Lecture 12: CS677

Sept 28, 2017

Admin

- HW3 posted, due Oct 5
- Exam 1, October 12
  - Class period (9-10:50AM)
    - “Local” DEN students must come to campus
    - “Distant” DEN students should arrange with their coordinator
  - Closed book, Closed notes
  - Topics to be covered in Lec 11 slides

Review

- Previous class
  - Stereo Geometry
    - Epipolar lines, essential matrix
    - Local Stereo matching
      - Adaptive Support Weighting (ASW)
- Today's objective
  - Global matching for stereo analysis
  - Intro to Structure from Motion

Global Optimization

- General formulation
  - Compute a *disparity space image* (DSI) which contains a value for each (x, y, d), *e.g.* cost of matching (x,y) at disparity d
  - In DSI, find a surface (*i.e.* d values for each x,y position) that minimizes an energy function such as

$$E(d) = E_{data}(d) + \lambda E_{smooth}(d).$$

$$E_{data}(d) = \sum_{(x,y)} C(x, y, d(x, y))$$

$$\begin{aligned} E_{smooth}(d) = \sum_{(x,y)} & \rho(d(x, y) - d(x{+}1, y)) + \\ & \rho(d(x, y) - d(x, y{+}1)), \end{aligned}$$

ρ is a monotonically increasing function of the disparity differences.,

Alternately: $E_s(d) = \rho_d(d(x,y) - d(x+1,y)) \cdot \rho_I(\|I(x,y) - I(x+1,y)\|)$

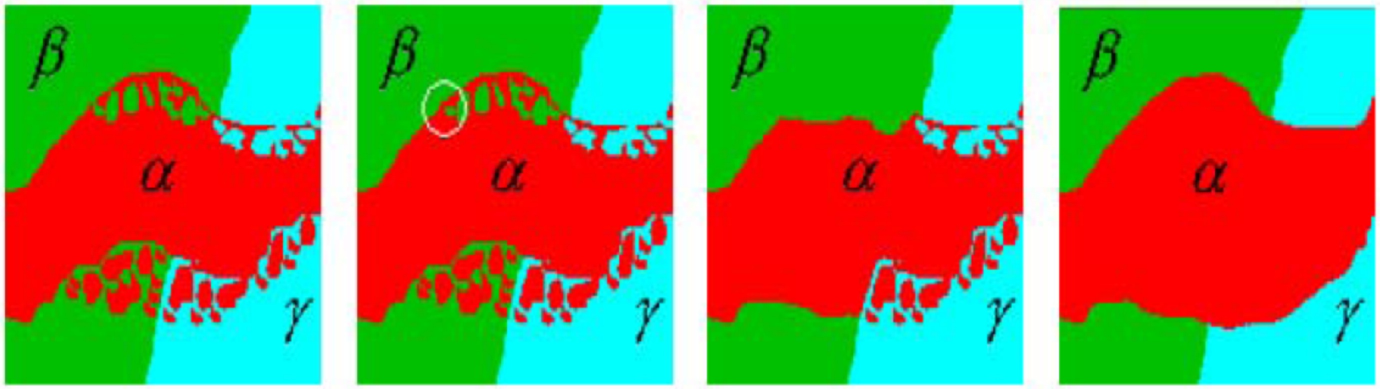
Prefers disparity discontinuities to coincide with the intensity discontinuities

Optimization Method

- In general, optimization problem is NP-hard so we compute approximations.
- One successful approach is to use graph-cut (max-flow) operations
  - An example algorithm based on work of Boykov et al is shown on next slide
  - Refs: papers by Boykov, Veksler and Zabih (ICCV 1999 is the short version, PAMI 2001 is a more detailed version; neither is required reading for cs677)
  - One major limitation is that disparity is limited to being an integer value

α-expansions and α-β swaps

- α–move; does changing a pixel disparity to α lower total energy?
- α–expansion: convert pixels to label α if it reduces energy. Compute all relabels in one step by using graph cuts (for certain energy functions)
- α-β swap: exchange dispartity values among a pair of pixels, check if energy is reduced.
  - Optimal set of all swaps (for a given α and a given β) can be computed simultaneously by a graph-cut construction
  - Repeat for all values of α-β pairs
- Note swap and expansion are two separate algorithms



(a) initial labeling (b) standard move (c) α-β-swap (d) α-expansion

Graph Cut for α-β Swap (Optional)

- Optimal α-β swap is given by min cut of the constructed graph (1-D example shown). Details omitted, see papers by Boykov, Veksler and Zabih.
- Weights depend on d_p and also on smoothness term to pixels in P_α or P_β

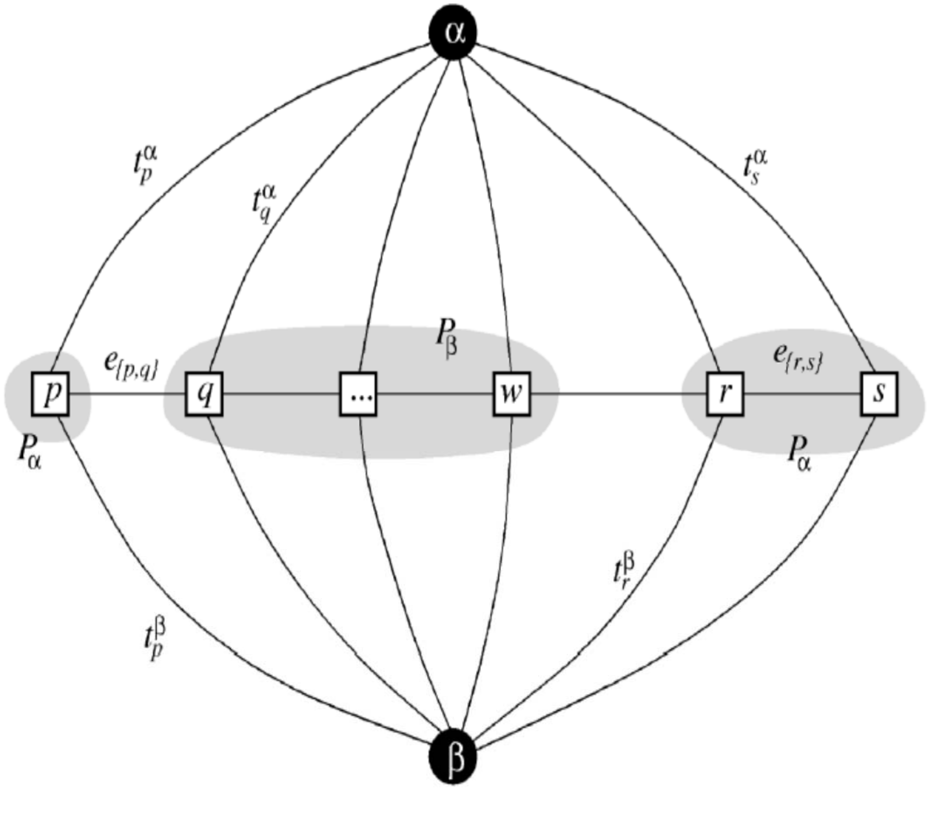
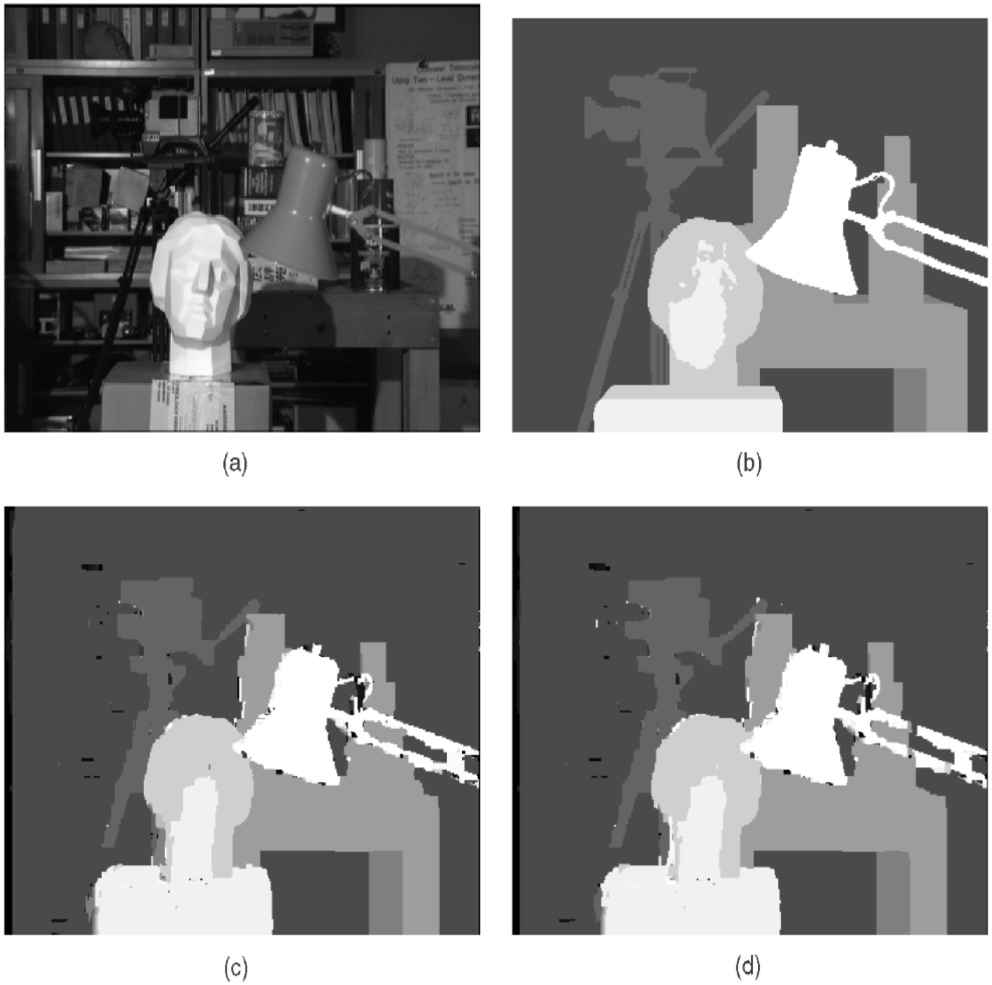


Fig. 4. An example of the graph $\mathcal{G}_{\alpha\beta}$ for a 1D image. The set of pixels in the image is $\mathcal{P}_{\alpha\beta} = \mathcal{P}_\alpha \cup \mathcal{P}_\beta$, where $\mathcal{P}_\alpha = \{p, r, s\}$ and $\mathcal{P}_\beta = \{q, \ldots, w\}$.

Results



a) Image,

b) Ground Truth

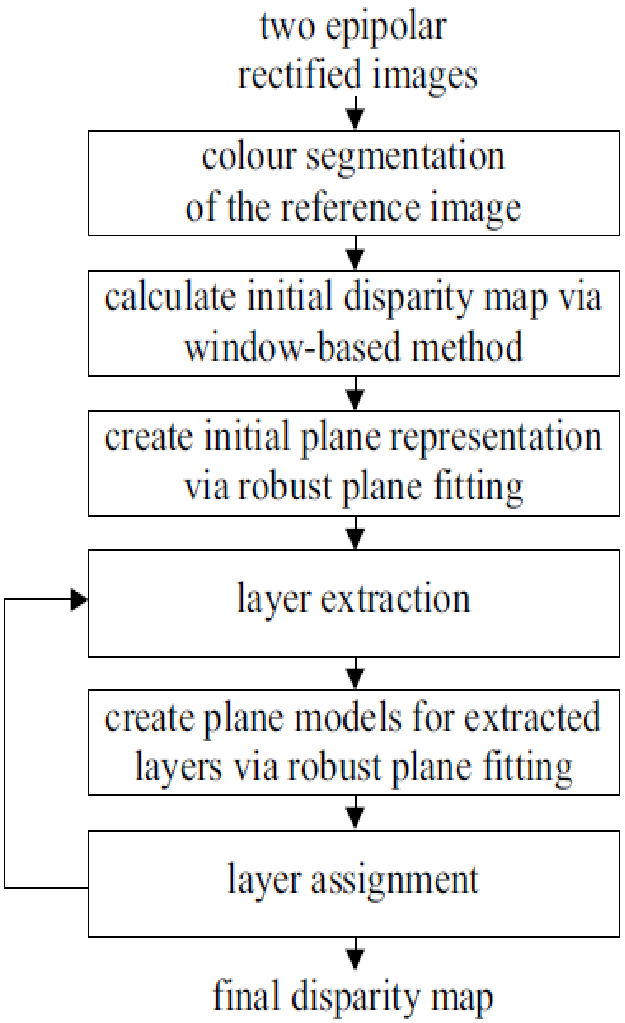
c) Using α-β swap

d) Using α expansion

Combining Segmentation and Matching (Optional)

- For better performance, we should use monocular information along with stereo information
  - Regions that are uniform in some image property (such as color) can be expected to be continuous in 3-D space as well. In particular, some algorithms assume that such regions are planar.
  - Many such algorithms exist, we briefly cover one by Bleyer and Gelautz (ICPR 2004).

BG Flow Chart (optional)



two epipolar rectified images

colour segmentation of the reference image

calculate initial disparity map via window-based method

create initial plane representation via robust plane fitting

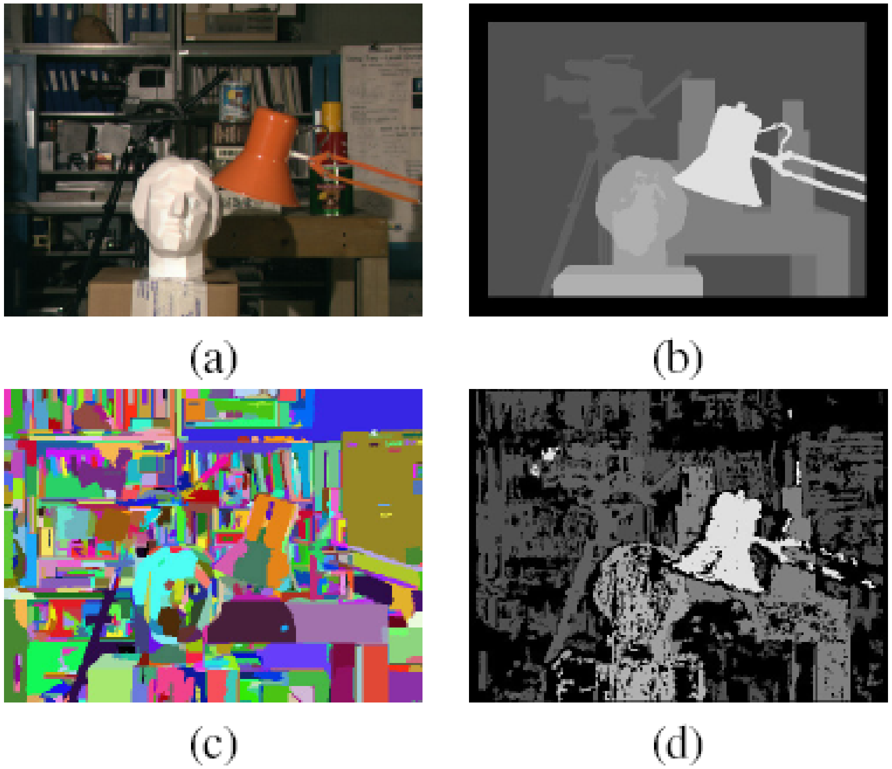
layer extraction

create plane models for extracted layers via robust plane fitting

layer assignment

final disparity map

Results: Initial (Optional)



(a)

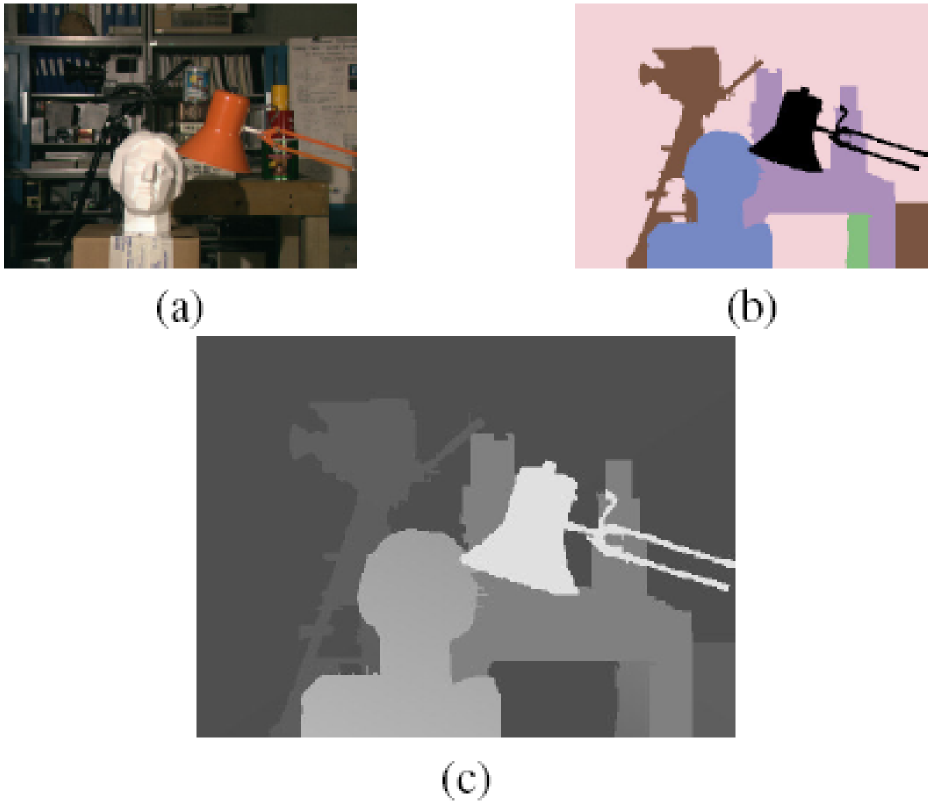
(b)

(c)

(d)

Fig. 1. Colour segmentation and initial disparity map. (a) Left image. (b) Ground truth provided with image pair. (c) Computed colour segmentation. (d) Computed initial disparity map. Invalid points are coloured black.

Results: Final (Optional)


(a)

(b)

(c)

Fig. 5. Results for the Tsukuba dataset. (a) Left image. (b) Final layer assignment. (c) Computed disparity map.

State-of-the-Art Methods

- Middelebury University maintains a website with standard datasets and results of submitted algorithms: updated regularly

  http://vision.middlebury.edu/stereo/eval3
- Top-ranking algorithm based on alpha-expansions
- Second one starts with feature matches and then densifies surface
- Several neural network algorithms ("CNN" in their name)
- Several algorithms that are not published in traditional CV literature but have high performance
- In general, state-of-the-art (SOTA) methods are quite complex
  - Out of scope of CS677

Structure from Motion: Intro

- Standard stereo methods assume that cameras are calibrated (intrinsic and extrinsic parameters are known)
  - 3-D can be recovered if correspondence problem is solved
- Requiring a calibrated stereo "rig" is not always feasible. Can we recover 3-D scene structure from multiple images taken from different view points but without knowledge of the camera positions or internal parameters?
  - A side effect would be to also recover camera parameters.
  - Such images are common:
    - a tourist taking two pictures by moving a short distance,
    - two or more images of famous architectural sites on the Internet; images may be taken by different photographers at different times
    - Robot Navigation (SLAM)
- This is called the "Structure from Motion" (SFM) problem

Is SFM Feasible?

- Given 3-D models of the environment, and matching points (between 3-D scene points and 2-D image points), we can estimate camera parameters: camera calibration
- Given matching points, in two or more 2-D images, and camera matrices, we can get 3-D of the environment (stereo problems)
- Given only the matching points in 2-D images, is the problem of inferring both camera parameters and scene properties really solvable?
  - This is also the SLAM problem in robotics
- Geometric analysis
- Algebraic Analysis

Geometric Analysis

- Image points in each camera define a bundle of rays
- We need to orient and translate cameras so that all the rays intersect
  - If all the matching points came from the same 3-D points, such intersections should exist (near intersections to account for noise)
  - How many point matches are neede; one is clearly not enough
  - Solutions may not be unique; at least scale will be unknown
- It is easier to examine ambiguities and minimal match requirement by algebraic analysis

Should SFM be possible?

- Consider *m* perspective cameras, *n* fixed points, say P_j, j=1,…,*n*
- In the i^{th} camera, image of P_j is $p_{ij}= (x_{ij}, y_{ij},1)^T$, i= 1,…,*m*
- We can write

$$\boldsymbol{p}_{ij} = \frac{1}{Z_{ij}}\begin{pmatrix}\mathcal{R}_i & \boldsymbol{t}_i\end{pmatrix}\begin{pmatrix}\boldsymbol{P}_j \\ 1\end{pmatrix},$$

- We want to recover
  - 3-D positions of the n points (3n parameters)
  - *m* camera matrices (*m**11 unknowns)
  - Total of 3*n* + 11m unknowns (ignoring inherent ambiguities)
  - Each image point gives 2 equations, so we have 2*mn* equations
  - Can solve if $2mn > 3n+11m$ (ignoring ambiguities)
  - Say $m=2$, $n \geq 22$; $m = 3$, $n \geq 11$
- If intrinsic parameters are known, only 6*m* camera unknowns
  - *m*=2, *n*= 12

Inherent Ambiguities

- Rigid Transformation
  - We can not fix object to a world coordinate frame; we can multiply all answers by an arbitrary rigid transformation and its inverse (rotation *R* and translation **t**) and get same image points back

$$p_{ij} = \frac{1}{Z_{ij}} \left(\begin{pmatrix} \mathcal{R}_i & t_i \end{pmatrix} \begin{pmatrix} \mathcal{R} & t \\ 0^T & 1 \end{pmatrix} \right) \left(\begin{pmatrix} \mathcal{R}^T & -\mathcal{R}^T t \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} P_j \\ 1 \end{pmatrix} \right) = \frac{1}{Z_{ij}} \begin{pmatrix} \mathcal{R}'_i & t'_i \end{pmatrix} \begin{pmatrix} P'_j \\ 1 \end{pmatrix},$$

$$\mathcal{R}'_i = \mathcal{R}_i \mathcal{R},\ t'_i = \mathcal{R}_i t + t_i, \text{ and } P'_j = \mathcal{R}^T (P_j - t).$$

- Scale also can not be recovered
  - Scale objects by constant, say λ; move cameras also by distance λ

$$p_{ij} = \frac{1}{\lambda Z_{ij}} \begin{pmatrix} \mathcal{R}_i & \lambda t_i \end{pmatrix} \begin{pmatrix} \lambda P_j \\ 1 \end{pmatrix} = \frac{1}{Z'_{ij}} \begin{pmatrix} \mathcal{R}_i & t'_i \end{pmatrix} \begin{pmatrix} P'_j \\ 1 \end{pmatrix}$$

- Fewer matches required for solution, known intrinsic parameters: $2nm \geq 6m + 3n - 7$; for m=2, n = 5
- If intrinsic parameters are unknown, ambiguities are deeper
  - Recovered shapes may be distorted; more details to follow later.

Non-linear Optimization

- If more than minimum number of point matches is available, solution may be found by optimizing:

$$E = \frac{1}{mn} \sum_{i,j} ||\boldsymbol{p}_{ij} - \frac{1}{Z_{ij}} (\mathcal{R}_i \quad \boldsymbol{t}_i) \begin{pmatrix} \boldsymbol{P}_j \\ 1 \end{pmatrix} ||^2$$

- This is non-linear optimization
- Good initial guess will help find a global minimum
- Much of what we will study is oriented towards finding good initial guesses by using linear optimization methods first

Two Camera Case: Outline of Method

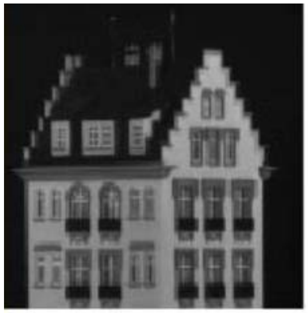
- Assume calibrated cameras, so known K and K'
- If we can estimate [R, **t**] and [R′, **t**′] problem reduces to stereo depth estimation: match points and triangulate
- We can align world coordinates with first camera so R=Id, **t** = 0
- How to estimate [R′, **t**′]?
- IF we can match points in two views, say, $\boldsymbol{p} = (u, v, 1)^T$ and $\boldsymbol{p}' = (u', v', 1)^T$ in the *normalized* camera coordinates, we know that they must be related by $\boldsymbol{p}^T \boldsymbol{\varepsilon}\, \boldsymbol{p}' = 0$; also that $\boldsymbol{\varepsilon} = [\mathbf{t}'_x]R'$
- We are not given $\boldsymbol{\varepsilon}$, **t**‘ or R′
- We can estimate $\boldsymbol{\varepsilon}$ from set of matches, similar to case for homography matrix
  - Linear solution requires 8 points (details later)
  - A popular non-linear solution needs only 5 points (but solves a 10th degree polynomial); we skip details of this method

Finding Matching Points

- Find sets of matching point pairs, by correlation or feature matching (but without benefit of epipolar constraint since $\boldsymbol{\varepsilon}$ is unknown at this time)
  - Use discriminative features, such as SIFT
  - Allow multiple matches for a keypoint
- Use RANSAC to choose consistent matches
  - Similar process as for homography but need 8 matches for the linear solution
- Once essential matrix has been computed, it can be decomposed into rotation and translation components
  - We will study this decomposition in combination with the uncalibrated camera case

Reconstruction

- Given matching points and camera matrices, 3-d positions of these points can be computed by triangulation, as in stereo.
- Euclidean reconstruction for internally calibrated cameras
  - 7 parameter ambiguities remain (global rotation, global translation and scale)

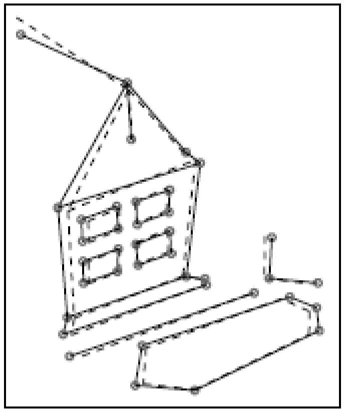


1

60

120

150



Result using only two images

Fundamental Matrix

- Essential matrix equation applies when cameras are calibrated
- Fortunately, a similar condition holds even without knowledge of the intrinsic parameters
- Let $p = \mathcal{K}\hat{p}$ and $p' = \mathcal{K}'\hat{p}'$; *p* and *p*' are the image coordinates; $\hat{p}$ and $\hat{p}'$ are the normalized coordinates, *K* and *K'* are the intrinsic matrices
- Substitute in essential matrix equation $\hat{p}^{\mathrm{T}}\boldsymbol{\varepsilon}\,\hat{p}' = 0$,we get: $\boldsymbol{p}^{\mathrm{T}}\boldsymbol{F}\boldsymbol{p}' = 0$;

  where $F = K^{-\mathrm{T}}\,\boldsymbol{\varepsilon}\,K'^{-1}$; ***F*** is called the *fundamental* matrix.

$$(u, v, 1)\begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0$$

- ***F*** is also of rank 2 (since $\boldsymbol{\varepsilon}$ is of rank 2) but the two eigenvalues are now not necessarily equal. Only 7 independent parameters even though it has 9 elements

Computing the Fundamental Matrix

- Linear solution can be found from 8 matching points (we can set F_{33} or any other coefficient to 1 since the equations are homogeneous and can be scaled)
- Results in following set of equations

$$\begin{pmatrix} u_1u'_1 & u_1v'_1 & u_1 & v_1u'_1 & v_1v'_1 & v_1 & u'_1 & v'_1 \\ u_2u'_2 & u_2v'_2 & u_2 & v_2u'_2 & v_2v'_2 & v_2 & u'_2 & v'_2 \\ u_3u'_3 & u_3v'_3 & u_3 & v_3u'_3 & v_3v'_3 & v_3 & u'_3 & v'_3 \\ u_4u'_4 & u_4v'_4 & u_4 & v_4u'_4 & v_4v'_4 & v_4 & u'_4 & v'_4 \\ u_5u'_5 & u_5v'_5 & u_5 & v_5u'_5 & v_5v'_5 & v_5 & u'_5 & v'_5 \\ u_6u'_6 & u_6v'_6 & u_6 & v_6u'_6 & v_6v'_6 & v_6 & u'_6 & v'_6 \\ u_7u'_7 & u_7v'_7 & u_7 & v_7u'_7 & v_7v'_7 & v_7 & u'_7 & v'_7 \\ u_8u'_8 & u_8v'_8 & u_8 & v_8u'_8 & v_8v'_8 & v_8 & u'_8 & v'_8 \end{pmatrix} \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \end{pmatrix} = - \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

- If more points are available, we can compute linear least squares solution, i.e. minimize: $\sum_{i=1}^{n} (\boldsymbol{p}_i^T \mathcal{F} \boldsymbol{p}'_i)^2$
- Note: 7 points suffice for non-linear solution, only 5 points needed if intrinsics known (i.e. to compute the essential matrix)

Numerical Issues and Refinement

- Standard linear least square minimization has been found to be unstable due to poor numerical conditioning. Changing origin and rescaling seem to help.
- One alternative is to minimize the geometric distance between the image points and the corresponding epipolar lines
  - This method gives more accurate results but requires non-linear optimization
  - Linear method could be used to initialize the non-linear method
- RANSAC to find good set of matching points
- Refine solution by imposing constraint that the matrix should be of rank 2
  - Method based on computing SVD (singular value decomposition)
  - Brief tutorial on SVD follows

Eigen-Decomposition

- Given an n x n square matrix, say A, it can be decomposed as:

  $\mathbf{A} = \mathbf{U}\,\mathbf{W}\,\mathbf{U}^{-1}$,

  where **W** is a diagonal matrix of eigenvalues along the diagonal;

  columns of **U** are the eigenvectors (in order of eigenvalues in **W**)
- Can be used to invert **A**: $\mathbf{A}^{-1} = \mathbf{U}\,\mathbf{W}^{-1}\,\mathbf{U}^{-1}$
- Can also be used to solve equations such as $A\boldsymbol{x} = \boldsymbol{b}$
- When **A** is not square (consider over determined set of linear equations), we can use singular valued decomposition (and pseudo-inverse of **A**).

Next Class

- Structure from Motion: Sections 8.1.2,8.2, 8.3