Stereo matching
Calibration

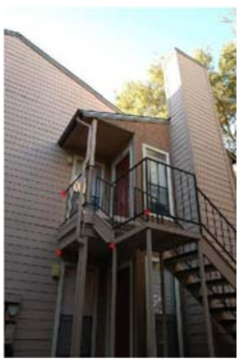
Thursday, Oct 23

Kristen Grauman
UT-Austin

Today

- Some Pset 2 results
- Correspondences, matching for stereo
  - A couple stereo applications
- Camera calibration
- Weak calibration
  - Fundamental matrix
  - 8-point algorithm

Ming-Jun Chen



Ming-Jun Chen

Ming-Jun Chen

Ming-Jun Chen

Andrew Harp

Wei-Cheng Su

Wei-Cheng Su

Wei-Cheng Su

Wei-Cheng Su

Andy Luong

Andy Luong

Andy Luong

Chia-Sheng Tsai

Chia-Sheng Tsai

Chia-Sheng Tsai

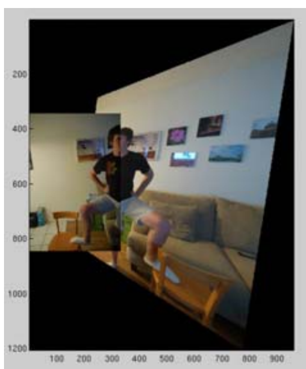
Chia-Sheng Tsai

Amirshahed Mehrtash

Amirshahed Mehrtash

Amirshahed Mehrtash

Cameron Davison



Jeffrey Dang

Fei Li

Sung Ju Hwang

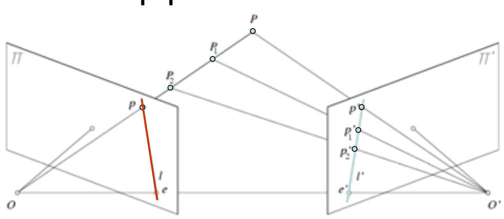
Sung Ju Hwang

Sung Ju Hwang

Sung Ju Hwang

Sung Ju Hwang

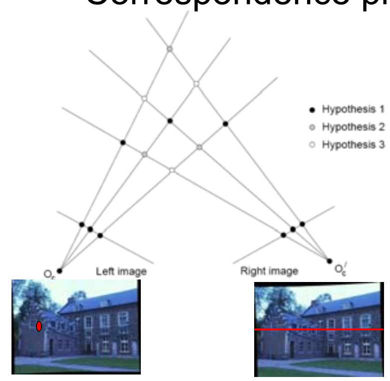
Epipolar constraint



- Potential matches for *p* have to lie on the corresponding epipolar line *l'*.
- Potential matches for *p'* have to lie on the corresponding epipolar line *l*.

Source: M. Pollefeys

Correspondence problem



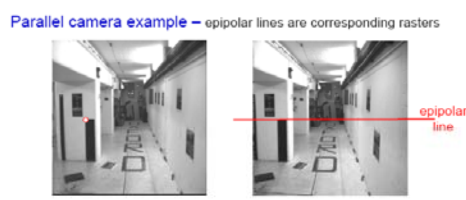
Multiple match hypotheses satisfy epipolar constraint, but which is correct?

Figure from Gee & Cipolla 1999

Correspondence problem

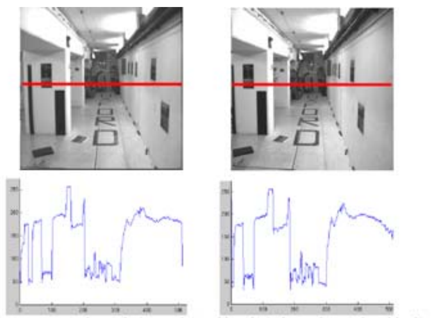
- Beyond the hard constraint of epipolar geometry, there are "soft" constraints to help identify corresponding points
  - Similarity
  - Uniqueness
  - Ordering
  - Disparity gradient
- To find matches in the image pair, we will assume
  - Most scene points visible from both views
  - Image regions for the matches are similar in appearance

Correspondence problem



Source: Andrew Zisserman

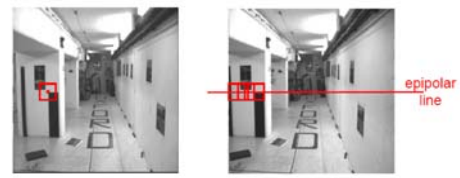
Intensity profiles



- Clear correspondence between intensities, but also noise and ambiguity

Source: Andrew Zisserman

Correspondence problem



Neighborhood of corresponding points are similar in intensity patterns.

Source: Andrew Zisserman

Normalized cross correlation

subtract mean: $A \leftarrow A - \langle A \rangle, B \leftarrow B - \langle B \rangle$

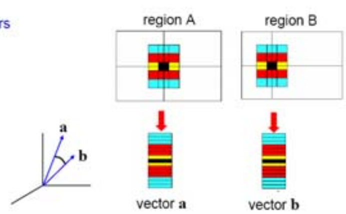
$$\text{NCC} = \frac{\sum_i \sum_j A(i,j)B(i,j)}{\sqrt{\sum_i \sum_j A(i,j)^2}\sqrt{\sum_i \sum_j B(i,j)^2}}$$

Write regions as vectors

$\mathtt{A} \rightarrow \mathbf{a}, \ \mathtt{B} \rightarrow \mathbf{b}$

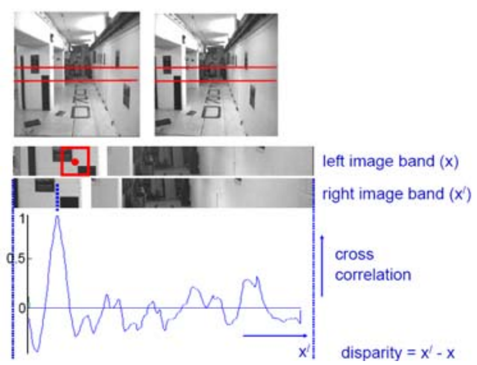
$$\text{NCC} = \frac{\mathbf{a}.\mathbf{b}}{|\mathbf{a}||\mathbf{b}|}$$

$$-1 \leq \text{NCC} \leq 1$$



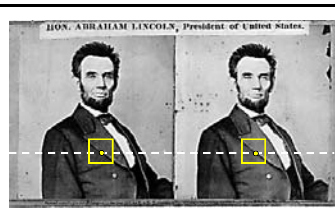
Source: Andrew Zisserman

Correlation-based window matching



Source: Andrew Zisserman

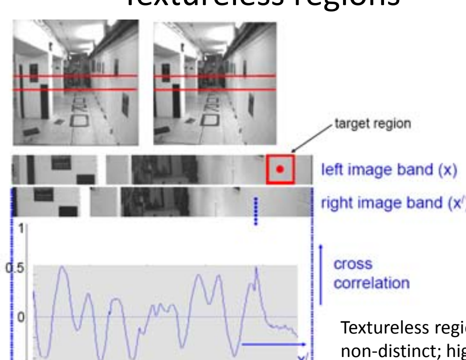
Dense correspondence search



For each epipolar line
- For each pixel / window in the left image
  - compare with every pixel / window on same epipolar line in right image
  - pick position with minimum match cost (e.g., SSD, correlation)

Adapted from Li Zhang

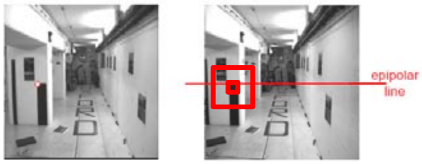
Textureless regions



Textureless regions are non-distinct; high ambiguity for matches.

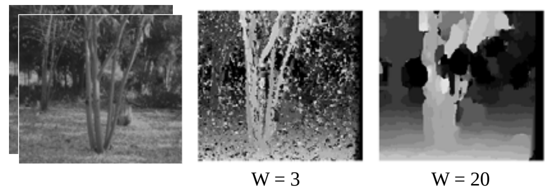
Source: Andrew Zisserman

Effect of window size



Source: Andrew Zisserman

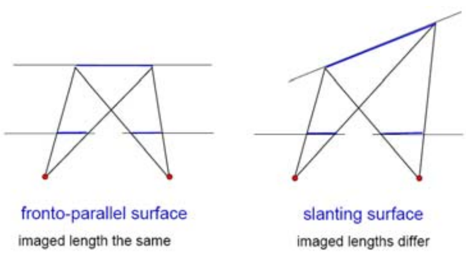
Effect of window size



Want window large enough to have sufficient intensity variation, yet small enough to contain only pixels with about the same disparity.

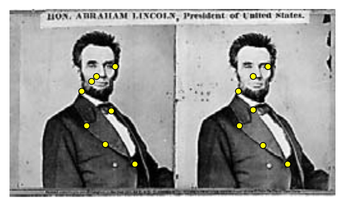
Figures from Li Zhang

Foreshortening effects



Source: Andrew Zisserman

Sparse correspondence search



- Restrict search to sparse set of detected features
- Rather than pixel values (or lists of pixel values) use *feature descriptor* and an associated *feature distance*
- Still narrow search further by epipolar geometry

Correspondence problem

- Beyond the hard constraint of epipolar geometry, there are “soft” constraints to help identify corresponding points
  - Similarity
  - Uniqueness
  - Ordering
  - Disparity gradient

Uniqueness

- For opaque objects, up to one match in right image for every point in left image

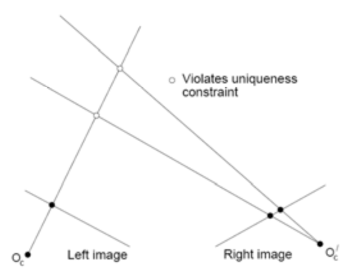


Figure from Gee & Cipolla 1999

Ordering constraint

- Points on **same surface** (opaque object) will be in same order in both views

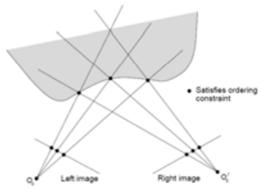
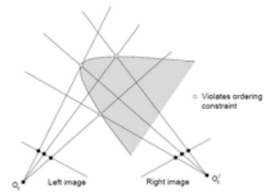
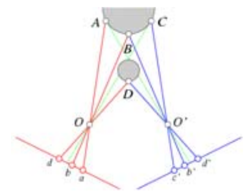


Figure from Gee & Cipolla 1999

Ordering constraint

- Won't always hold, e.g. consider transparent object, or an occluding surface

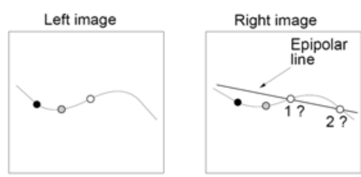




Figures from Forsyth & Ponce

Disparity gradient

- Assume piecewise continuous surface, so want disparity estimates to be locally smooth

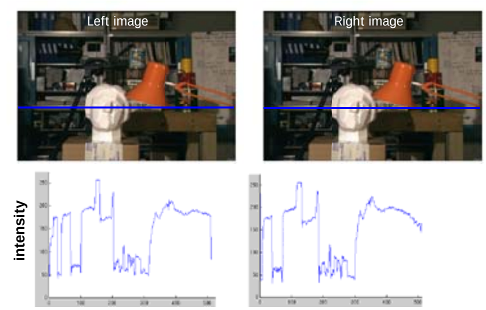


Given matches ● and ○, point ○ in the left image must match point 1 in the right image. Point 2 would exceed the disparity gradient limit.

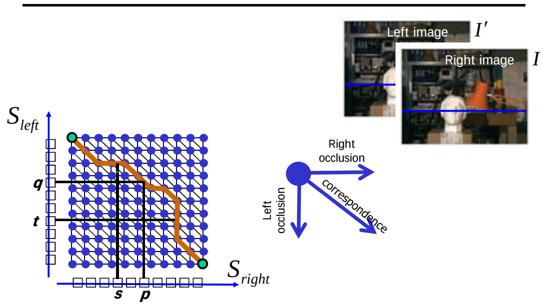
Figure from Gee & Cipolla 1999

Scanline stereo

- Try to coherently match pixels on the entire scanline
- Different scanlines are still optimized independently



"Shortest paths" for scan-line stereo



Can be implemented with dynamic programming

Ohta & Kanade '85, Cox et al. '96

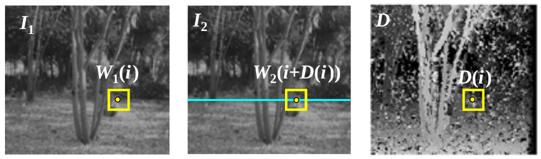
Slide credit: Y. Boykov

Coherent stereo on 2D grid

- Scanline stereo generates streaking artifacts

- Can't use dynamic programming to find spatially coherent disparities/ correspondences on a 2D grid

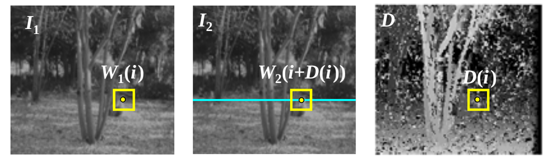
Stereo matching as energy minimization



$$E = \alpha E_{\text{data}}(I_1, I_2, D) + \beta E_{\text{smooth}}(D)$$

$$E_{\text{data}} = \sum_i \left(W_1(i) - W_2(i + D(i))\right)^2 \qquad E_{\text{smooth}} = \sum_{\text{neighbors } i,j} \rho\left(D(i) - D(j)\right)$$

Stereo matching as energy minimization



$$E = \alpha E_{\text{data}}(I_1, I_2, D) + \beta E_{\text{smooth}}(D)$$

$$E_{\text{data}} = \sum_i \left(W_1(i) - W_2(i + D(i))\right)^2 \qquad E_{\text{smooth}} = \sum_{\text{neighbors } i,j} \rho\left(D(i) - D(j)\right)$$

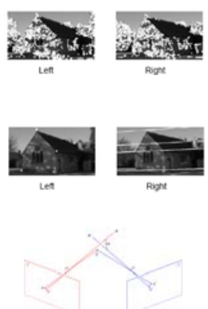
- Energy functions of this form can be minimized using *graph cuts*

Y. Boykov, O. Veksler, and R. Zabih, [Fast Approximate Energy Minimization via Graph Cuts](), PAMI 2001

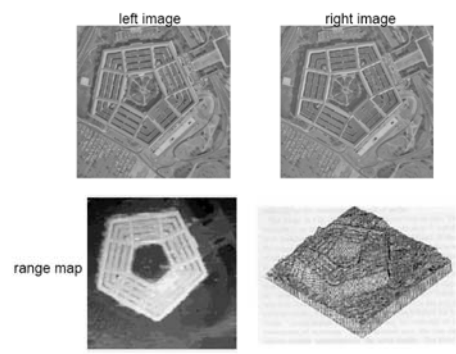
Source: Steve Seitz

Recap: stereo with calibrated cameras

- Image pair
- Detect some features
- Compute **E** from given **R** and **T**
- Match features using the epipolar and other constraints
- Triangulate for 3d structure



•



Z-keying for virtual reality

- Merge synthetic and real images given depth maps

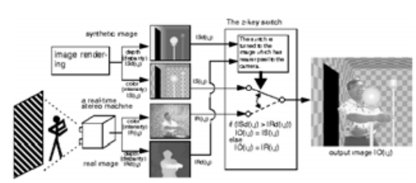
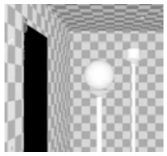


Figure 1: A schema of the z-key method

Kanade et al., CMU, 1995

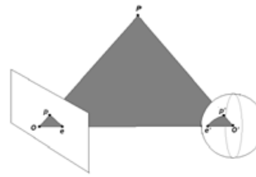
Z-keying for virtual reality



Kanade et al., CMU, 1995

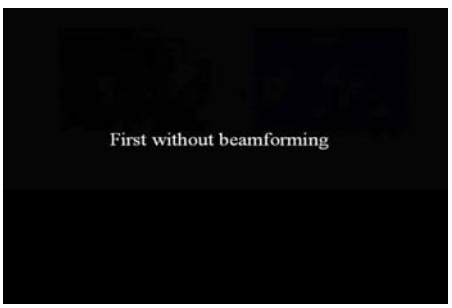
http://www.cs.cmu.edu/afs/cs/project/stereo-machine/www/z-key.html

An audio camera & epipolar geometry



Spherical microphone array

Adam O' Donovan, Ramani Duraiswami and Jan Neumann
Microphone Arrays as Generalized Cameras for Integrated Audio Visual Processing, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Minneapolis, 2007



- Adam O' Donovan, Ramani Duraiswami and Jan Neumann. Microphone Arrays as Generalized Cameras for Integrated Audio Visual Processing, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Minneapolis, 2007

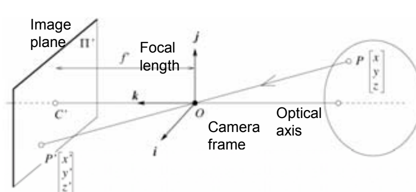
Uncalibrated case

- What if we don't know the camera parameters?

Today

- Some Pset 2 results
- Correspondences, matching for stereo
  - A couple stereo applications
- Camera calibration
- Weak calibration
  - Fundamental matrix
  - 8-point algorithm

Perspective projection



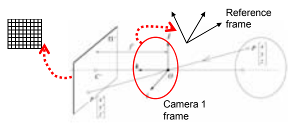
$$(x,y,z) \rightarrow (f\frac{x}{z}, f\frac{y}{z})$$

Scene point $\longrightarrow$ Image coordinates

Thus far, in camera's reference frame only.

Camera parameters

- **Extrinsic: location and orientation of camera frame with respect to reference frame**
- Intrinsic: how to map pixel coordinates to image plane coordinates



Extrinsic camera parameters

$$\mathbf{P}_c = \mathbf{R}(\mathbf{P}_w - \mathbf{T})$$

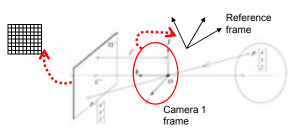
Camera reference frame

World reference frame

$$\mathbf{P}_c = (X, Y, Z)^T$$

Camera parameters

- Extrinsic: location and orientation of camera frame with respect to reference frame
- **Intrinsic: how to map pixel coordinates to image plane coordinates**



Intrinsic camera parameters

- Ignoring any geometric distortions from optics, we can describe them by:

$$x = -(x_{im} - o_x)s_x$$

$$y = -(y_{im} - o_y)s_y$$

Coordinates of projected point in camera reference frame

Coordinates of image point in pixel units

Coordinates of image center in pixel units

Effective size of a pixel (mm)

Camera parameters

- We know that in terms of camera reference frame:

$$x = f\frac{X}{Z} \qquad y = f\frac{Y}{Z} \qquad \text{and} \qquad \begin{array}{l}\mathbf{P}_c = \mathbf{R}(\mathbf{P}_w - \mathbf{T}) \\ \mathbf{P}_c = (X, Y, Z)^T\end{array}$$

- Substituting previous eqns describing intrinsic and extrinsic parameters, can relate *pixels coordinates* to *world points:*

$$-(x_{im} - o_x)s_x = f\frac{\mathbf{R}_1 \cdot (\mathbf{P}_w - \mathbf{T})}{\mathbf{R}_3 \cdot (\mathbf{P}_w - \mathbf{T})}$$

$$-(y_{im} - o_y)s_y = f\frac{\mathbf{R}_2 \cdot (\mathbf{P}_w - \mathbf{T})}{\mathbf{R}_3 \cdot (\mathbf{P}_w - \mathbf{T})}$$

$\mathbf{R}_i$ = Row i of rotation matrix

Projection matrix

- This can be rewritten as a matrix product using homogeneous coordinates:

$$\begin{pmatrix} wx_{im} \\ wy_{im} \\ w \end{pmatrix} = \mathbf{M}_{int}\,\mathbf{M}_{ext} \underbrace{\begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}}_{\text{point in camera coordinates}}$$

$$x_{im} = wx_{im} / w$$

$$y_{im} = wy_{im} / w$$

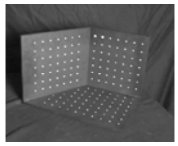
$$\mathbf{M}_{int} = \begin{pmatrix} -f/s_x & 0 & o_x \\ 0 & -f/s_y & o_y \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{M}_{ext} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & -\mathbf{R}_1^T\mathbf{T} \\ r_{21} & r_{22} & r_{23} & -\mathbf{R}_2^T\mathbf{T} \\ r_{31} & r_{32} & r_{33} & -\mathbf{R}_3^T\mathbf{T} \end{pmatrix}$$

Calibrating a camera

- Compute intrinsic and extrinsic parameters using observed camera data

Main idea

- Place “calibration object” with known geometry in the scene
- Get correspondences
- Solve for mapping from scene to image: estimate **M=M**$_{int}$**M**$_{ext}$



The Opti-CAL Calibration Target Image

Projection matrix

- This can be rewritten as a matrix product using homogeneous coordinates:

$$\begin{pmatrix} wx_{im} \\ wy_{im} \\ w \end{pmatrix} = \underbrace{\mathbf{M}_{int}\,\mathbf{M}_{ext}}_{\mathbf{M}} \overbrace{\begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}}^{\mathbf{P}_w \text{ in homog.}}$$

product **M** is single **projection matrix** encoding both extrinsic and intrinsic parameters

$$x_{im} = \frac{\mathbf{M}_1 \cdot \mathbf{P}_w}{\mathbf{M}_3 \cdot \mathbf{P}_w}$$

$$y_{im} = \frac{\mathbf{M}_2 \cdot \mathbf{P}_w}{\mathbf{M}_3 \cdot \mathbf{P}_w}$$

Let $\mathbf{M}_i$ be row i of matrix **M**

Estimating the projection matrix

For a given feature point

$$x_{im} = \frac{\mathbf{M}_1 \cdot \mathbf{P}_w}{\mathbf{M}_3 \cdot \mathbf{P}_w} \longrightarrow 0 = (\mathbf{M}_1 - x_{im}\mathbf{M}_3)\cdot \mathbf{P}_w$$

$$y_{im} = \frac{\mathbf{M}_2 \cdot \mathbf{P}_w}{\mathbf{M}_3 \cdot \mathbf{P}_w} \longrightarrow 0 = (\mathbf{M}_2 - y_{im}\mathbf{M}_3)\cdot \mathbf{P}_w$$

Estimating the projection matrix

$$\boxed{\begin{array}{l} 0 = (\mathbf{M}_1 - x_{im}\mathbf{M}_3)\cdot \mathbf{P}_w \\ 0 = (\mathbf{M}_2 - y_{im}\mathbf{M}_3)\cdot \mathbf{P}_w \end{array}}$$

Expanding this first equation, we have:

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \end{bmatrix} - x_{im}\begin{bmatrix} m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} * \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = 0$$

$$(X_w \boxed{m_{11}} - X_w x_{im} \boxed{m_{31}}) + (Y_w \boxed{m_{12}} - Y_w x_{im} \boxed{m_{32}})\ldots$$

$$\ldots + (Z_w \boxed{m_{13}} - Z_w x_{im} \boxed{m_{33}}) + (\boxed{m_{14}} - x_{im} \boxed{m_{34}}) = 0$$

Estimating the projection matrix

$$\boxed{\begin{array}{l} 0 = (\mathbf{M}_1 - x_{im}\mathbf{M}_3)\cdot \mathbf{P}_w \\ 0 = (\mathbf{M}_2 - y_{im}\mathbf{M}_3)\cdot \mathbf{P}_w \end{array}}$$

$$\begin{pmatrix} X_w & Y_w & Z_w & 1 & 0 & 0 & 0 & 0 & -x_{im}X_w & -x_{im}Y_w & -x_{im}Z_w & -x_{im} \\ 0 & 0 & 0 & 0 & X_w & Y_w & Z_w & 1 & -y_{im}X_w & -y_{im}Y_w & -y_{im}Z_w & -y_{im} \end{pmatrix} \begin{pmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Estimating the projection matrix

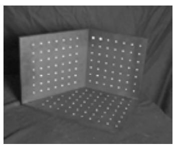
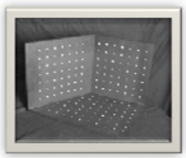
This is true for every feature point, so we can stack up *n* observed image features and their associated 3d points in single equation: **Pm = 0**

$$\underbrace{\begin{pmatrix} X_w^{(1)} & Y_w^{(1)} & Z_w^{(1)} & 1 & 0 & 0 & 0 & 0 & -x_{im}^{(1)}X_w^{(1)} & -x_{im}^{(1)}Y_w^{(1)} & -x_{im}^{(1)}Z_w^{(1)} & -x_{im}^{(1)} \\ 0 & 0 & 0 & 0 & X_w^{(1)} & Y_w^{(1)} & Z_w^{(1)} & 1 & -y_{im}^{(1)}X_w^{(1)} & -y_{im}^{(1)}Y_w^{(1)} & -y_{im}^{(1)}Z_w^{(1)} & -y_{im}^{(1)} \\ & & \ldots & & & \ldots & & & \ldots & & \ldots & \\ X_w^{(n)} & Y_w^{(n)} & Z_w^{(n)} & 1 & 0 & 0 & 0 & 0 & -x_{im}^{(n)}X_w^{(n)} & -x_{im}^{(n)}Y_w^{(n)} & -x_{im}^{(n)}Z_w^{(n)} & -x_{im}^{(n)} \\ 0 & 0 & 0 & 0 & X_w^{(n)} & Y_w^{(n)} & Z_w^{(n)} & 1 & -y_{im}^{(n)}X_w^{(n)} & -y_{im}^{(n)}Y_w^{(n)} & -y_{im}^{(n)}Z_w^{(n)} & -y_{im}^{(n)} \end{pmatrix}}_{\mathbf{P}} \underbrace{\begin{pmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{pmatrix}}_{\mathbf{m}} = \begin{pmatrix} 0 \\ 0 \\ \ldots \\ 0 \\ 0 \end{pmatrix}$$

Solve for m_{ij}'s (the calibration information) [F&P Section 3.1]

Summary: camera calibration

- Associate image points with scene points on object with known geometry
- Use together with perspective projection relationship to estimate projection matrix
- (Can also solve for explicit parameters themselves)

When would we calibrate this way?

- Makes sense when geometry of system is not going to change over time
- …When would it change?

Weak calibration

- Want to estimate world geometry without requiring calibrated cameras
  - Archival videos
  - Photos from multiple unrelated users
  - Dynamic camera system
- Main idea:
  - Estimate epipolar geometry from a (redundant) set of point correspondences between two uncalibrated cameras

Uncalibrated case

For a given camera: $\overline{\mathbf{p}} = \mathbf{M}_{\text{int}} \mathbf{p}$ Camera coordinates

So, for two cameras (left and right):

$$\mathbf{p}_{(left)} = \mathbf{M}^{-1}_{left,\text{int}} \overline{\mathbf{p}}_{(left)}$$

$$\mathbf{p}_{(right)} = \underbrace{\mathbf{M}^{-1}_{right,\text{int}}}_{\text{Internal calibration matrices, one per camera}} \overline{\mathbf{p}}_{(right)}$$

Camera coordinates

Image pixel coordinates

Uncalibrated case: fundamental matrix

$$\mathbf{p}_{(left)} = \mathbf{M}^{-1}_{left,\text{int}} \overline{\mathbf{p}}_{(left)}$$

$$\mathbf{p}_{(right)} = \mathbf{M}^{-1}_{right,\text{int}} \overline{\mathbf{p}}_{(right)}$$

$$\boxed{\mathbf{p}_{(right)}^{\mathrm{T}} \mathbf{E} \mathbf{p}_{(left)} = 0}$$

From before, the **essential** matrix **E.**

$$\left(\mathbf{M}^{-1}_{right,\text{int}} \overline{\mathbf{p}}_{right}\right)^{\mathrm{T}} \mathbf{E} \left(\mathbf{M}^{-1}_{left,\text{int}} \overline{\mathbf{p}}_{left}\right) = 0$$

$$\overline{\mathbf{p}}^{\mathrm{T}}_{right} \underbrace{\left(\mathbf{M}^{-\mathrm{T}}_{right,\text{int}} \mathbf{E} \mathbf{M}^{-1}_{left,\text{int}}\right)}\overline{\mathbf{p}}_{left} = 0$$

$$\overline{\mathbf{p}}^{\mathrm{T}}_{right} \mathbf{F} \overline{\mathbf{p}}_{left} = 0$$

Fundamental matrix

Fundamental matrix

- Relates pixel coordinates in the two views
- More general form than essential matrix: we remove need to know intrinsic parameters
- If we estimate fundamental matrix from correspondences in pixel coordinates, can reconstruct epipolar geometry without intrinsic or extrinsic parameters

Computing F from correspondences

$$\mathbf{F} = \left(\mathbf{M}_{right,\text{int}}^{-\text{T}} \mathbf{E} \mathbf{M}_{left.\text{int}}^{-1}\right)$$

$$\overline{\mathbf{p}}_{right}^{\text{T}} \mathbf{F} \overline{\mathbf{p}}_{left} = 0$$

- Cameras are uncalibrated: we don't know **E** or left or right $\mathbf{M}_{\text{int}}$ matrices
- Estimate F from 8+ point correspondences.

Computing F from correspondences

Each point correspondence generates one constraint on F

$$\overline{\mathbf{p}}_{right}^{\text{T}} \mathbf{F} \overline{\mathbf{p}}_{left} = 0$$

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

Collect n of these constraints

$$\begin{bmatrix} u'_1u_1 & u'_1v_1 & u'_1 & v'_1u_1 & v'_1v_1 & v'_1 & u_1 & v_1 & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = \mathbf{0}$$

Solve for f , vector of parameters.

Stereo pipeline with weak calibration

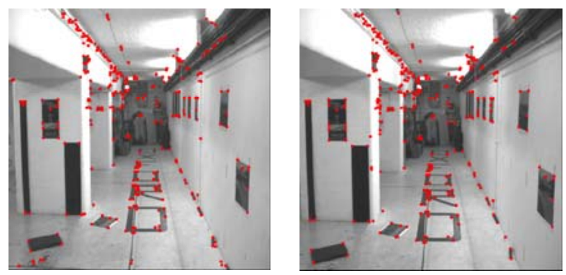
- So, where to start with uncalibrated cameras?
  - Need to find fundamental matrix F **and** the correspondences (pairs of points (u',v') ↔ (u,v)).

- 1) Find interest points in image (more on this later)
- 2) Compute correspondences
- 3) Compute epipolar geometry
- 4) Refine

Example from Andrew Zisserman

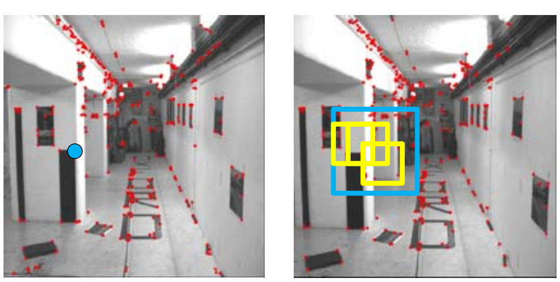
Stereo pipeline with weak calibration

1) Find interest points (next week)

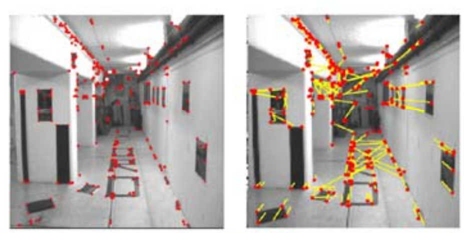


Stereo pipeline with weak calibration

2) Match points only using proximity



Putative matches based on correlation search

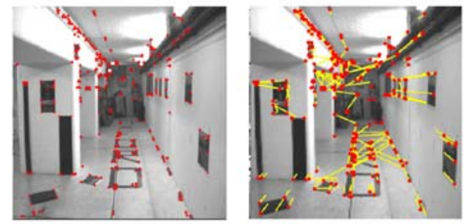


- Many wrong matches (10-50%), but enough to compute F

RANSAC for robust estimation of the fundamental matrix

- Select random sample of correspondences
- Compute F using them
  - This determines epipolar constraint
- Evaluate amount of support – inliers within threshold distance of epipolar line
- Choose F with most support (inliers)

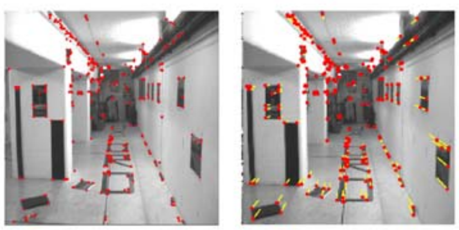
Putative matches based on correlation search



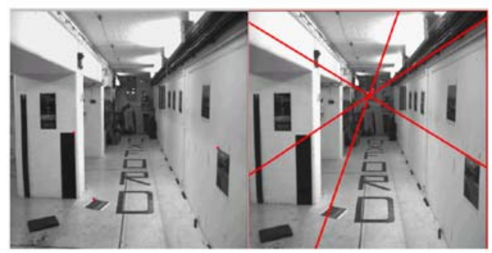
• Many wrong matches (10-50%), but enough to compute F

Pruned matches

- Correspondences consistent with epipolar geometry



- Resulting epipolar geometry



Next

- How to find interest points?
- How to describe local neighborhoods more robustly than with a list of pixel intensities?