Homography

\[
\begin{bmatrix}
X_h \\
Y_h \\
W_h
\end{bmatrix} =
\begin{bmatrix}
h_{oo} & h_{o1} & h_{o2} \\
h_{10} & h_{11} & h_{12} \\
h_{20} & h_{21} & h_{22}
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
1
\end{bmatrix}
\]

Make this h22 then rewrite

\[
X' = \frac{h_{oo}X + h_{o1}Y + h_{o2}}{h_{20}X + h_{21}Y + h_{22}}
\]

\[
Y' = \frac{h_{10}X + h_{11}Y + h_{12}}{W_h}
\]

Not linear!

\[
\frac{h_{oo}X + h_{o1}Y + h_{o2}}{h_{20}X + h_{21}Y + h_{22}} = X'
\]

\[
h_{oo}X + h_{o1}Y + h_{o2} = X'(h_{20}X + h_{21}Y + h_{22})
\]

\[
X \text{ Residual: } \left(h_{oo}X + h_{o1}Y + h_{o2}\right) - X'(h_{20}X + h_{21}Y + h_{22})
\]

\[
= (h_{oo}X + h_{o1}Y + h_{o2}) - (h_{20}X'y' + h_{21}Y'y' + h_{22}Y')
\]

\[
h_{10}X + h_{11}Y + h_{12} = Y'
\]

\[
h_{10}X + h_{11}Y + h_{12} = Y'(h_{20}X + h_{21}Y + h_{22})
\]

\[
y_{\text{residual}}: h_{10}X + h_{11}Y + h_{12} - (h_{20}X'y' + h_{21}Y'y' + h_{22}Y')
\]

We had: \( X' = \frac{X_h}{W_h} \) now \( X_h = X'W_h \)

\( y' = \frac{Y_h}{W_h} \) now \( Y_h = Y'W_h \)
Residuals: $X_h$

$$
X = \begin{bmatrix} h_{02} x + h_{01} y + h_{00} - h_{21} x' y - h_{22} x' \\
 h_{10} x + h_{11} y + h_{12} - h_{20} x' y' - h_{21} y' - h_{22} y' \\
\end{bmatrix}
$$

$$\min \| Ah - b \|^2$$

$$
\begin{bmatrix}
x, y, 1 & 0 & 0 & 0 & 0 & x x' & x y' & x' \\
0 & 0 & 0 & 0 & 0 & x x' & x y' & x' \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
X_h, y_h, 1 & 0 & 0 & 0 & 0 & x x' & x y' & x' \\
0 & 0 & 0 & 0 & 0 & x x' & x y' & x' \\
\end{bmatrix}
$$

$$
\begin{bmatrix}
h_{00} \\
h_{01} \\
h_{02} \\
h_{10} \\
h_{11} \\
h_{12} \\
h_{20} \\
h_{21} \\
h_{22} \\
\end{bmatrix}
$$

$$
\begin{bmatrix}
0 \\
0 \\
\end{bmatrix}
$$
Minimizing $\|Ah - o\|^2$

Trivial: $h = 0$

Solution: constrain $\|h\| = 1$

Min $\|Ah - o\|^2$ s.t. $\|h\| = 1$

$\min \|Ah\|^2$ s.t. $\|h\| = 1$

$\|Ah\|^2 = (Ah)^T(Ah) = h^TA^TAh$

Singular Value decomposition: $A = U\Sigma V^T$

$\Sigma V^T h = h^T U \Sigma V^T V^T h$

$= h^T V \Sigma V^T h$

$\|h\| = 1$

$\|h\|^2 = 1$

$V_i^T V_i = 1$

$V_i^T V_j = 0$

$h = v_3$
In Practice:
1. Compute SVD of $A$.
2. Find index of smallest $\sigma_i$ in $\Sigma$.
3. Take the $i$th column of $V$ (i.e., row of $VT$) as solution $h$. 
RANSAC: The key Idea:

Points that fit the "true" model will agree.
Points that don't will not agree on some other, wrong model.

An idea: Generate all possible lines, see which one has the most agreeing points.
Runtime: $O(\infty)$

A better idea: all possible lines through 2 points:

for $p_1, p_2 \in P$: 
RANSAC - Random Sample Consensus

Motivation: Fitting geometric transformations with least squares works well if there are no outliers.

Fitting a transformation is a model fitting problem:

Given matches, find a transformation

Analogy: Given points, find a line.

When does least squares work well?

When errors are small and random.
(Math answer): When errors are i.i.d Gaussian with zero mean.

Outliers have an influence effect because of the square in the objective

\[
\min_x \|Ax-b\| = \min_y \|Ax-b\|/\|Ax-b\|
\]
RANSAC: Algorithm

\[
\text{for } i = 0 \text{ to } K:\n\]
\[
d; \in \text{ random data points} \\
M; \leftarrow \text{fit} \cdot \text{model}(d;)
\]
\[
inlier \cdot count \leftarrow \sum_{x; , y; \in D} \mathbb{I}(M(x; ) - y; \leq \delta) \quad \# \text{data points in agreement with } M;
\]
\[
\begin{cases}
\text{if inlier} \cdot \text{count } < \text{best} \cdot \text{count} : \\
\quad \text{best} \cdot \text{count} \leftarrow \text{inlier} \cdot \text{count}
\end{cases}
\]
\[
\quad \text{best} \cdot M \leftarrow M;
\quad \text{best} \cdot \text{data} \leftarrow \{(x; , y; ) : |M(x; ) - y; | < \delta\}
\]
\[
M \cdot \text{final} \leftarrow \text{fit} \cdot \text{model}(\text{best} \cdot \text{data})
\]

Parameters

- \(K\) \# iterations (hypotheses)
- \(S\) \# data points needed to fit a model
- \(\delta\) inlier threshold

Choosing parameter values:

- \(\delta\): based on expected inlier noise. Common case:
  
  Assume Gaussian with variance \(\sigma^2\)
  
  Let \(\delta = 1.0 \sigma^2\)

- \(S\): based on specific problem:
  
  - Linear regression - 2 \((x, y)\) points
  - Translation fitting - one match \((x, y) \mapsto (x', y')\)
  - Affine - 3 matches
  - Homography - 4 matches
  - Ellipse (why not?) - 3 \((x, y)\) points
Choosing Parameter Values (cont):

\( K \text{ (# iterations)} \) - suppose we want to find a set of \( S \) inliers with probability \( \geq P \). Assume we can estimate the inlier ratio:

\[
\gamma = \frac{\# \text{ inliers}}{\# \text{ data points}}
\]

In one hypothesis,

\[
P(\text{choose all inliers}) = r^S
\]

\[
P(\text{at least one outlier}) = 1 - r^S
\]

Over \( K \) trials,

\[
P(\text{at least one outlier all } K \text{ trials}) = (1 - r^S)^K
\]

\[
P = P(\text{no outliers in at least one trial}) = 1 - (1 - r^S)^K
\]

What \( K \) do I need to make \( P(\text{success}) \geq P \)?

\[
P \geq 1 - (1 - r^S)^K
\]

\[
1 - P \geq (1 - r^S)^K
\]

\[
\log(1 - P) \geq K \log(1 - r^S)
\]

\[
\frac{\log(1 - P)}{\log(1 - r^S)} \geq K
\]

Sanity checks:
Lower prob. of success \( \rightarrow \) fewer iterations
More points to fit a model \( \rightarrow \) more iterations
Forward Warping

\[ I_1(x, y) \quad I_{10}(x', y') \]

\[ \begin{array}{c}
(x, y) \\
T \\
(y') \\
(x', y')
\end{array} \]

for \( x, y \) in \( I_1 \):

\[ x', y' = T(x, y) \]

\[ I_{10}(x', y') = I(x, y) \]

Problem: What if \( x', y' \) are floats?

Possible answer: "Splat" \( I(x, y) \) to multiple pixels in \( I(x', y') \)

Issues with:

- Scale (e.g., \( T \) is a 16x uniform scale)
- Holes remaining after splatting
Linear Interpolation

\[ y = y_1(x_2 - x) + y_2(x - x_1) \]

If \( x_1 = 0, x_2 = 1 \),

\[ y_1(1-x) + y_2(x) \]
Inverse Warping

For each \((x', y')\) in \(I'_1\),
\[
x, y = T^{-1}(x', y')
\]
\[
I'_1[x, y] = \text{interpolate}(I_1, T^{-1}(x', y'))
\]

Bilinear Interpolation - placing a tent filter at non-integer coordinates!

Interpretations:
- A tent filter at non-integer cords
- Weights determined by areas of rectangles at opposite corners
- Interpolate linearly on two sides, then interpolate linearly between the two

\[
I'(x', y') = \frac{(x - x_1)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)} I(x_1, y_1) + \frac{(x_2 - x)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)} I(x_2, y_1) + \frac{(x - x_1)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)} I(x_1, y_2) + \frac{(x_2 - x)(y_2 - y_2)}{(x_2 - x_1)(y_2 - y_1)} I(x_2, y_2)
\]