3D Manipulation of 2D Images*

Alexandru Vasile (alexv@ll.mit.edu) & Peter Cho (cho@ll.mit.edu)

MIT Lincoln Laboratory

MIT IAP Course
January 2012

Class 3 notes

*This work was sponsored by the Department of the Air Force under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

Course Outline

- Class 1: Single-view geometry
  - Pinhole camera model
  - 2D image insertion into 3D map
  - Camera calibration
- Class 2: Panorama formation
  - Homographies
  - 2D & 3D mosaics
  - Geometric propagation of knowledge
- Class 3: Two-view geometry
  - Feature matching & RANSAC
  - Epipolar geometry
  - Fundamental matrix
- Class 4: 3D reconstruction
  - Structure from motion
  - Bundle adjustment
  - Photo tourism

Feature Matching

- **Problem: Given two images with partial overlap, find interesting points in each image and determine pair-wise point correspondences.**

- **Challenges:**
  - **Need to find features that are robust to camera pose changes (rotation, translation, scaling).**
  - **Need method to reject bad correspondences / outliers.**

- **Importance:**
  - **Allows for higher level scene understanding (object tracking, camera pose estimation, 3D triangulation for scene reconstruction).**

Feature Matching Example

- **First, find interesting features in each image using some feature detector.**

MITLeft image

MITMiddle image

Feature Matching Example

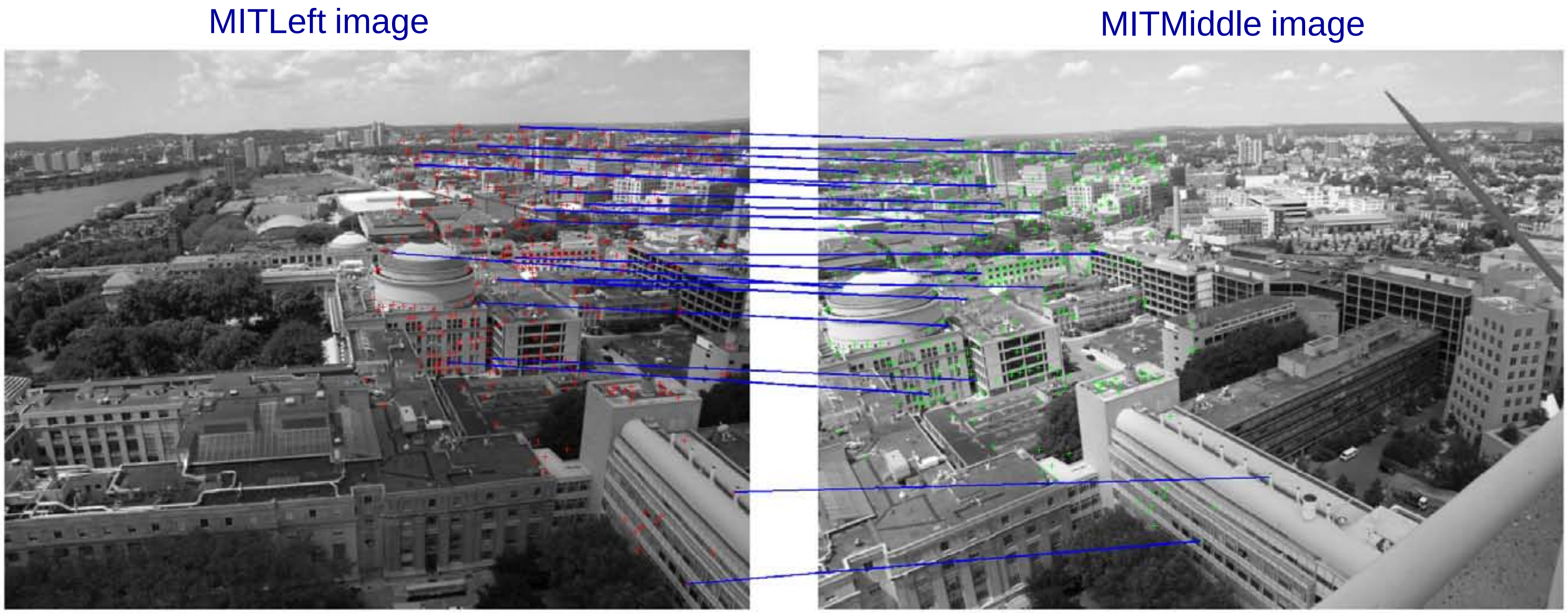
- **First, find interesting features in each image using some feature detector.**

MITLeft image

MITMiddle image

Feature Matching Example

- First, find interesting features in each image using some feature detector.
- **Second, do feature matching.**
  - **For each feature in image 1, find top N best matches in image 2**



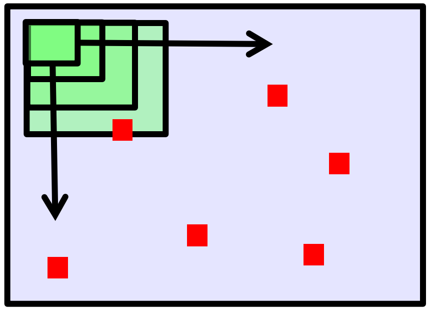
Local Feature Detectors

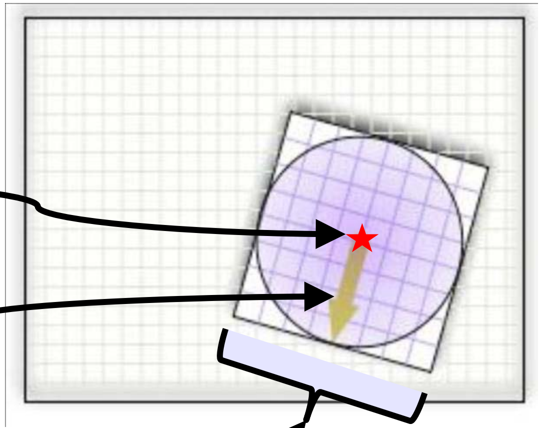
- **Key properties of a good local feature descriptor:**
  - **Selects interesting points that are highly distinctive.**
  - **Easy to extract and match against a large database of features.**
    - **Compact representation of local pixel neighborhood.**
  - **Invariance to :**
    - **Image noise.**
    - **Changes in illumination.**
    - **Camera pose (scaling, rotation, translation).**

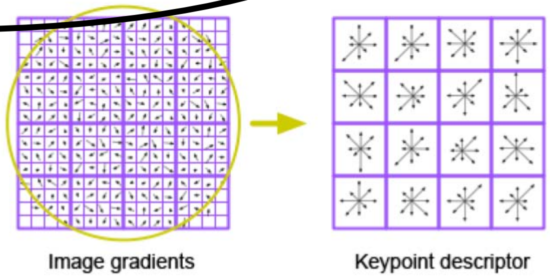
- **Will talk about one that has become a standard in computer vision, named Scale Invariant Feature Transform (SIFT).**
  - **Others exist, such as SURF, RIFT …**

Scale Invariant Feature Transform (SIFT)

- **Detection stage for SIFT features:**
  - **Scale-space extrema detection.**
    - ➢ Convolve image with template at 4 different scales.
    - ➢ Difference of Gaussians template picks out location with lots of gradient changes.
    - ➢ Get both scale invariance and u-v locations of points.
  - **Keypoint localization.**
    - ➢ Do sub-pixel u-v localization of points.
  - **Orientation Assignment.**
    - ➢ Find major gradient direction(s).
    - ➢ Get rotation invariance.
  - **Generation of keypoint descriptors.**
    - ➢ Based on scale, orientation assignment, pick out size and rotation of local support region.
    - ➢ Divide into 4x4 sub-regions and compute 8-bin histogram per region.
    - ➢ Concatenate histograms to form 128 element keypoint descriptor.







SIFT Matching Method

- **For each SIFT feature found in image 1, match against all features in image 2.**
  - **Compute dot-products of 128 element descriptor.**
  - **Higher dot-product indicates better match.**
  - **Decide to keep a match if it is unique.**
    - **Use ratio of second best dot product to best.**
    - **Keep if ratio below 0.6 (heuristic value, can be changed).**

- **Advantages of SIFT descriptor**
  - **Selects interesting points that are highly distinctive.**
  - **Easy to extract and match against a large database of features.**
  - **Invariant/tolerant to noise, scale, rotation, illumination.**

- **Practical limitations.**
  - **Can handle about 30 degrees out-of-image plane pose change before breaking down.**

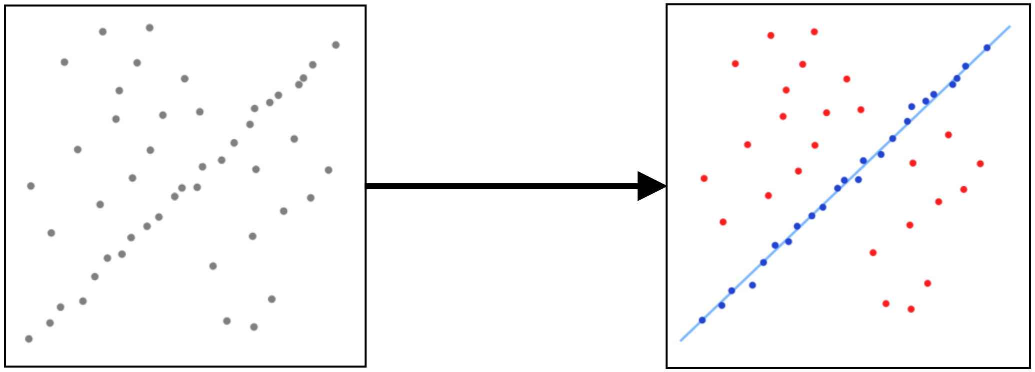
Have some matches: now what?

- Might still have bad correspondences, outliers.
- Need method to score quality of match.
  - Can score match quality individually or by its consensus with other matches.
  - Consensus typically works better, more statistics used.
- How to measure consensus between matches?
  - Can fit a model to the data points.
- Two widely used techniques:
  - RANdom SAmple Consensus (RANSAC).
  - Hough Transform.

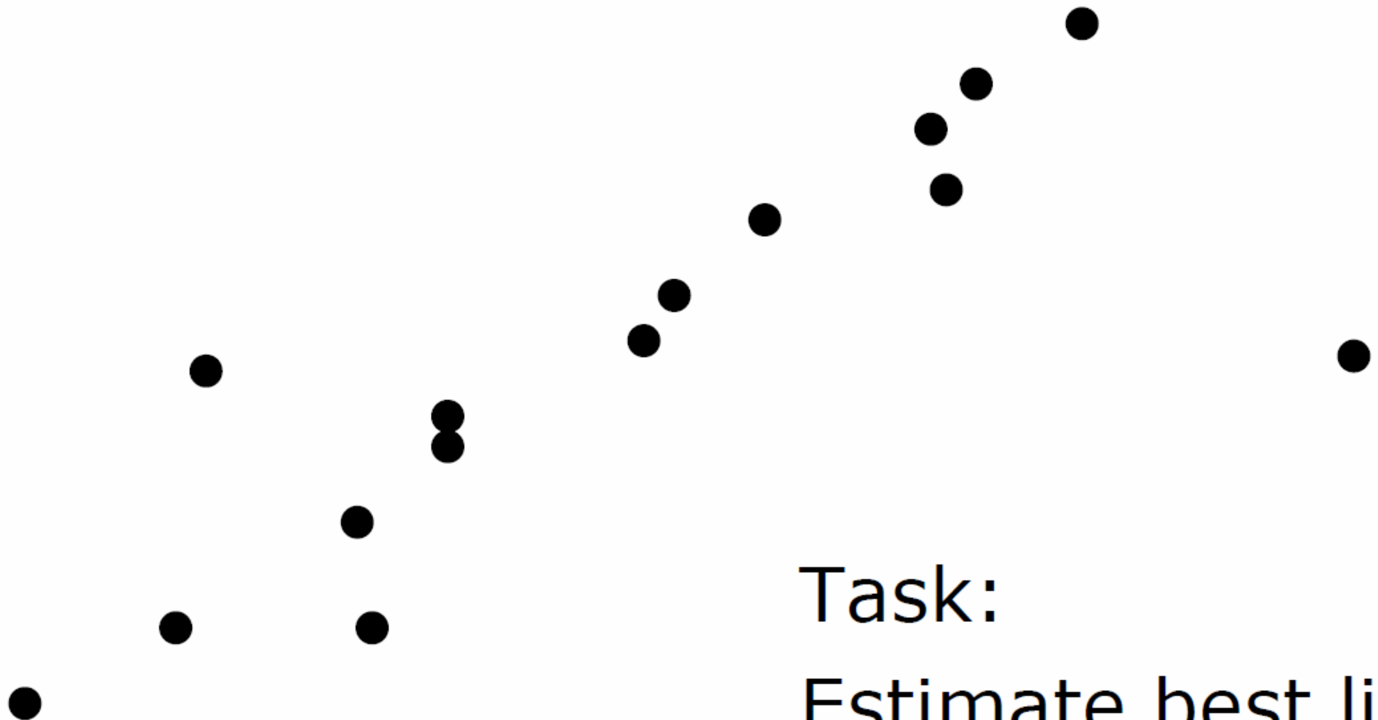
RANdom SAmple Consensus (RANSAC)

- **Idea:**
  1. **Randomly select small number of data points and use to generate instance of model.**
  2. **Check number of data points consistent with this fit**
  3. **Repeat Step 1+2 until "good enough" consistent set found or hit some max iteration number.**
  4. **Generate new fit from this consistent set.**

- **Toy-problem to motivate algorithm:**
  - **2D line fitting in presence of noise/outliers.**



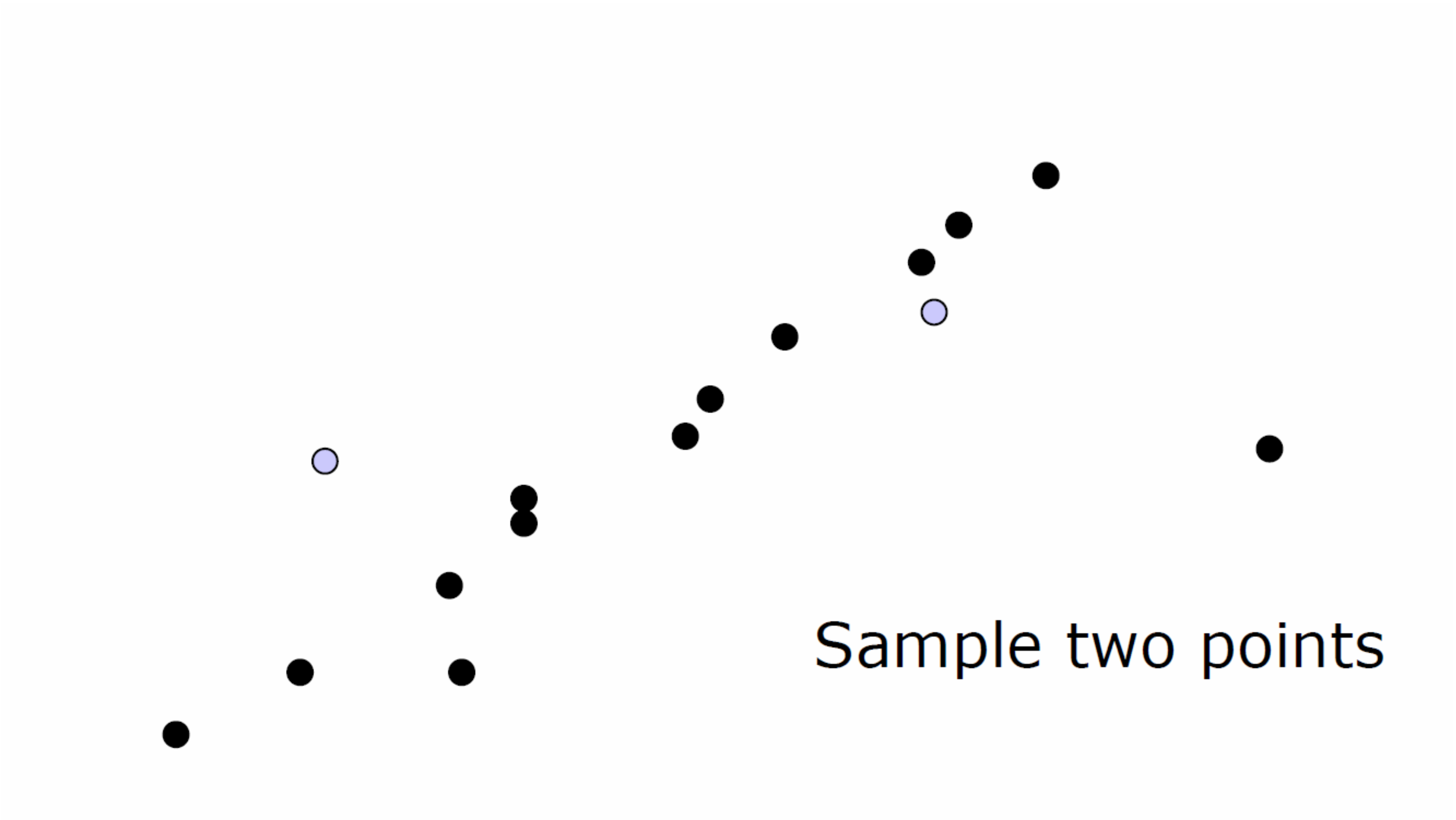
RANSAC Line Fitting Example



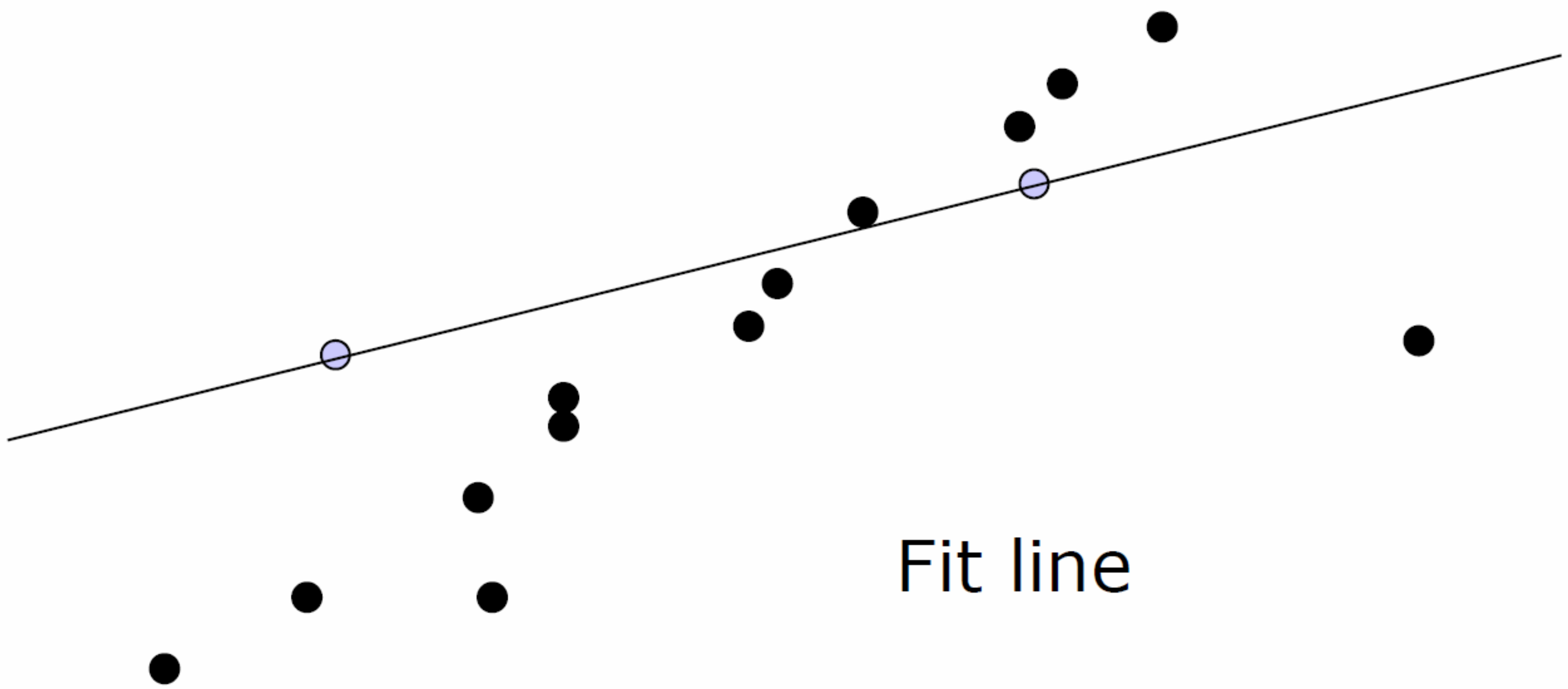
Task:

Estimate best line

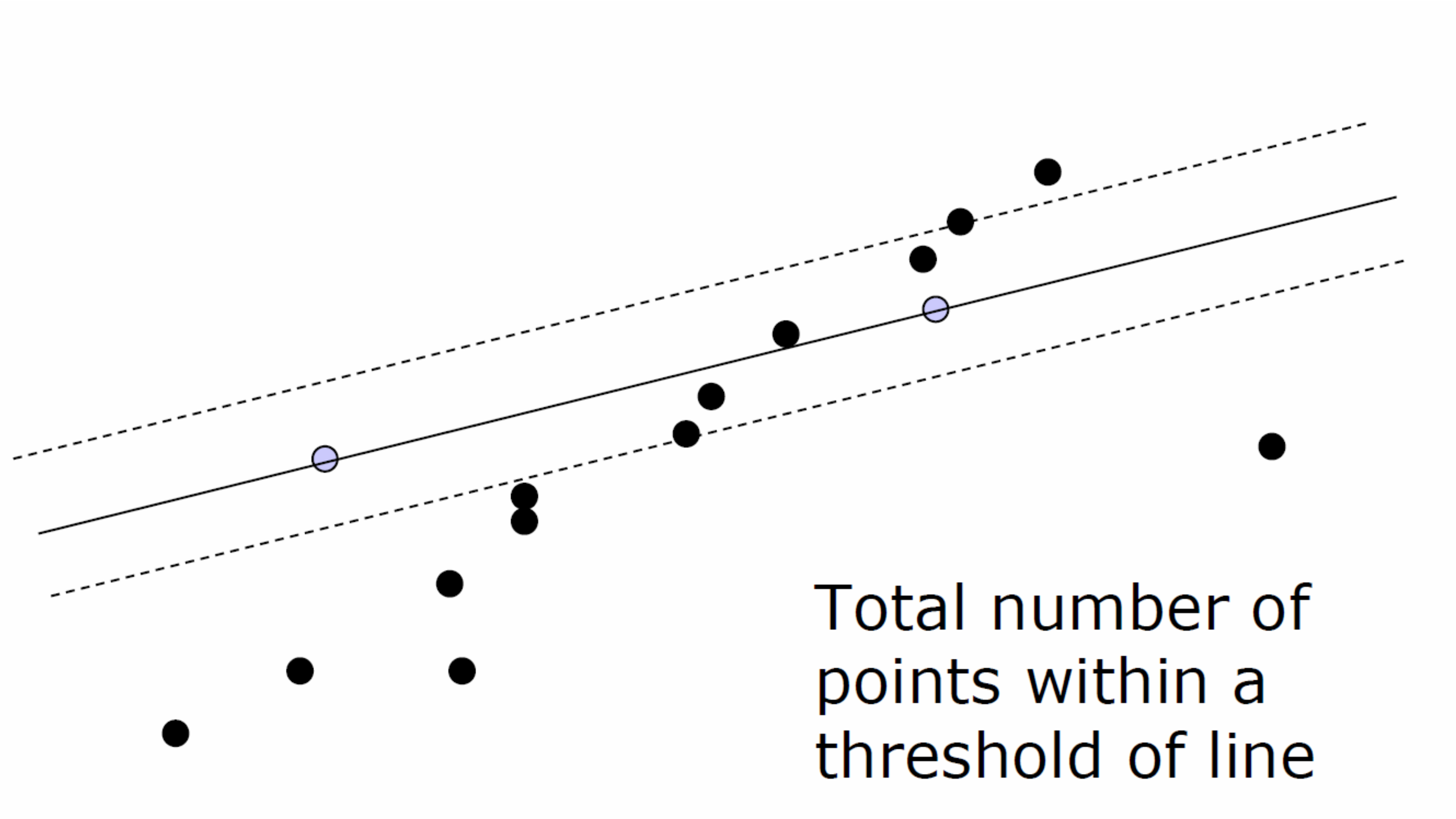
RANSAC Line Fitting Example



RANSAC Line Fitting Example



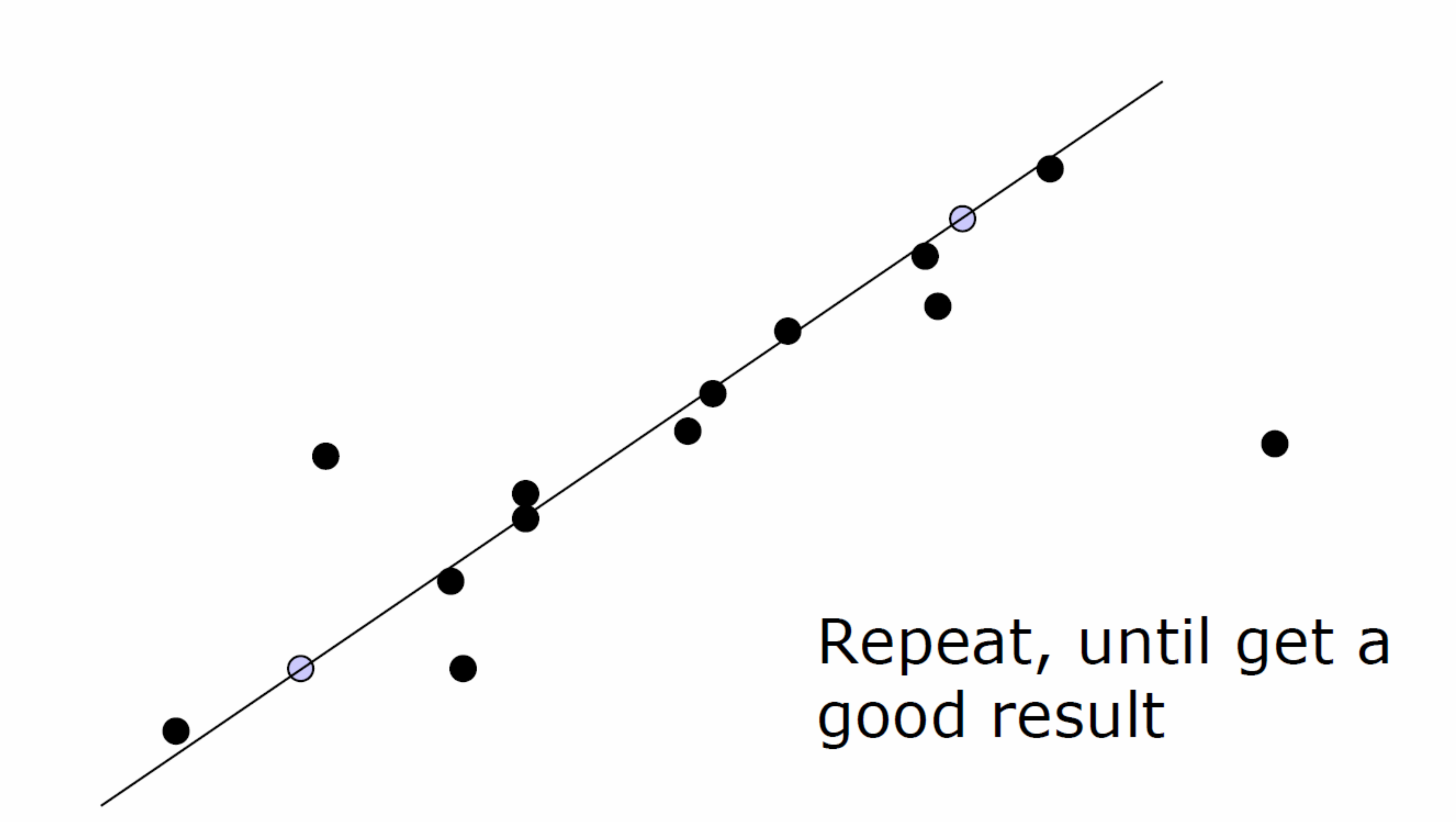
RANSAC Line Fitting Example



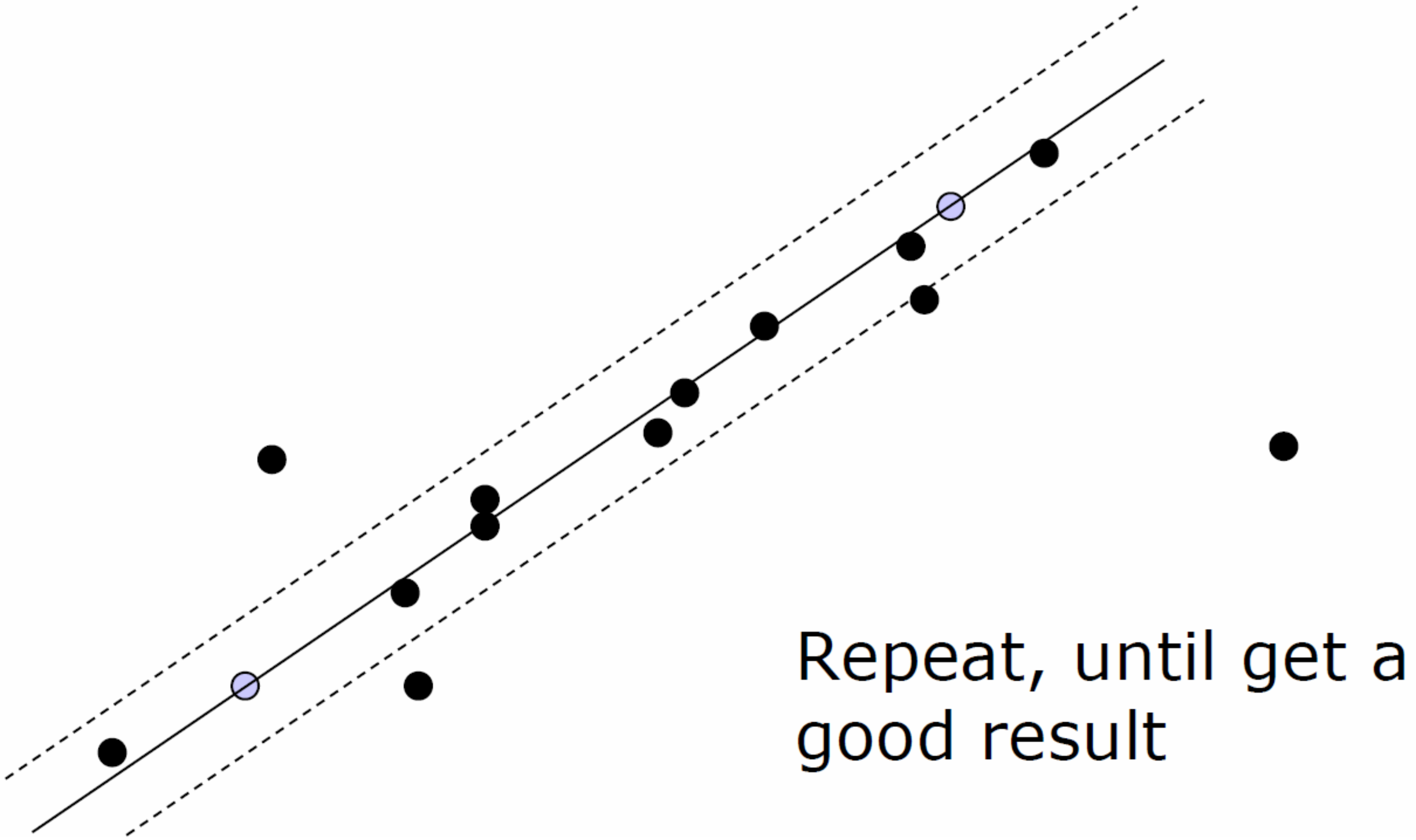
RANSAC Line Fitting Example

Repeat, until get a good result

RANSAC Line Fitting Example



RANSAC Line Fitting Example



Geometric Fitting Models

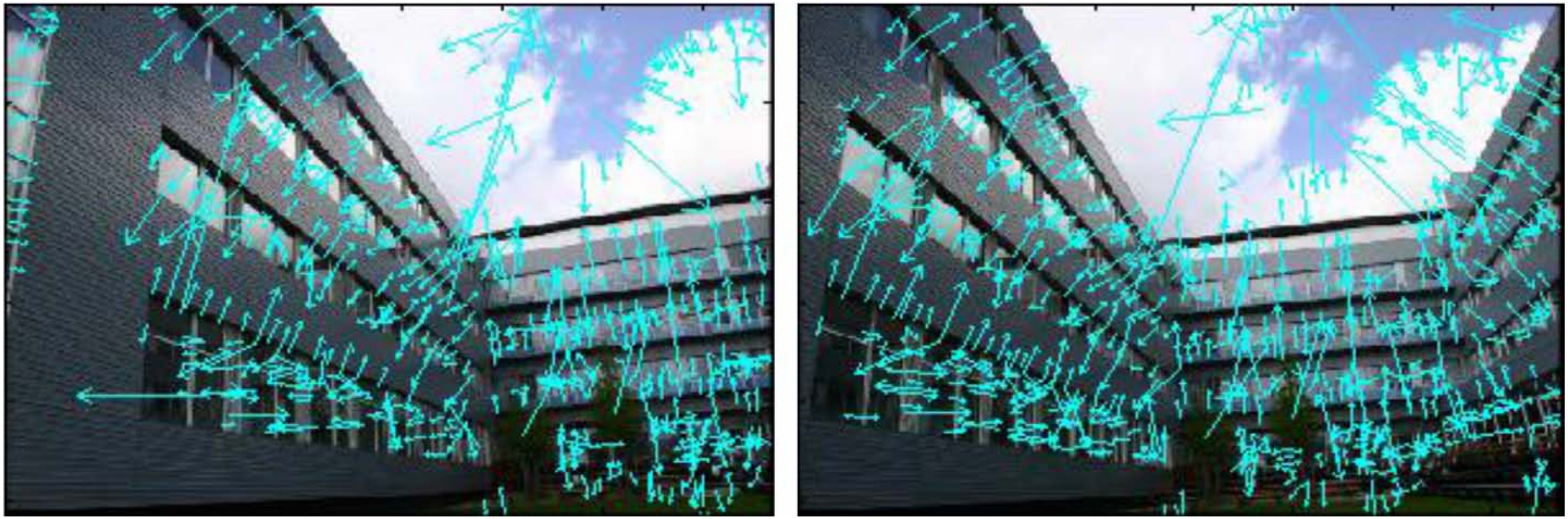
- **Use RANSAC with an geometric fitting model.**
- **Homography Model**
  - **For general 3D scene, holds for only small pose changes.**
  - **In case of dominant planar region in image, holds for most pose changes.**
  - **In case of pure rotation, holds for all 3D scenes.**
- **Epipolar Geometry Model**
  - **For general 3D scene, holds for most pose changes.**
  - **Degenerate cases if scene is planar or camera motion is pure rotation.**
- **Models are complimentary …**
  - **Choose one based on prior knowledge of camera motion.**

SIFT + RANSAC for Homography with Pure Camera Rotation

- **Choose 2 overlapping images.**

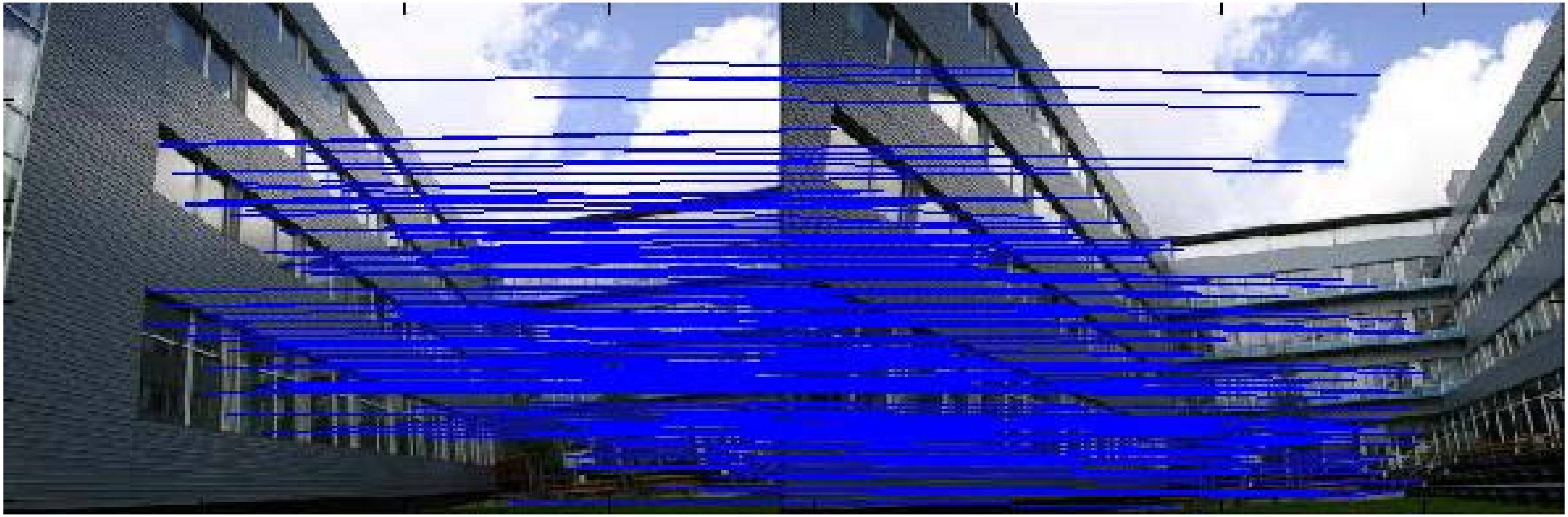
SIFT + RANSAC for Homography with Pure Camera Rotation

- Choose 2 overlapping images.
- Find SIFT features for each image.



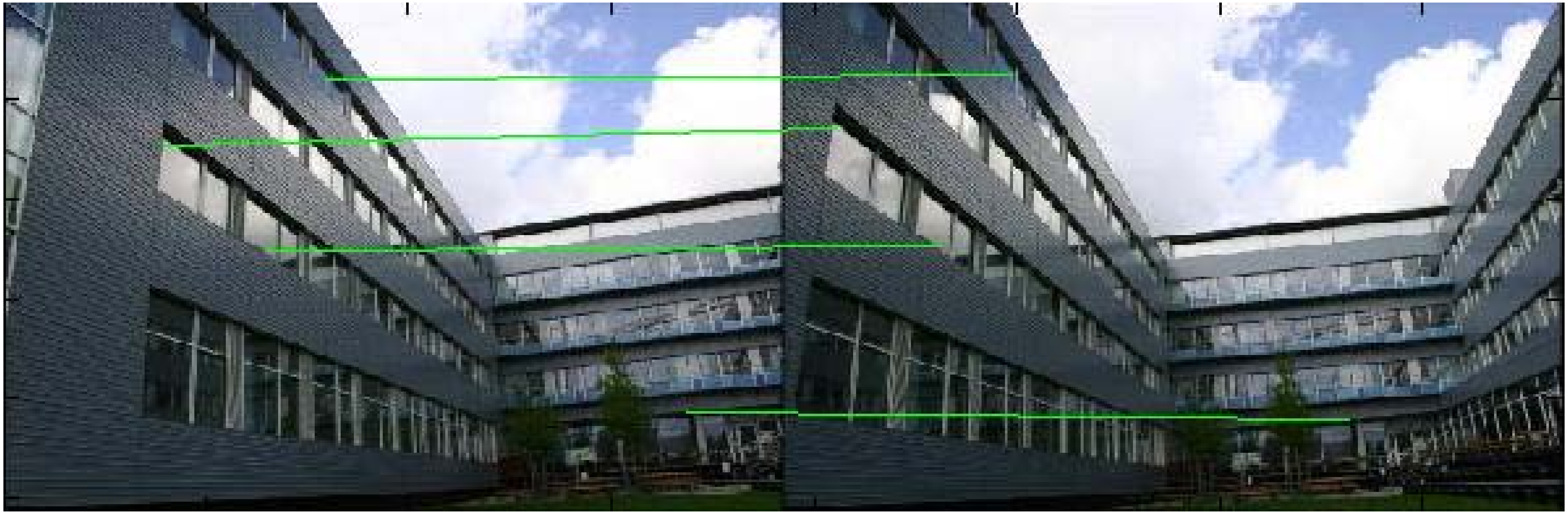
SIFT + RANSAC for Homography with Pure Camera Rotation

- Choose 2 overlapping images.
- Find SIFT features for each image.
- **Match SIFT features to get initial point correspondences.**



SIFT + RANSAC for Homography with Pure Camera Rotation

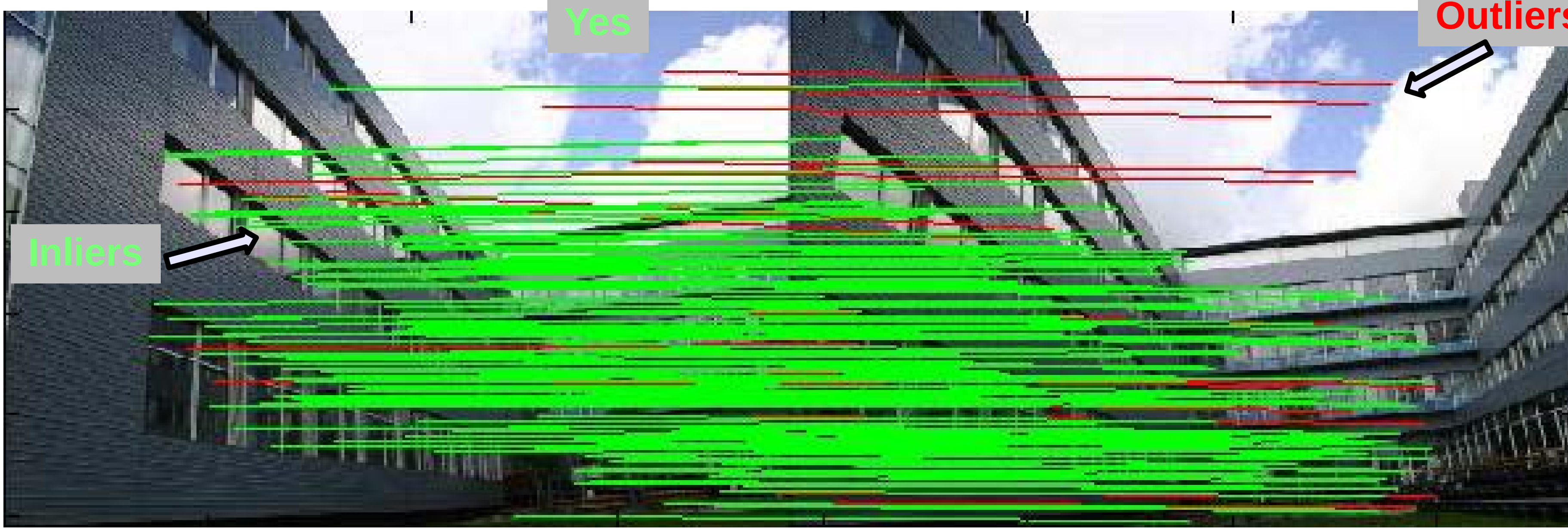
- Choose 2 overlapping images.
- Find SIFT features for each image.
- Match SIFT features to get initial point correspondences.
- **Run RANSAC:**
  1. **Select minimal number of points (4), find homography.**



SIFT + RANSAC for Homography with Pure Camera Rotation

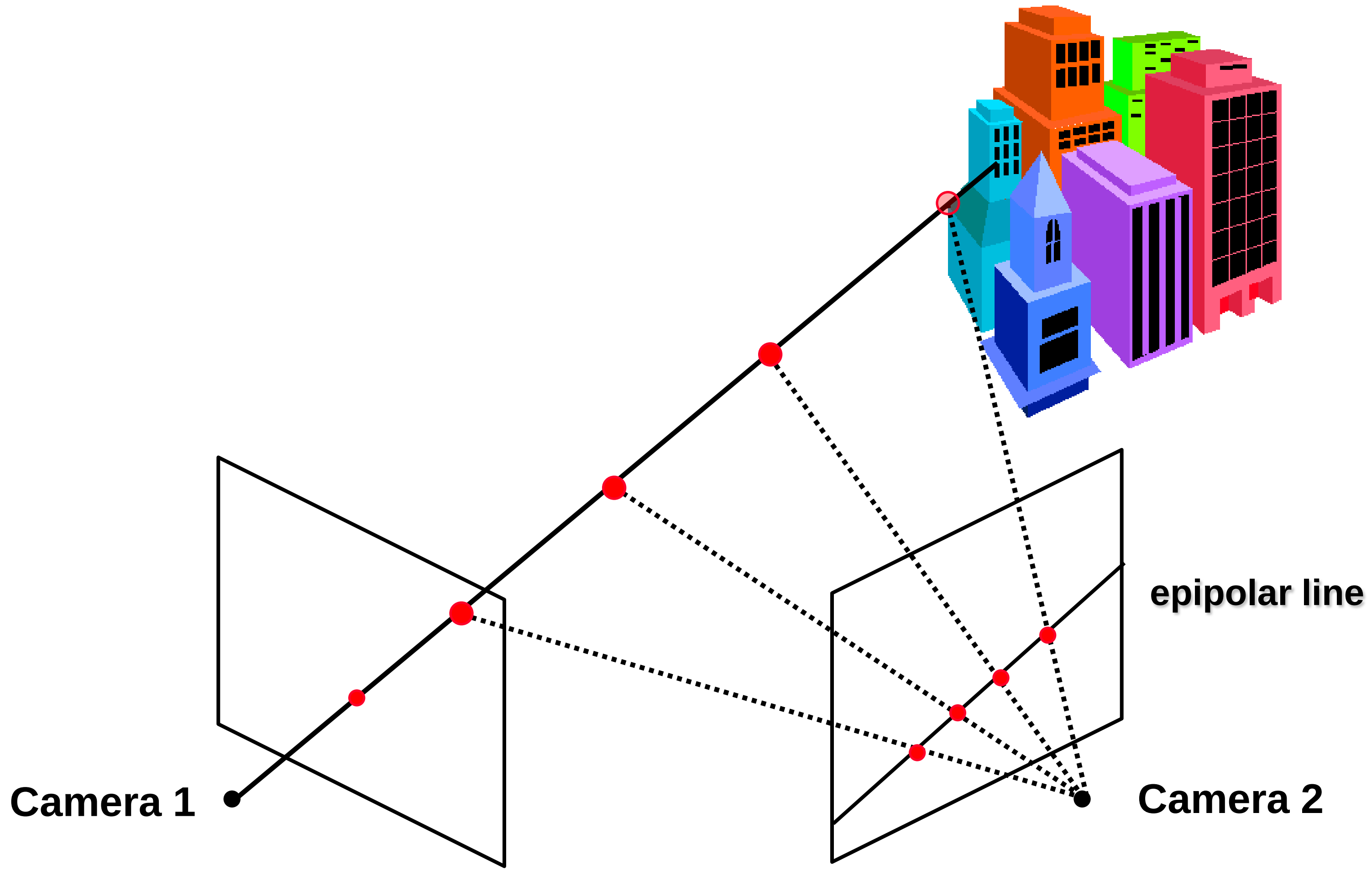
- Choose 2 overlapping images.
- Find SIFT features for each image.
- Match SIFT features to get initial point correspondences.
- **Run RANSAC:**
  1. Select minimal number of points (4), find homography.
  2. **Check number of data points consistent with this fit.**
  3. **Good enough? ⟹ Find homography using all inliers.**

No



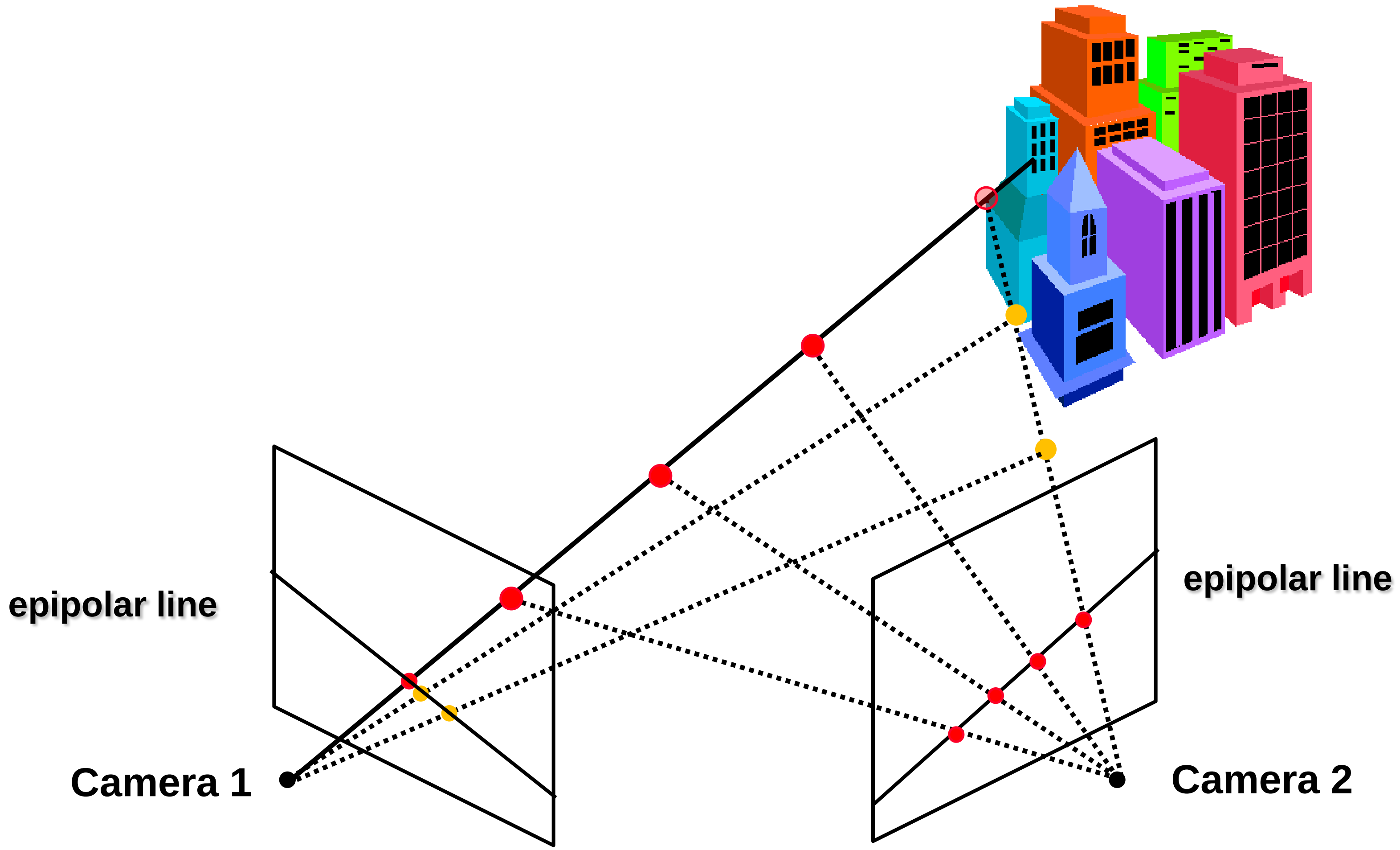
Epipolar Geometry Model

- **Point in camera 1 has a corresponding point that lies somewhere along an epipolar line in camera 2.**



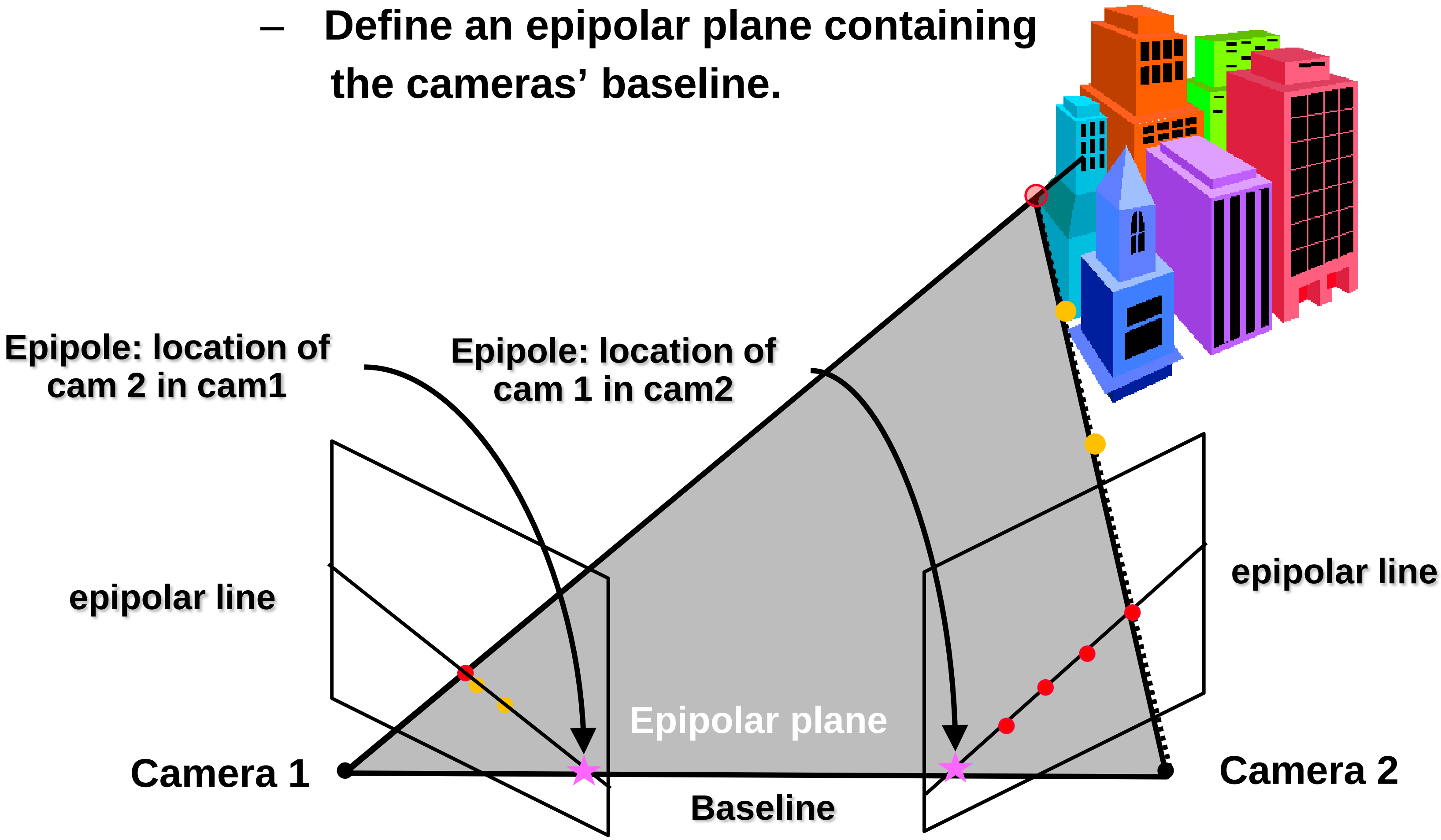
Epipolar Geometry Model

- **Corresponding points:**
  - **Lie on conjugate epipolar lines.**



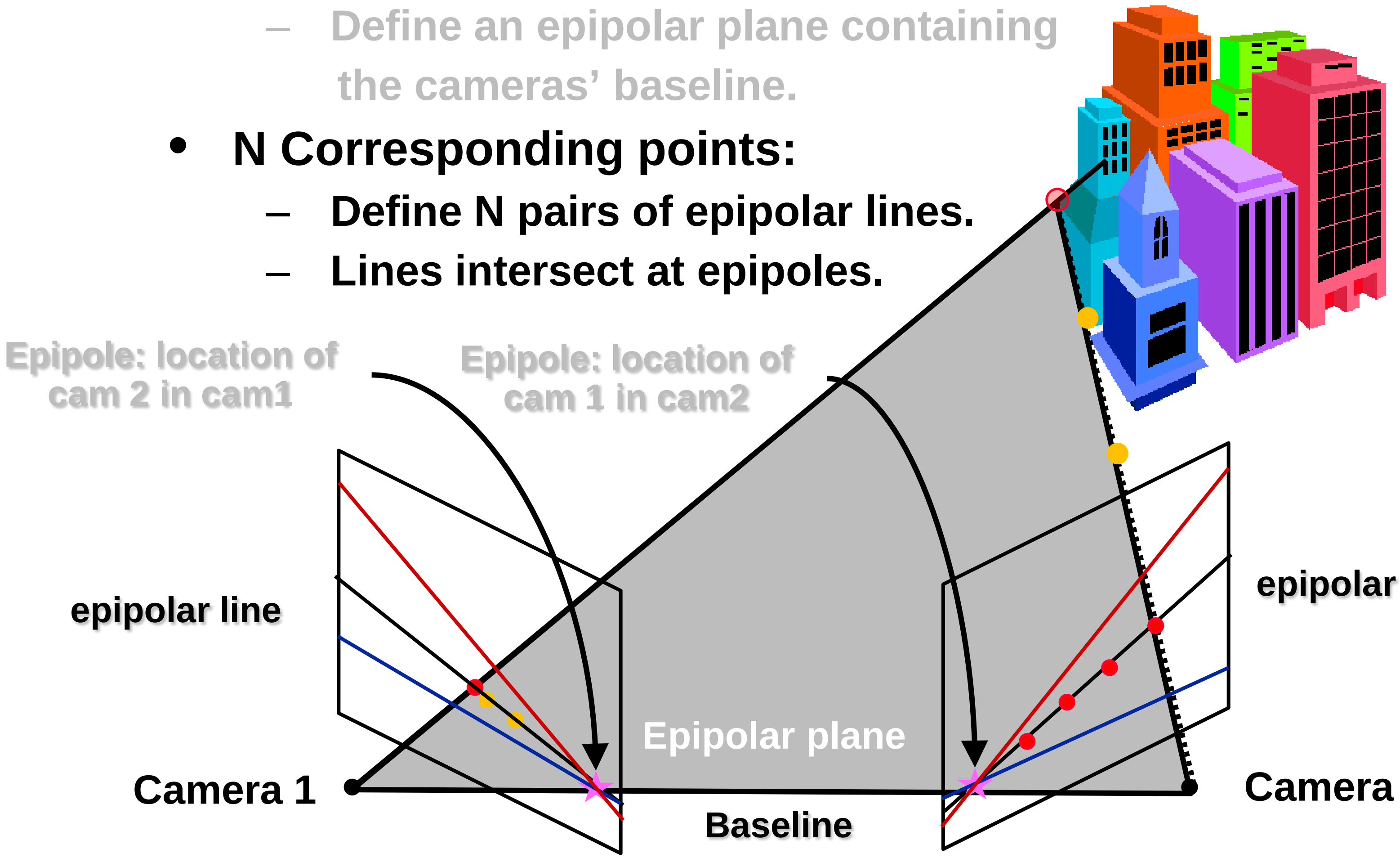
Epipolar Geometry Model

- **Corresponding points:**
  - Lie on conjugate epipolar lines.
  - Define an epipolar plane containing the cameras' baseline.



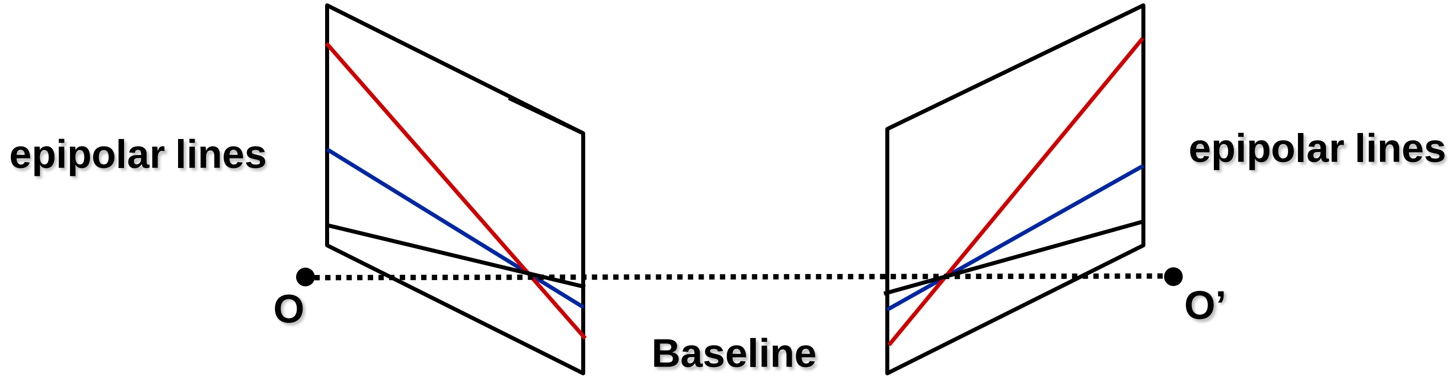
Epipolar Geometry Model

- Corresponding points:
  - Lie on conjugate epipolar lines.
  - Define an epipolar plane containing the cameras' baseline.
- **N Corresponding points:**
  - **Define N pairs of epipolar lines.**
  - **Lines intersect at epipoles.**



Epipolar Geometry Recap

- Every plane through the baseline is an epipolar plane.
- It determines a pair of epipolar lines (one in each image).
- Two systems of epipolar lines are obtained.
- Each system intersects in a point, the *epipole.*
- The epipole is the projection of the center of the other camera.



Epipolar Geometry Model & Correspondence Matching

- **Question:**
  - How do we use this epipolar geometry to constrain/filter some prior computed correspondence matches?
- **Answer:**
  - Given a correspondence point in one image, need to determine method to find the epipolar line in the second image.
  - If prior corresponding point in second image is close to the line, then it fits the epipolar geometry model.
- **What is missing:**
  - Method to determine how to go from point to epipolar line.

Fundamental Matrix

- **Fundamental matrix allows us to go from point p_R in right image to epipolar line L_L in the left image.**

p_r

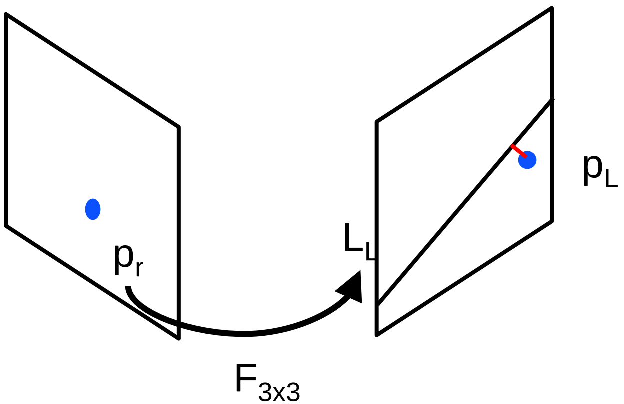
$$[u_R, v_R, 1]\ F_{3x3} = L_L$$

Eqn 3.1

- **Dot product of line L_l and left image point p_L, which should be on the line should equal zero.**

$$L_L \begin{bmatrix} u_L \\ v_L \\ 1 \end{bmatrix} = 0$$

p_L



$$[u_R, v_R, 1]\ F_{3x3} \begin{bmatrix} u_L \\ v_L \\ 1 \end{bmatrix} = 0$$

Eqn 3.2

Properties of Fundamental Matrix

- *F* is homogeneous
- Has rank 2.
- Its (right and left) null spaces are the two epipoles.
- *9 parameters*
- *F* can be recovered up to scale using 8 points.

Computing F: The 8 Point Algorithm

- **Assume that we have m correspondences.**
- **Each correspondence i [p_R, p_L], satisfies:**

$$p_{R\,i}^{T}\ F_{3x3}\ \ p_{L\,i} = 0 \qquad \text{Eqn 3.3}$$

- **F is a 3x3 matrix (9 entries), but rank 2.**
- **Homogenous linear system with 9 unknowns.**
- **Need $m \geq 8$; solution will be up to a constant.**

Computing F: The 8 Point Algorithm

- **Let** $p_{Ri} = [u'_i, v'_i, 1],\ and\ p_{Li} = [u_i, v'_i, 1]$

$$p_{Ri}^{\ T}\ F\ \ p_{Li} = 0 \qquad i = 1...m$$

- **Then:**

$$[u'_i, v'_i, 1]\ F_{3x3} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = 0$$

Eqn 3.4

$$[u'_i, v'_i, 1] \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = 0$$

Eqn 3.5

Computing F: The 8 Point Algorithm

$$[u'_i, v'_i, 1] \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = 0$$

Eqn 3.6

$$\underbrace{\begin{bmatrix} u_1u'_1 & u_1v'_1 & u_1 & v_1u'_1 & v_1v'_1 & v_1 & u'_1 & v'_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_mu'_m & u_mv'_m & u_m & v_mu'_m & v_mv'_m & v_m & u'_m & v'_m & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} f11 \\ f12 \\ f13 \\ f21 \\ f22 \\ f23 \\ f31 \\ f32 \\ f33 \end{bmatrix}}_{x} = 0$$

Eqn 3.7

- **Need to find non-trivial solution to Ax = 0, rank (A) = m-1**
- **Solve** $\min_x \|Ax\|^2 \;\; such\ that \;\; \|x\|^2 = 1$

Computing F: The 8 Point Algorithm

- Construct the m x 9 matrix A
- Find the SVD entries of A = U D V' Eqn 3.8
- The entries of F are the components of the last column of V corresponding to the least singular value.

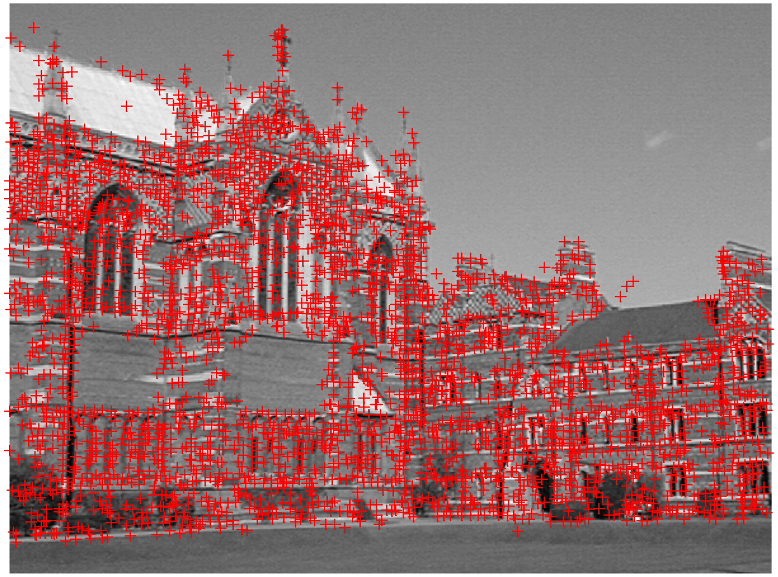
- F must be singular (3x3 matrix of rank 2). Due to noise in correspondences, will not be singular. To enforce it:
  - Find SVD of $F = U_f D_f V_f'$. Eqn 3.9
  - Set smallest singular value of D_f to 0 to create D_f'.
  - Recompute $F = U_f D_f' V_f'$. Eqn 3.10

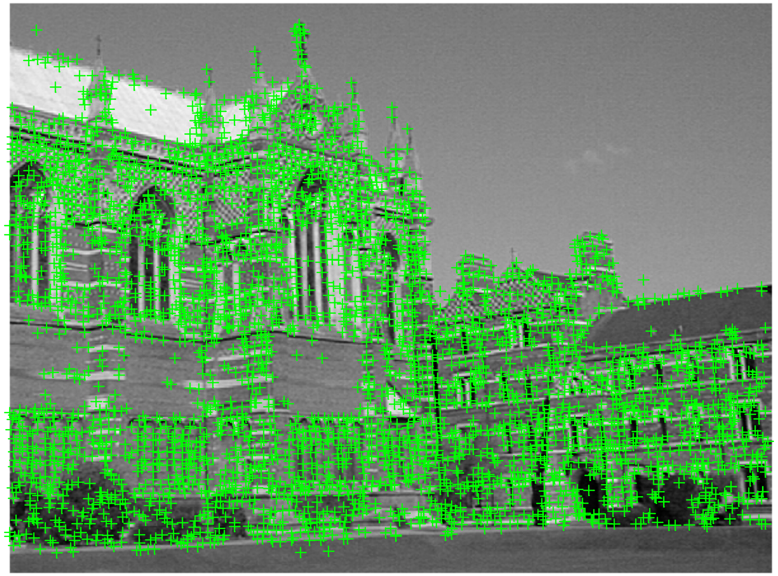
SIFT + RANSAC for computing the Fundamental Matrix

- Choose 2 overlapping images.

SIFT + RANSAC for computing the Fundamental Matrix

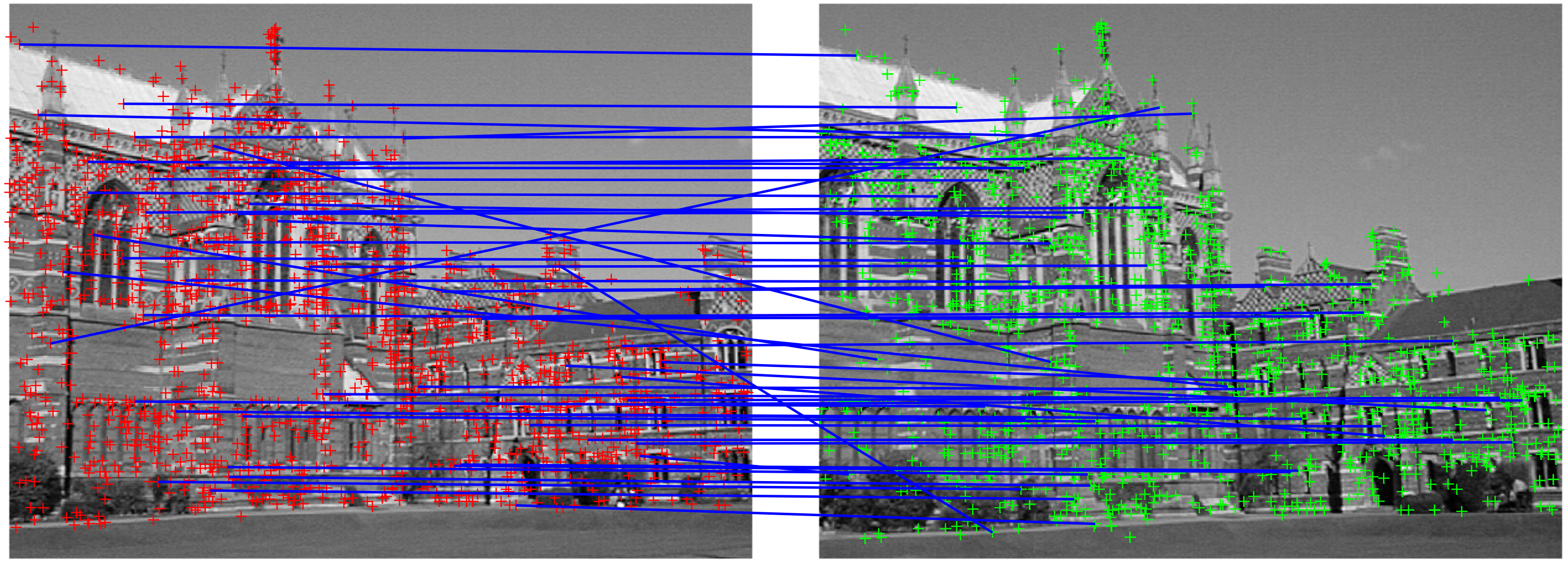
- Choose 2 overlapping images.
- Find SIFT features for each image.





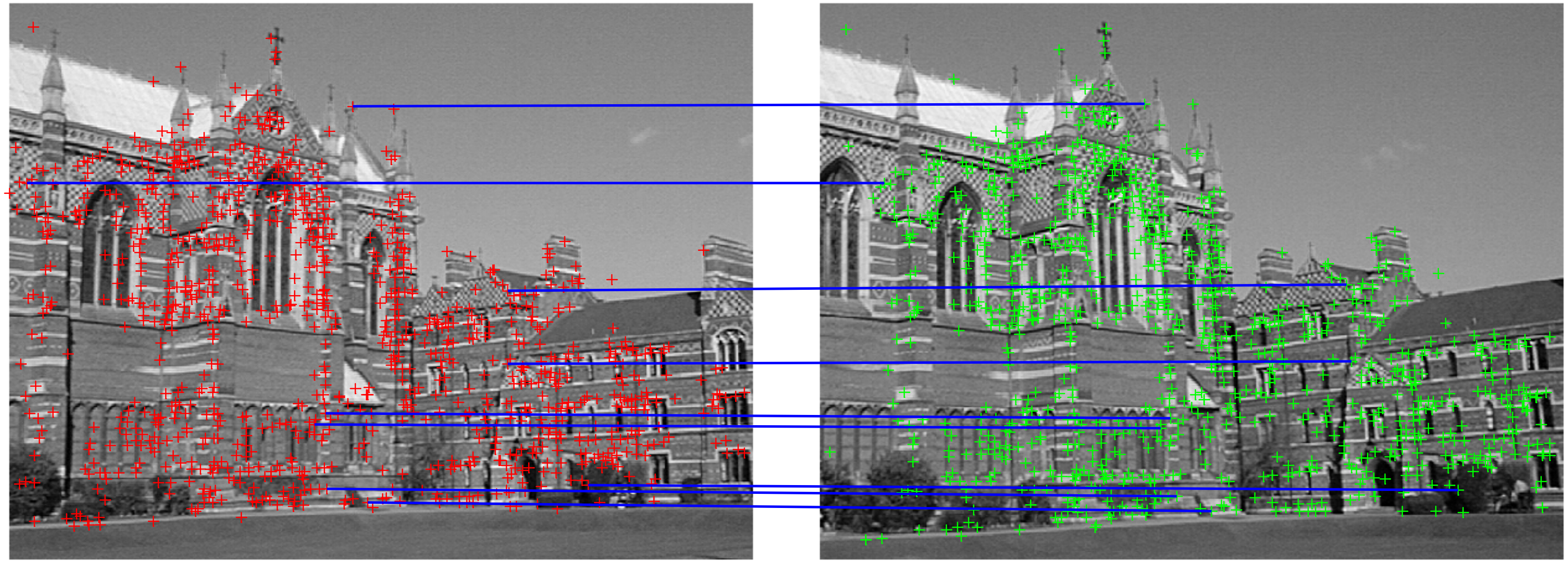
SIFT + RANSAC for Homography with Pure Camera Rotation

- Choose 2 overlapping images.
- Find SIFT features for each image.
- **Match SIFT features to get initial point correspondences.**



SIFT + RANSAC for computing the Fundamental Matrix

- Choose 2 overlapping images.
- Find SIFT features for each image.
- Match SIFT features to get initial point correspondences.
- **Run RANSAC:**
  1. **Select minimal number of points (8), find fundamental matrix.**

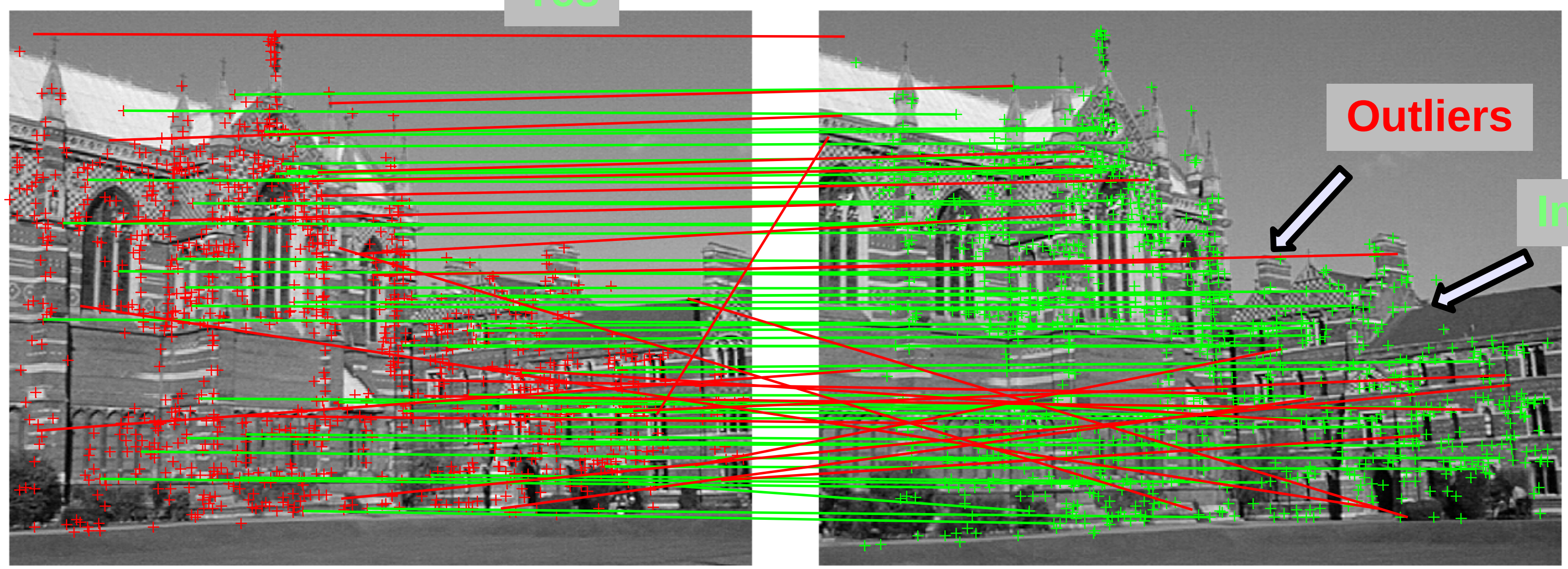


SIFT + RANSAC for computing the Fundamental Matrix

- Choose 2 overlapping images.
- Find SIFT features for each image.
- Match SIFT features to get initial point correspondences.
- **Run RANSAC:**
  1. Select minimal number of points (8), find fundamental matrix.
  2. **Check number of data points consistent with this fit.**
  3. **Good enough? ⟹ Find F matrix using all inliers.**

No

Yes



Lab 3A: SIFT + RANSAC for Fundamental Matrix Estimation

- **Compute the fundamental matrix and estimate best F matrix using RANSAC.**
- **Go to web.mit.edu/alexv/Public/IAP_2012/class03/Lab03A/**
  - **Download code for SIFT/RANSAC (Set path with subfolders).**
  - **Complete code for 8-point algorithm under function fundmatrix.m (See Equations 3.8 to 3.10).**
- **Provided two example images from which to compute the fundamental matrix:**
  - **image01.bmp, image02.bmp**
- **Main function is testFund.m**

Lab 3B: SIFT + RANSAC for Homography Estimation

- Redo Lab 2B using SIFT+ RANSAC to automatically compute the homographies.
- Go to web.mit.edu/alexv/Public/IAP_2012/class03/Lab03B/
  - Download code ransac, 4 point homography.
- Provided same 3 images as in Lab 2, down-sampled to reduce computation time:
  - MITLeft_downsampled.jpg, MITMiddle_downsampled.jpg MITRight_downsampled.jpg
- Main function is testHomographies.m

Preparation for Lab 04
Structure from Motion

- **Find a place / static object that you would like to reconstruct in 3D.**
  - Could be inside or outside a building.
  - Objects that reconstruct well typically have lots of textures, not a lot of symmetry, no shiny surfaces.
- **Take a bunch of pictures (~20-30) of the place of interest from various camera poses (say in a 45-90^{o} frustrum).**
- **Try to keep the object in the center of the image.**
  - Move around on the ground to see object from various perspectives.
  - Try higher/lower vantage points.

- **Install Visual SFM:**
  - http://www.cs.washington.edu/homes/ccwu/vsfm/

References

- Lowe, D. G., “Distinctive Image Features from Scale-Invariant Keypoints”, International Journal of Computer Vision, 60, 2, pp. 91-110, 2004
- Martin A. Fischler and Robert 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 (6): 381–395
- R. I. Hartley. In defence of the 8-point algorithm. In Proceedings of the IEEE International Conference on Computer Vision, 1995.
- R. Hartley and A. Zisserman, “Multiple view geometry in computer vision (2nd edition)”, Cambridge University Press, 2003.
  - Pedagogical material for today’s class was drawn from chapters 9, 10 and 11.
  - Excerpts of book can be found online here: http://www.robots.ox.ac.uk/~vgg/hzbook/