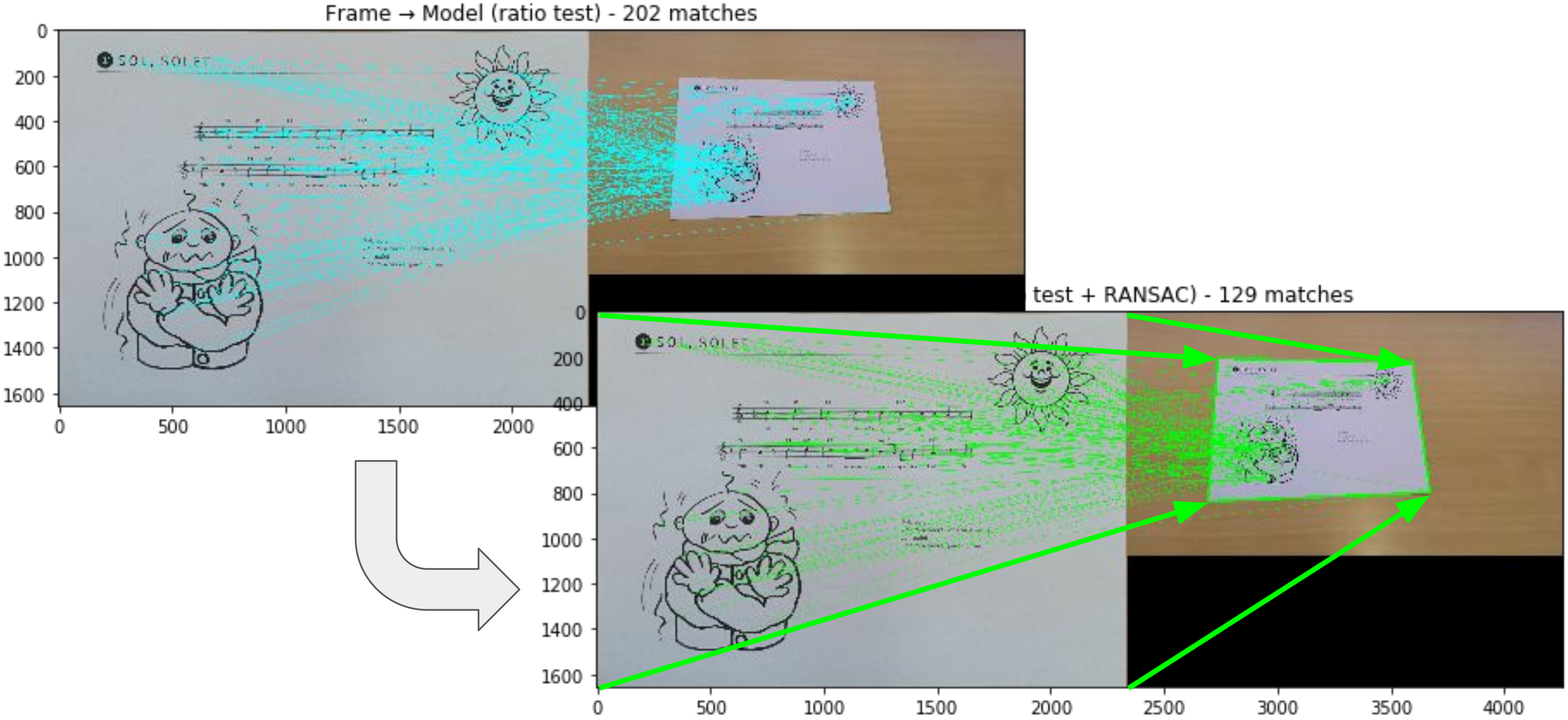
MLRF Lecture 03

J. Chazalon, LRDE/EPITA, 2020

Homography estimation
Geometric validation

Lecture 03 part 05

So we want to recover H from keypoint matches



Recover the parameters of a perspective transform

From our matched points we want to estimate A that maps from x to x'

xH = x'

$$\begin{bmatrix} \tilde{x}' \\ \tilde{y}' \\ \tilde{w}' \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{w} \end{bmatrix}$$

How many degrees of freedom?

>> 8 (**<u>not 9</u>** because $h_{22} = 1$ **OR** $||H|| = \Sigma h_{ij}^2 = 1$)

How many knowns do we get with one match mH = n?

>> 2

$$n_x = (h_{00}*m_x + h_{01}*m_y + h_{02}*m_w) / (h_{20}*m_x + h_{21}*m_y + h_{22}*m_w)$$
$$n_y = (h_{10}*m_x + h_{11}*m_y + h_{12}*m_w) / (h_{20}*m_x + h_{21}*m_y + h_{22}*m_w)$$

How many correspondences are needed?

Depends on the type of transform:

- How many for translation?
- For rotation?
- …
- For general projective transform?

Reminded: we have 2 knowns for each match

How many correspondences are needed?

Transformation	Matrix	# DoF	Min. # of independent matches required
translation	$\begin{bmatrix} \mathbf{I} \mid \mathbf{t} \end{bmatrix}_{2\times 3}$	2	1
rigid (Euclidean)	$\begin{bmatrix} \mathbf{R} \mid \mathbf{t} \end{bmatrix}_{2\times 3}$	3	2
similarity	$\begin{bmatrix} s\mathbf{R} \mid \mathbf{t} \end{bmatrix}_{2\times 3}$	4	2
affine	$\begin{bmatrix} \mathbf{A} \end{bmatrix}_{2\times 3}$	6	3
projective	$\begin{bmatrix} \tilde{\mathbf{H}} \end{bmatrix}_{3\times 3}$	8	4

from R.Szeliski

Enforcing 8 DOF

Approach 1: set $h_{22} = 1$

$$x' = \frac{h_{00}x + h_{01}y + h_{02}}{h_{20}x + h_{21}y + 1}$$

$$y' = \frac{h_{10}x + h_{11}y + h_{12}}{h_{20}x + h_{21}y + 1}$$

Approach 2: Impose unit vector constraint

$$x' = \frac{h_{00}x + h_{01}y + h_{02}}{h_{20}x + h_{21}y + h_{22}}$$

$$y' = \frac{h_{10}x + h_{11}y + h_{12}}{h_{20}x + h_{21}y + h_{22}}$$

subject to the constraint

$$h_{00}^2 + h_{01}^2 + h_{02}^2 + \cdots + h_{22}^2 = 1$$

Build an equation system to solve

Assuming $h_{22} = 1$ here:

$$x' = \frac{h_{00}x + h_{01}y + h_{02}}{h_{20}x + h_{21}y + 1}$$

Multiplying through by denominator:

$$(h_{20}x + h_{21}y + 1)\,x' = h_{00}x + h_{01}y + h_{02}$$

Rearrange:

$$h_{00}x + h_{01}y + h_{02} - h_{20}xx' + h_{21}yx' = x'$$

Same for y:

$$h_{10}x + h_{11}y + h_{12} - h_{20}xy' + h_{21}yy' = y'$$

Linear system with h_{22} = 1

$$
\begin{array}{cccc}
\mathbf{M} & \mathbf{a} & & \mathbf{b} \\
\begin{bmatrix}
x_0 & y_0 & 1 & 0 & 0 & 0 & -x_0x'_0 & -y_0x'_0 \\
0 & 0 & 0 & x_0 & y_0 & 1 & -x_0y'_0 & -y_0y'_0 \\
x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x'_1 & -y_1x'_1 \\
0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y'_1 & -y_1y'_1 \\
x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x'_2 & -y_2x'_2 \\
0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y'_2 & -y_2y'_2 \\
x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3x'_3 & -y_3x'_3 \\
0 & 0 & 0 & x_3 & y_3 & 1 & -x_3y'_3 & -y_3y'_3 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots
\end{bmatrix}
&
\begin{bmatrix}
h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \\ h_{22}
\end{bmatrix}
& = &
\begin{bmatrix}
x'_0 \\ y'_0 \\ x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \\ \vdots
\end{bmatrix}
\end{array}
$$

Use Linear Least Square to solve ***M a = b***

Still works if overdetermined: minimize squared error $|| b - M a ||^2$

$$|| b - M a ||^2 = (b - M a)^T(b - M a)$$

$$= b^Tb - a^TM^Tb - b^TMa + a^TM^TMa$$

$$= b^Tb - 2a^TM^Tb + a^TM^TMa$$

This is convex and minimized when gradient = 0.
So we take the derivative wrt a and set = 0.

$$-M^Tb + (M^TM)a = 0 \;\Leftrightarrow\; (M^TM)a = M^Tb \;\Leftrightarrow\; a = (M^TM)^{-1}M^Tb$$

Not always numerically stable though, and what if h_{22} = 0?

Build the equation system with **||H|| = 1**

||H|| = 1:

$$x' = \frac{h_{00}x + h_{01}y + h_{02}}{h_{20}x + h_{21}y + h_{22}}$$

Multiplying through by denominator: $$(h_{20}x + h_{21}y + h_{22})\,x' = h_{00}x + h_{01}y + h_{02}$$

Rearrange:

$$h_{00}x + h_{01}y + h_{02} - h_{20}xx' + h_{21}yx' - h_{22}x' = 0$$

Same for y:

$$h_{00}x + h_{01}y + h_{02} - h_{20}xy' + h_{21}yy' - h_{22}y' = 0$$

Linear system with **||H|| = 1**

M **a** **=** **0**

$$
\begin{bmatrix}
x_0 & y_0 & 1 & 0 & 0 & 0 & -x_0x'_0 & -y_0x'_0 & -x'_0 \\
0 & 0 & 0 & x_0 & y_0 & 1 & -x_0y'_0 & -y_0y'_0 & -y'_0 \\
x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x'_1 & -y_1x'_1 & -x'_1 \\
0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y'_1 & -y_1y'_1 & -y'_1 \\
x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x'_2 & -y_2x'_2 & -x'_2 \\
0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y'_2 & -y_2y'_2 & -y'_2 \\
x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3x'_3 & -y_3x'_3 & -x'_3 \\
0 & 0 & 0 & x_3 & y_3 & 1 & -x_3y'_3 & -y_3y'_3 & -y'_3 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots &
\end{bmatrix}
\begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \\ h_{22} \end{bmatrix}
=
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}
$$

Solve the system

Challenges:

- Overcomplete system
- Probably no exact solution because of noise

Solutions:

Use total least square with singular value decomposition (SVD)

[enough math here]

So: given enough matches, we get an estimate of H's parameters.

How reliable is the estimate? (Least square output)

(Example on fitting 2 parameters)

Perfect data ⇒ Everything is fine

but…

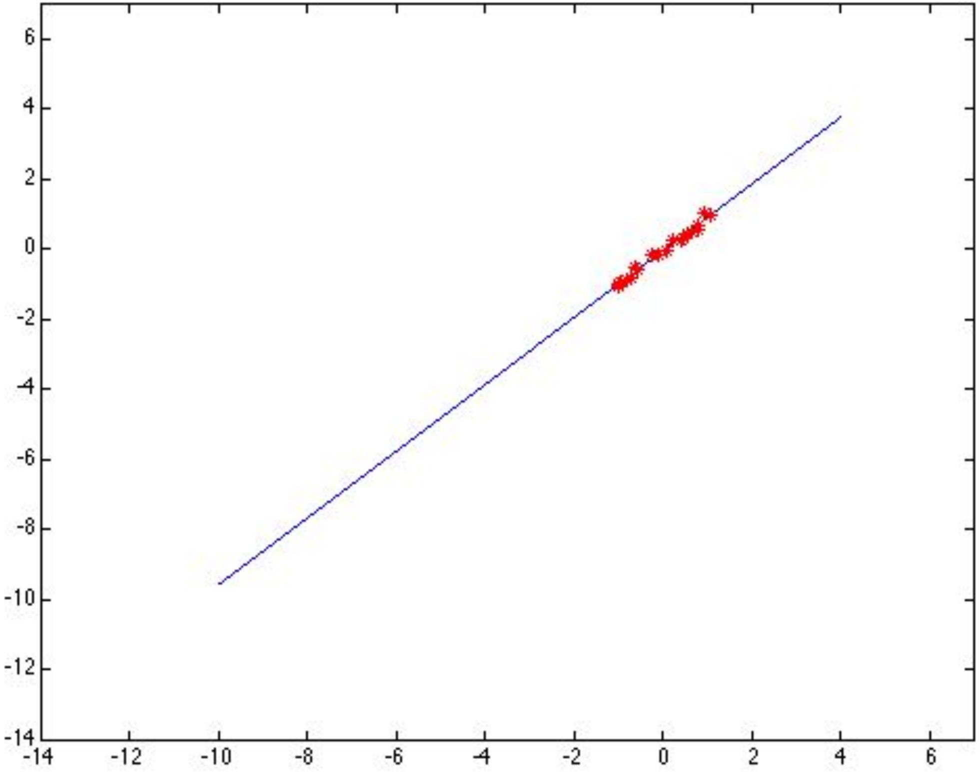


Image credit: JP Reddie

Is our data perfect?

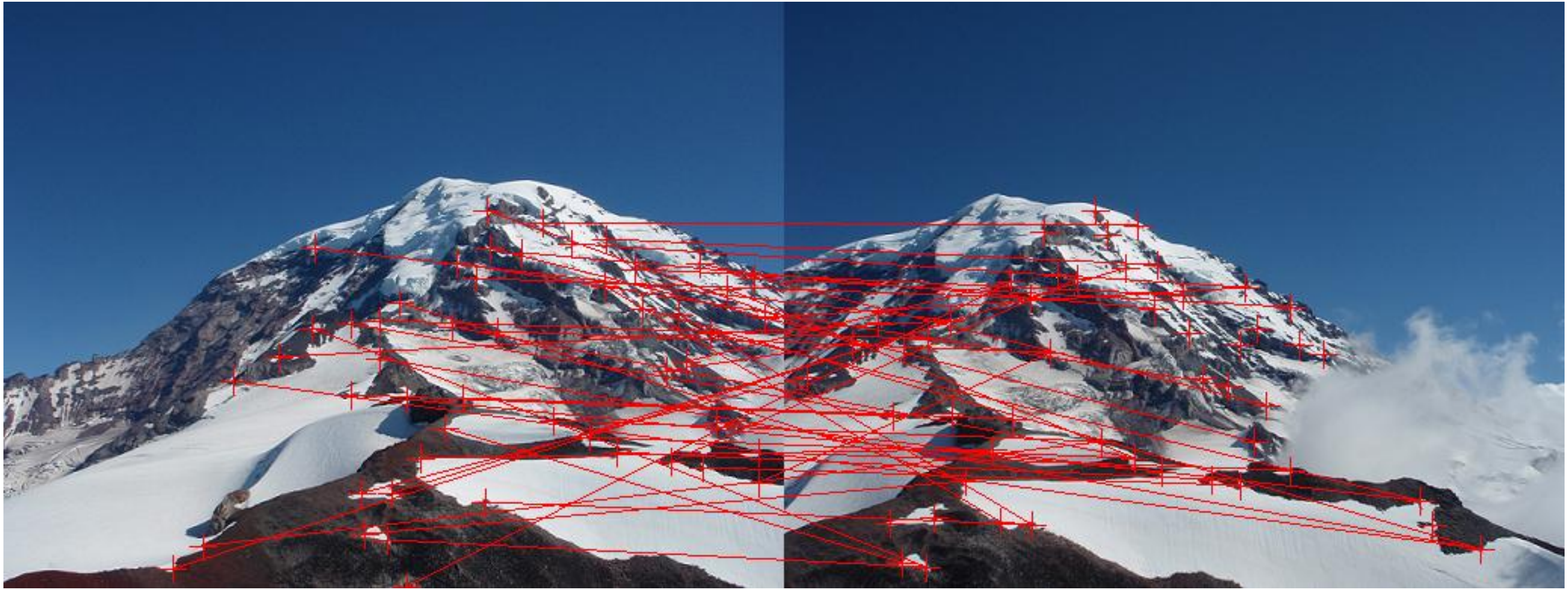


Image credit: JP Reddie

How reliable is the estimate? (Least square output)

(Example on fitting 2 parameters)

Error based on squared residual

Very scared of being wrong, even for just one point

Very bad at handling outliers in data

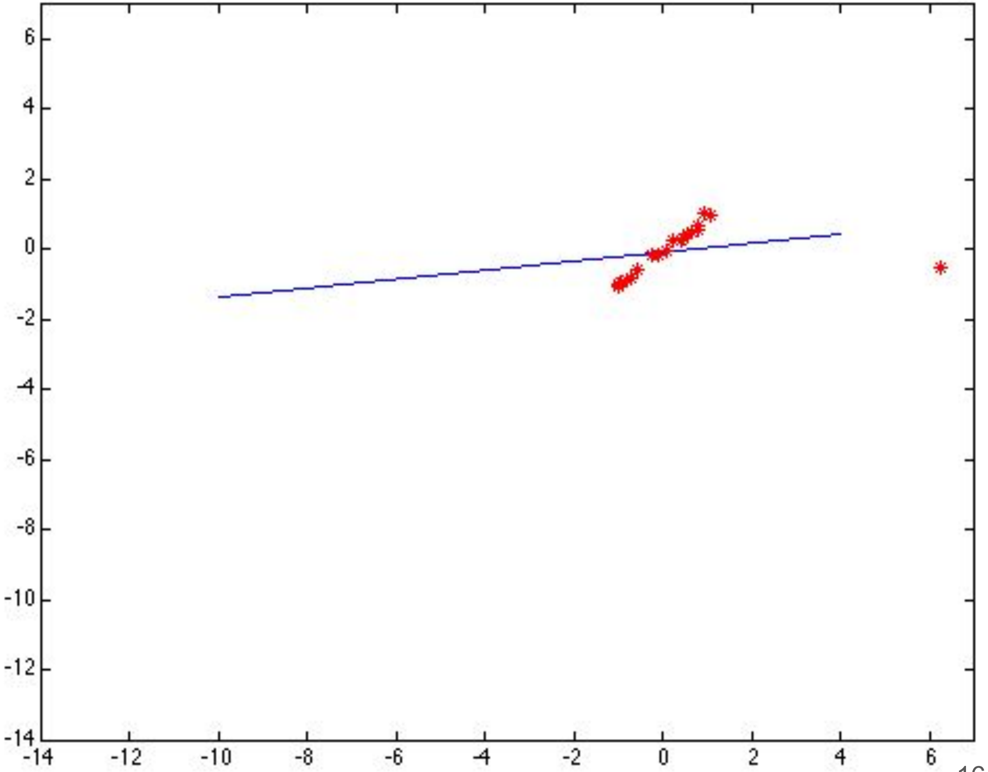


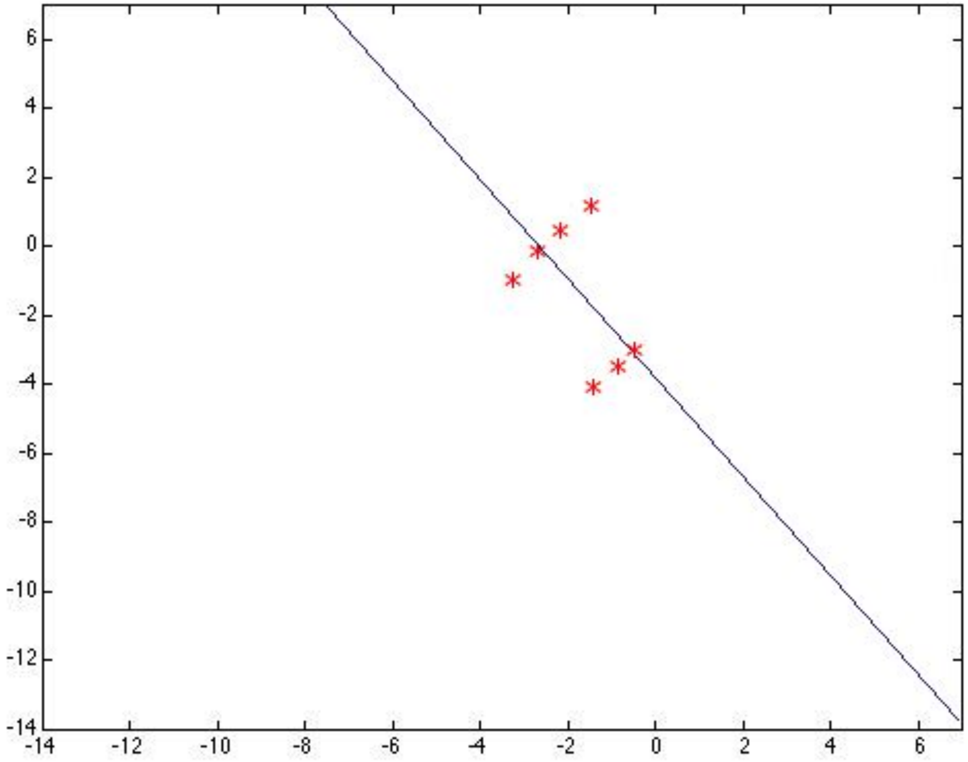
Image credit: JP Reddie

Even worse

(Example on fitting 2 parameters)

Multiple structures can also skew the results.

The fit procedure **implicitly assumes** there is **only one instance** of the model in the data.



Overcoming Least Square limitations

We need a robust estimation.

Approach: view estimation as a two-stage process:

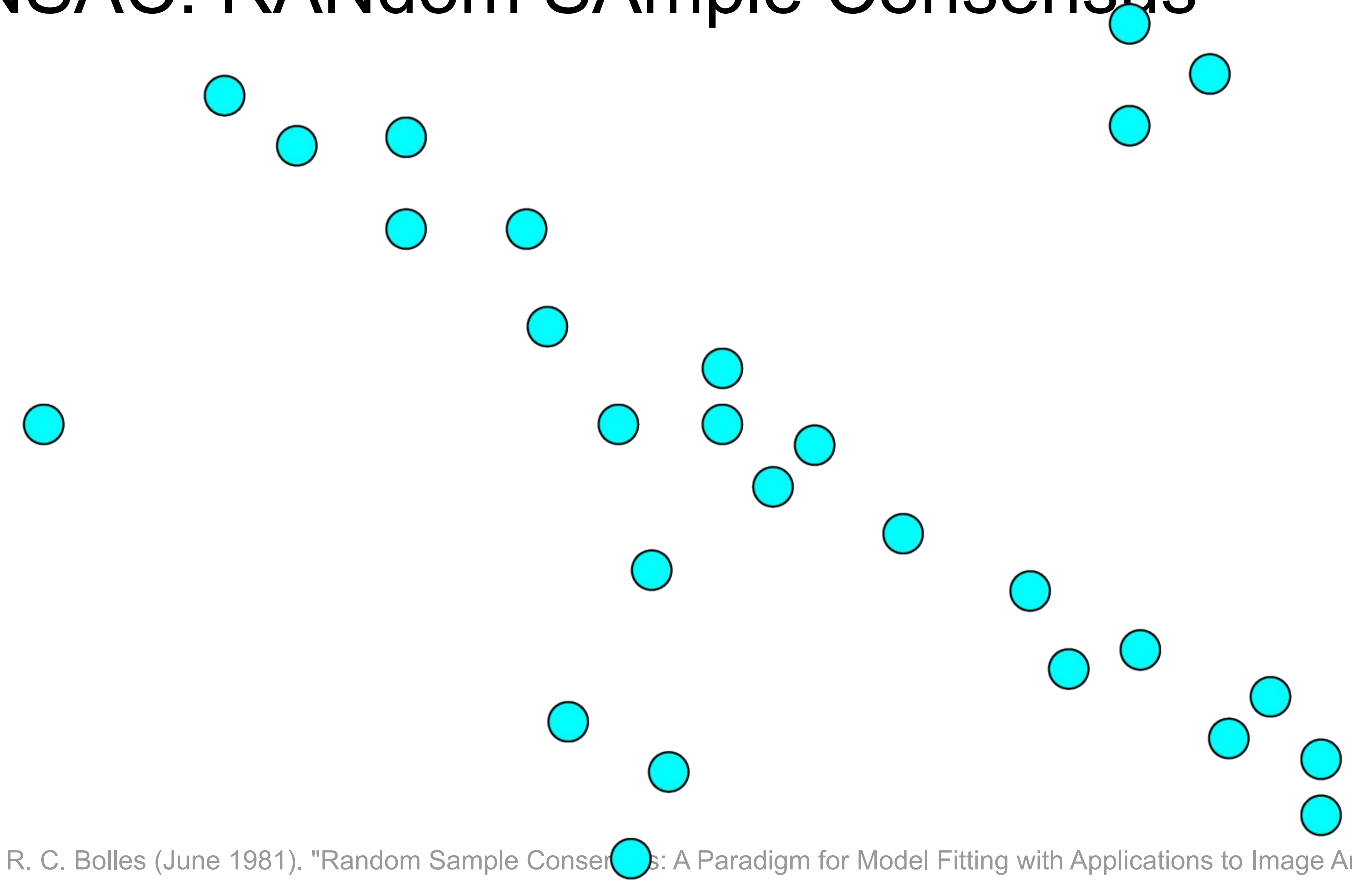
1. Classify data points as outliers or inliers
2. Fit model to inliers while ignoring outliers

Assumptions: outliers are random and will not agree

How? Try many models on subsets of the dataset and keep the best.

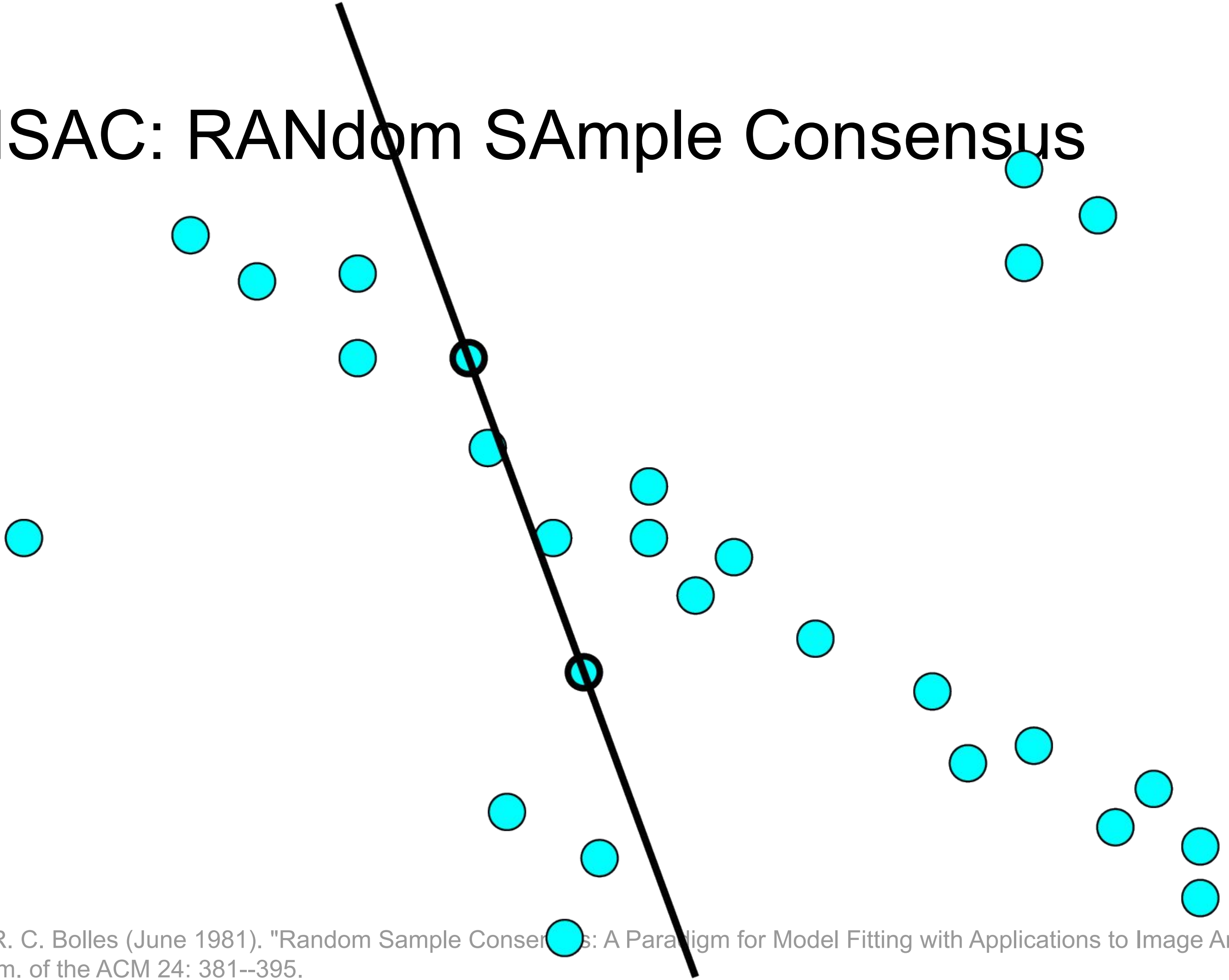
Several approaches: RANSAC, Hough transform, clustering…

RANSAC: RANdom SAmple Consensus



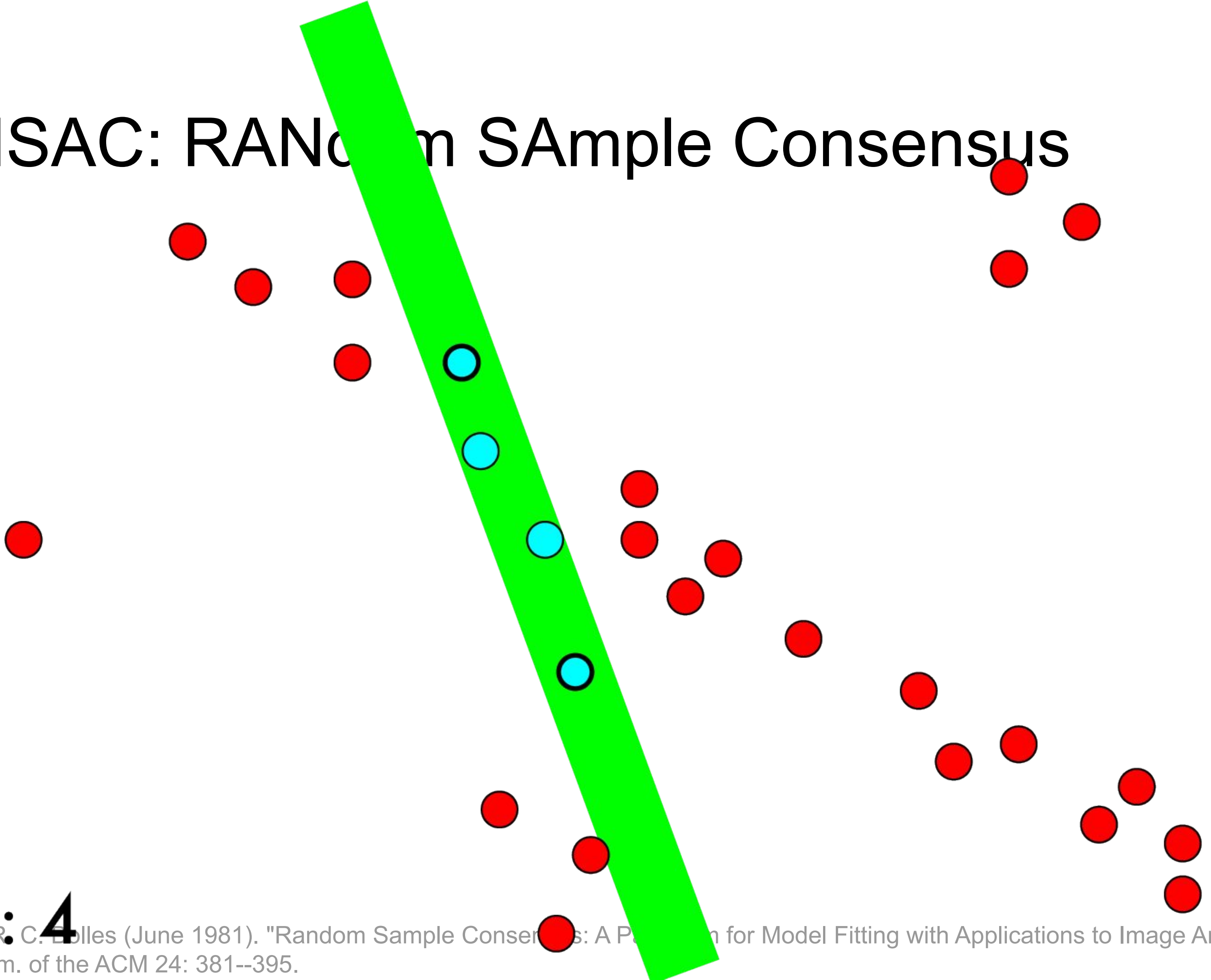
M. A. Fischler and R. C. Bolles (June 1981). "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". Comm. of the ACM 24: 381--395.

RANSAC: RANdom SAmple Consensus



M. A. Fischler and R. C. Bolles (June 1981). "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". Comm. of the ACM 24: 381--395.

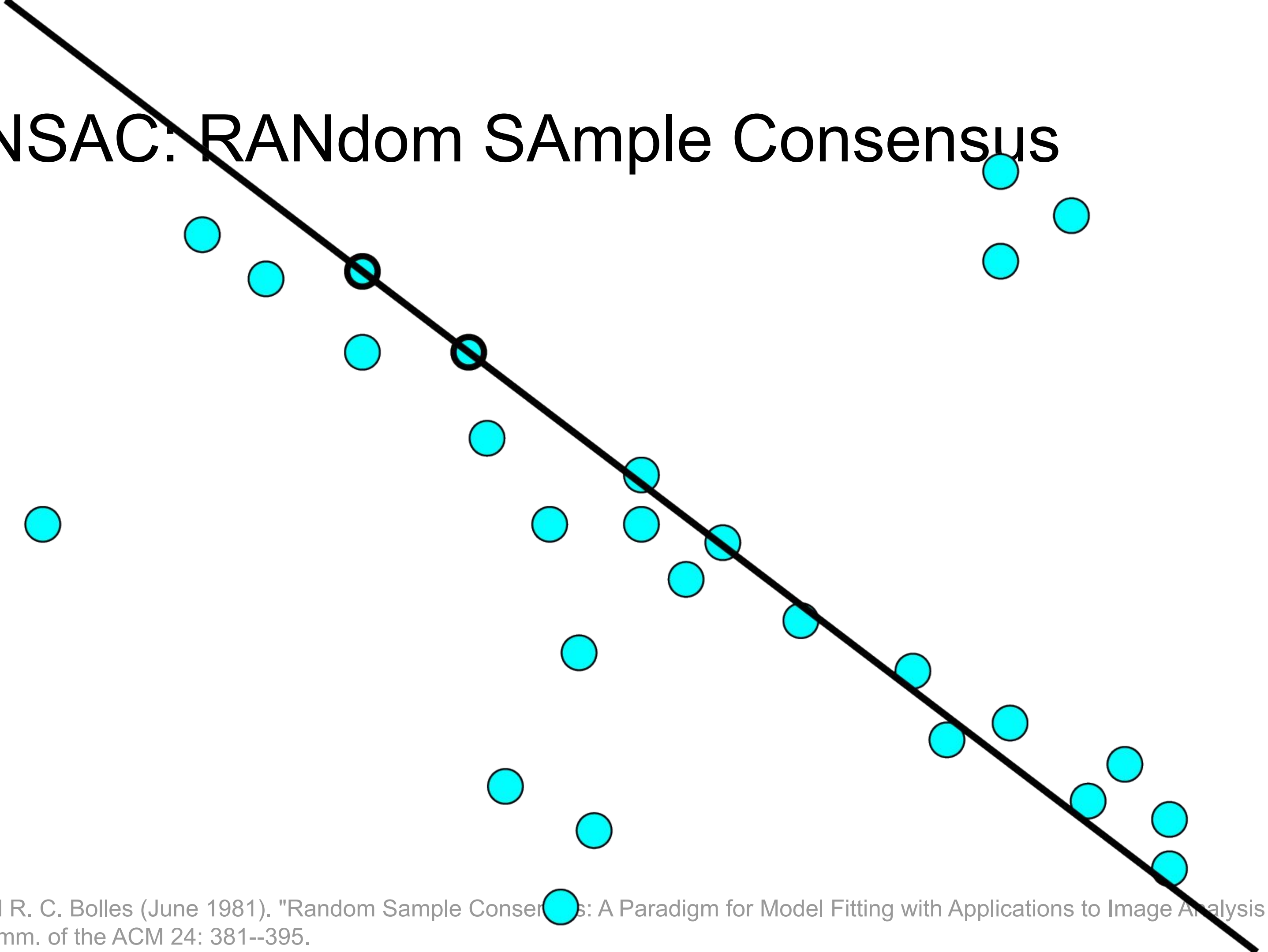
RANSAC: RANdom SAmple Consensus



Inliers: 4

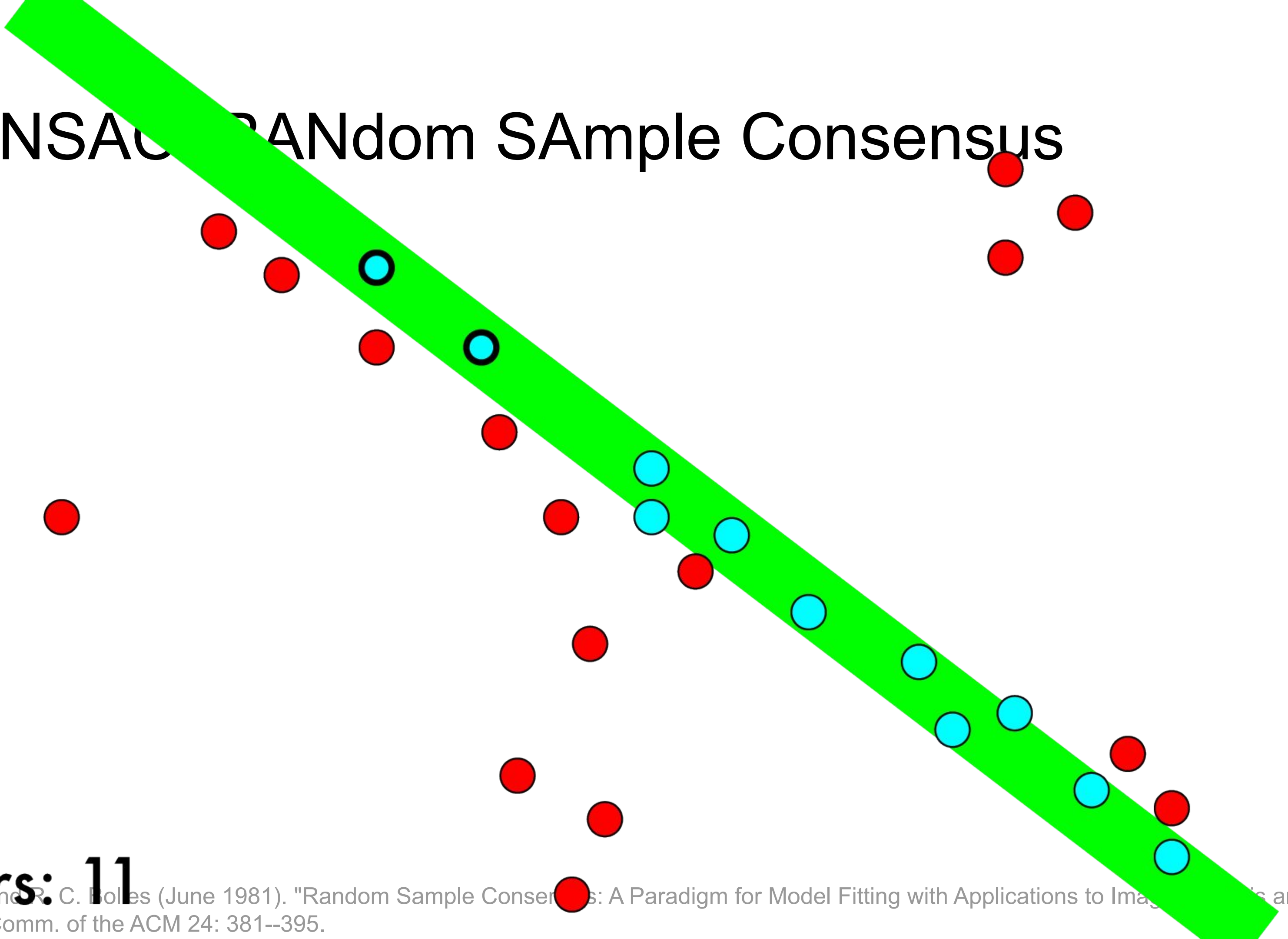
M. A. Fischler and R. C. Bolles (June 1981). "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". Comm. of the ACM 24: 381--395.

RANSAC: RANdom SAmple Consensus



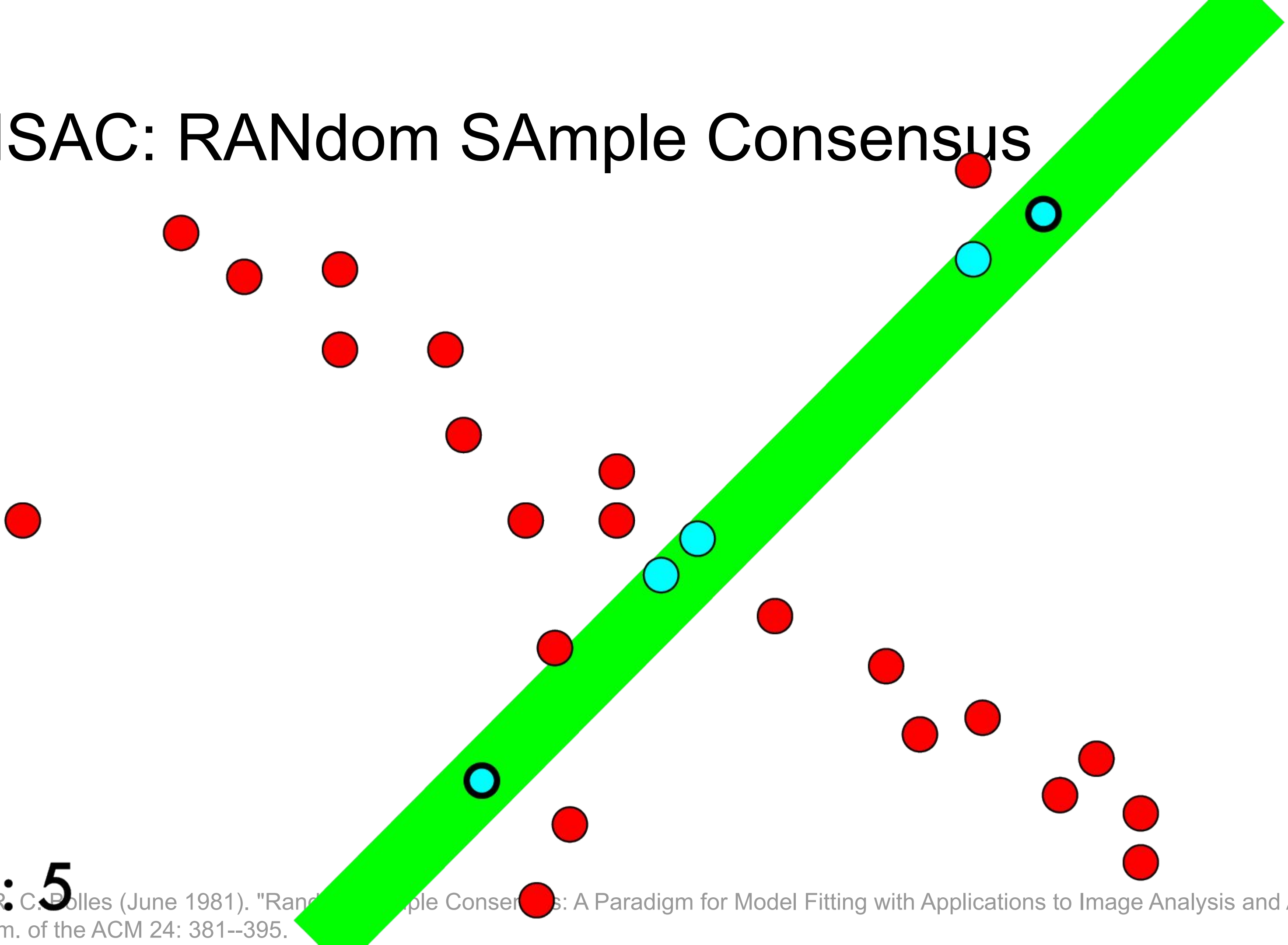
M. A. Fischler and R. C. Bolles (June 1981). "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". Comm. of the ACM 24: 381--395.

RANSAC: RANdom SAmple Consensus



M. A. Fischler and R. C. Bolles (June 1981). "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". Comm. of the ACM 24: 381--395.

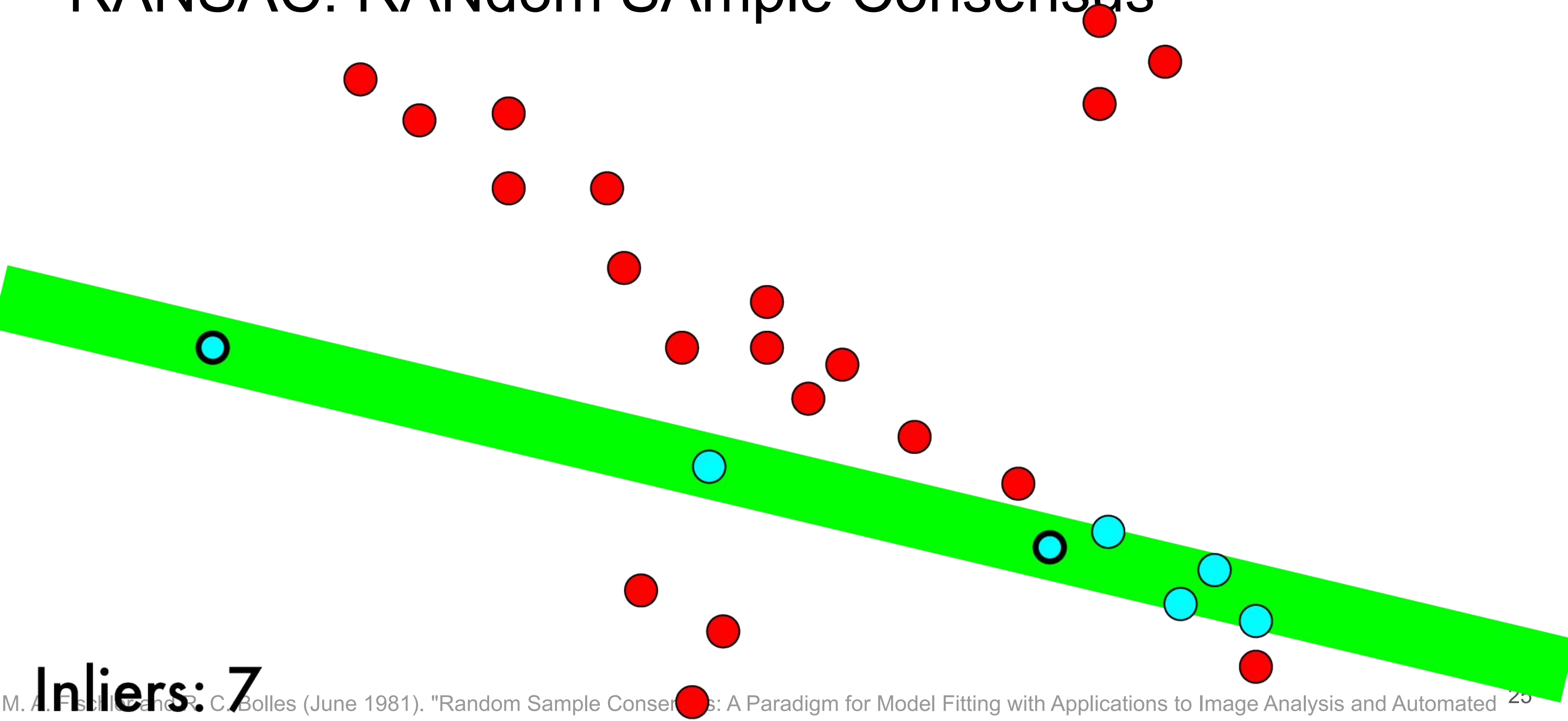
RANSAC: RANdom SAmple Consensus



Inliers: 5

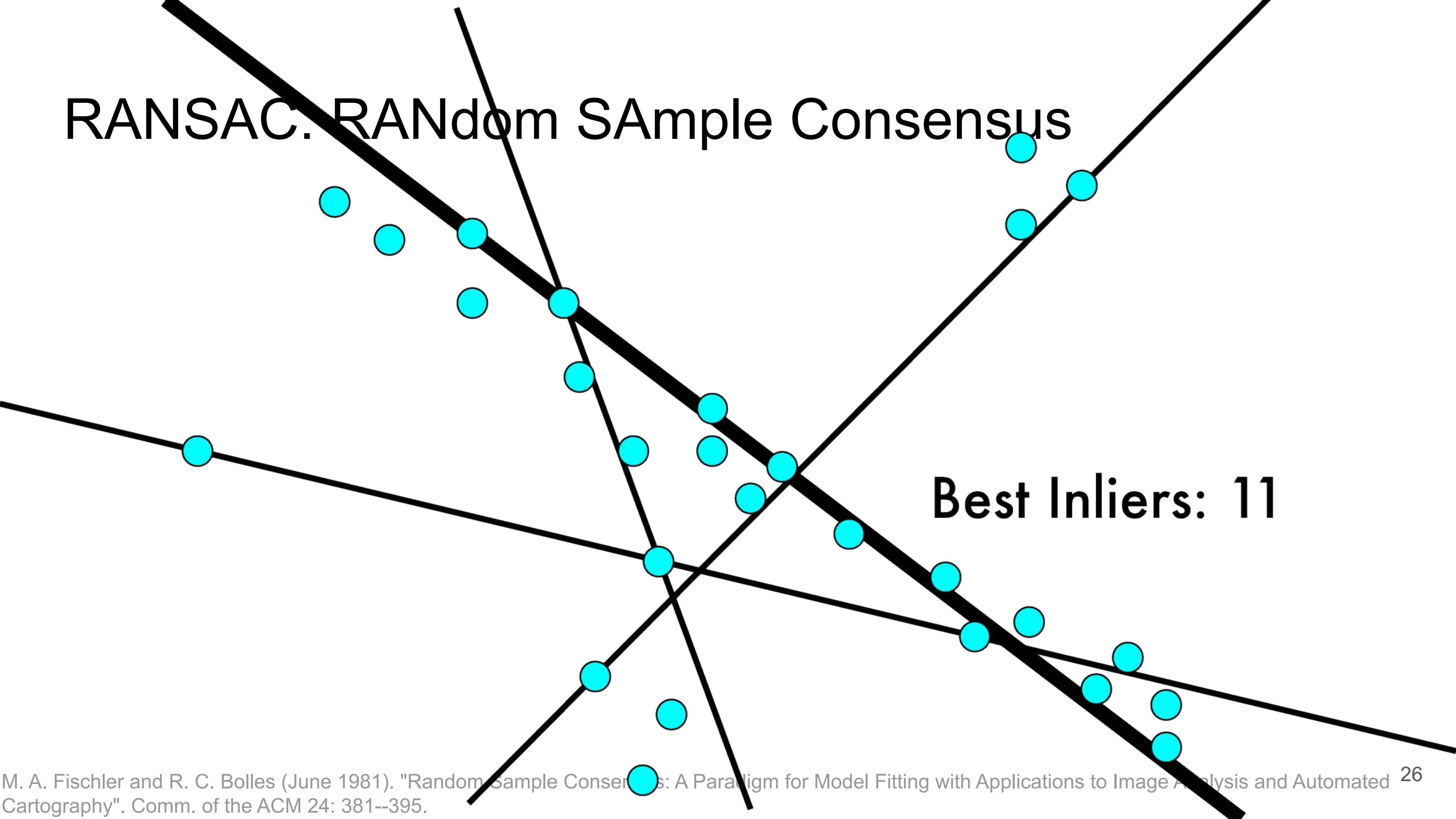
M. A. Fischler and R. C. Bolles (June 1981). "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". Comm. of the ACM 24: 381--395.

RANSAC: RANdom SAmple Consensus



M. A. Fischler and R. C. Bolles (June 1981). "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". Comm. of the ACM 24: 381--395.

RANSAC: RANdom SAmple Consensus



M. A. Fischler and R. C. Bolles (June 1981). "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". Comm. of the ACM 24: 381--395.

RANSAC algorithm

Parameters: **data**, **n**: num. points required to fit model, **k**: max iterations,
d: distance threshold to belong to fitted model, **m**: min. # inliers for early stop

$$\sum_i \left(x'_i - \frac{h_{00}x_i + h_{01}y_i + h_{02}}{h_{20}x_i + h_{21}y_i + h_{22}}\right)^2 + \left(y'_i - \frac{h_{10}x_i + h_{11}y_i + h_{12}}{h_{20}x_i + h_{21}y_i + h_{22}}\right)^2$$

```
bestmodel = None
bestfit = INF
While niter < k:
    sample = draw n random points from data
    Fit model to sample ← using Least Square here
    inliers = data within distance d of model ← OpenCV uses retroprojection error, ie ||dstpoint - proj(srcpoint)||_2 < d
    if inliers > bestfit:
        Fit model to all inliers
        bestfit = fit
        bestmodel = model
        if inliers > m:
            return model
return bestmodel
```

RANSAC algorithm

How to set the parameters?

Default values are usually fine in toolkits.

- **n**: num. points required to fit model
  ⇒ set to minimum necessary (max 8)
- **k**: max iterations
  ⇒ 2000 permits to be sure that at least 1 sample without outliers will be drawn with a probability > 99%
- **d**: distance threshold to belong to fitted model
  ⇒ keep small but **may depend on problem**
- **m**: min. # inliers for early stop
  ⇒ should be >> **n**, but you do not care to reach **k** anyway

RANSAC works well with extreme noise

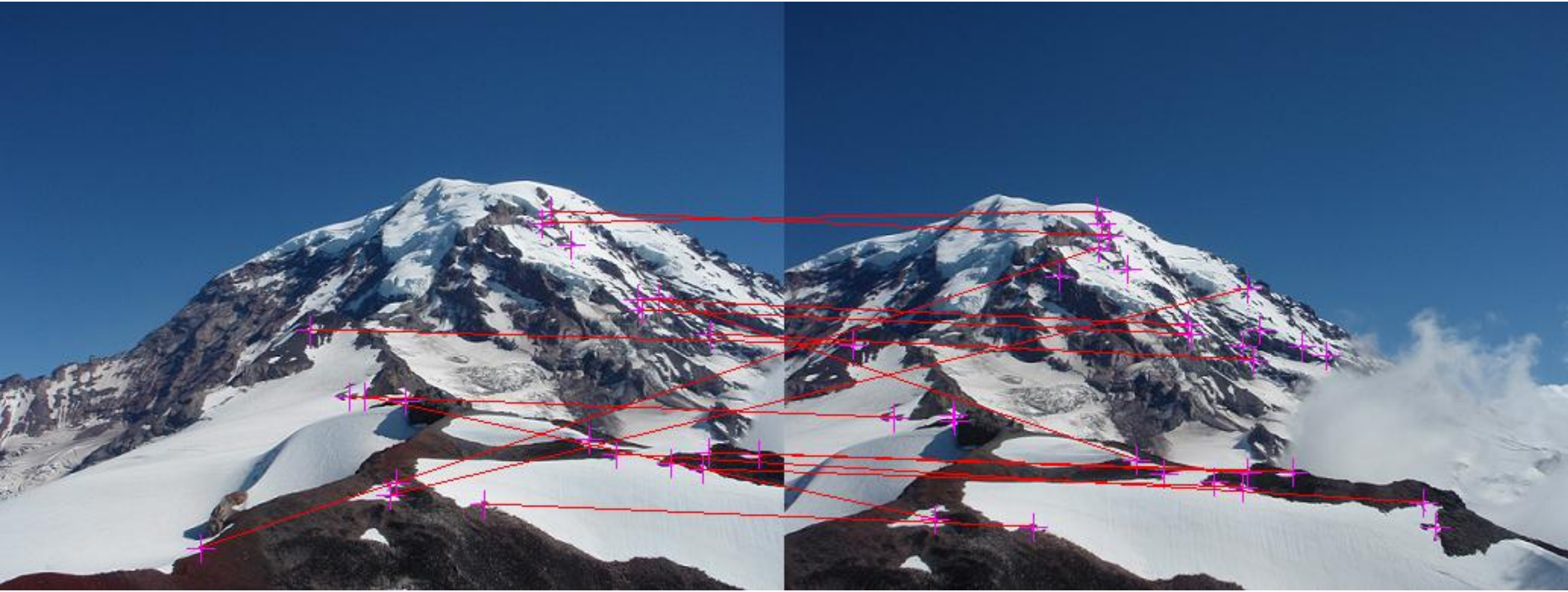


Image credit: JP Reddie

RANSAC works well with extreme noise

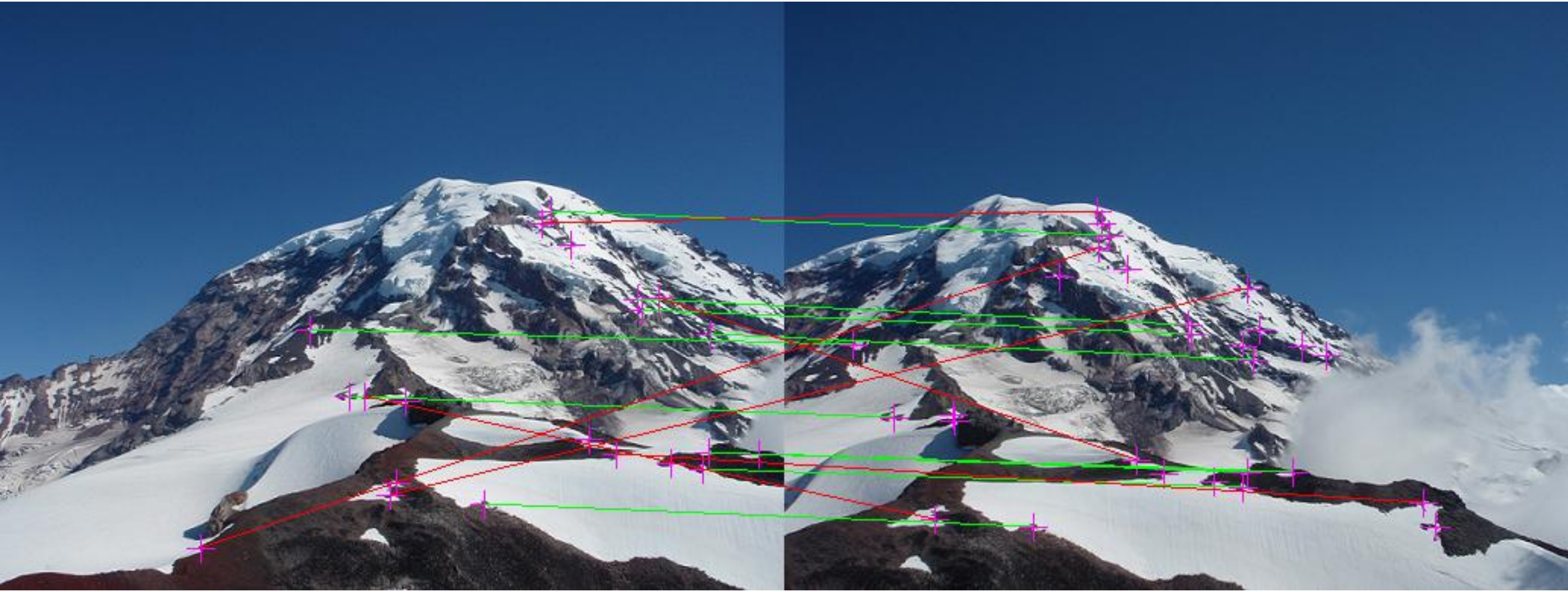


Image credit: JP Reddie

Other approaches

Compatible with **multiple instance detection**.

Naive implementation:

1. (opt.) When matching descriptors, accept more than 1 neighbor
   Recommended: use a background model to set a radius threshold
2. Estimate all possible homographies,
   keeping track of the support points for each
3. Run a clustering (or bin counting) algorithm in the parameter space of the homography to identify the best candidates