)
\[\leftrightarrow\] Point in image coordinate: \(KR[f \cos(\theta), h, f \sin(\theta)]^T\)
where \(R\) is given by \(R = K^{-1}HK\)
Image Panorama (Cylindrical Projection)
Image Panorama (Cylindrical Projection)