Image-based Modeling and Rendering

6. Basic Concepts of 3D Modeling from Images

National Chiao Tung Univ, Taiwan

By: I-Chen Lin, Assistant Professor

Objectives

- Estimate R, T from E.
- 3D position estimation
- Feature matching and extraction

Introduction

- How to determine geometry from images?
  - Passive approaches
    - Reconstruction from stereo images
    - Reconstruction from motion
  - Active approaches
    - Structured-light-based modeling

Ref:

• Image-based Modeling and Rendering, SIGGRAPH'99 course notes.

• J. Weng, T.S. Huang, N. Ahuja, Motion and Structure from Image Sequences, 1993.

• Camera calibration toolbox for matlab

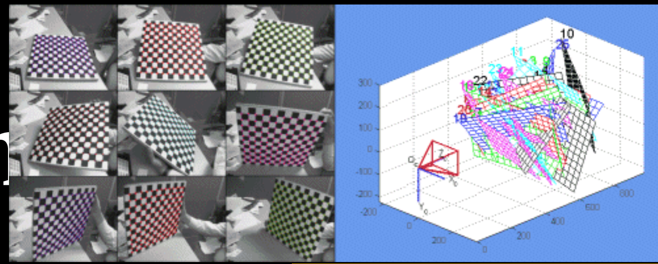
•D. Frolova, D. Simakov, Slides of "Matching with Invariant Features".

•, IEEE T. PAMI, IJCV, Proc. ICCV, Proc. CVPR ….

3D Geometry from Images

- Computer vision is the inverse of computer graphics.
- This talk describes some techniques for recovering 3D geometry from images.
- These techniques will be extended for image-based modeling.

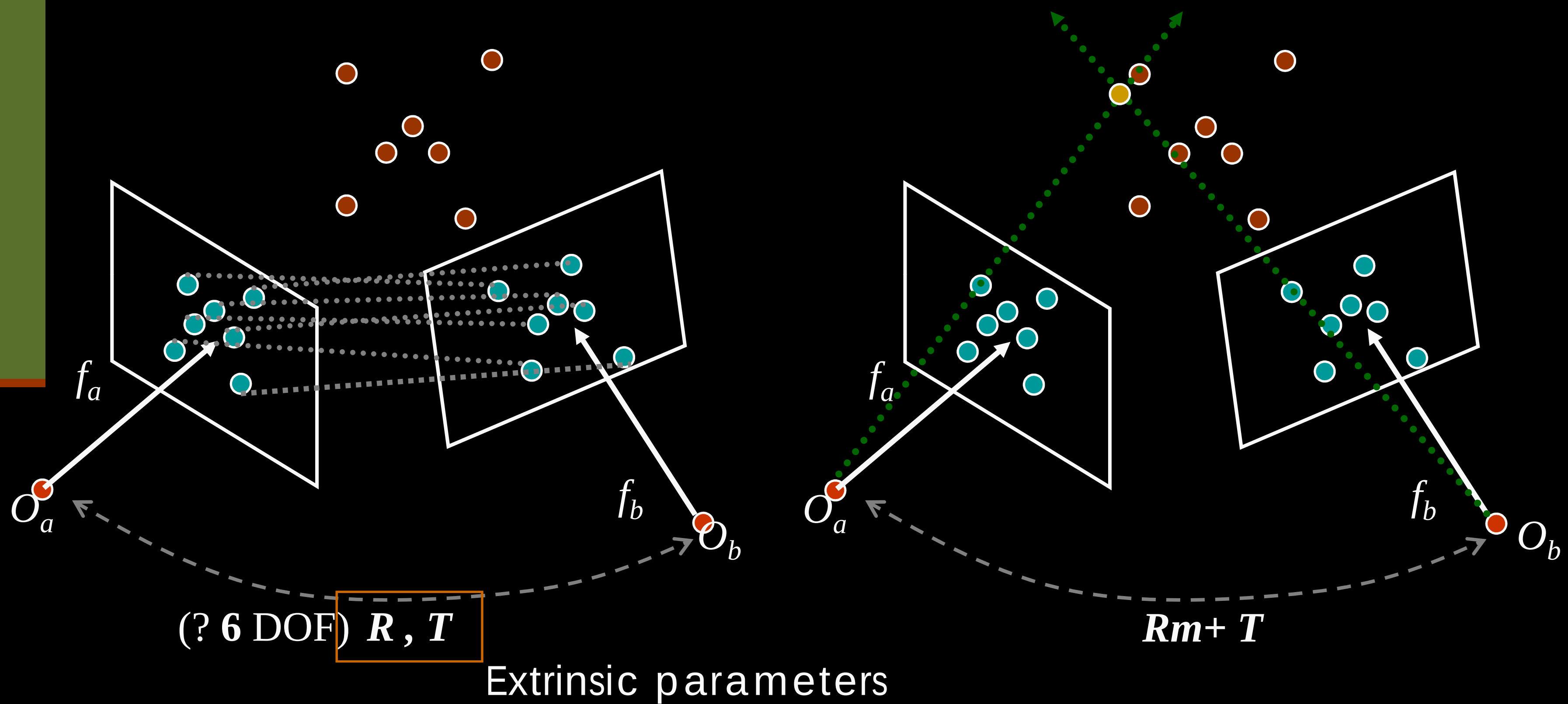
Camera Calibration



- Public camera calibration tools
  - A flexible new technique for camera calibration
    - http://research.microsoft.com/~zhang/calib/
    - Z. Zhang. A flexible new technique for camera calibration. IEEE Trans. Pattern Analysis and Machine Intelligence, 22(11):1330-1334, 2000.
  - Camera calibration toolbox for matlab
    - http://www.vision.caltech.edu/bouguetj/calib_doc/
  - Tsai’s camera model
    - http://www.cs.cmu.edu/~rgw/TsaiDesc.html
    - "A versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses", Roger Y. Tsai, IEEE J. Robotics and Automation, Vol. RA-3, No. 4, 1987, pages 323-344.

Triangulation (Stereo)

- Given some points in correspondence across two or more images (taken from calibrated cameras), $\{(u_j, v_j)\}$, compute the 3D location X



Triangulation (Stereo)

- Constructing 3D structure from two views.
  - H.C. Longuet-Higgins (Nature′81).
  - J.Weng et al. The two-view approach (PAMI′89).

- Given some points in correspondence across two images (in a normalized camera model), $\{(u_j, v_j)\}$,
  - Estimate R, T from corresponding points.
  - 3D position estimation from triangulation.
  - (optional) non-linear optimization

The Two-view Approach

- Without loss of generality, the images of different view direction *d1*, *d2* is regarded as a rigid-body motion of an object between *t1*, *t2*.

$\boldsymbol{x_i} = (x_i, y_i, z_i)$t is the 3D position of point P_i at time t_1.

$\boldsymbol{x_i'}=(x_i', y_i', z_i')$t is the 3D position of point P_i at time t_2.

$\boldsymbol{X_i}=(u_i, v_i, 1)$t is the projected vector of P_i at time t_1.

$\boldsymbol{X_i'}=(u_i', v_i', 1)$t = $(x_i'/z_i', y_i'/z_i', 1)$t is the projected vector of P_i at time t_2.

The Two-view Approach (1)

- Step (1). Solving for essential matrix E.

$$A = \begin{bmatrix} u_1u'_1 & u_1v'_1 & u_1 & v_1u'_1 & v_1v'_1 & v_1 & u'_1 & v'_1 & 1 \\ u_2u'_2 & u_2v'_2 & u_2 & v_2u'_2 & v_2v'_2 & v_2 & u'_2 & v'_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_nu'_n & u_nv'_n & u_n & v_nu'_n & v_nv'_n & v_n & u'_2 & v'_n & 1 \end{bmatrix}$$

- $\min_h \| Ah \| = 1$, subject to $\| h \| = 1$.

$$E = \begin{bmatrix} E_1 & E_2 & E_3 \end{bmatrix} = \sqrt{2} \begin{bmatrix} h_1 & h_4 & h_7 \\ h_2 & h_5 & h_8 \\ h_3 & h_6 & h_9 \end{bmatrix}$$

The solution of h is the unit eigenvector of A^tA associated with the smallest eigenvalue.

The Two-view Approach (2)

- Step (2). Determining a unit vector T_s with $T_0 = \pm T_s$.
  - $\min_{T_s} \| E^t T_s \|$, subject to $\| T_s \| = 1$.

The solution of T_s is the unit eigenvector of EE^t associated with the smallest eigenvalue.

$$E = T_\times R = \begin{bmatrix} E_1 & E_2 & E_3 \end{bmatrix} = \begin{bmatrix} T_\times R_1 & T_\times R_2 & T_\times R_3 \end{bmatrix}$$

$$\therefore E_1, E_2, E_3 \perp T \quad \Rightarrow \quad E^t T_s = 0$$

  - if($\Sigma_i (T_s \times X_i') \bullet (E X_i) < 0$), $T_s = -T_s$.

The Two-view Approach (3)

- Step (3). Determining rotation matrix *R*.
  - Without noise, $W=R$

    $$W = [(E_1 \times T_s + E_2 \times E_3) \ (E_2 \times T_s + E_3 \times E_1) \ (E_3 \times T_s + E_1 \times E_2)]$$

    Using the identity equation: (a x b) x c = (a · c) b – (b · c) a

  - With noise,

    $\min_R \| R - W \|$, subject to: *R* is a rotation matrix.

The Two-view Approach (3 app.)

- $\min_R \| RC - D \|$, subject to: R is a rotation matrix.
- Define a 4x4 matrix B by $B = \sum_{i=1}^{3} B_i^t B_i$

  where $B_i = \begin{bmatrix} 0 & (C_i - D_i)^t \\ D_i - C_i & [D_i + C_i]_\times \end{bmatrix}_{4\times 4}$
- $\mathbf{q} = (q_0, q_1, q_2, q_3)^t$ is the unit eigenvector of B associated with the smallest eigenvalue.

$$R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_2 q_1 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_3 q_1 - q_0 q_2) & 2(q_3 q_2 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

The Two-view Approach (4)

- Step (4). Checking $T = 0$, If $T \neq 0$, determine the sign of T_0.

if for all $i = 1 \sim n$, then report $T \approx 0$.

else if ($\Sigma_i (T_s \times X_i') \bullet (R X_i) > 0$), then $T_0 = T_s$, otherwise $T_0 = -T_s$.

The Two-view Approach (5)

- Step (5). If $T \neq 0$, estimate relative depths.
- To find $$Z_i = \left(\frac{z_i'}{\|T\|}, \frac{z_i}{\|T\|} \right)^t = \left(\tilde{z}_i', \tilde{z}_i \right)$$

by $$\min \left\| \left[X_i' \quad -RX_i \right] Z_i - T^0 \right\|$$

The Nonlinear Optimization

- Two-view linear algorithms are often easily disturbed by noise.
  - More calibration points.
  - Nonlinear optimization.
- First, take the result of the two-view linear algorithm as an initial guess.
- Approximate the *R*, *T* by $\min_m \{\| f(u,m) \|\}$ in a nonlinear least square approach
  - E.g. the Levenberg-Marquardt method, or the Gauss-Newton method.
  - $f(u,m) = prj(m, y(u,m)) - u$

where *u* is the observed projected position,
m is the motion parameters(*R*,*T*), *y*(*u*,*m*) is the best 3D positions of *P*,
and *prj*(*m*, *x*) is the projected position of the input structure x and motion m.

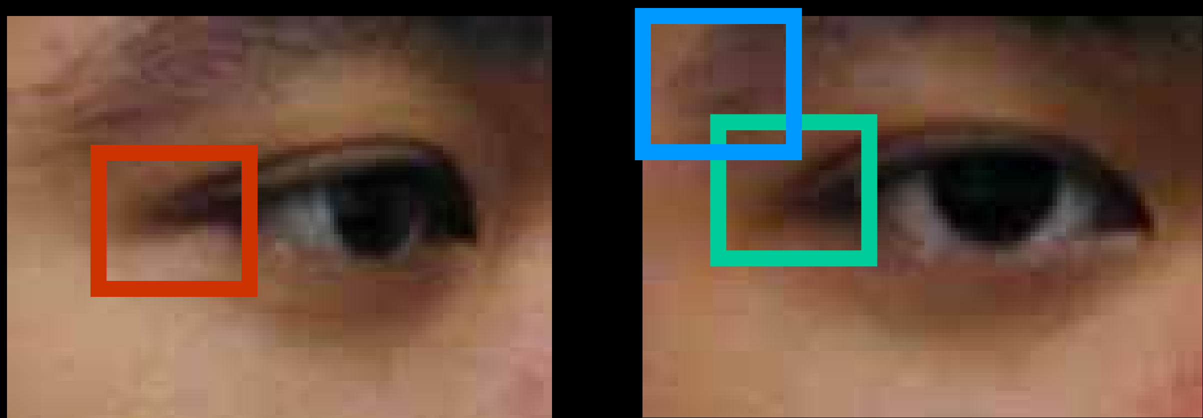
Triangulation limits

- Difficult to reliably estimate structure and motion unless:
  - large (*x* or *y*) rotation
  - large field of view and depth variation
- Camera calibration is important
- Need good feature trackers or manual assistance
- Post-processing of the resulting 3-D points?

Feature Matching (correlation)

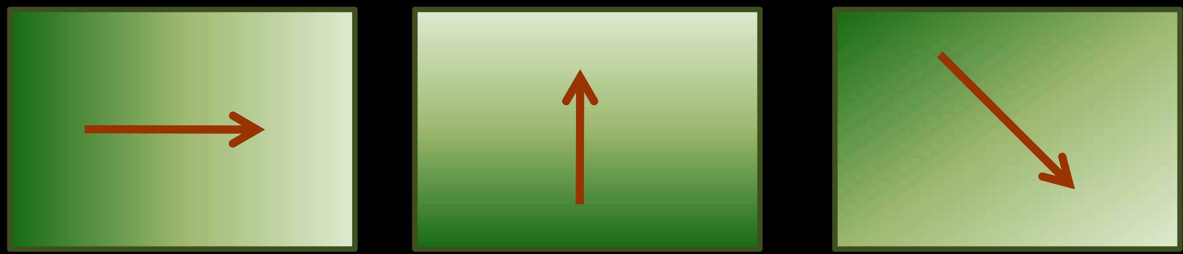
- Find corresponding points in image video sequence
  - one simple technique: find two patches with minimal summed squared error.

$$E_{x,y}(u,v) = \sum_{k=x-w}^{x+w} \sum_{l=y-w}^{y+w} \left[I_1(k+u, l+v) - I_0(k,l) \right]^2$$



Feature Extraction

- The gradient of an image: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$
  - E.g.



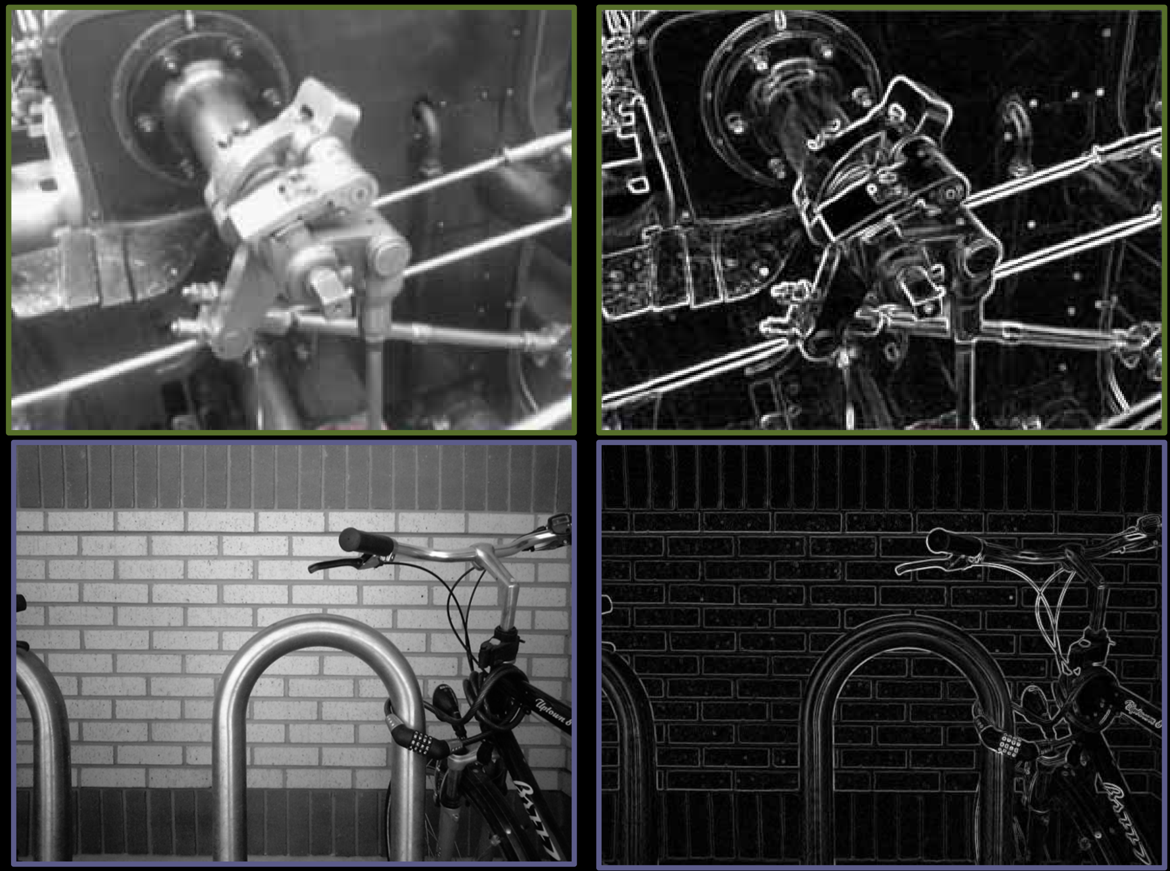
$$\nabla f = \left[\frac{\partial f}{\partial x}, 0 \right] \qquad \nabla f = \left[0, \frac{\partial f}{\partial y} \right] \qquad \nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$

- The gradient direction: $\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} \Big/ \frac{\partial f}{\partial x} \right)$

- The gradient magnitude: $\|\nabla f\| = \sqrt{(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2}$

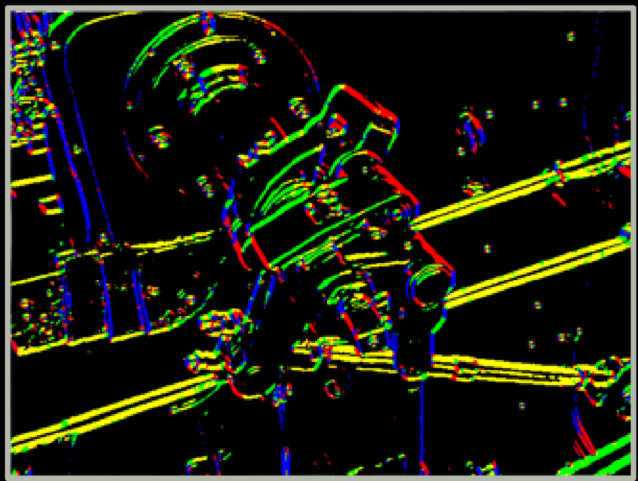
Sobel Edge Detection

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

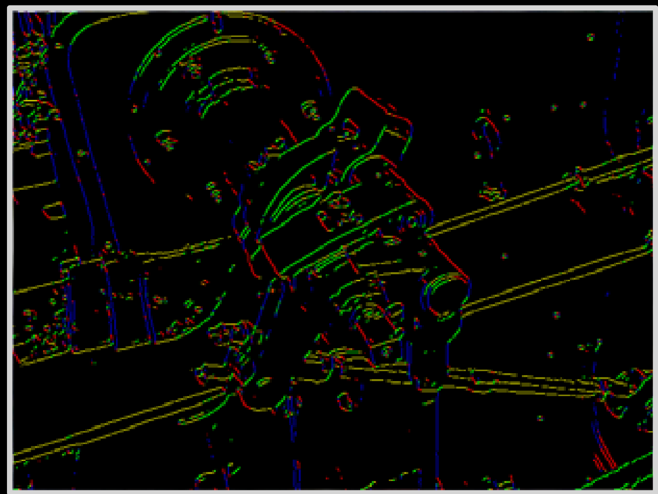


Canny Edge Detection

- Noise removal (Gaussian filtering)
- Gradients of the image (similar to Sobel)
- Non-maximum suppression
  - Check whether a pixel is a local maximum along gradient direction



Yellow for 0 degrees; green for 45 degrees; blue for 90 degrees ; red for 135 degrees



After non-maximum suppression

Corner Detection (Harris)

- Consider the matrix for a small square around (*x*,*y*)

$$A = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$

- The simplest case

$$A = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

  - If λ_1 0 and λ_2 0 then there are no features of interest at this pixel (*x*,*y*).
  - If λ_1 and λ_2 is some large positive values, then an edge is found.
  - If λ_1 and λ_2 are both large, distinct positive values, then a corner is found.

Corner Detection (Harris)

- More general cases:

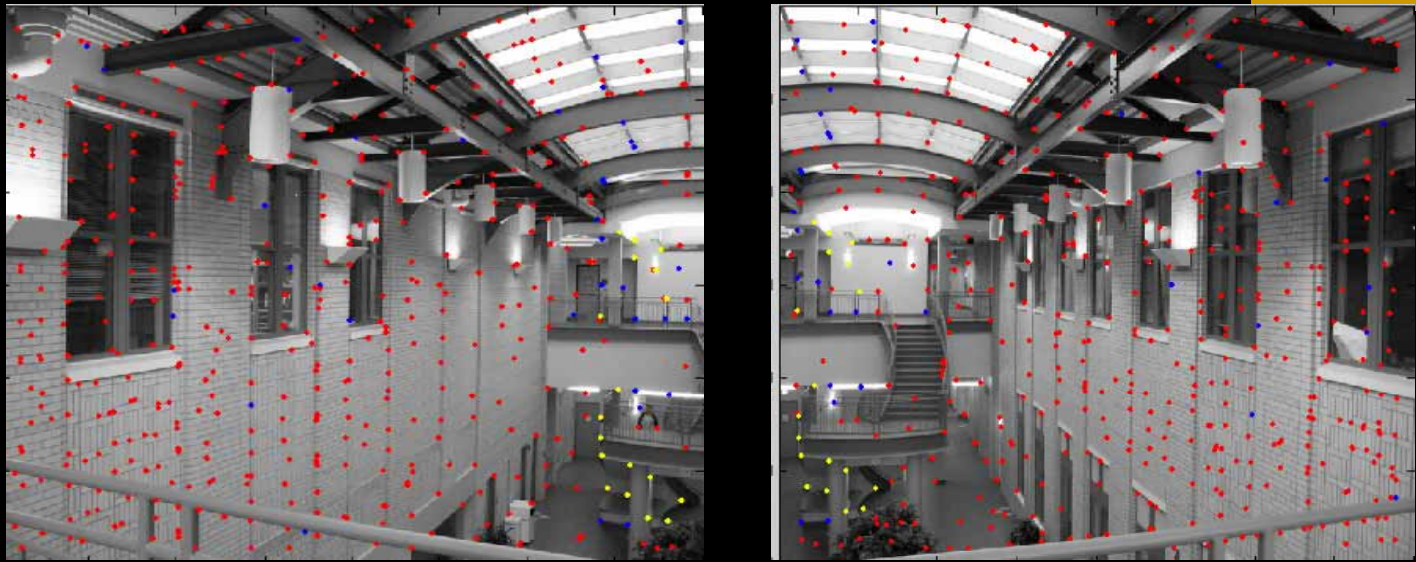
$$A = R^{-1}\begin{bmatrix}\lambda_1 & 0 \\ 0 & \lambda_2\end{bmatrix}R$$

where R is a rotation matrix.

  - Remind "Eigenvalues" or "Singular Value Decomposition (SVD)".

- Process steps
  - Apply Gaussian filter.
  - Evaluate magnitudes of the gradients.
  - Construct A.
  - Find λ_1 and λ_2 by evaluation of eigen values or SVD.
  - If they are both big, we have a corner.

Matching with Features



Figures from Alexei Efros, Computational Photography