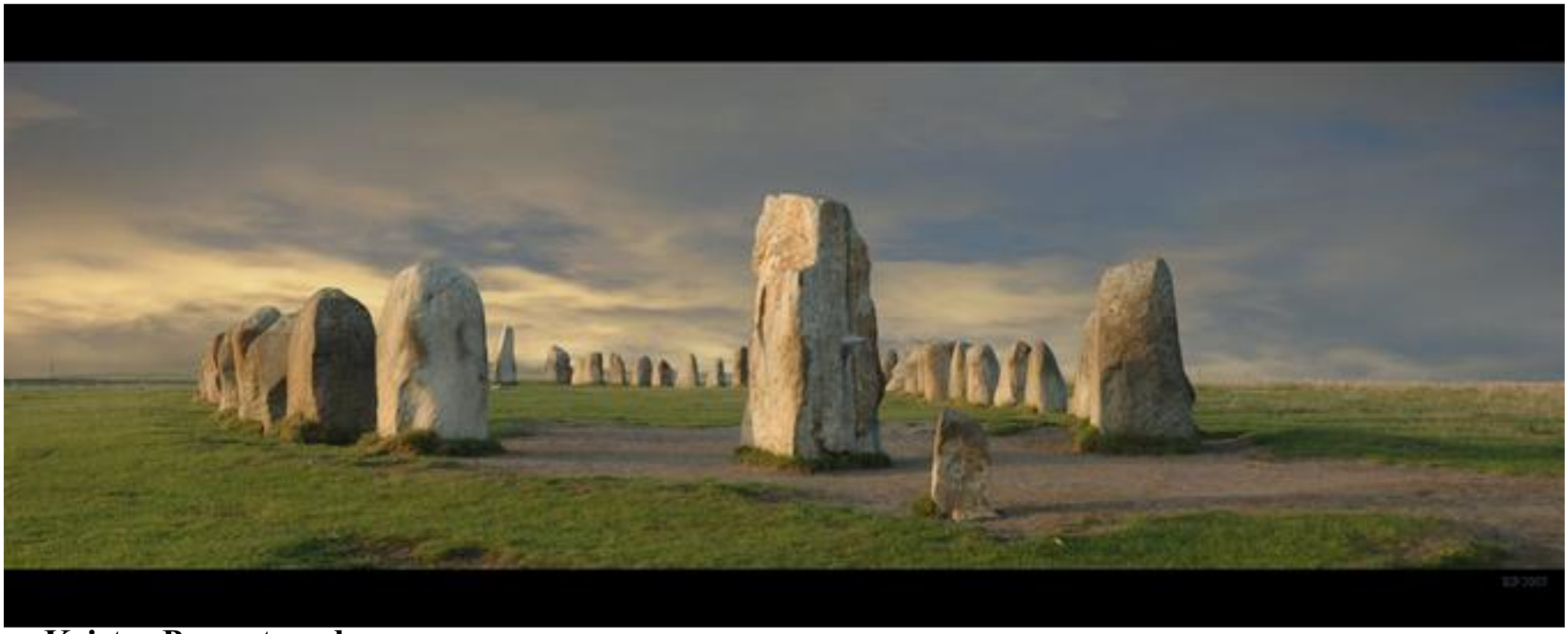
RANSAC

Fredo Durand
MIT EECS 6.8370/6.8371

Feature Matching and RANSAC



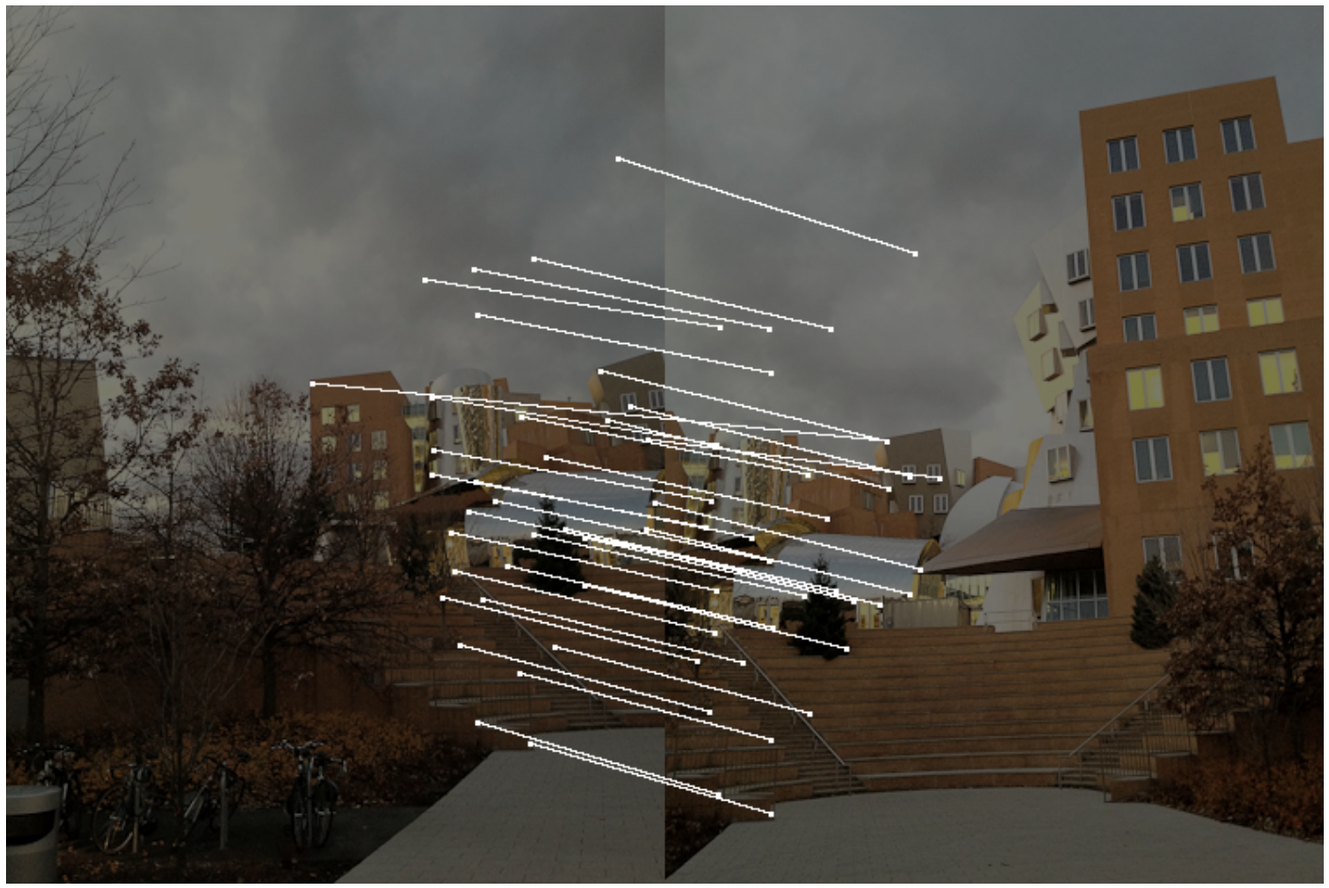
© **Krister Parmstrand**

Nikon D70. Stitched Panorama. The sky has been retouched. No other image manipulation.

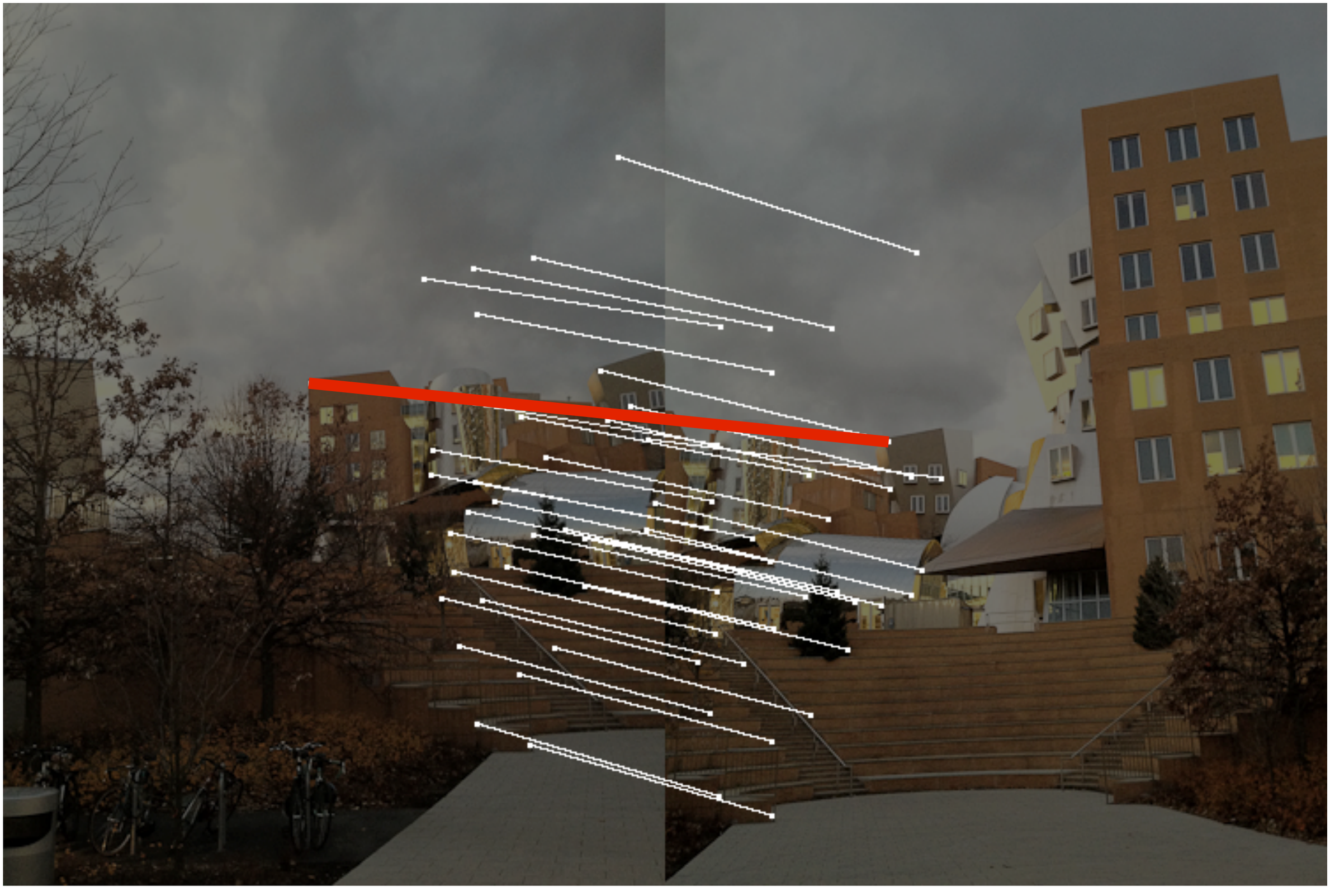
with a lot of slides stolen from
Alyosha Efros, Steve Seitz and Rick Szeliski

Status: after 2NN test

- We have feature points
- Matches between two images
  - Most of them are good
  - But not all of them. There can still be error.
  - **Today: RANSAC to remove remaining outliers**



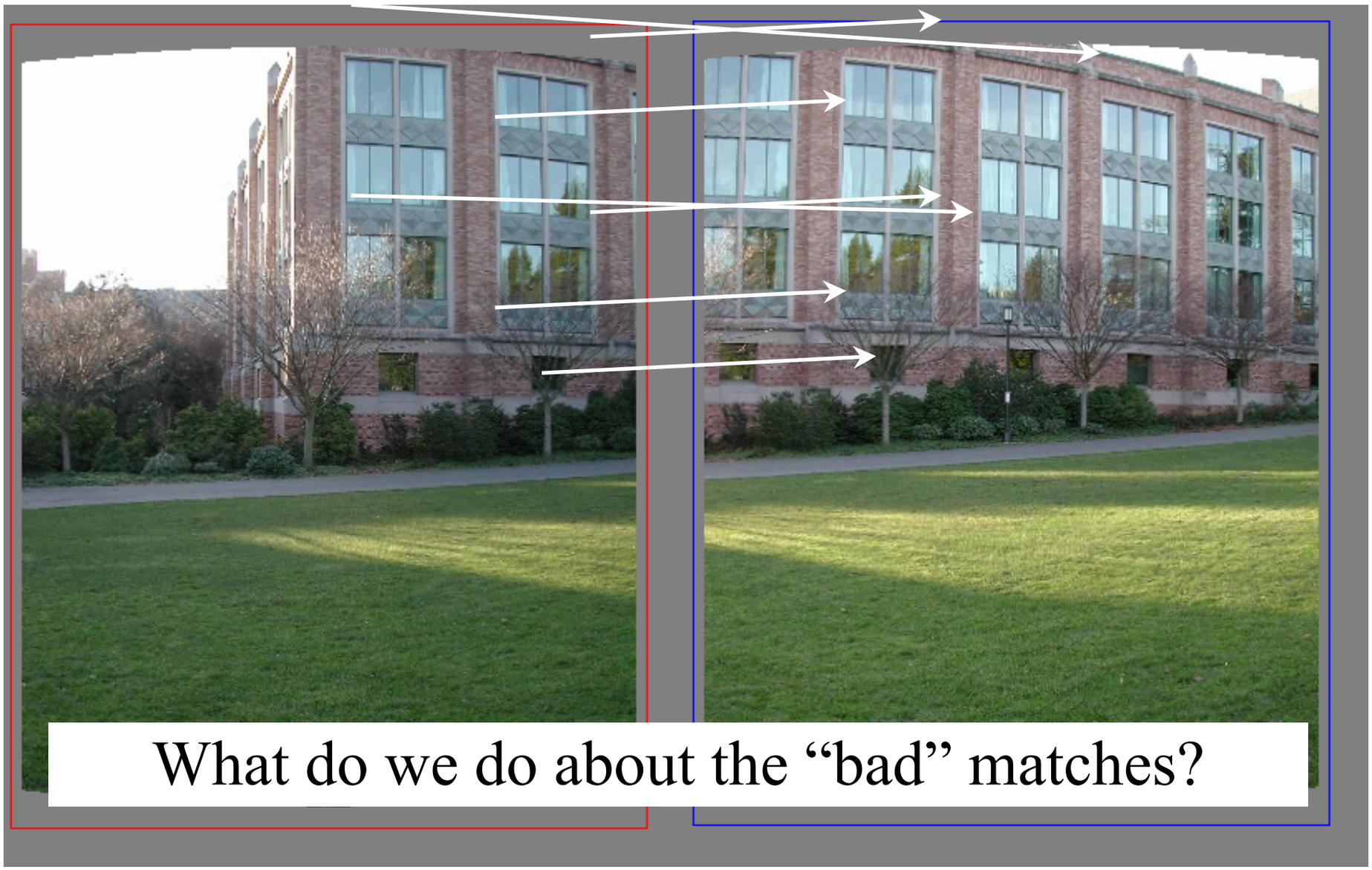
Example of outlier



Ransac idea

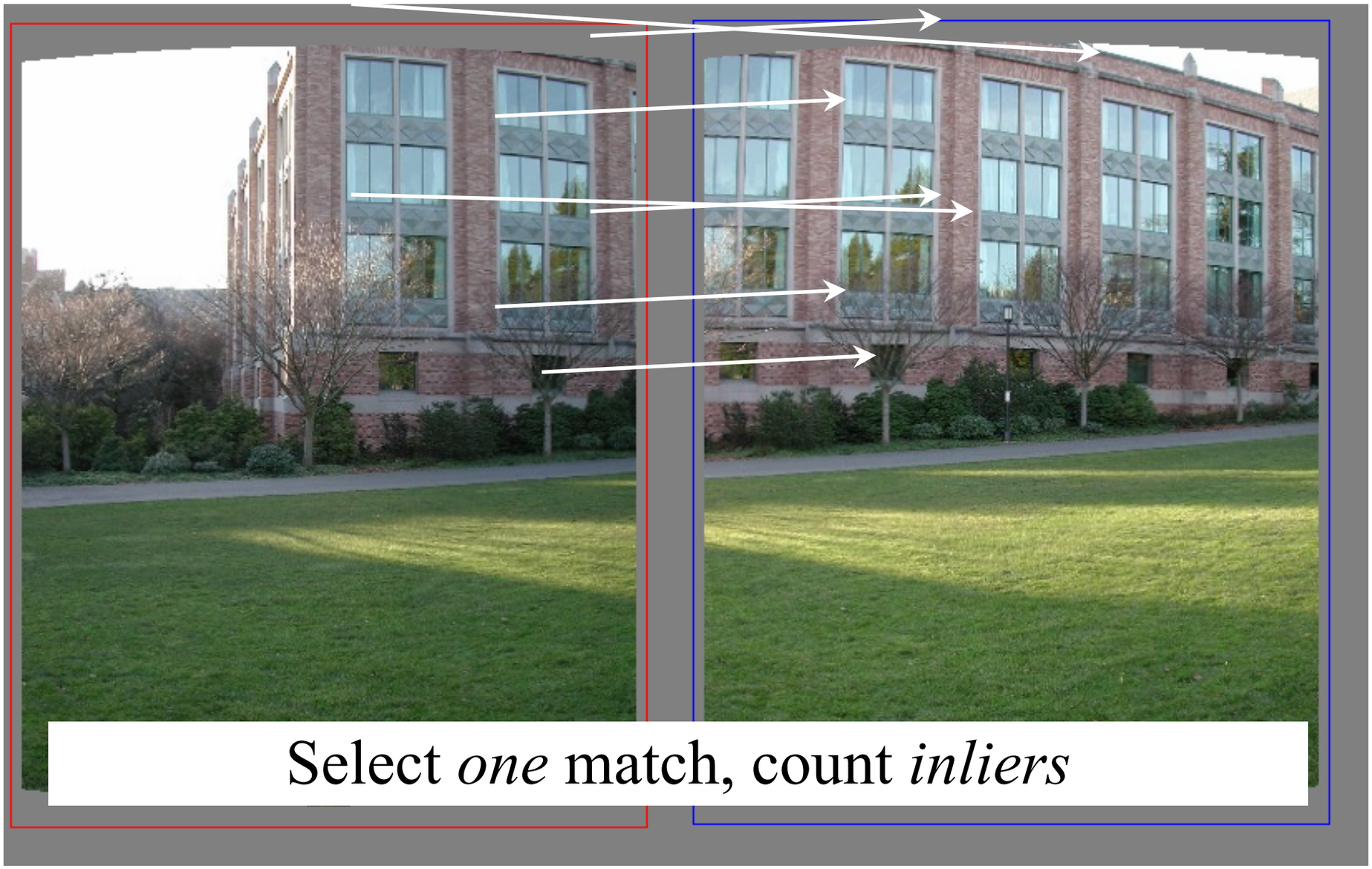
(b) Matching features at the image level--looking for a good homography

Simplified illustration with translation instead of homography

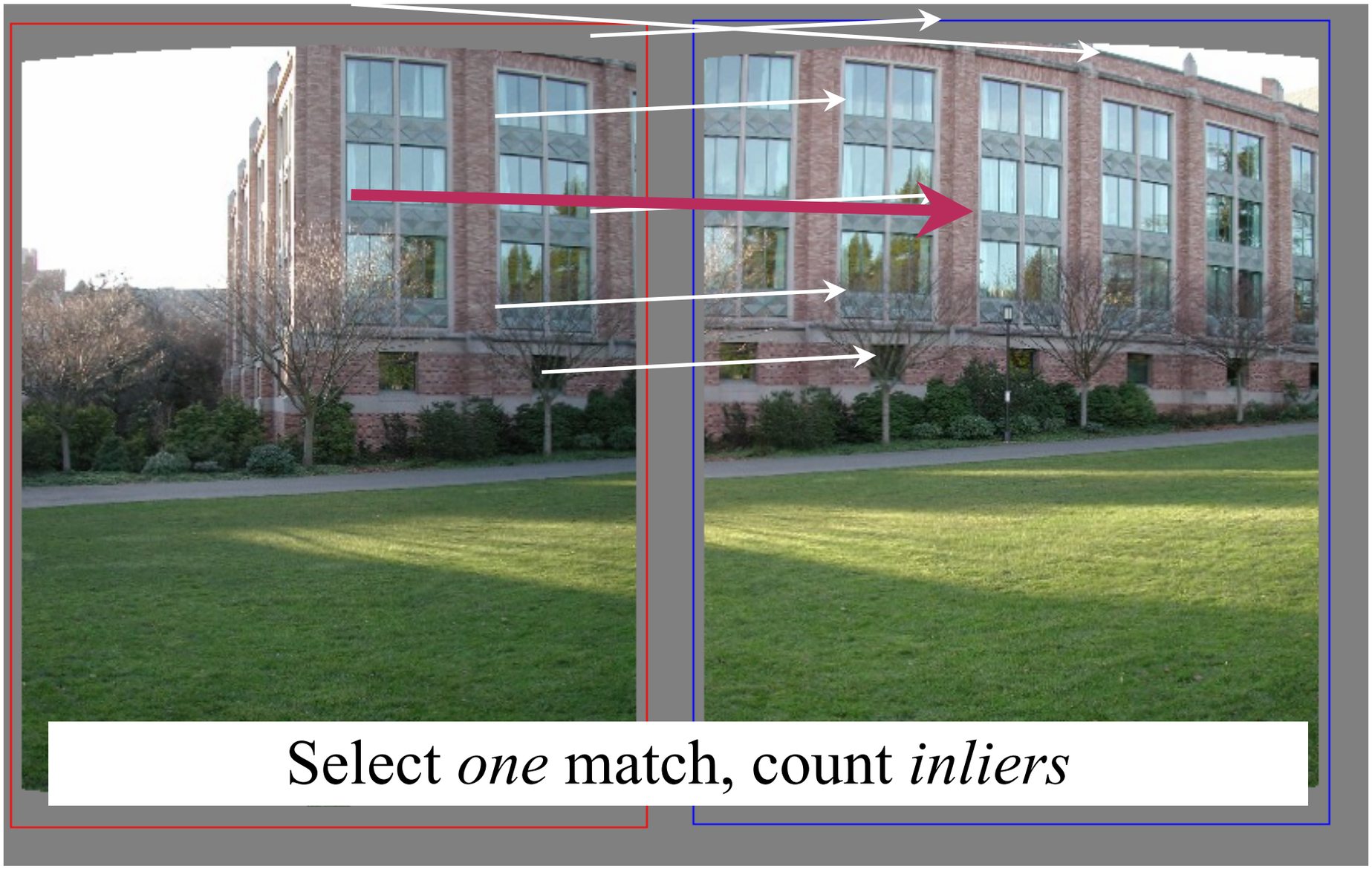


Note: at this point we don’t know which ones are good/bad

RAndom SAmple Consensus

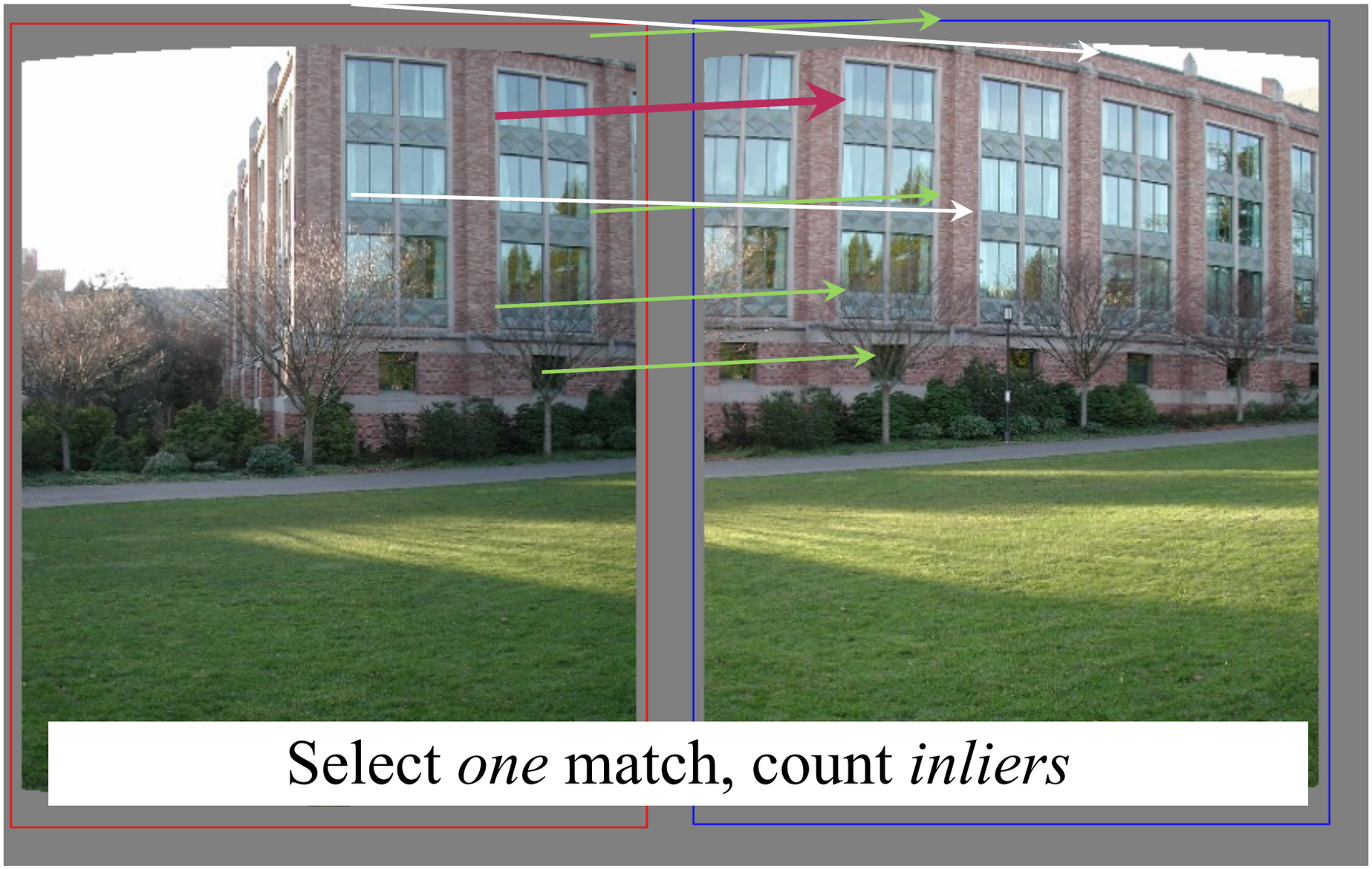


RAndom SAmple Consensus



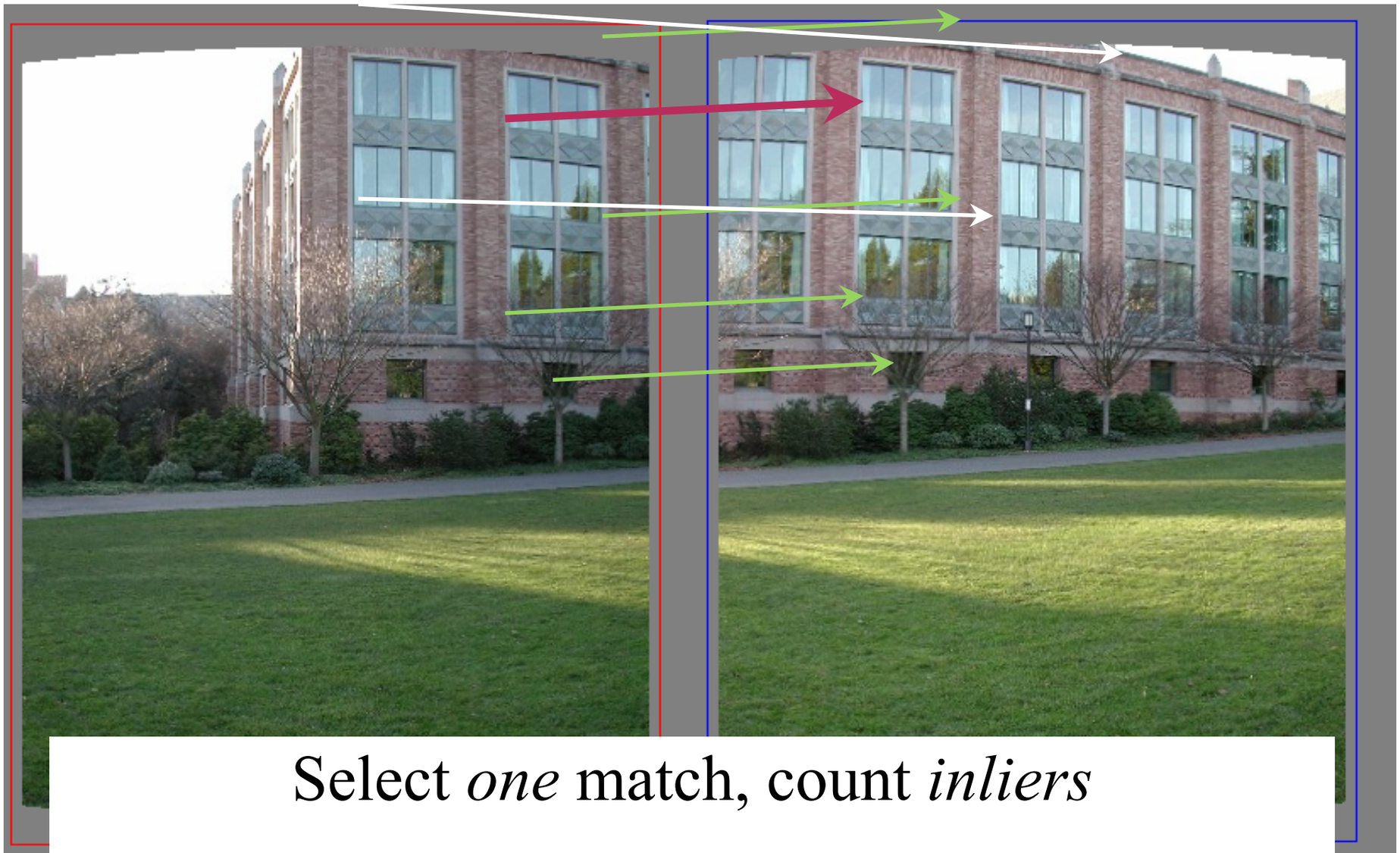
1 inliers

RAndom SAmple Consensus



5 inliers

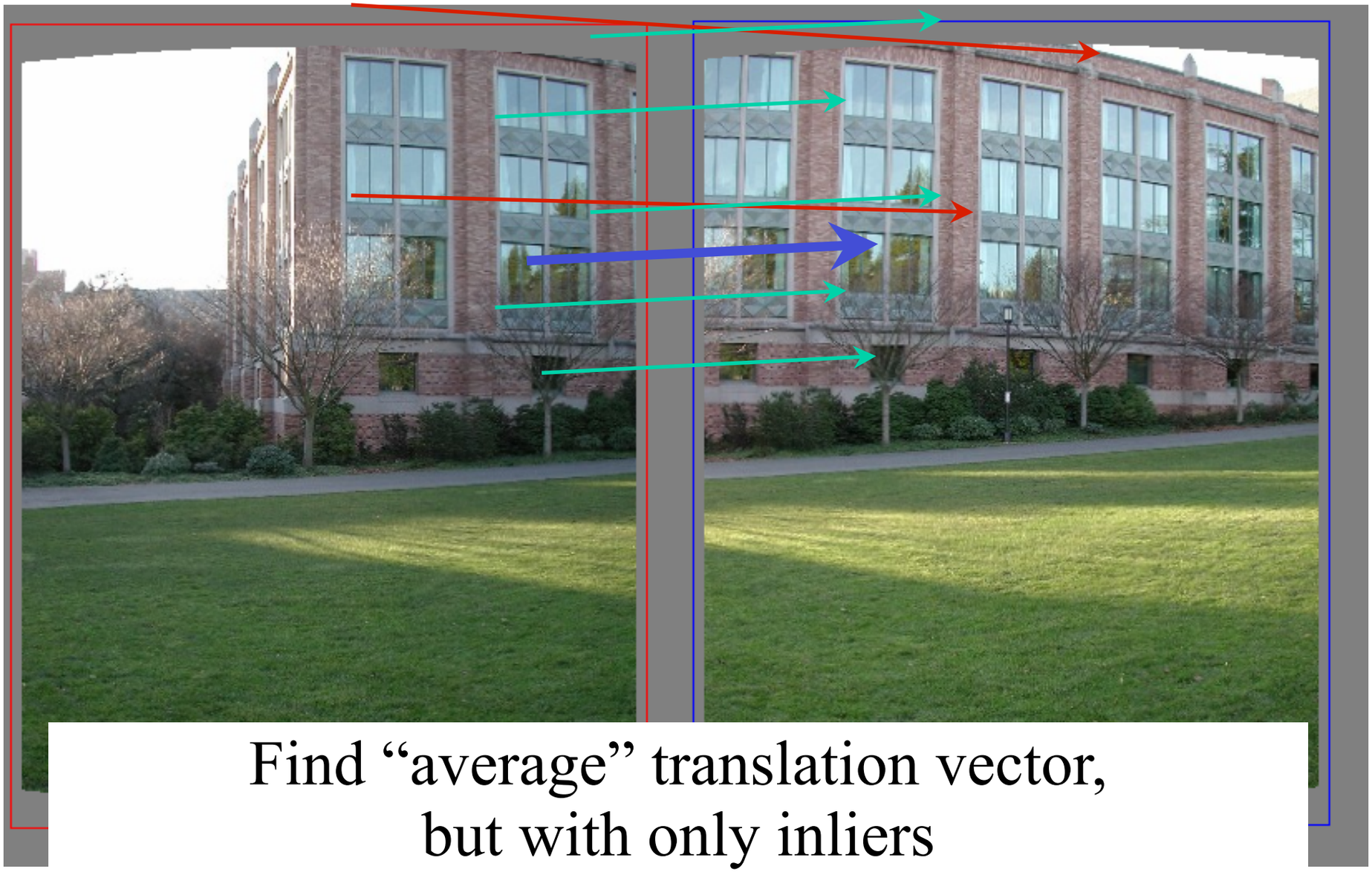
RAndom SAmple Consensus



Select *one* match, count *inliers*

Keep match with largest set of inliers

At the end: Least squares fit

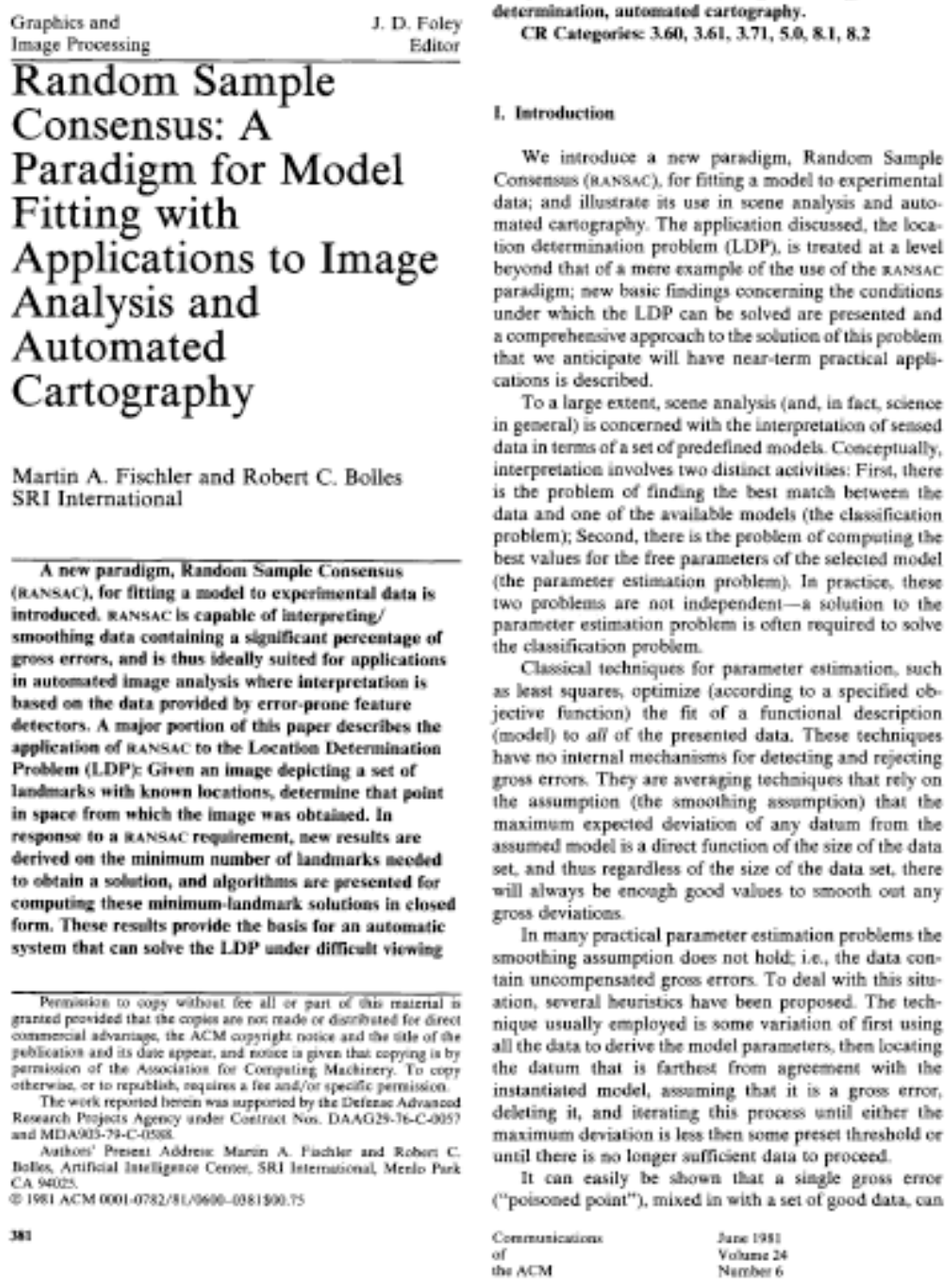


Find “average” translation vector,
but with only inliers

RANSAC

Reference

- M. A. Fischler, R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Comm. of the ACM, Vol 24, pp 381-395, 1981.
- http://portal.acm.org/citation.cfm?id=358692



Graphics and Image Processing — J. D. Foley Editor

Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography

Martin A. Fischler and Robert C. Bolles
SRI International

A new paradigm, Random Sample Consensus (RANSAC), for fitting a model to experimental data is introduced. RANSAC is capable of interpreting/smoothing data containing a significant percentage of gross errors, and is thus ideally suited for applications in automated image analysis where interpretation is based on the data provided by error-prone feature detectors. A major portion of this paper describes the application of RANSAC to the Location Determination Problem (LDP): Given an image depicting a set of landmarks with known locations, determine that point in space from which the image was obtained. In response to a RANSAC requirement, new results are derived on the minimum number of landmarks needed to obtain a solution, and algorithms are presented for computing these minimum-landmark solutions in closed form. These results provide the basis for an automatic system that can solve the LDP under difficult viewing and analysis conditions. Implementation details and computational examples are also presented.

Key Words and Phrases: model fitting, scene analysis, camera calibration, image matching, location determination, automated cartography.
CR Categories: 3.60, 3.61, 3.71, 5.0, 8.1, 8.2

I. Introduction

We introduce a new paradigm, Random Sample Consensus (RANSAC), for fitting a model to experimental data; and illustrate its use in scene analysis and automated cartography. The application discussed, the location determination problem (LDP), is treated at a level beyond that of a mere example of the use of the RANSAC paradigm; new basic findings concerning the conditions under which the LDP can be solved are presented and a comprehensive approach to the solution of this problem that we anticipate will have near-term practical applications is described.

To a large extent, scene analysis (and, in fact, science in general) is concerned with the interpretation of sensed data in terms of a set of predefined models. Conceptually, interpretation involves two distinct activities: First, there is the problem of finding the best match between the data and one of the available models (the classification problem); Second, there is the problem of computing the best values for the free parameters of the selected model (the parameter estimation problem). In practice, these two problems are not independent—a solution to the parameter estimation problem is often required to solve the classification problem.

Classical techniques for parameter estimation, such as least squares, optimize (according to a specified objective function) the fit of a functional description (model) to *all* of the presented data. These techniques have no internal mechanisms for detecting and rejecting gross errors. They are averaging techniques that rely on the assumption (the smoothing assumption) that the maximum expected deviation of any datum from the assumed model is a direct function of the size of the data set, and thus regardless of the size of the data set, there will always be enough good values to smooth out any gross deviations.

In many practical parameter estimation problems the smoothing assumption does not hold; i.e., the data contain uncompensated gross errors. To deal with this situation, several heuristics have been proposed. The technique usually employed is some variation of first using all the data to derive the model parameters, then locating the datum that is farthest from agreement with the instantiated model, assuming that it is a gross error, deleting it, and iterating this process until either the maximum deviation is less then some preset threshold or until there is no longer sufficient data to proceed.

It can easily be shown that a single gross error ("poisoned point"), mixed in with a set of good data, can

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

The work reported herein was supported by the Defense Advanced Research Projects Agency under Contract Nos. DAAG29-76-C-0057 and MDA903-79-C-0588.

Authors' Present Address: Martin A. Fischler and Robert C. Bolles, Artificial Intelligence Center, SRI International, Menlo Park CA 94025.

© 1981 ACM 0001-0782/81/0600-0381$00.75

RANSAC for estimating homography

RANSAC for estimating homography

RANSAC loop:

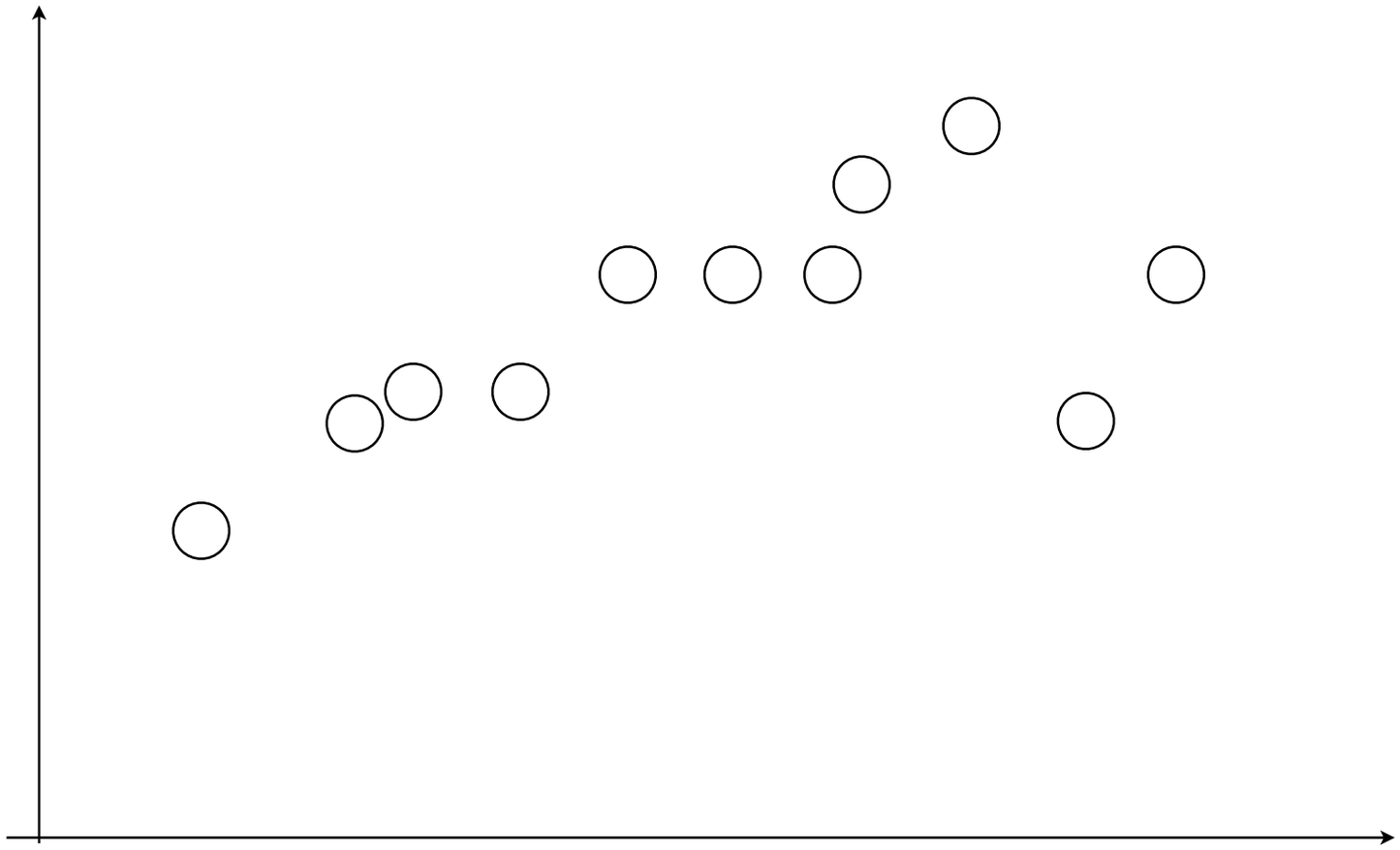
- Select four feature pairs (at random)
- Compute homography H (exact)
- Compute *inliers* where $\|p_i', \boldsymbol{H}\, p_i\| < \varepsilon$

Keep largest set of inliers

Re-compute least-squares H estimate on all of the inliers

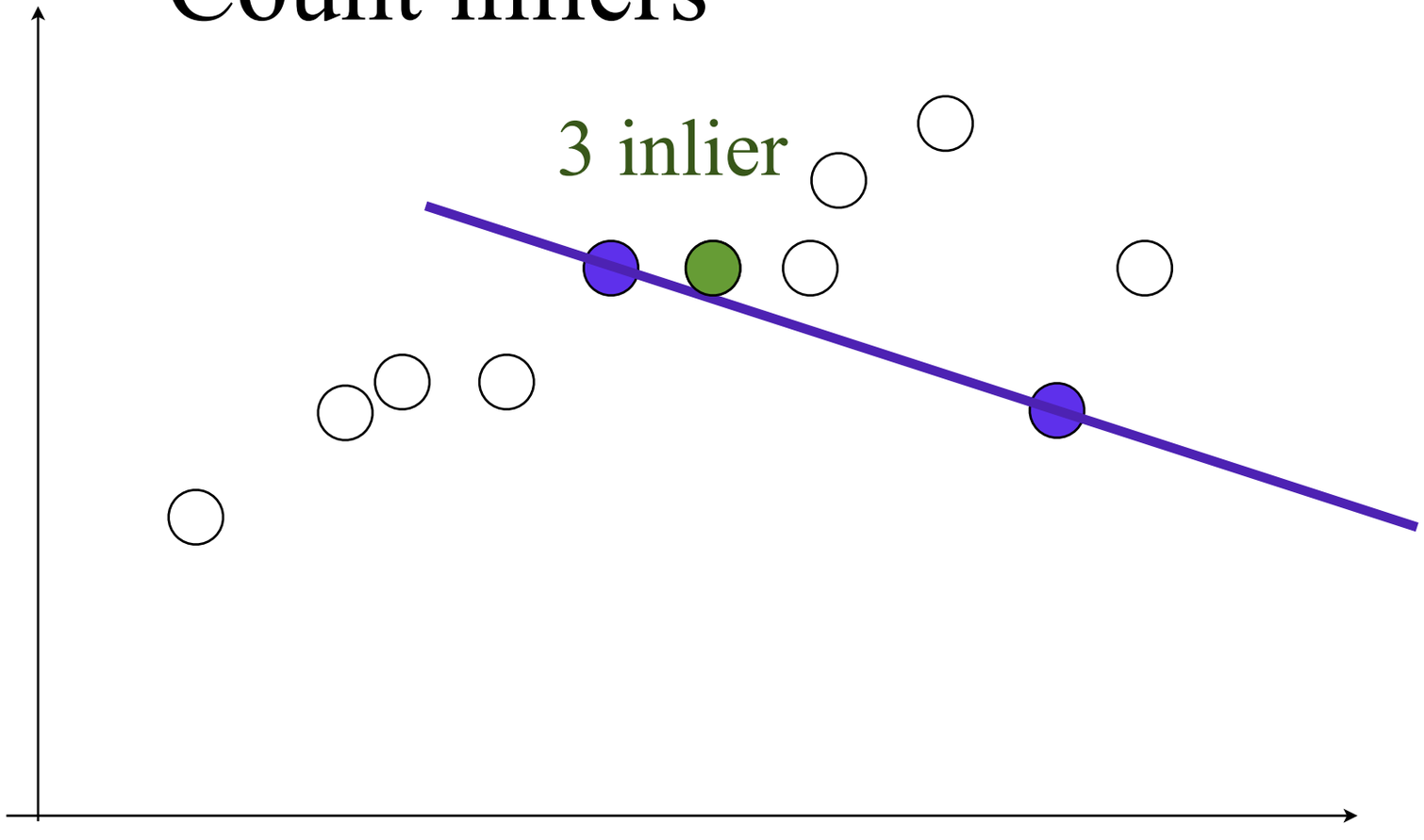
Simple example: fit a line

- Rather than homography H (8 numbers) fit y=ax+b (2 numbers a, b) to 2D pairs



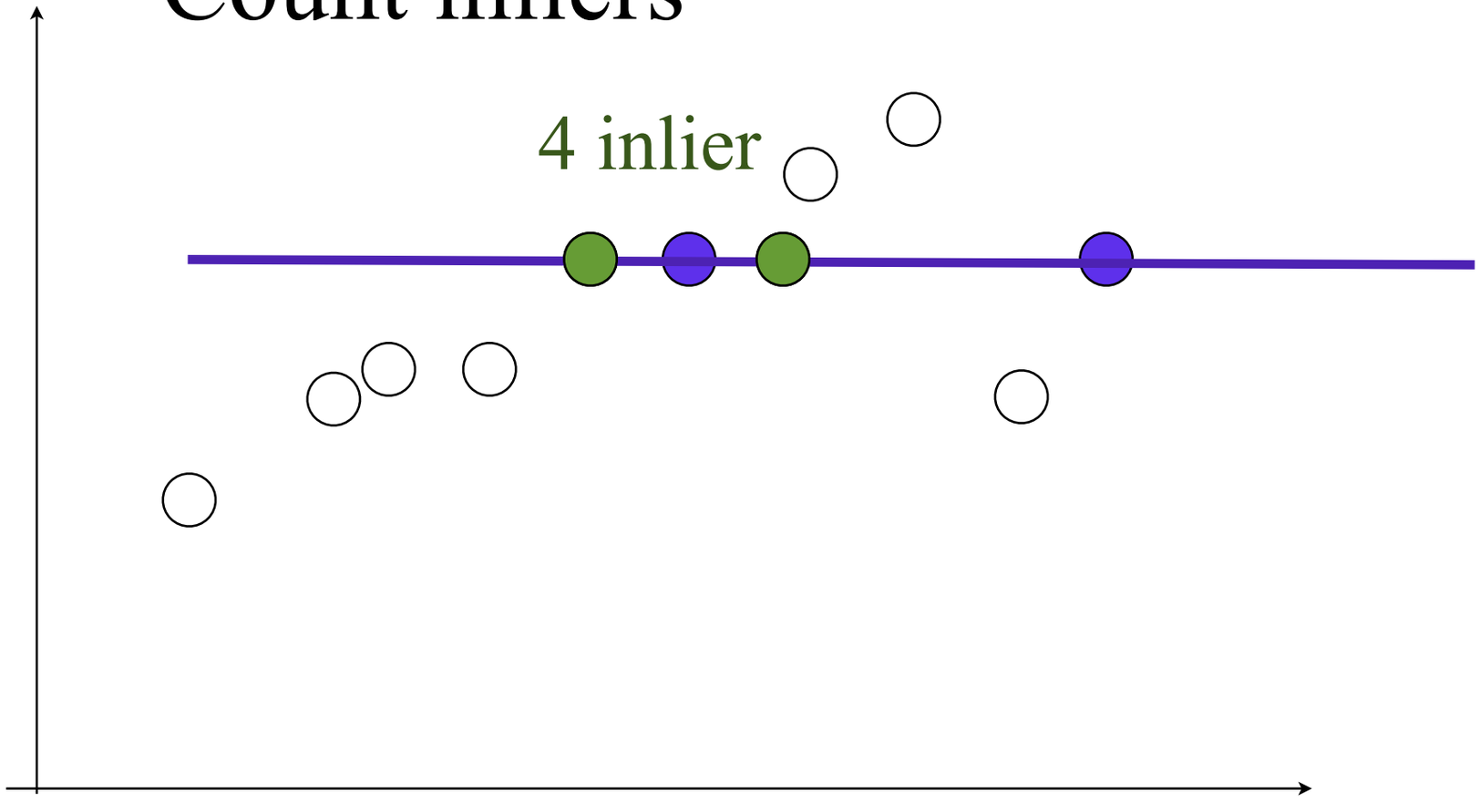
Simple example: fit a line

- Pick 2 points
- Fit line
- Count inliers



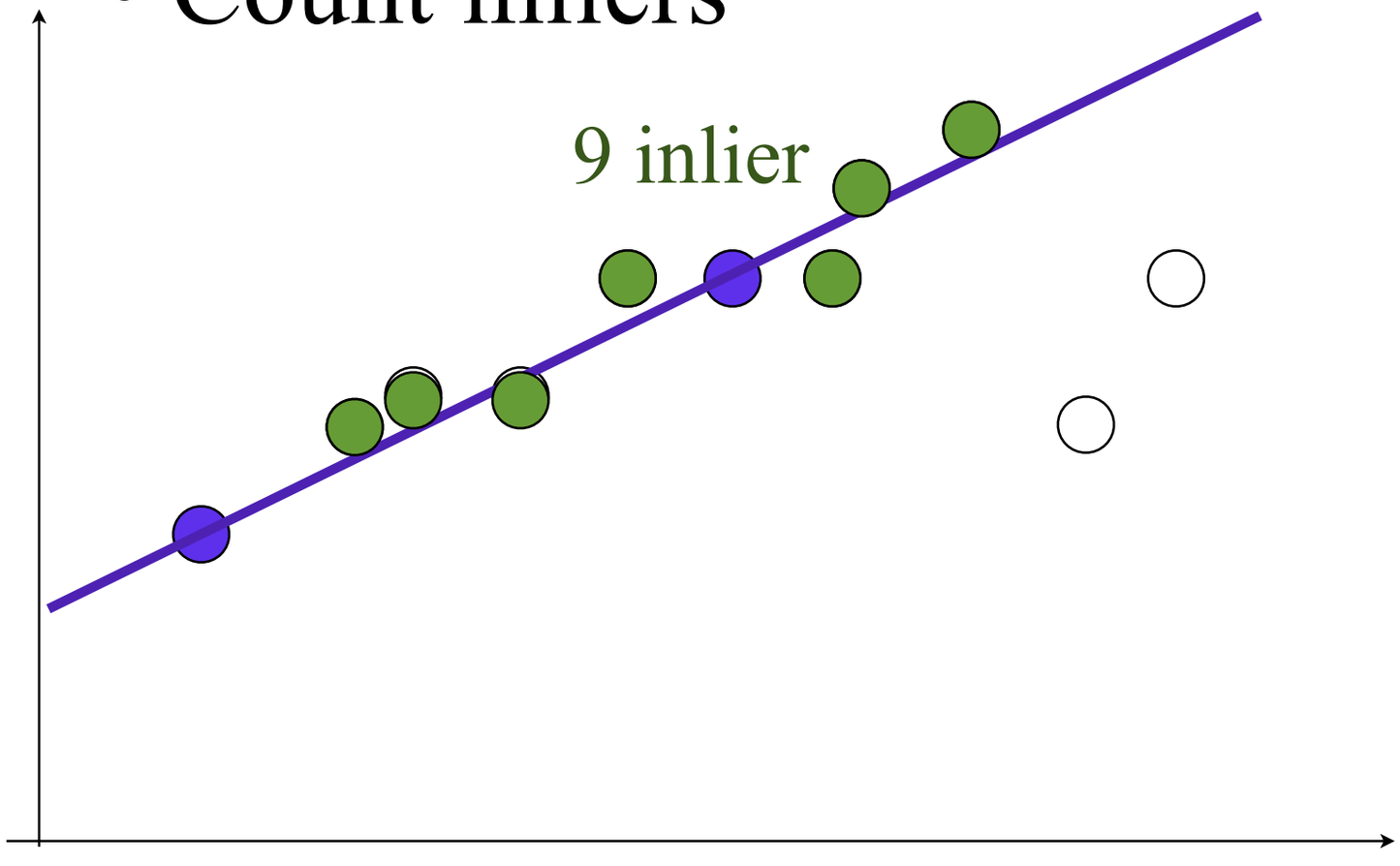
Simple example: fit a line

- Pick 2 points
- Fit line
- Count inliers



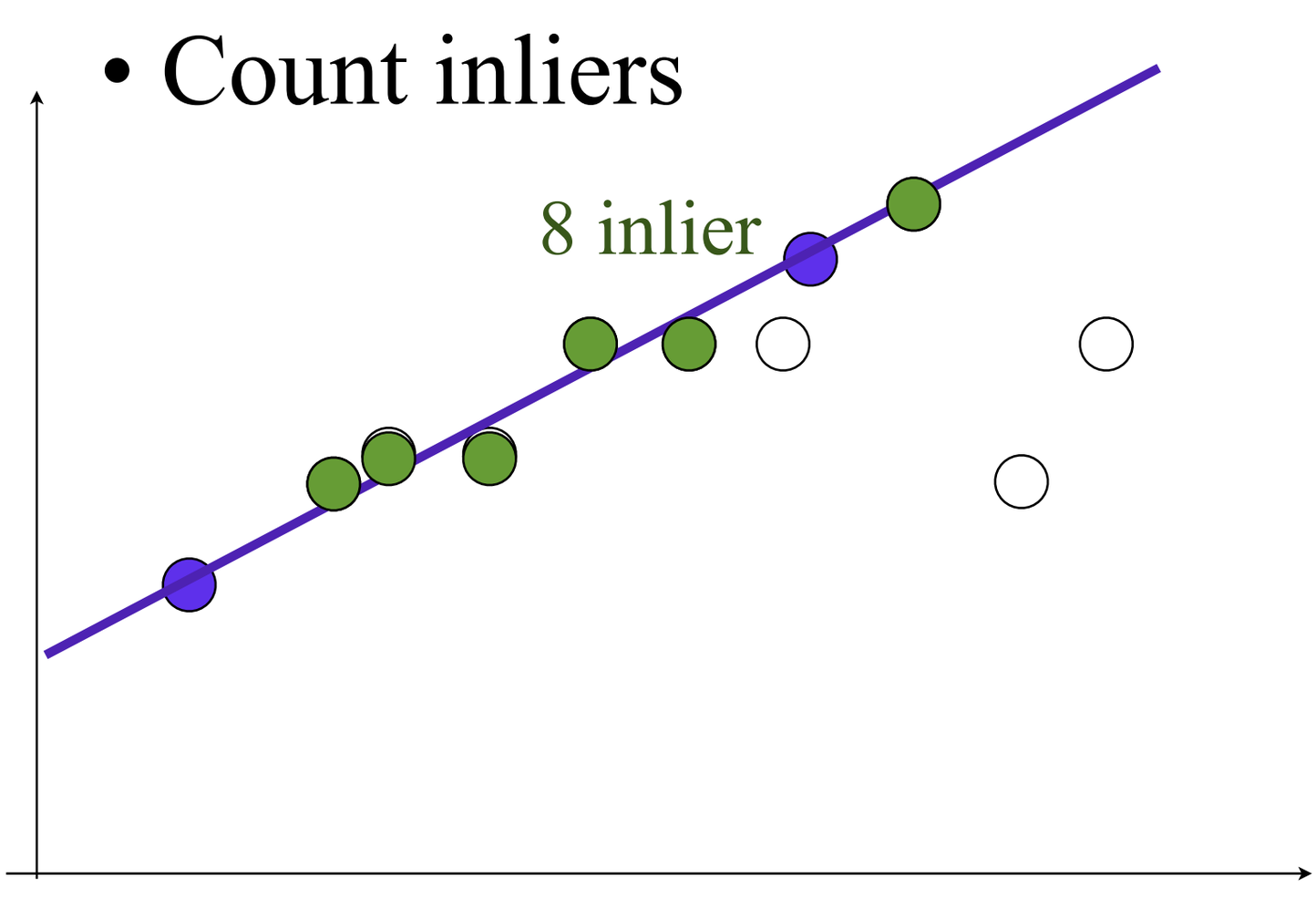
Simple example: fit a line

- Pick 2 points
- Fit line
- Count inliers



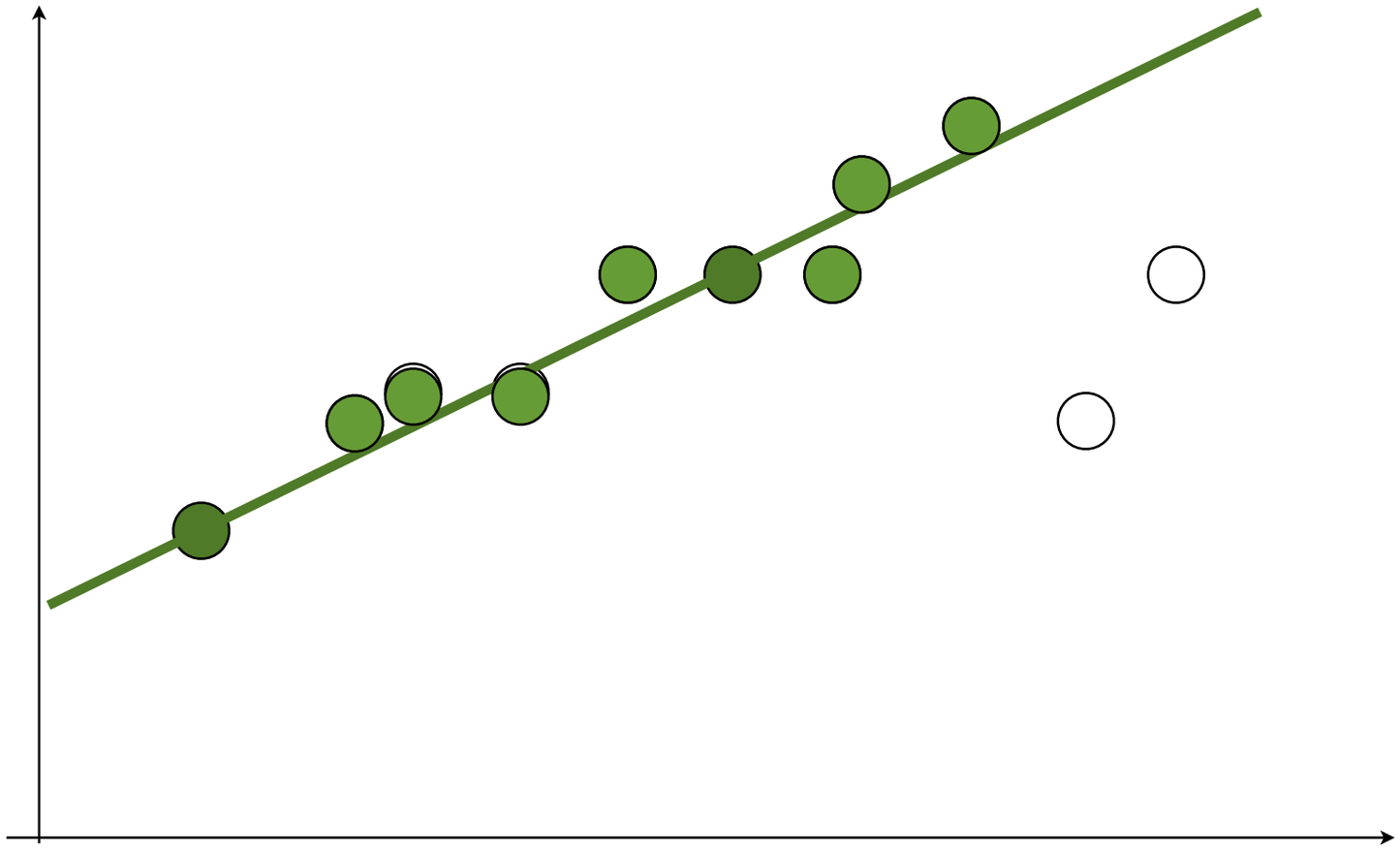
Simple example: fit a line

- Pick 2 points
- Fit line
- Count inliers



Simple example: fit a line

- Use biggest set of inliers
- Do least-square fit



RANSAC for estimating homography

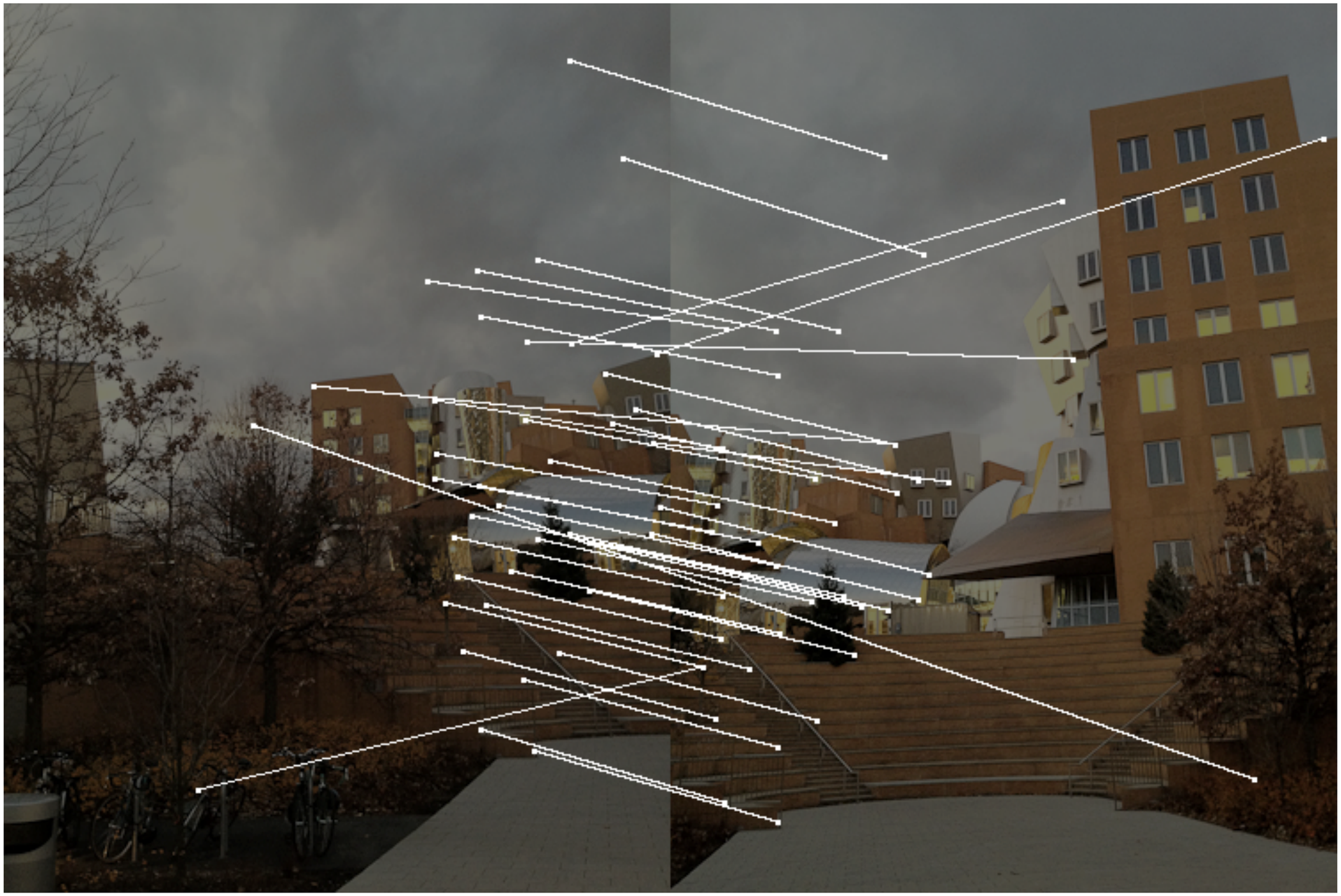
RANSAC loop:

- Select four feature pairs (at random)
- Compute homography H (exact)
- Compute *inliers* where $\|p_i', \boldsymbol{H}\, p_i\| < \varepsilon$

Keep largest set of inliers

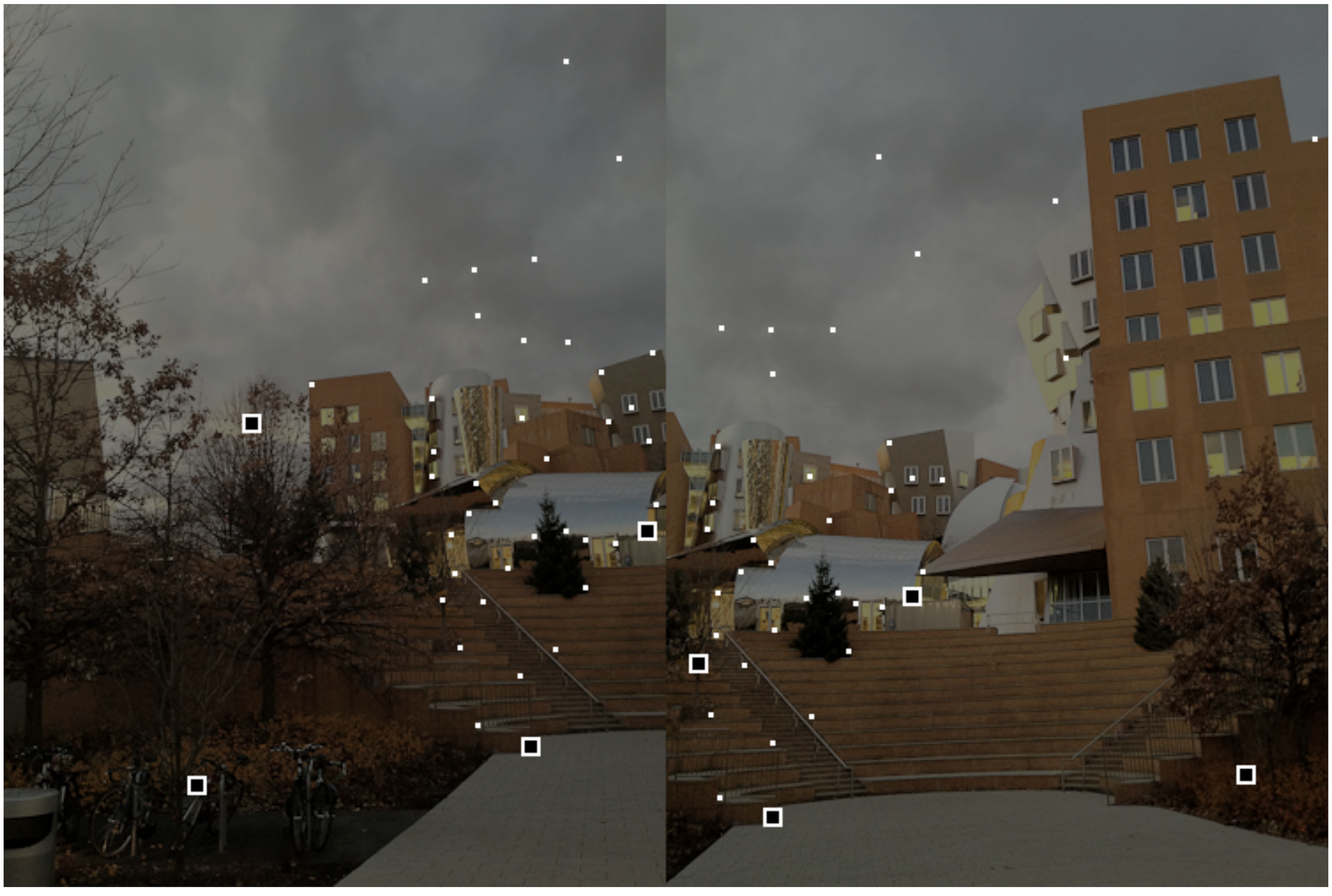
Re-compute least-squares H estimate on all of the inliers

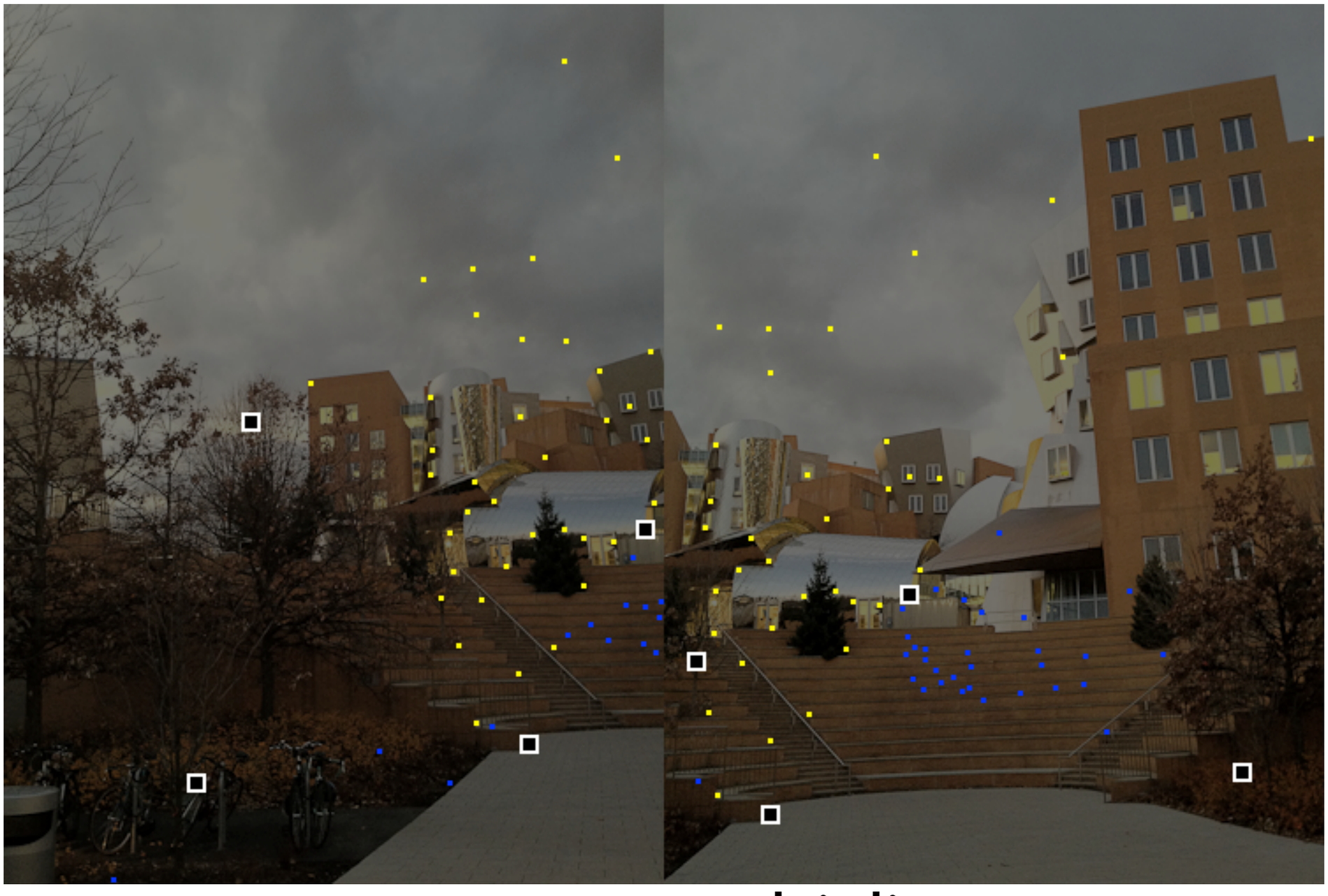
Correspondences



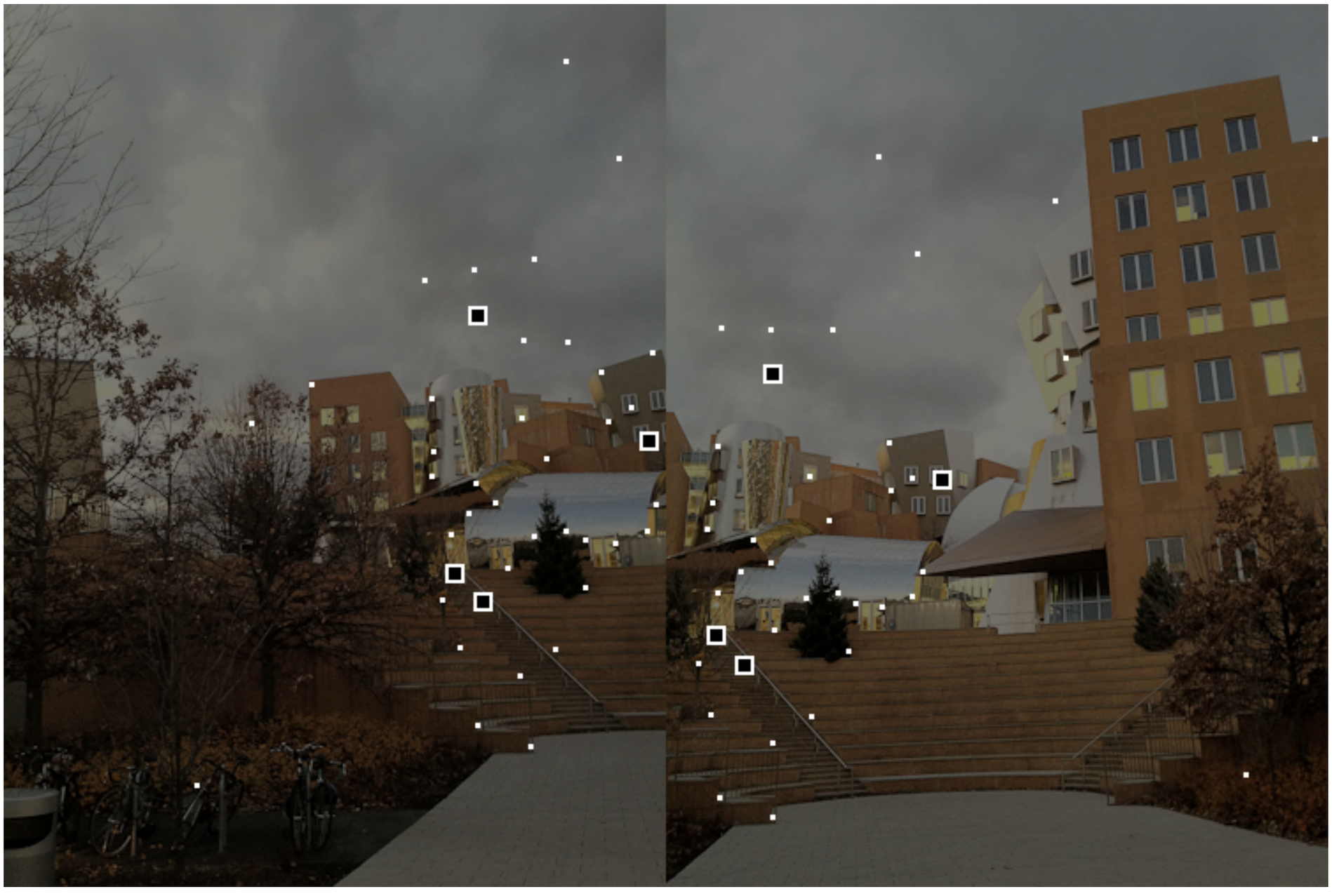
- I lowered the 2NN threshold to make things more interesting...

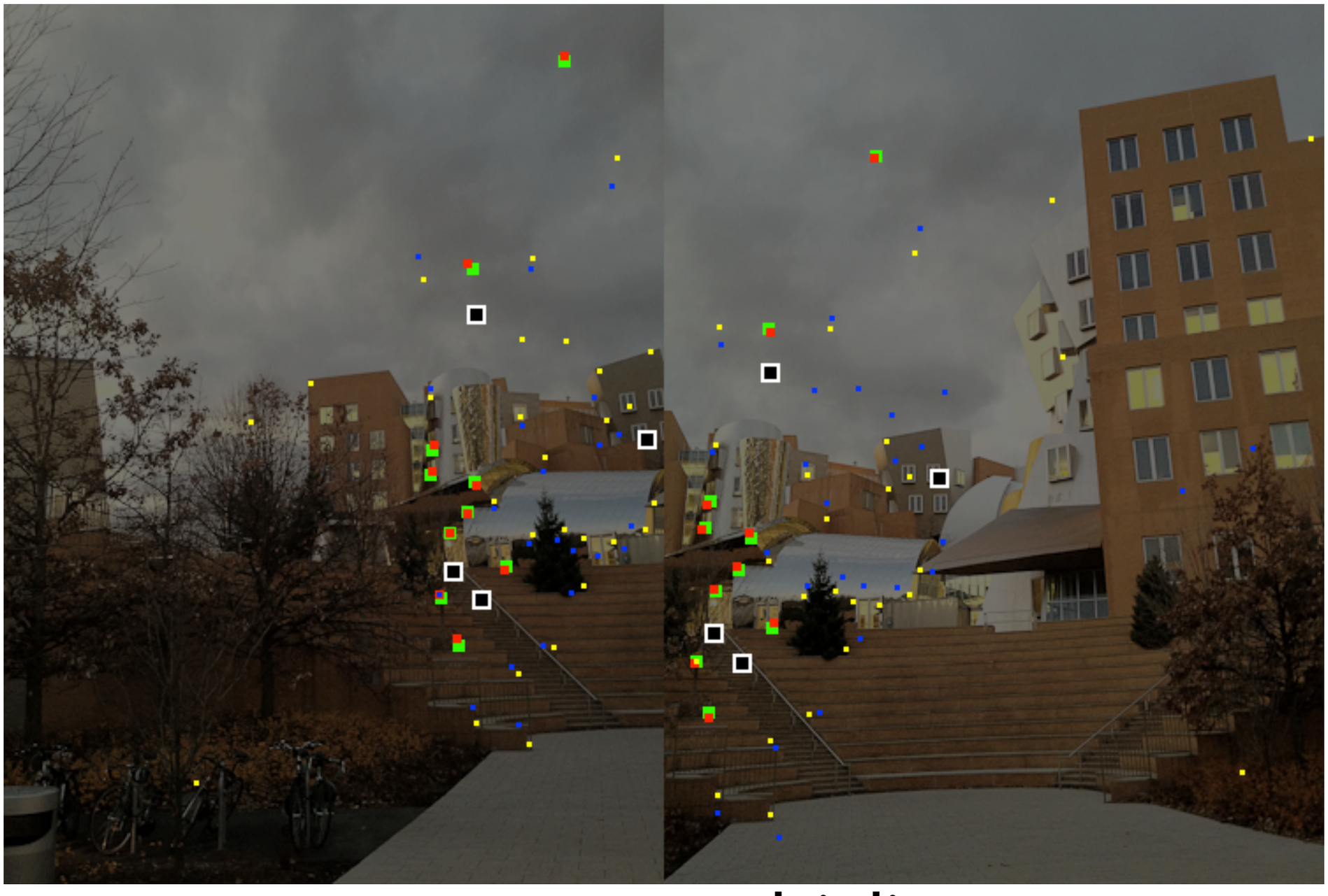
RANSAC iterations





green-on-red: inlier
yellow and blue: outlier

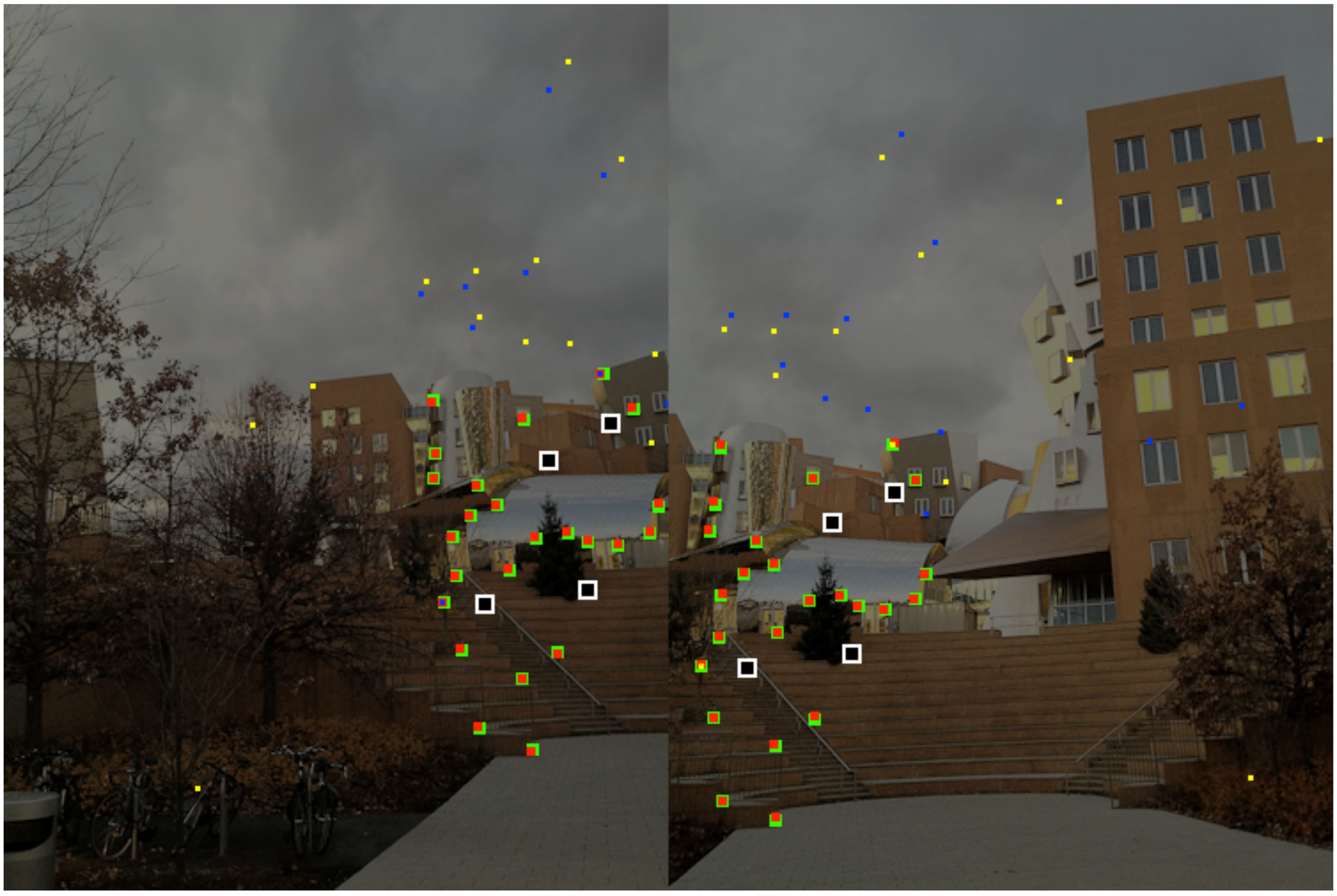




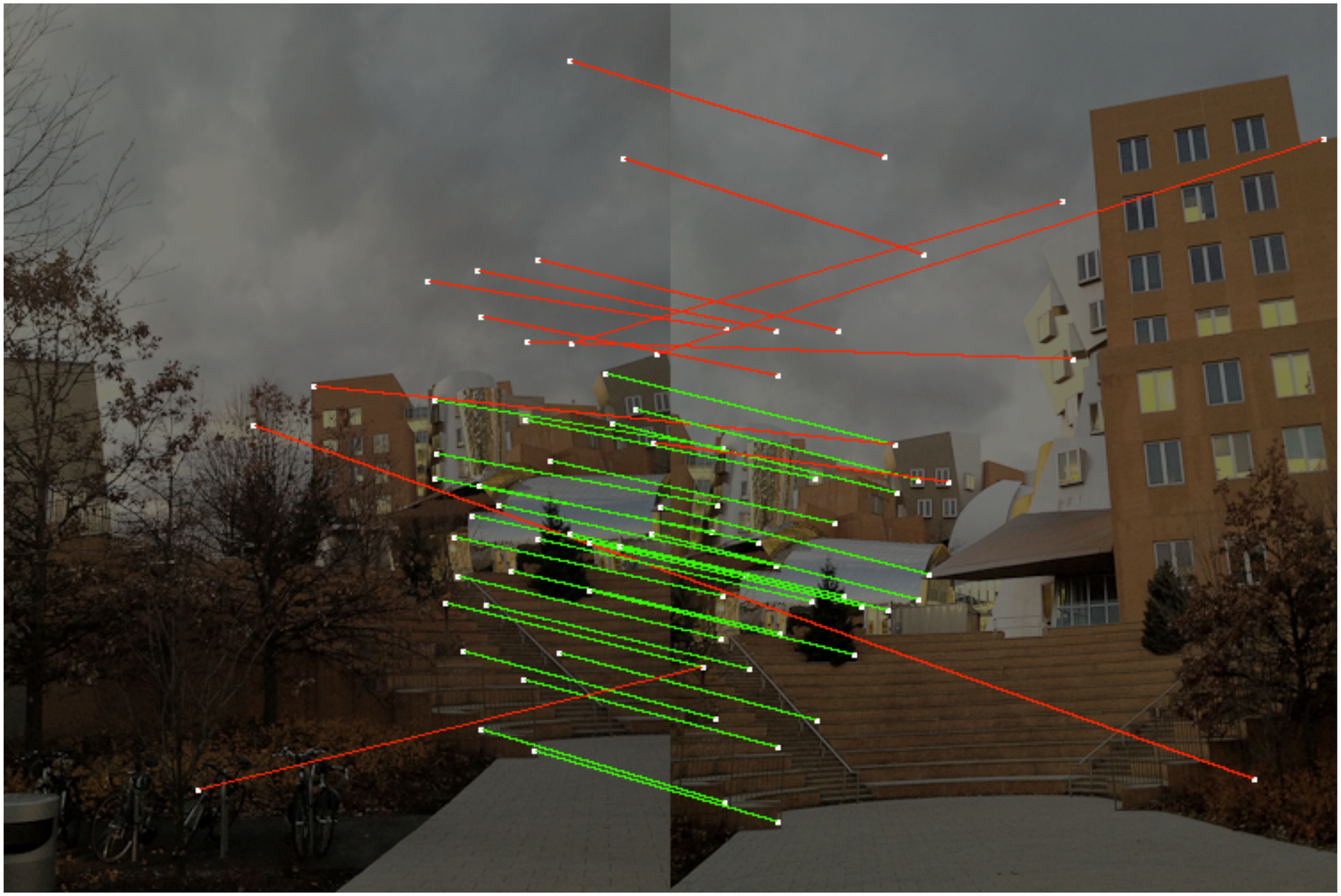
green-on-red: inlier

yellow and blue: outlier

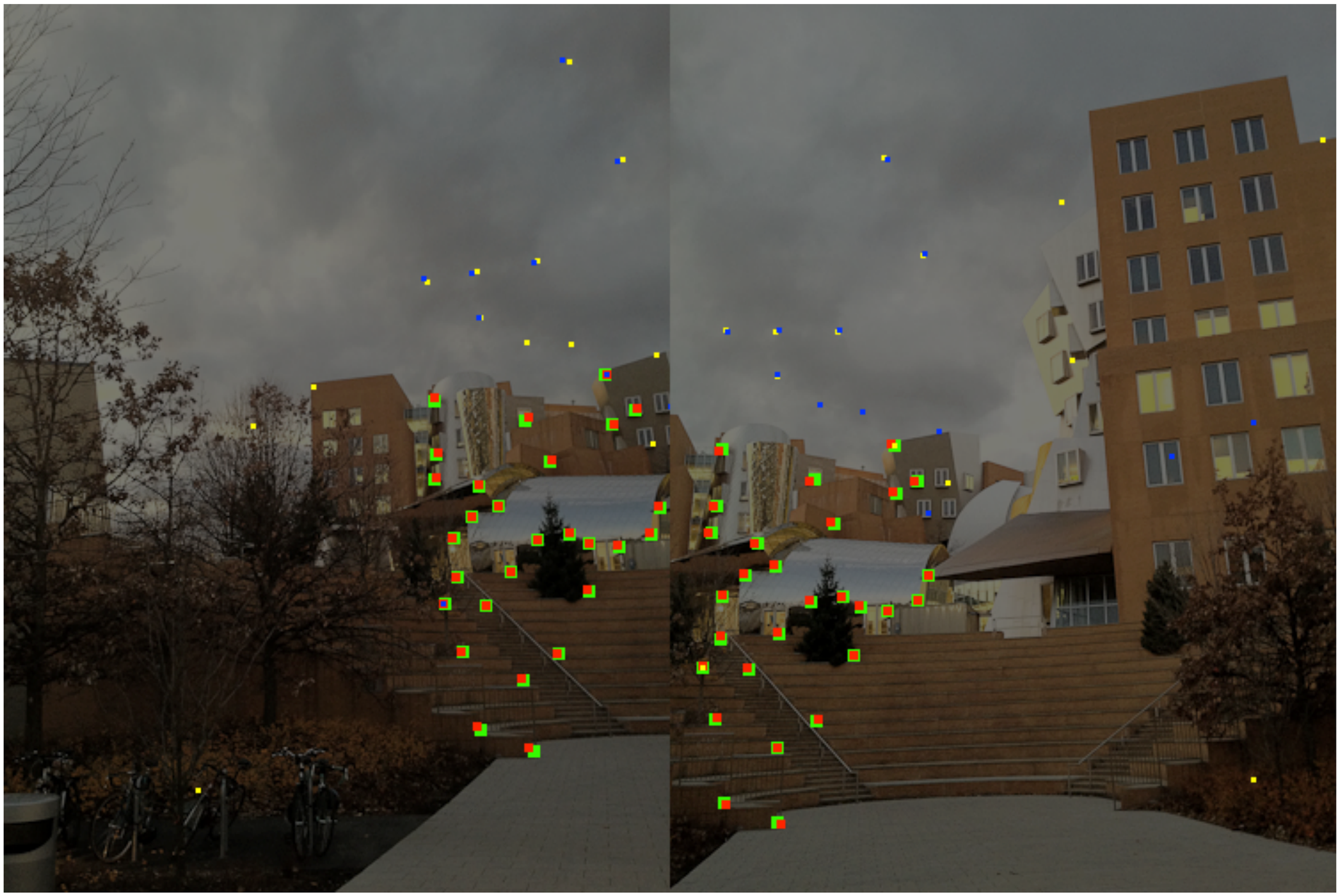




Resulting inliers

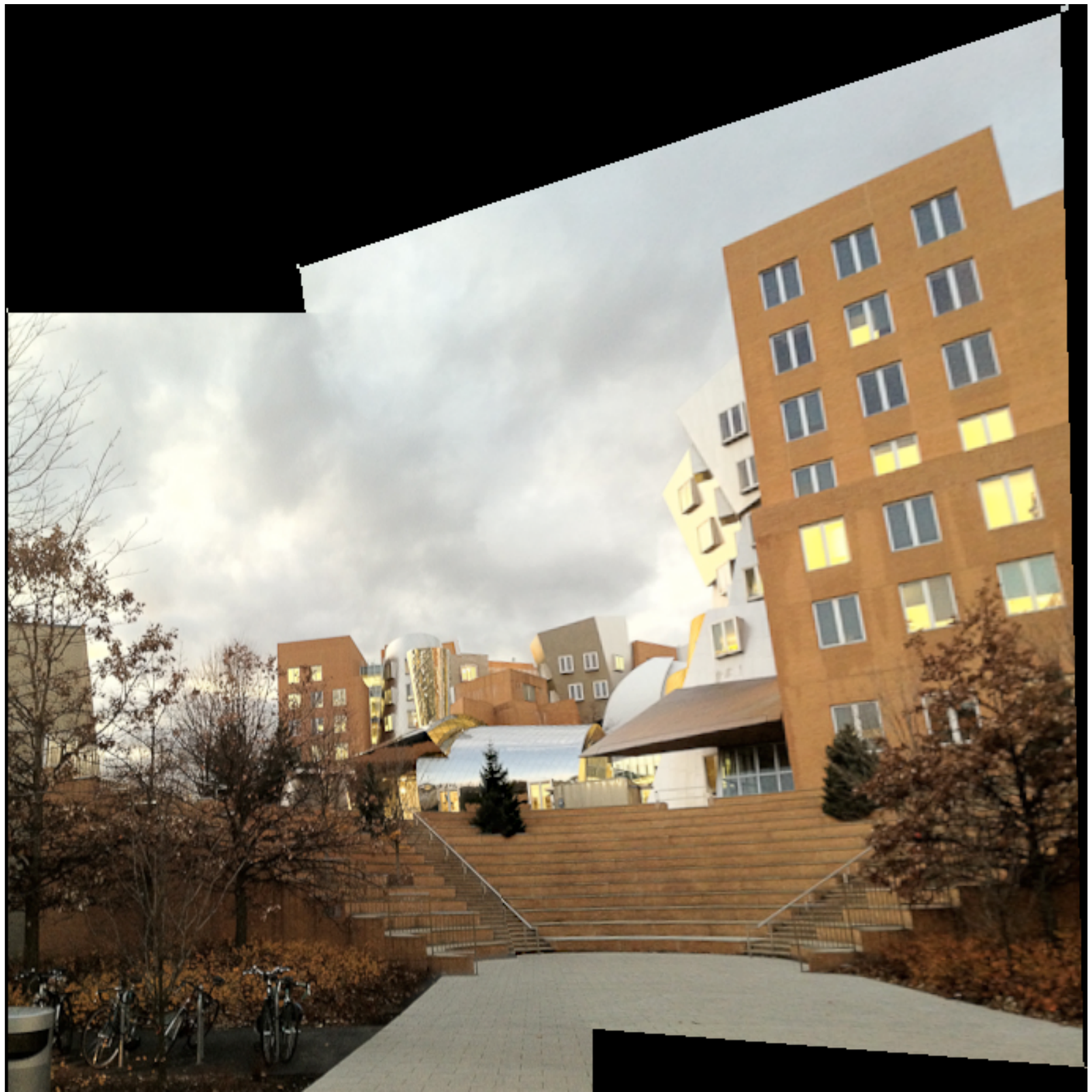


After reweighted least square

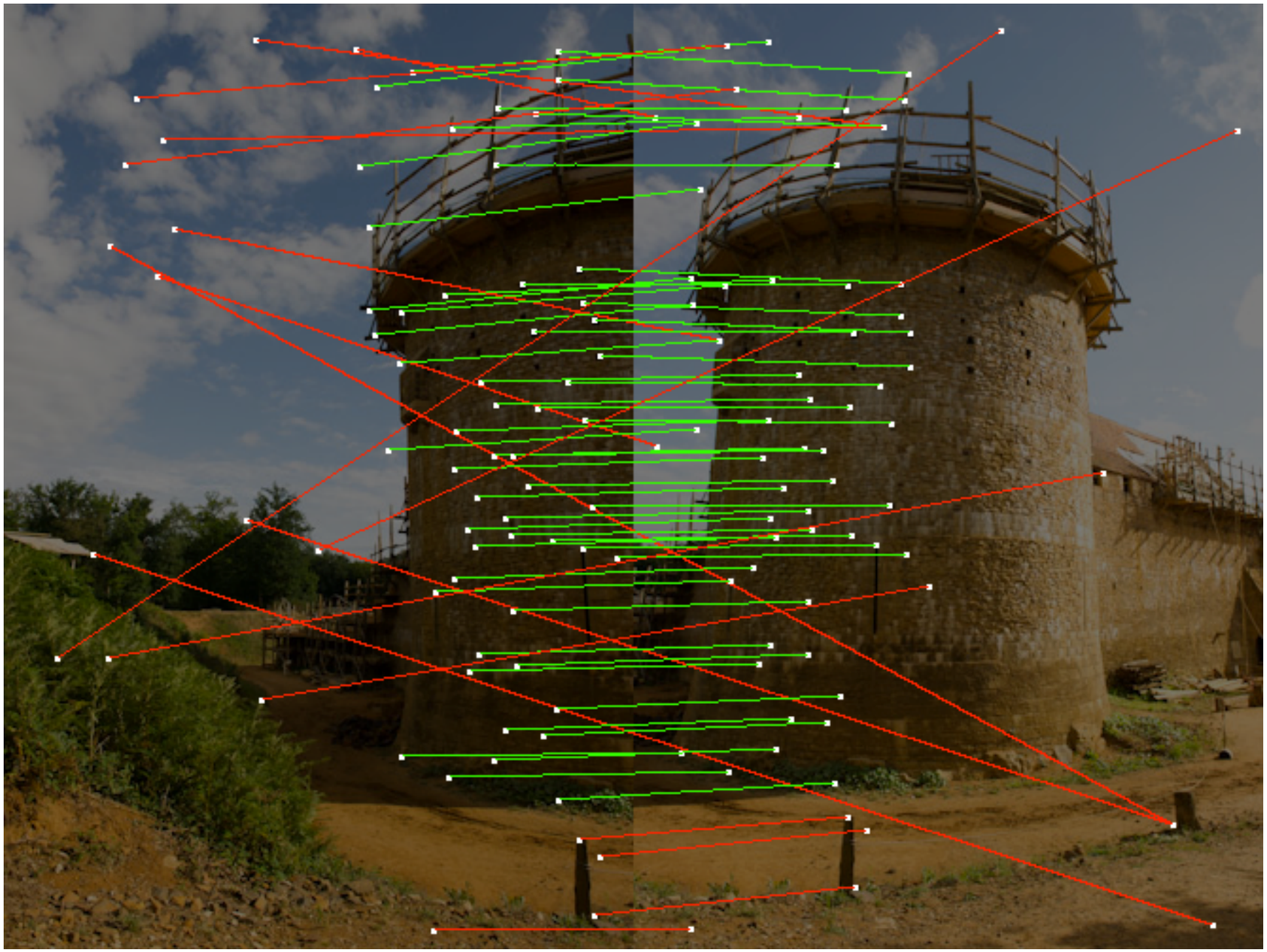


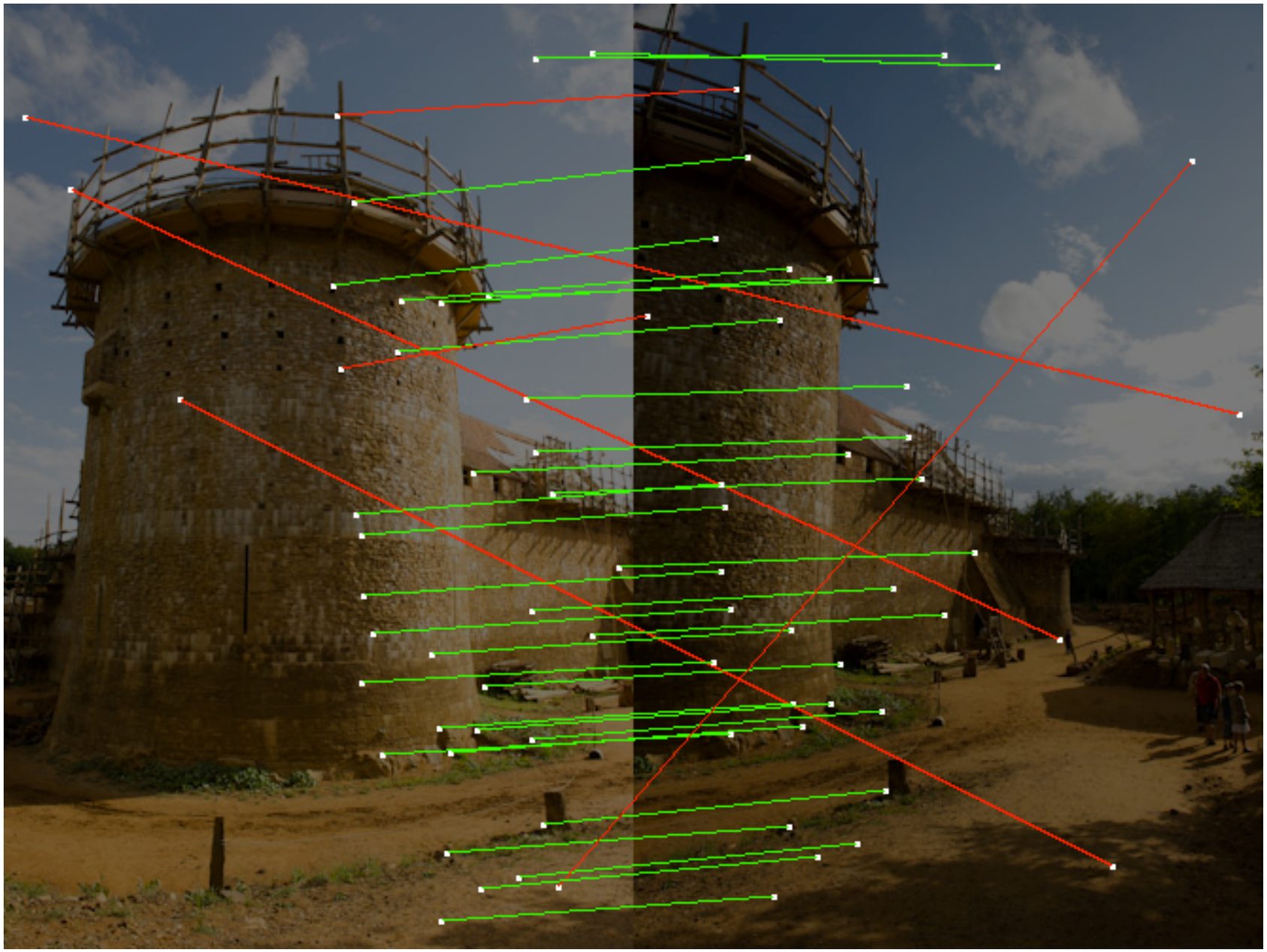
- Took into account near outliers

Reprojection



More results









Questions?

- http://en.wikipedia.org/wiki/Infinite_monkey_theorem



Probabilistic analysis

Robustness - problem statement

Robustness

- For now, assume we know proportion of inliers in our pairs is X
- Our model needs P pairs
  - P=4 for homography
- Probability that after N RANSAC iterations we succeed?

Probabilities

- Failure vs. Success
- AND vs. OR

Robustness

- Proportion of inliers in our pairs is X
- Our model needs P pairs
  - P=4 for homography

Robustness

- Proportion of inliers in our pairs is X
- Our model needs P pairs
  - P=4 for homography
- Probability that we pick P inliers?
  - X^P
- Probability that after N RANSAC iterations we have **not** picked a set of inliers?
  - $(1-X^P)^N$

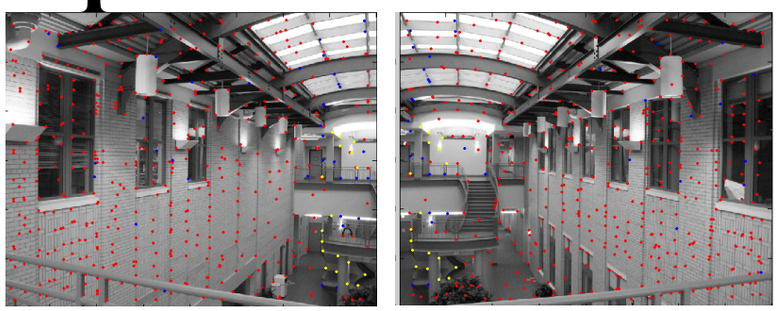
Robustness: example

- `Matlab: p=4; x=0.5; n=1000; (1-x^p)^n`
- Proportion of inliers **X=0.5**
- Probability that we pick P=4 inliers?
  - 0.5^4=0.0625 (6% chances)

Robustness: example

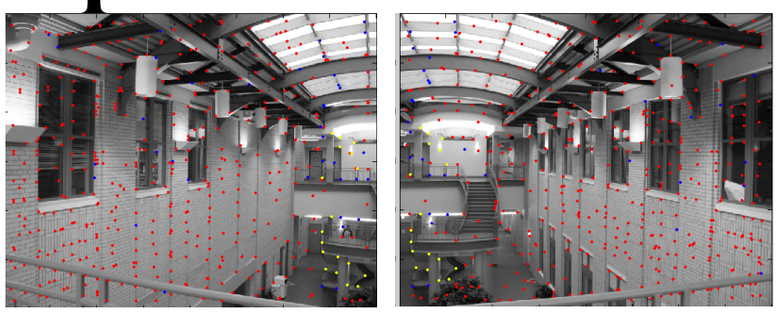
- `Matlab: p=4; x=0.5; n=1000; (1-x^p)^n`
- Proportion of inliers **X=0.5**
- Probability that we pick P=4 inliers?
  - 0.5^4=0.0625 (6% chances)
- Probability that we have **not** picked a set of inliers?
  - N=100 iterations:
    $(1-0.5^4)^{100}$=0.00157 (1 chance in 600)
  - N=1000 iterations:
    1 chance in 1e28

Robustness: example



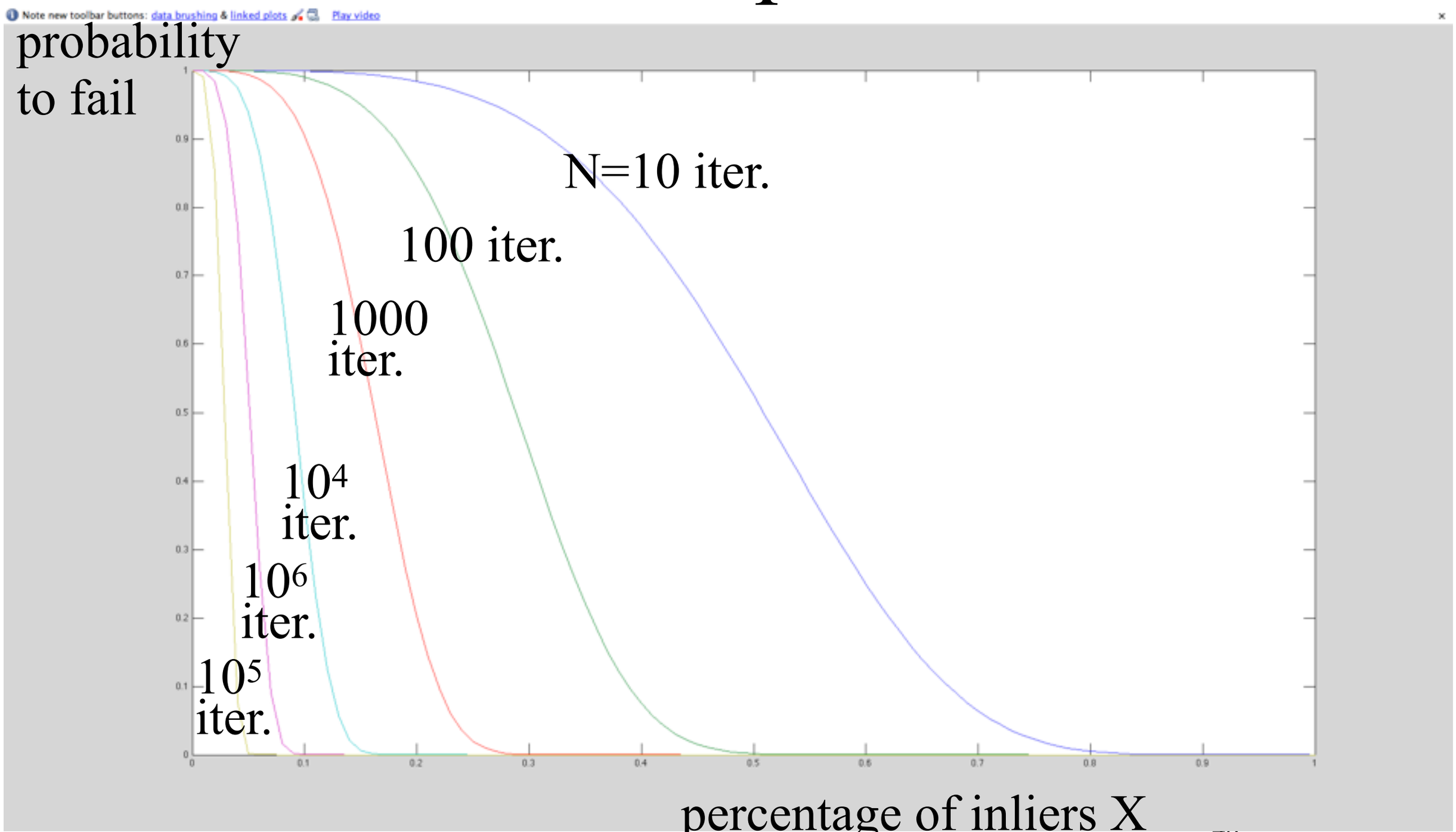
- Proportion of inliers **X=0.3**
- Probability that we pick P=4 inliers?
  - 0.3^4=0.0081 (0.8% chances)
- Probability that we have **not** picked a set of inliers?
  - N=100 iterations:
    $(1-0.3^4)^{100}$=0.44 (1 chance in 2)
  - N=1000 iterations:
    1 chance in 3400

Robustness: example



- Proportion of inliers **X=0.1**
- Probability that we pick P=4 inliers?
  - 0.1^4=0.0001 (0.01% chances, 1 in 10,000)
- Probability that we have **not** picked a set of inliers?
  - N=100 iterations: $(1-0.1^4)^{100}$=0.99
  - N=1000 iterations: 90%
  - N=10,000: 36%
  - N=100,000: 1 in 22,000

Graph


Note new toolbar buttons: data brushing & linked plots
Play video
probability to fail
N=10 iter.
100 iter.
1000 iter.
10^4 iter.
10^6 iter.
10^5 iter.
percentage of inliers X

Robustness: conclusions

- Effect of number of parameters of model/ number of necessary pairs
  - Bad exponential
- Effect of percentage of inliers
  - Base of the exponential
- Effect of number of iterations
  - Good exponential

RANSAC recap

- For fitting a model with low number P of parameters (8 for homographies)
- Loop
  - Select P random data points
  - Fit model
  - Count inliers
    (other data points well fit by this model)
- Keep model with largest number of inliers

Every line of vanilla RANSAC has been studied

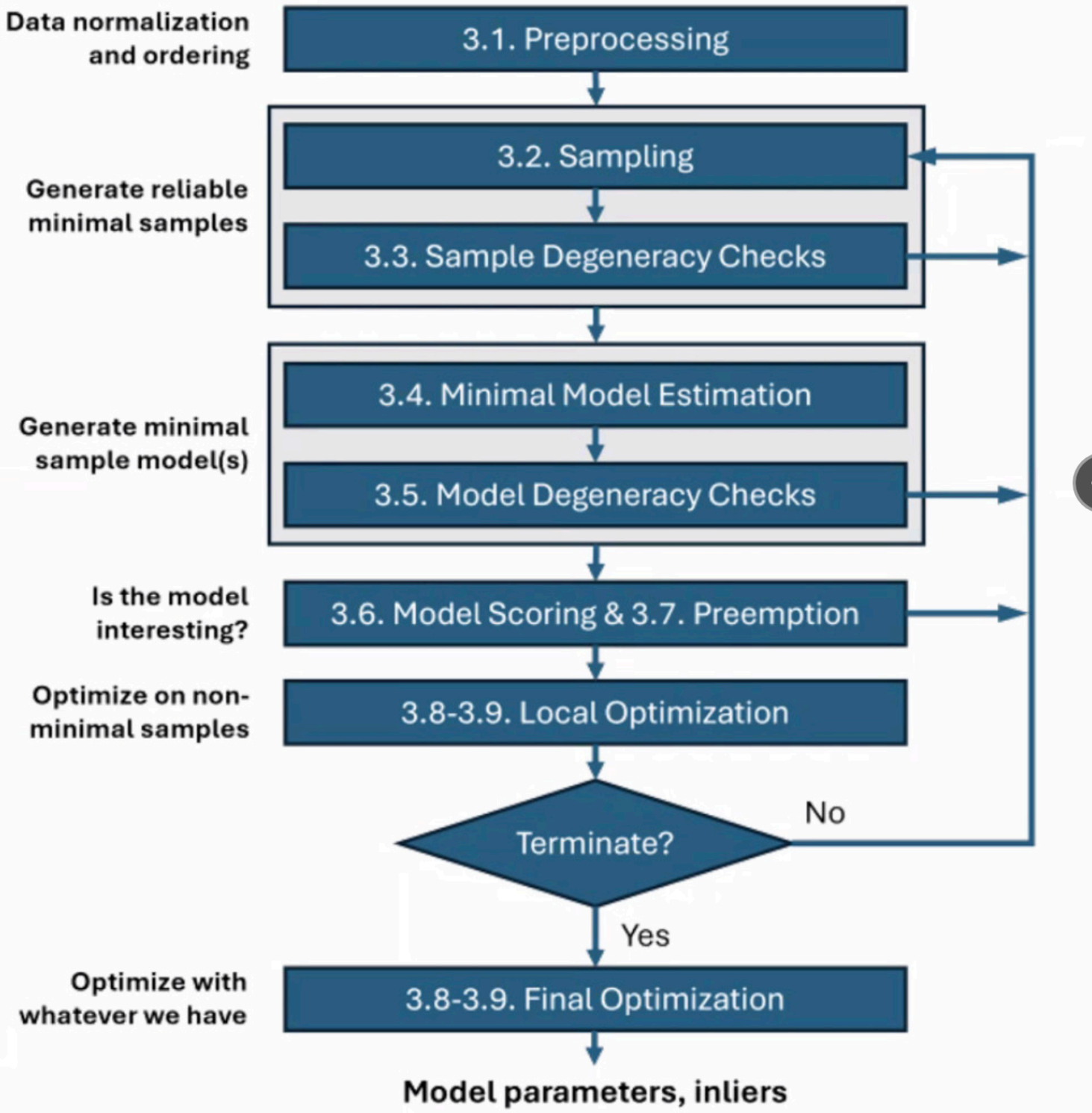
... and modification proposed.

The SOTA is covered by Daniel Barath's



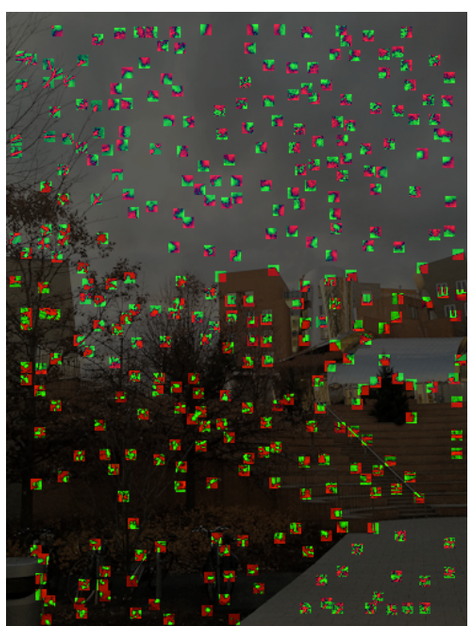
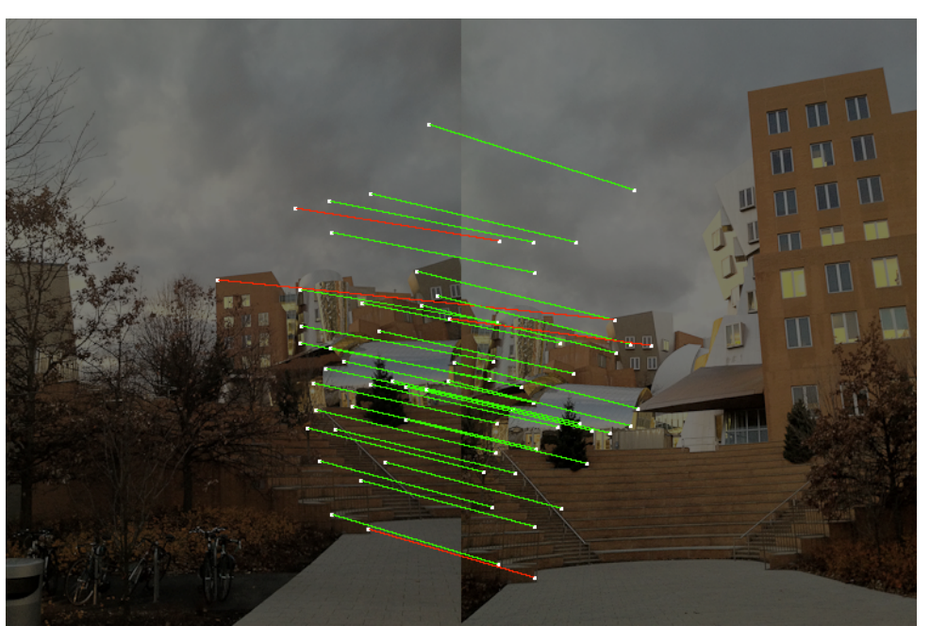
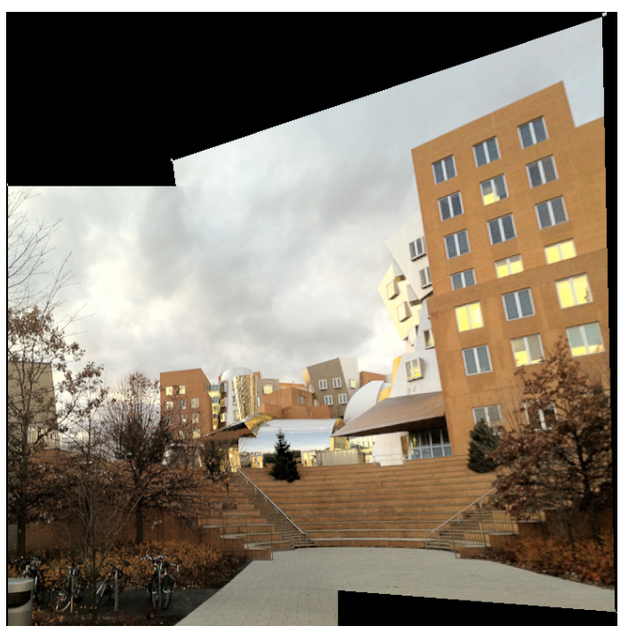
Local and Final Optimization Covered by Viktor Larsson



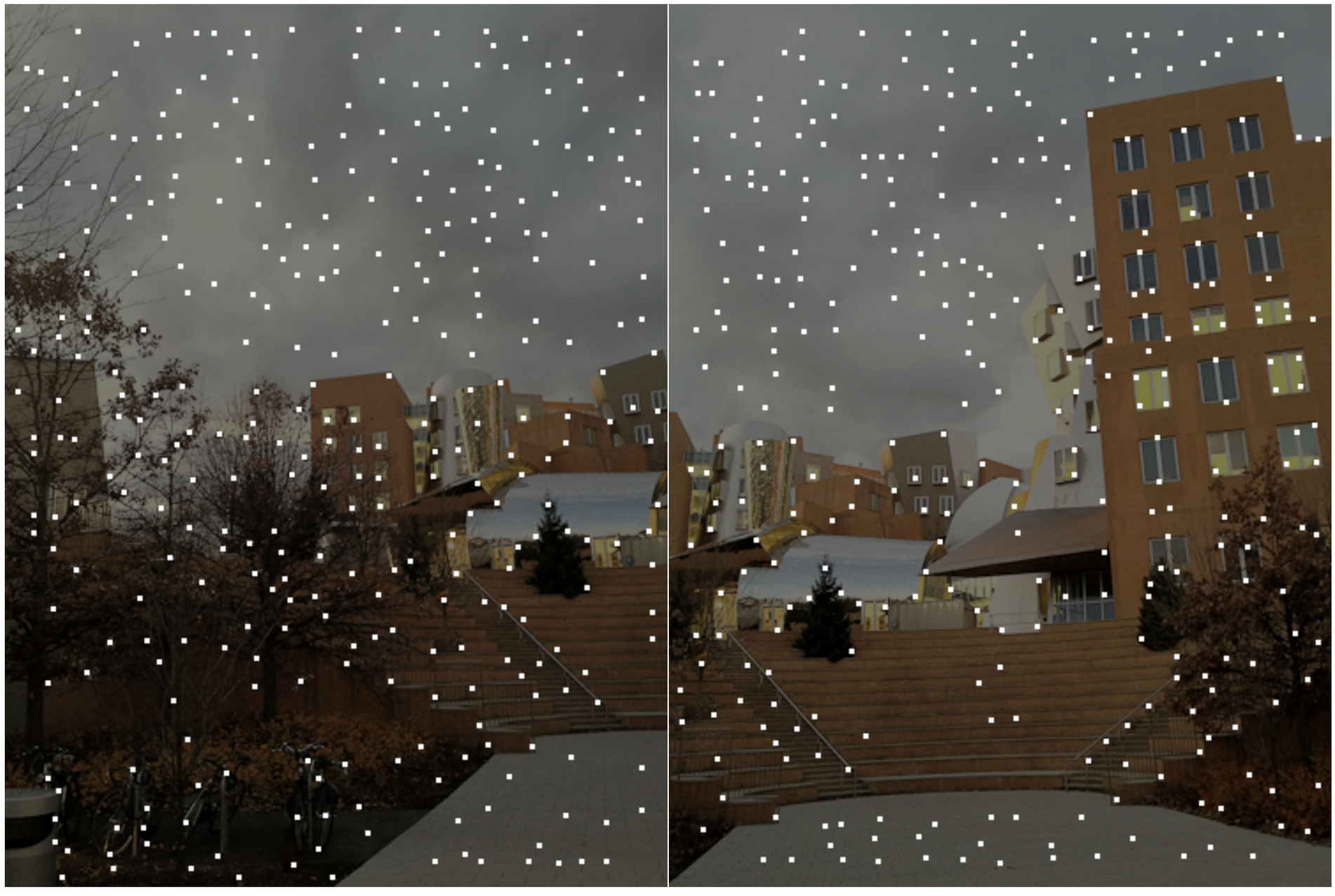


Recap: Auto pano

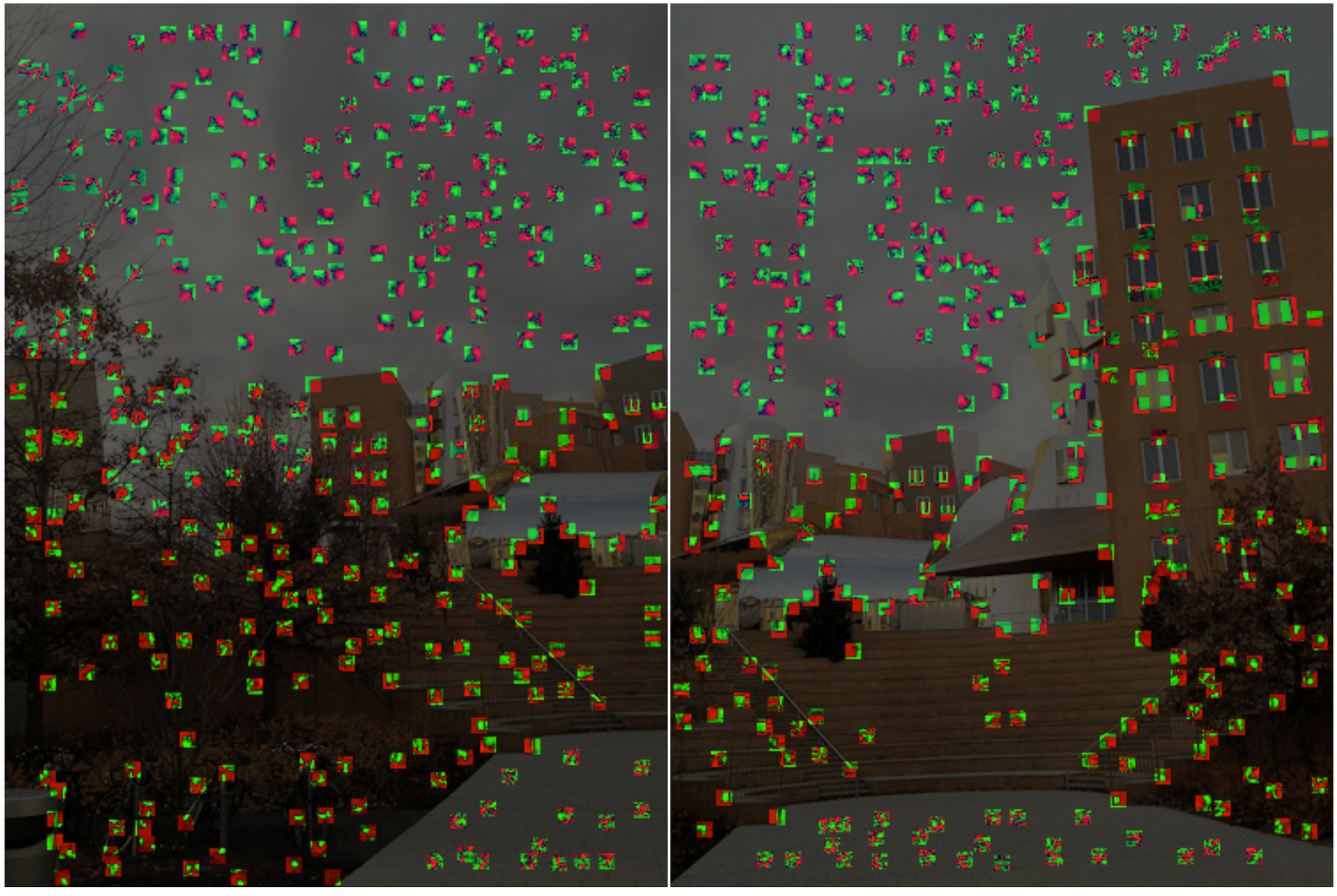
- Extract Interest points (Harris coner)
- Match with descriptors (9x9 patches)
- Second-Nearest-Neighbor test rejects bad matches
- RANSAC fits homography, robust to outliers

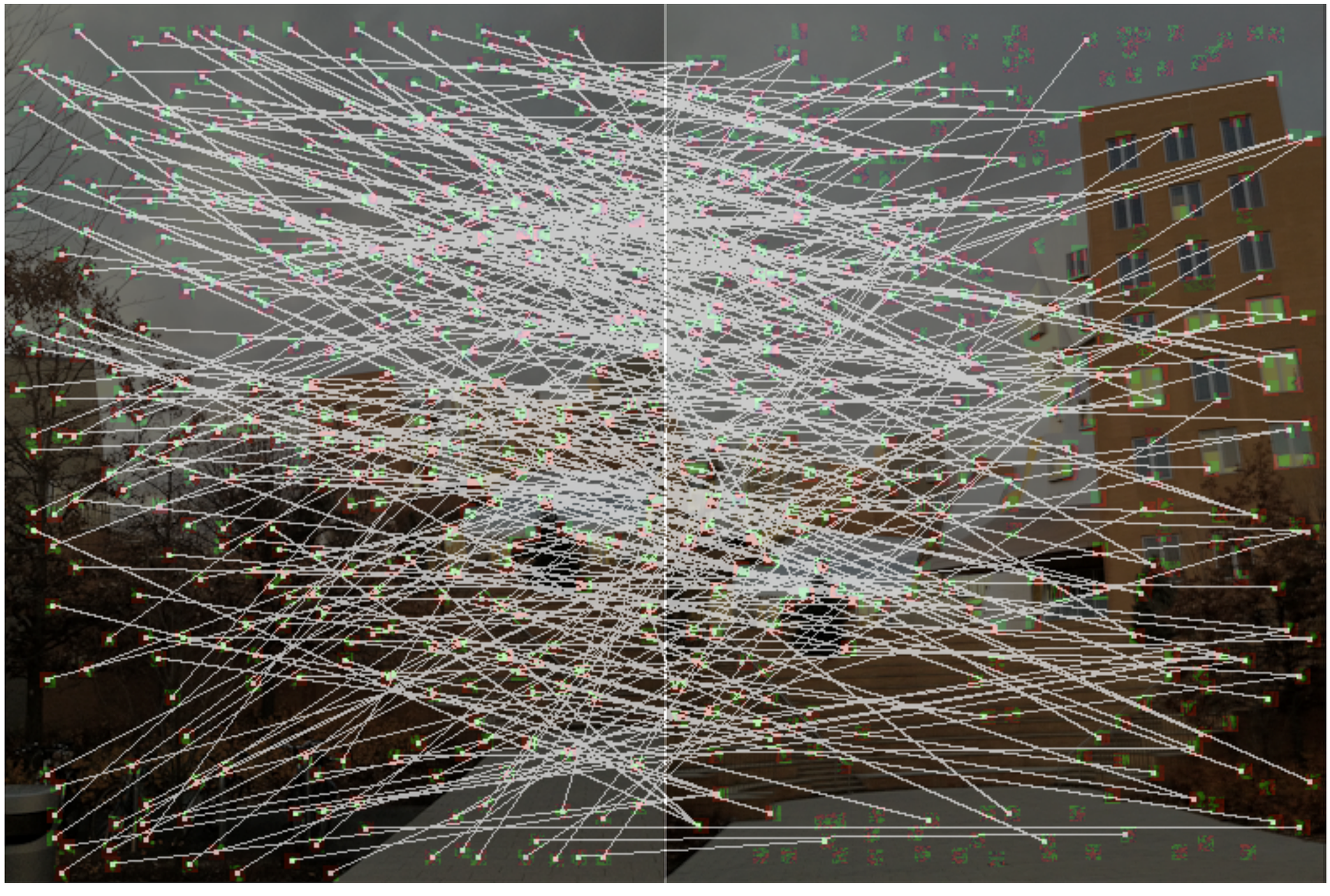
Extract Interest points (Harris)



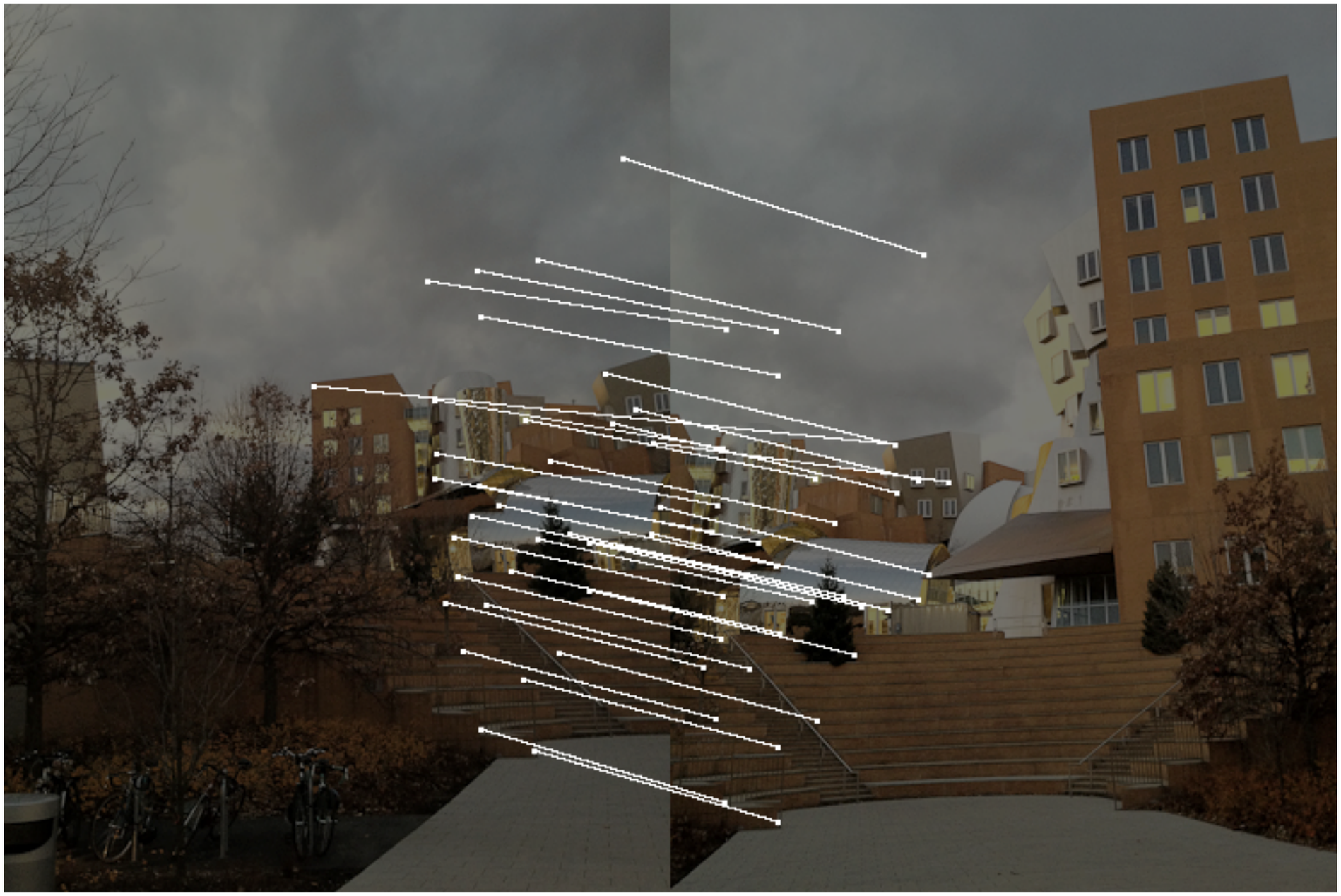
Descriptors



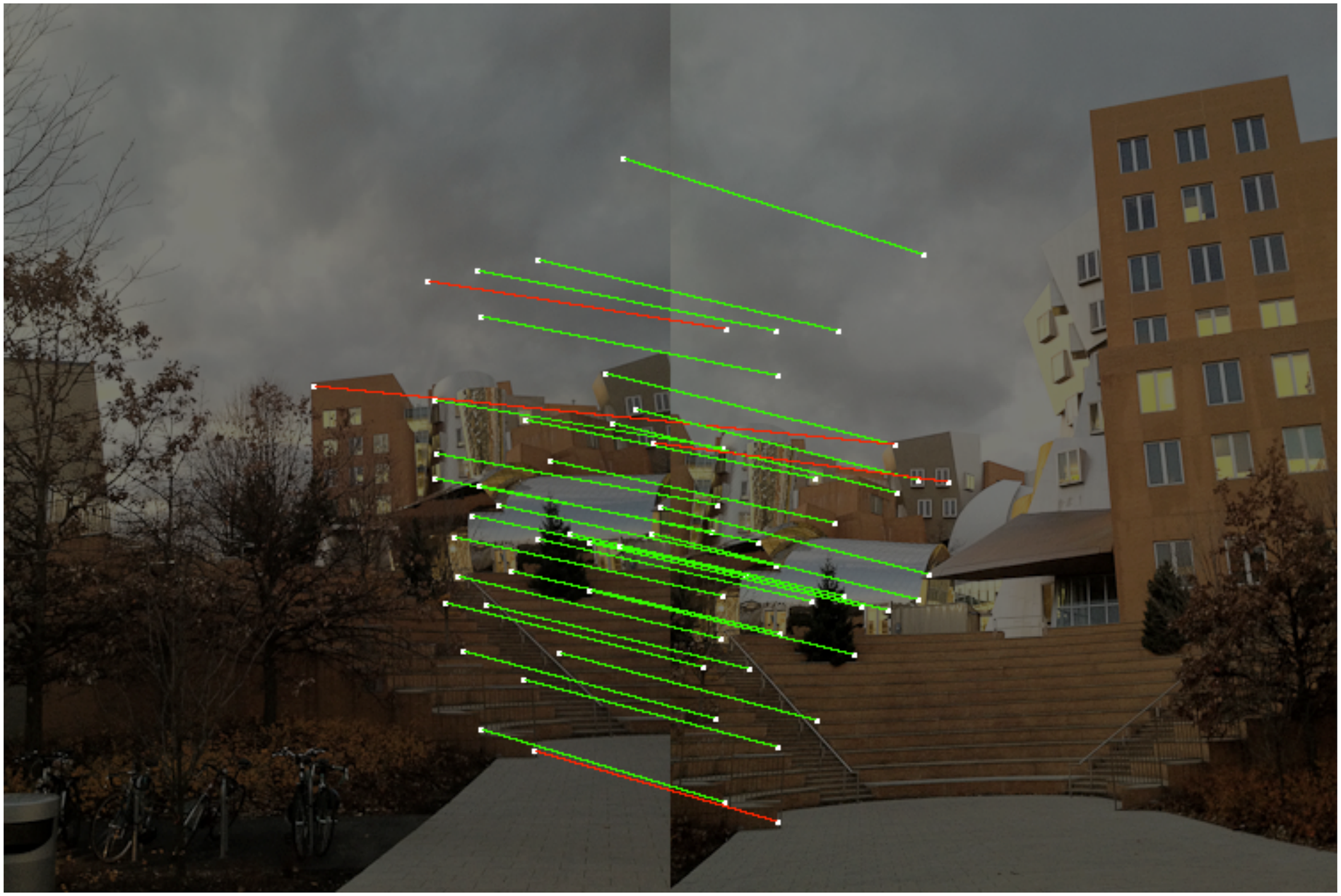
After NN match



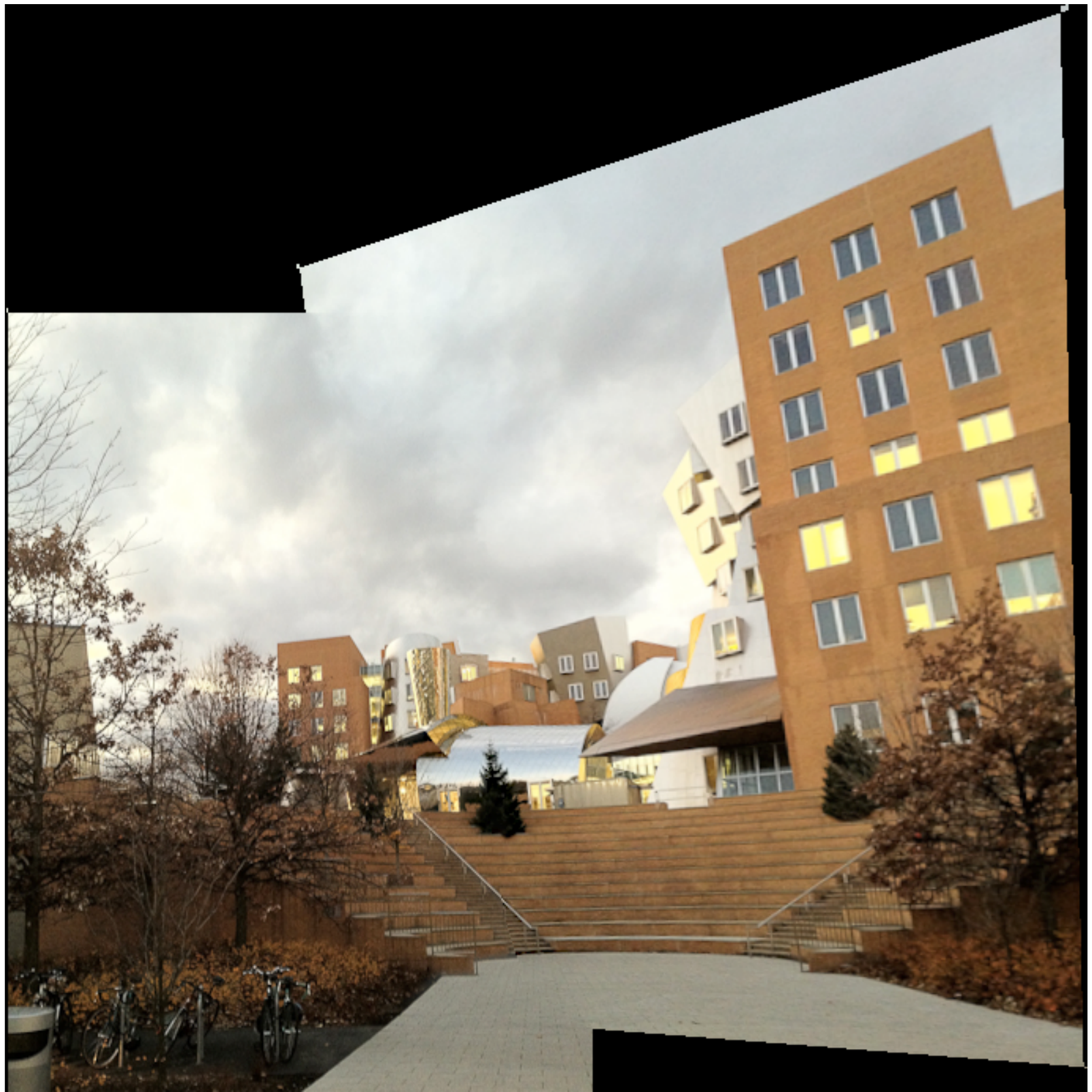
After match & 2NN test



After Ransac



Result with blending



Mini break

More fun with homographies

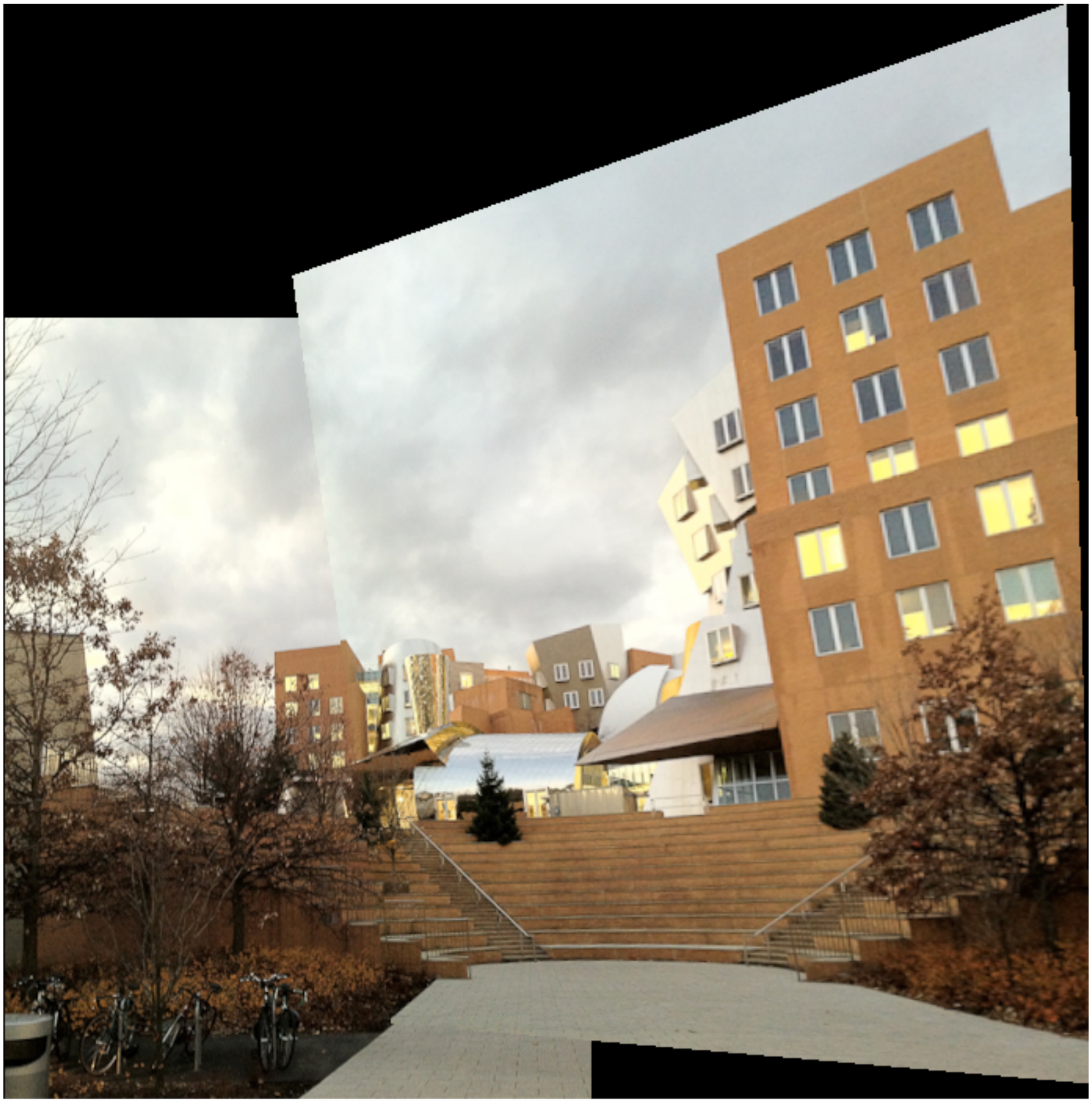
- Frame of Mind from Steven Alan http://vimeo.com/38940289



- http://www.youtube.com/watch?v=TOnif6BC0kc&feature=youtu.be

Blending

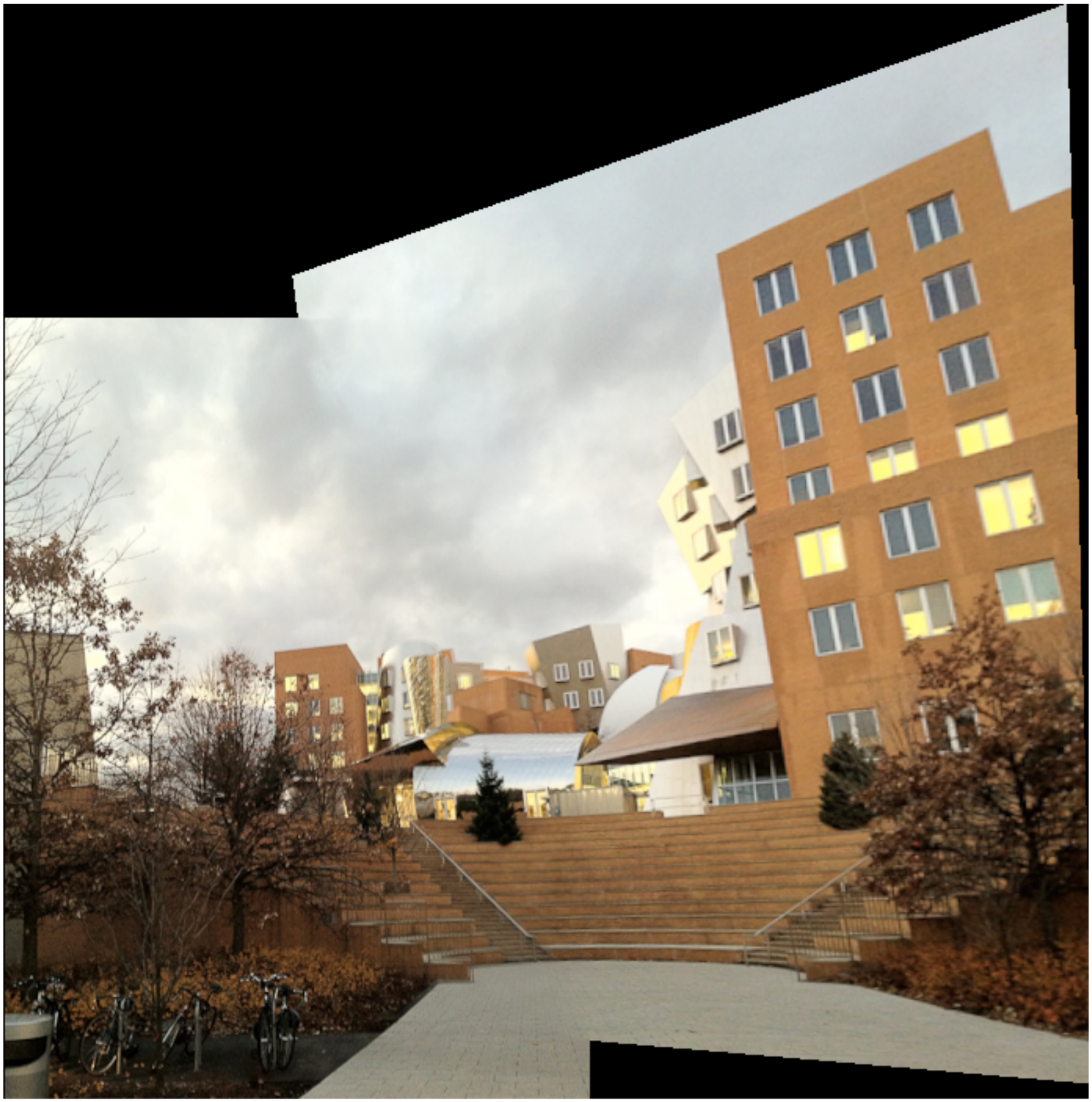
The problem



The problem



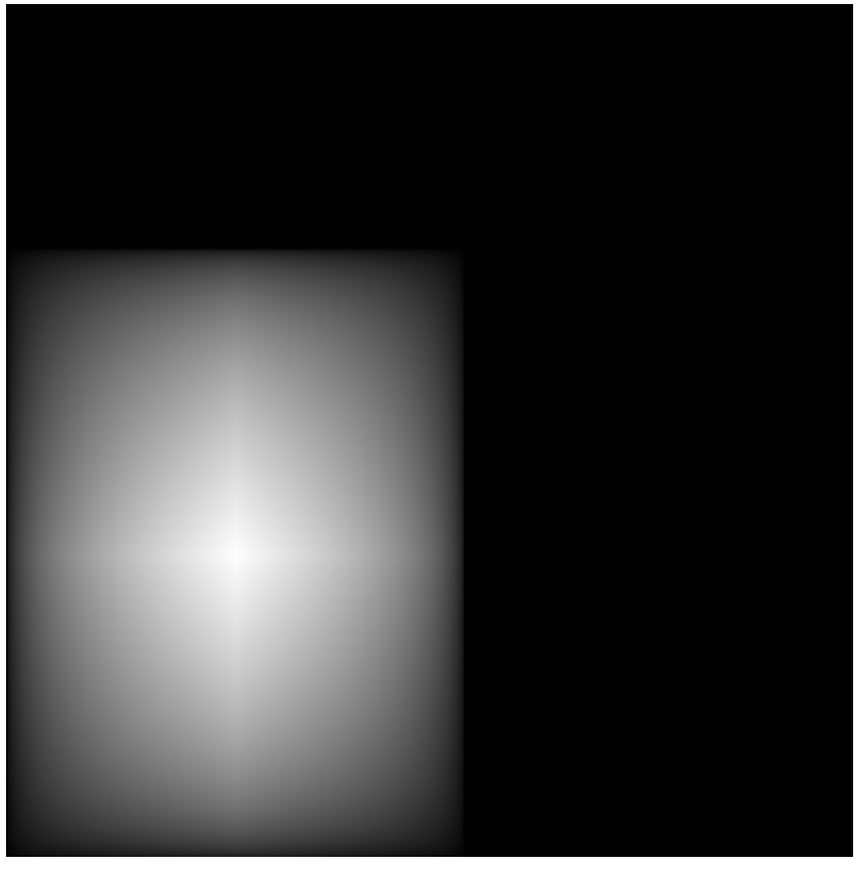
Solution 1: smooth blending



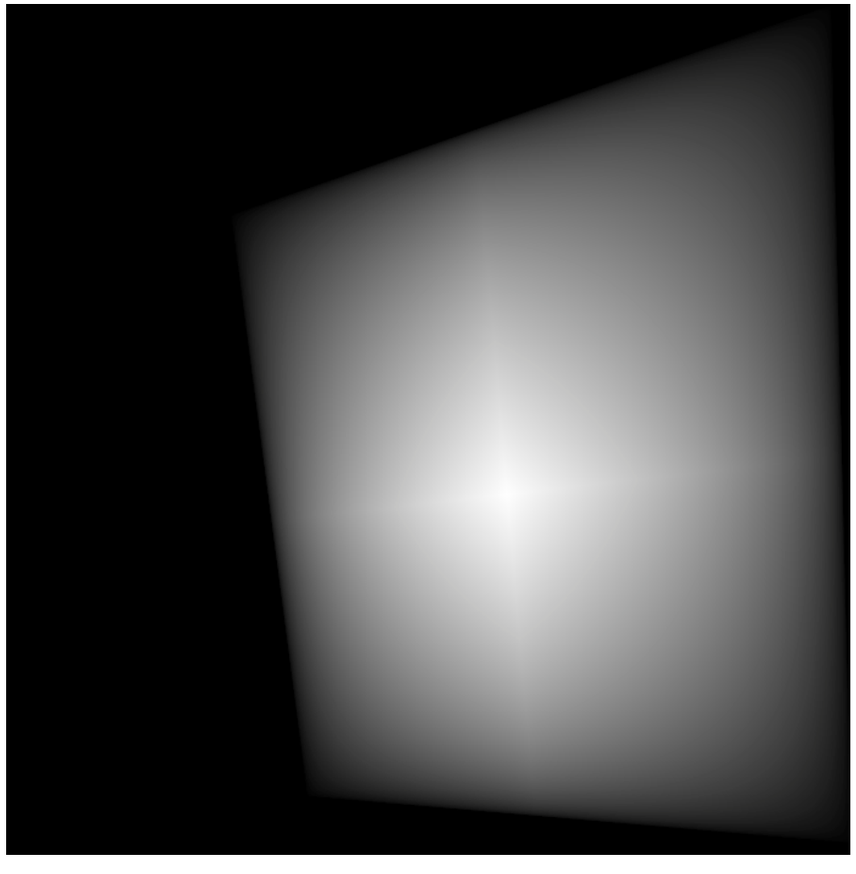
Solution 1: smooth blending

- Each output pixel is weighted average of images
- Weight depends on distance to source boundary





Weights of image 1

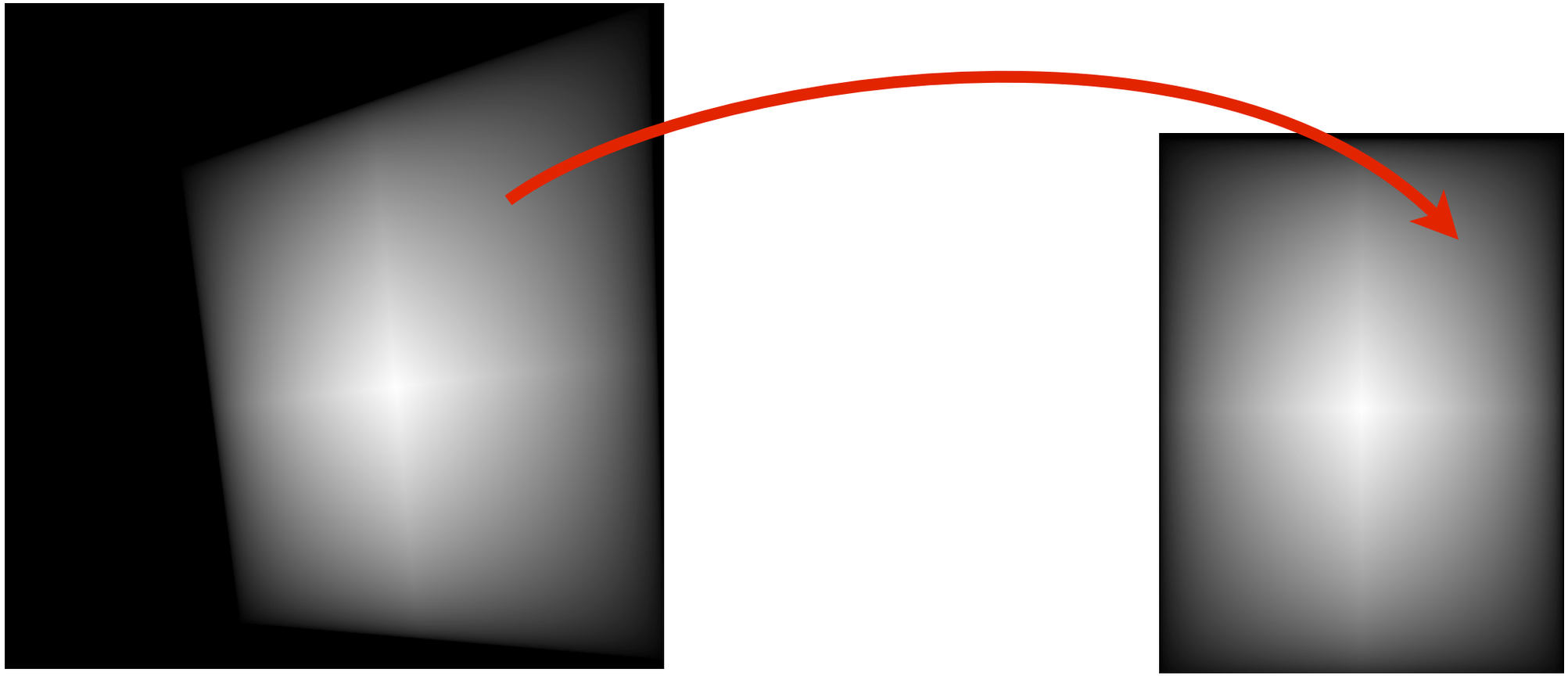


Weights of image 2



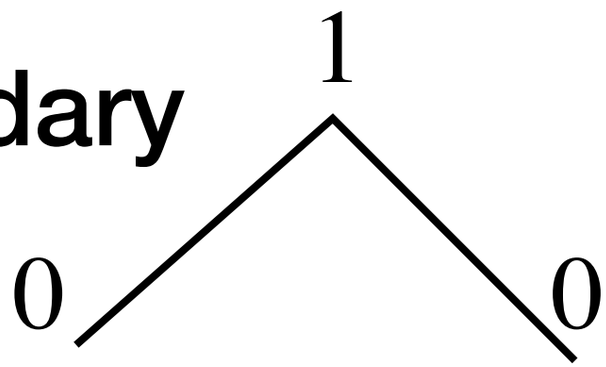
Computing distance

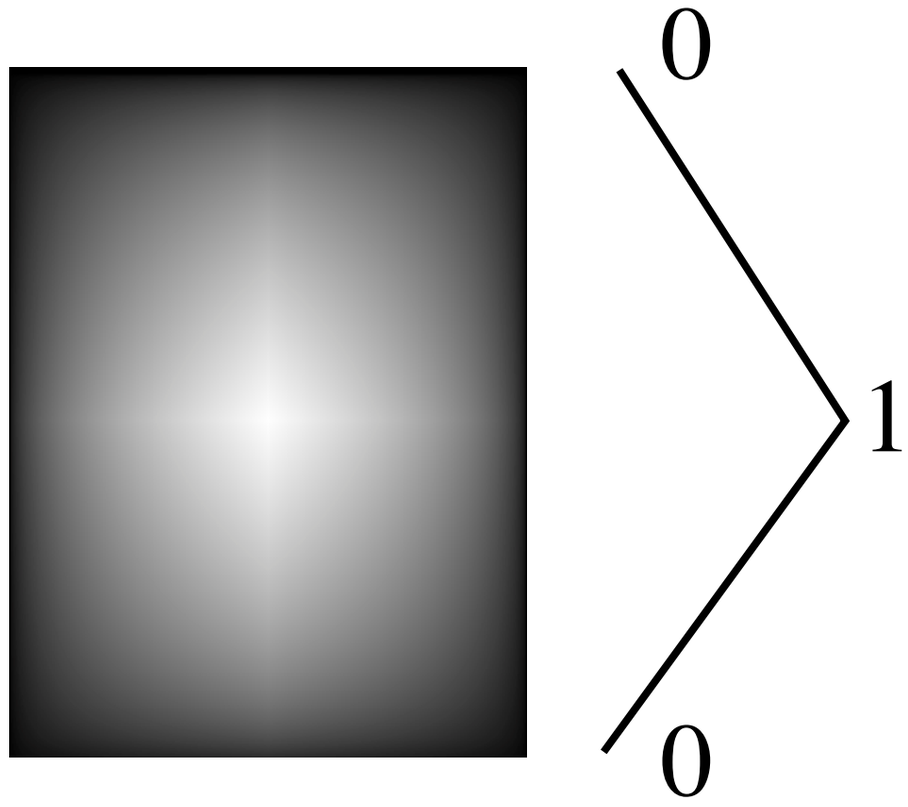
- Complex in output image because of homography
- But easy in the source space



Weights

- Separable in y and x
- Piecewise-linear
- 1 in the middle, 0 at boundary





Keep track of sum of weights

- My advice: have an extra image that maintains the sum of the weights for each pixel.

Basic reprojection



Smooth blending



Smooth blending: problem



Smooth blending: problem







Smooth blending: problem

- Ghosting
- Blurring





Two-scale Blending

Lessons learnt

- Basic reprojection has transition issues
- Smooth blending leads to ghosting & blurring





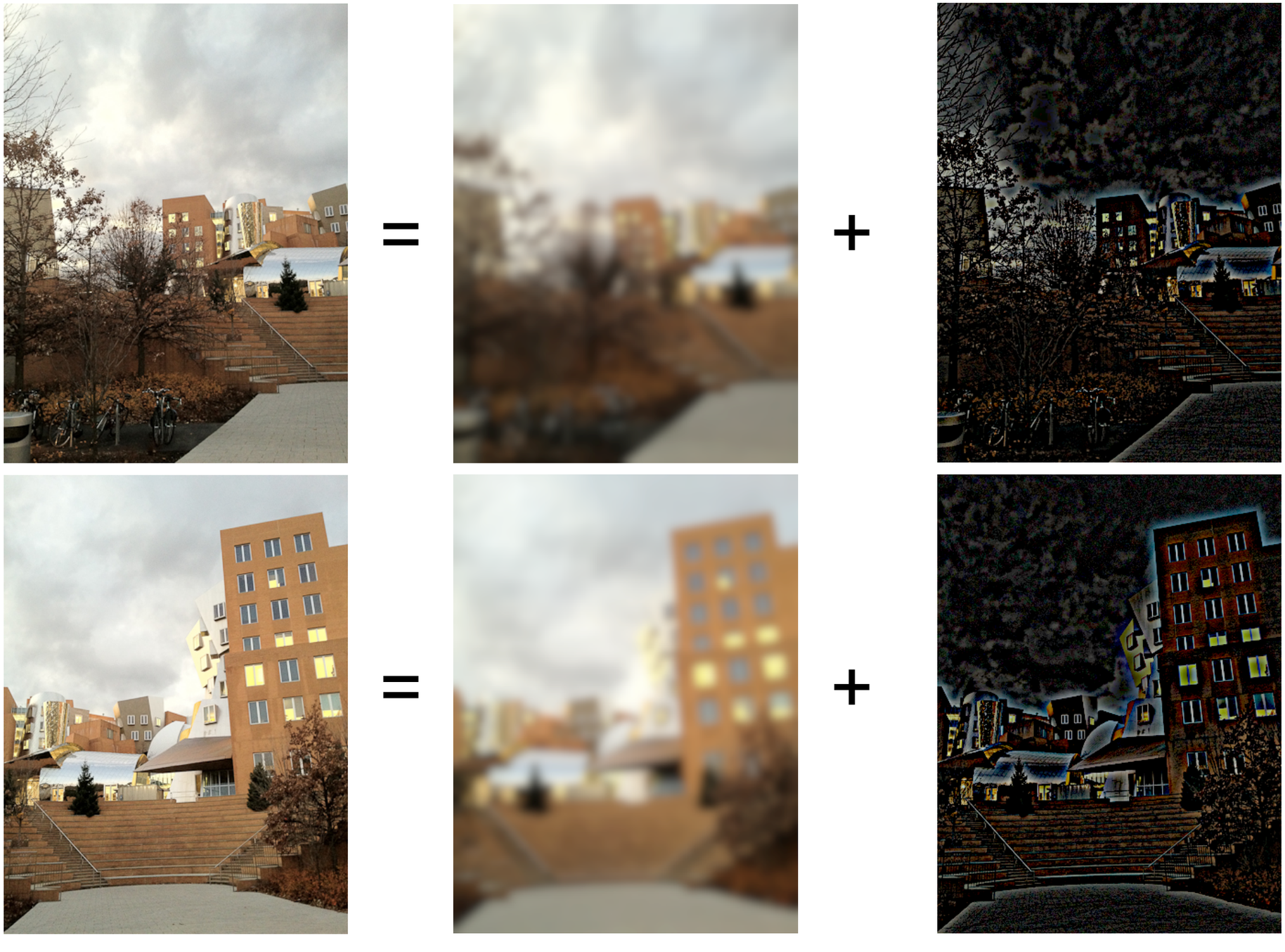


Solution: multiscale

- Combine smooth and abrupt transition
- Depending on scale:
  - smooth for low frequencies
    - Corrects effects like vignetting or exposure differences
  - abrupt for high frequencies
    - preserve sharpness, avoids duplicated objects

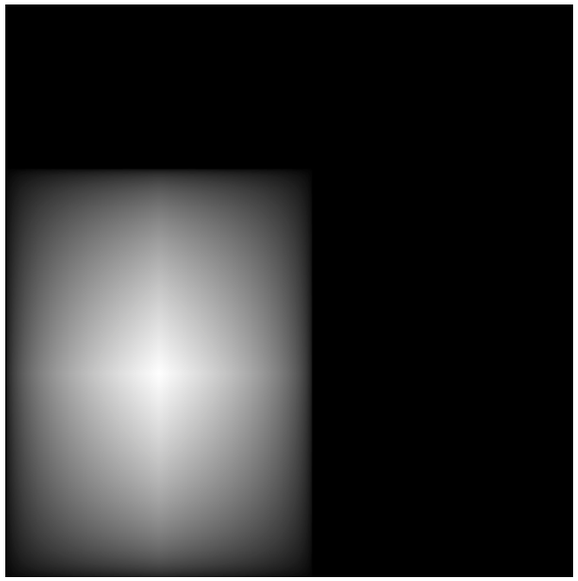
2-scale separation

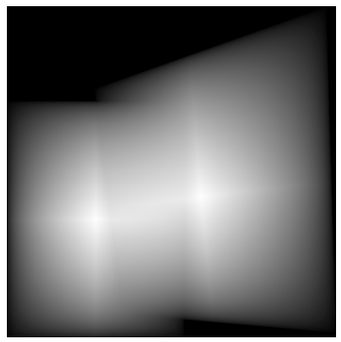
- Use Gaussian blur to split

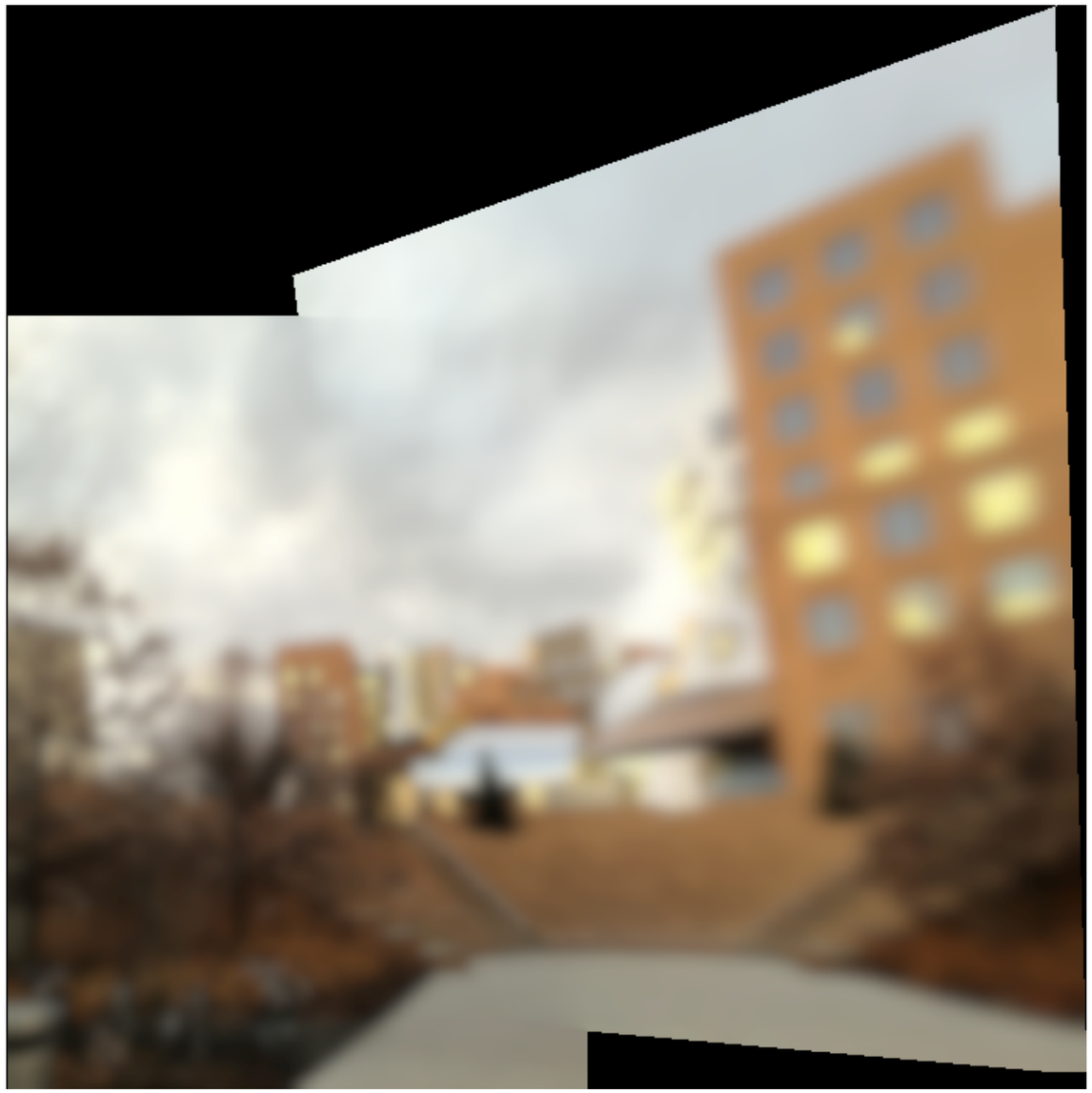


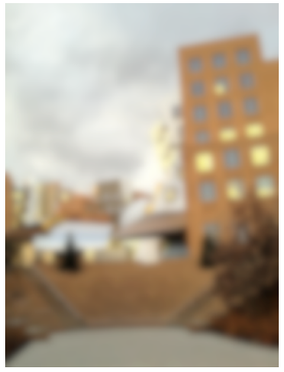
Low frequency: smooth blending

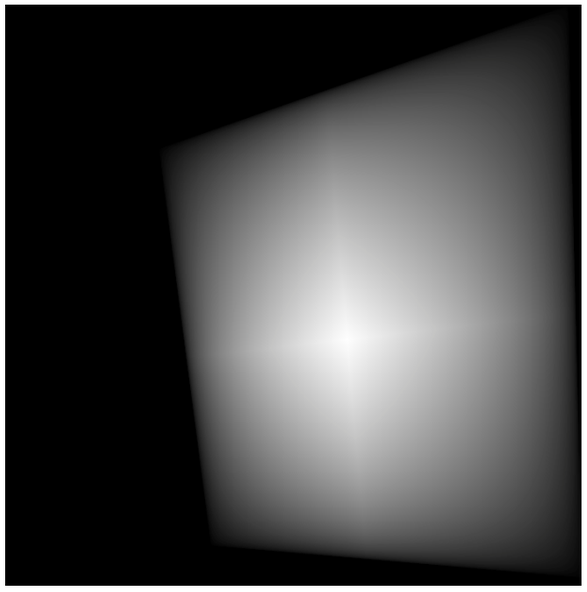










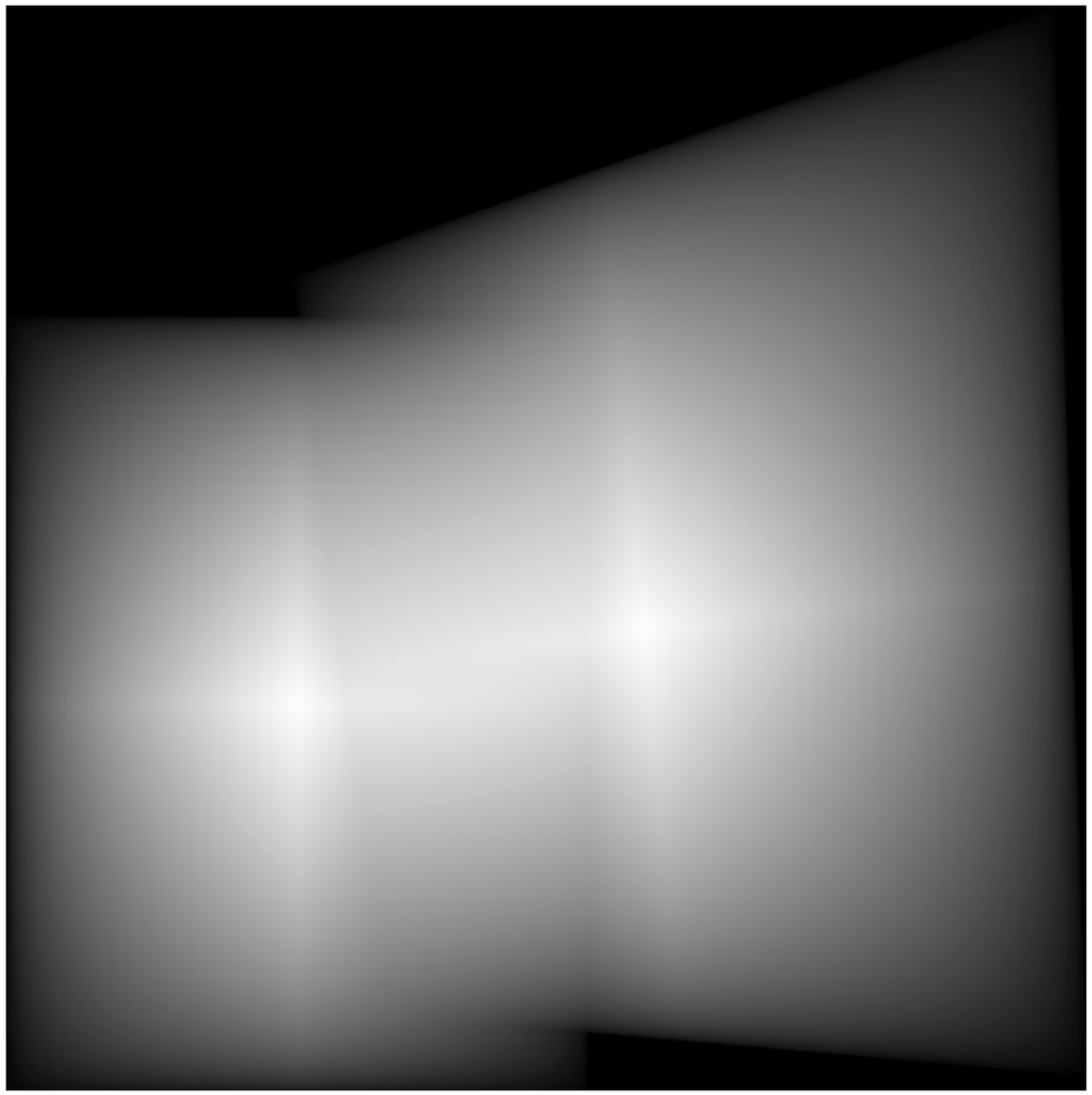


Low frequency: smooth blending

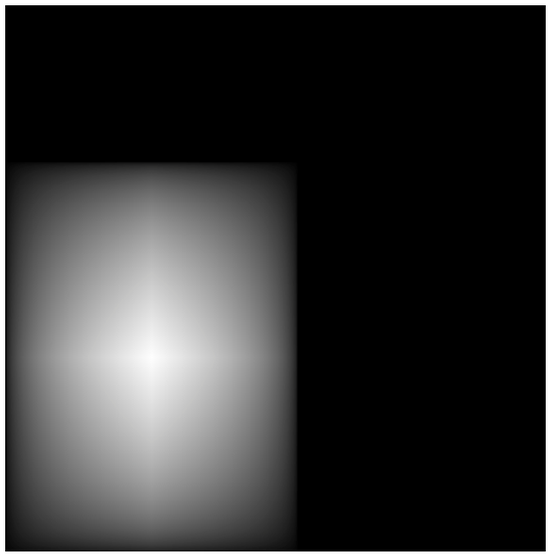
Low frequencies

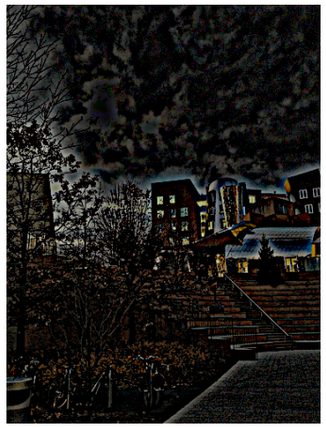


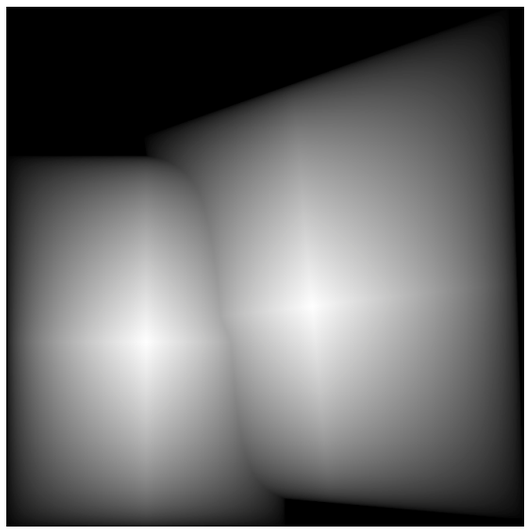
(sum of) Weights



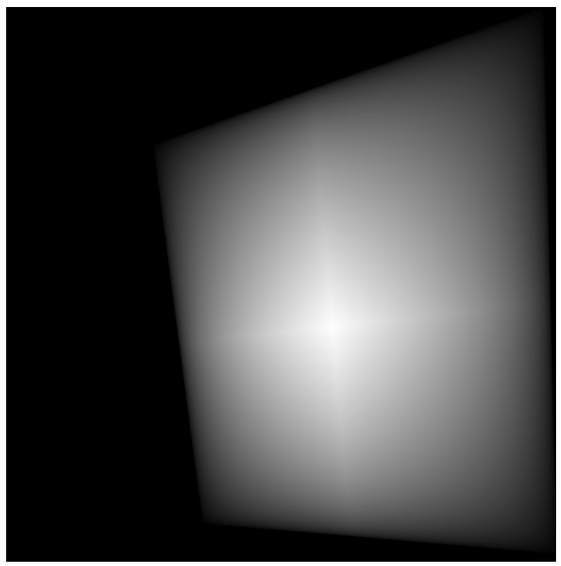
High frequency: abrupt

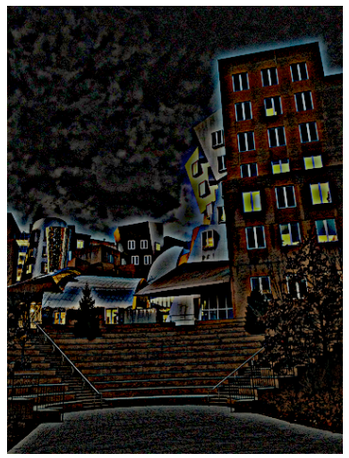












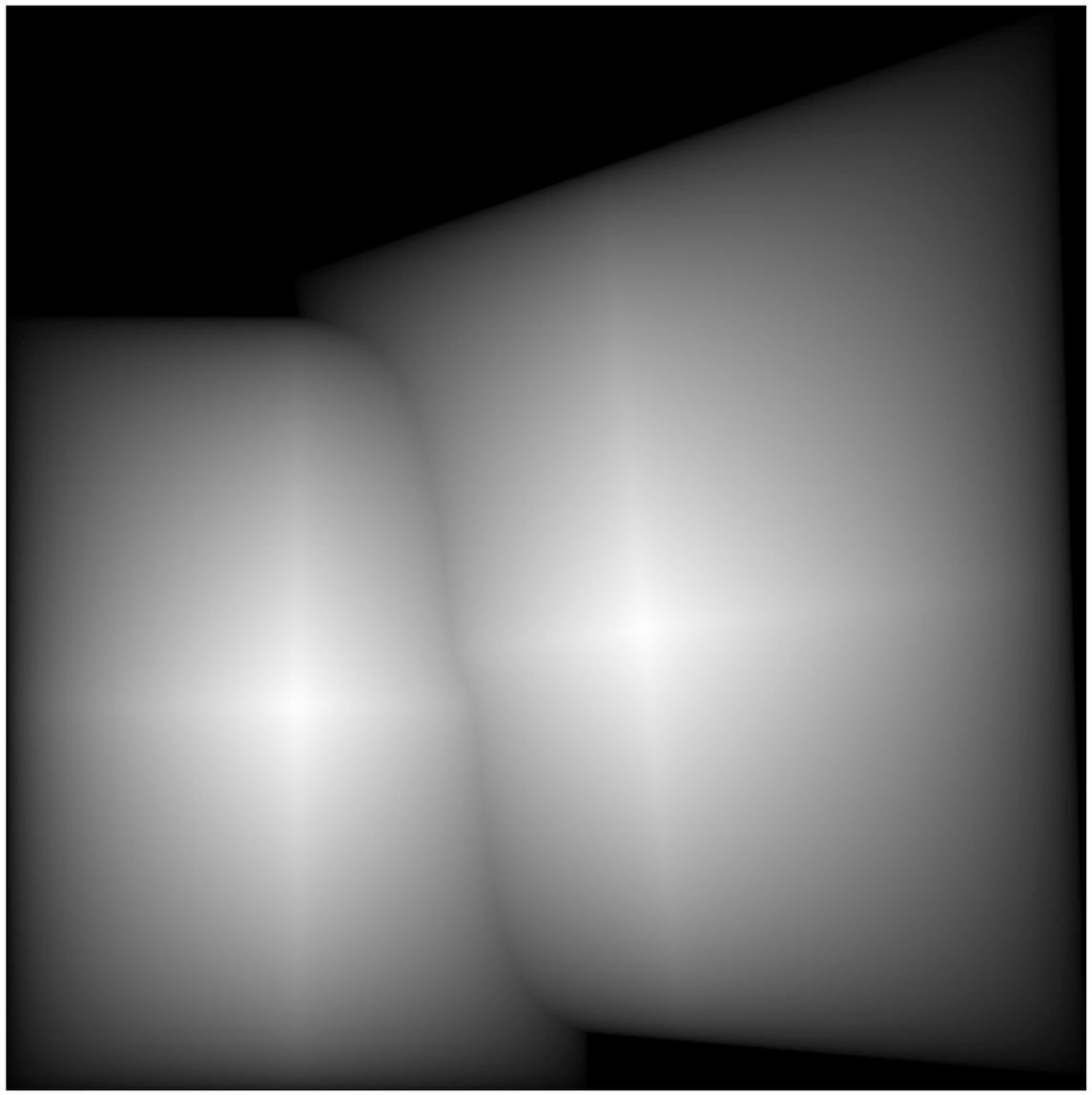
High frequency: abrupt

- For a pixel, only keep the source value with the highest weight

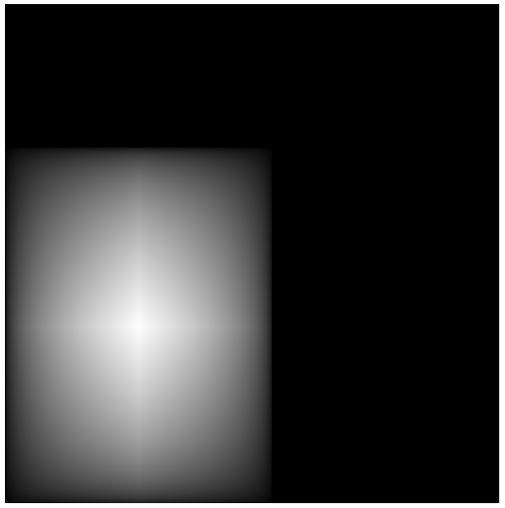
High frequencies

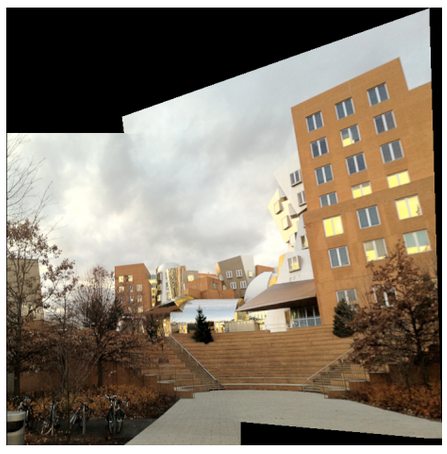
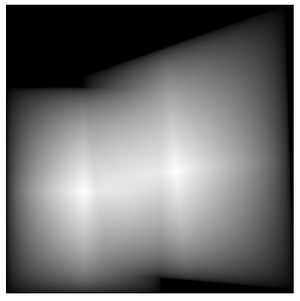
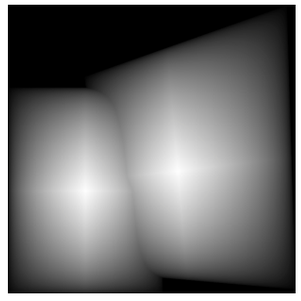


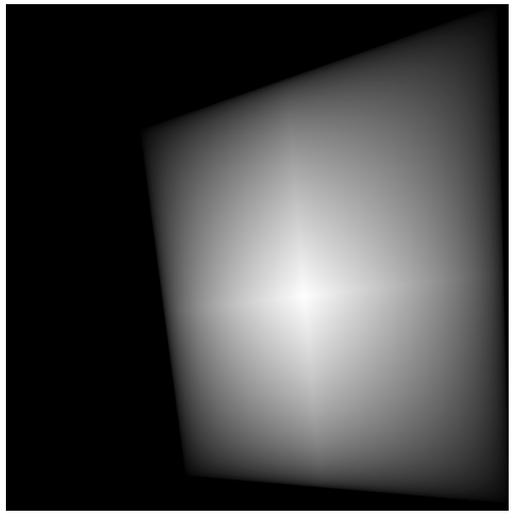
Max weights

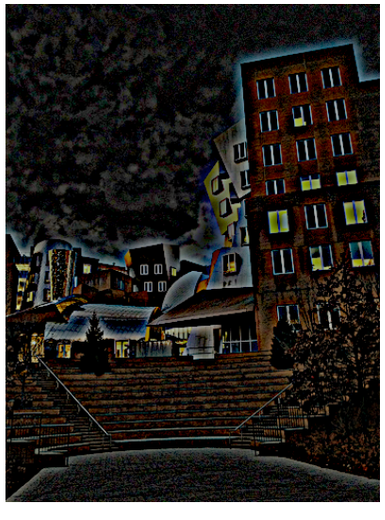


2-scale separation





Result



Basic reprojection



Smooth blending



2-scale blending



Comparison







Multiscale blending

- Proposed by Adelson and Burt
  - http://persci.mit.edu/pub_pdfs/spline83.pdf
- Simplified to two scales by Brown and Lowe
  - http://cs.bath.ac.uk/brown/papers/iccv2003.pdf

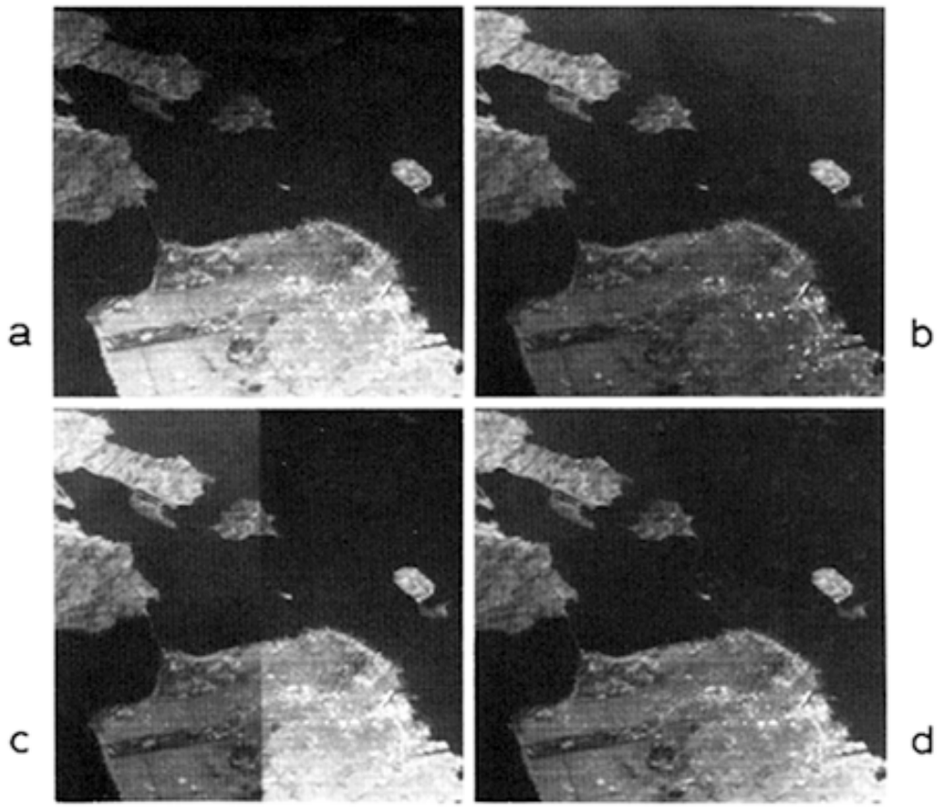


Fig. 6. The spline applied to Landsat images of San Francisco. When the left half of Figure 6a is joined to the right half of Figure 6b without a spline, the boundary is clearly visible (Figure 6c). No boundary is visible when the multiresolution spline is used (Figure 6d).



(b) All 80 images registered



(c) Rendered with multi-band blending