Computer Vision – Lecture 15

Epipolar Geometry & Stereo Basics

01.07.2019

Bastian Leibe
Visual Computing Institute
RWTH Aachen University
http://www.vision.rwth-aachen.de/

leibe@vision.rwth-aachen.de

Course Outline

- Image Processing Basics
- Segmentation & Grouping
- Object Recognition & Categorization
- Local Features & Matching
- Deep Learning
- 3D Reconstruction
  - Epipolar Geometry and Stereo Basics
  - Camera calibration & Uncalibrated Reconstruction
  - Multi-view Stereo

Topics of This Lecture

- Geometric vision
  - Visual cues
  - Stereo vision
- Epipolar geometry
  - Depth with stereo
  - Geometry for a simple stereo system
  - Case example with parallel optical axes
  - General case with calibrated cameras
- Stereopsis & 3D Reconstruction
  - Correspondence search
  - Additional correspondence constraints
  - Possible sources of error
  - Applications

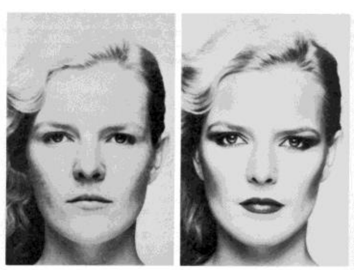
Geometric vision

- Goal: Recovery of 3D structure
  - What cues in the image allow us to do this?

Slide credit: Svetlana Lazebnik

Visual Cues

- Shading



Merle Norman Cosmetics, Los Angeles

Slide credit: Steve Seitz

Visual Cues

- Shading
- Texture

The Visual Cliff, by William Vandivert, 1960

Slide credit: Steve Seitz

Visual Cues

- Shading
- Texture
- Focus

From *The Art of Photography*, Canon

Visual Cues

- Shading
- Texture
- Focus
- Perspective

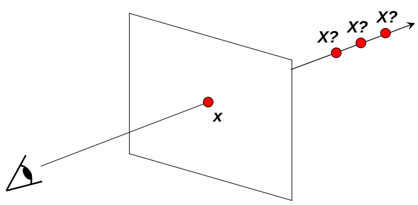
Visual Cues

- Shading
- Texture
- Focus
- Perspective
- Motion

Figures from L. Zhang

Our Goal: Recovery of 3D Structure

- We will focus on perspective and motion
- We need *multi-view geometry* because recovery of structure from one image is inherently ambiguous



To Illustrate This Point…

- Structure and depth are inherently ambiguous from single views.

Stereo Vision

http://www.well.com/~jimg/stereo/stereo_list.html

What Is Stereo Vision?

- Generic problem formulation: given several images of the same object or scene, compute a representation of its 3D shape

Computer Vision Summer'18

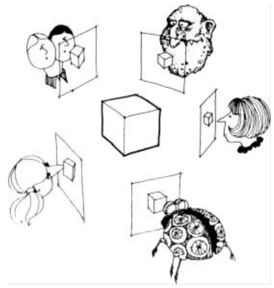
Slide credit: Svetlana Lazebnik, Steve Seitz

B. Leibe

13

What Is Stereo Vision?

- Generic problem formulation: given several images of the same object or scene, compute a representation of its 3D shape



Computer Vision Summer'18

Slide credit: Svetlana Lazebnik, Steve Seitz

B. Leibe

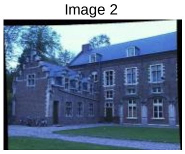
14

What Is Stereo Vision?

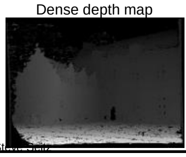
- Narrower formulation: given a calibrated binocular stereo pair, fuse it to produce a depth image

Image 1

Image 2



Dense depth map



Computer Vision Summer'18

Slide credit: Svetlana Lazebnik, Steve Seitz

15

What Is Stereo Vision?

- Narrower formulation: given a calibrated binocular stereo pair, fuse it to produce a depth image.
  - Humans can do it

Stereograms: Invented by Sir Charles Wheatstone, 1838

Computer Vision Summer'18

Slide credit: Svetlana Lazebnik, Steve Seitz

B. Leibe

16

What Is Stereo Vision?

- Narrower formulation: given a calibrated binocular stereo pair, fuse it to produce a depth image.
  - Humans can do it

Autostereograms: http://www.magiceye.com

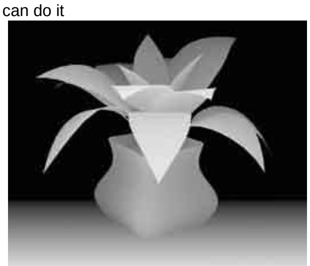
Computer Vision Summer'18

Slide credit: Svetlana Lazebnik, Steve Seitz

17

What Is Stereo Vision?

- Narrower formulation: given a calibrated binocular stereo pair, fuse it to produce a depth image.
  - Humans can do it



Autostereograms: http://www.magiceye.com

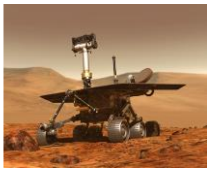
Computer Vision Summer'18

Slide credit: Svetlana Lazebnik, Steve Seitz

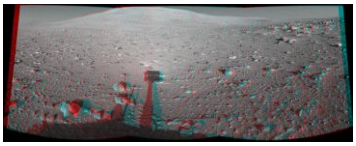
18

Application of Stereo: Robotic Exploration

Nomad robot searches for meteorites in Antartica



Real-time stereo on Mars

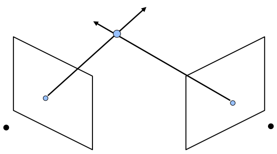


Slide credit: Steve Seitz

Topics of This Lecture

- Geometric vision
  - Visual cues
  - Stereo vision
- Epipolar geometry
  - Depth with stereo
  - Geometry for a simple stereo system
  - Case example with parallel optical axes
  - General case with calibrated cameras
- Stereopsis & 3D Reconstruction
  - Correspondence search
  - Additional correspondence constraints
  - Possible sources of error
  - Applications

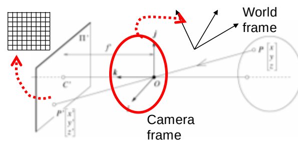
Depth with Stereo: Basic Idea



- Basic Principle: Triangulation
  - Gives reconstruction as intersection of two rays
  - Requires
    - Camera pose (calibration)
    - Point correspondence

Slide credit: Steve Seitz

Camera Calibration



Extrinsic parameters:
Camera frame ↔ Reference frame

Intrinsic parameters:
Image coordinates relative to camera ↔ Pixel coordinates

- Parameters
  - *Extrinsic*: rotation matrix and translation vector
  - *Intrinsic*: focal length, pixel sizes (mm), image center point, radial distortion parameters

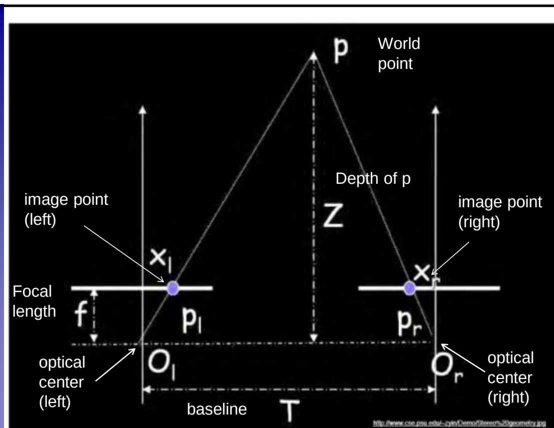
We'll assume for now that these parameters are given and fixed.

Slide credit: Kristen Grauman

Geometry for a Simple Stereo System

- First, assuming parallel optical axes, known camera parameters (i.e., calibrated cameras):

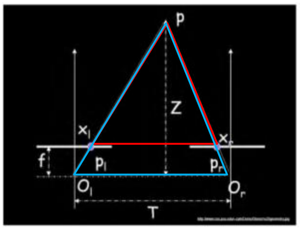
Slide credit: Kristen Grauman



Slide credit: Kristen Grauman

Geometry for a Simple Stereo System

- Assume parallel optical axes, known camera parameters (i.e., calibrated cameras). We can triangulate via:



Similar triangles (p_l, P, p_r) and (O_l, P, O_r):

$$\frac{T - (x_r - x_l)}{Z - f} = \frac{T}{Z}$$

$$Z = f \frac{T}{x_r - x_l}$$

"disparity"

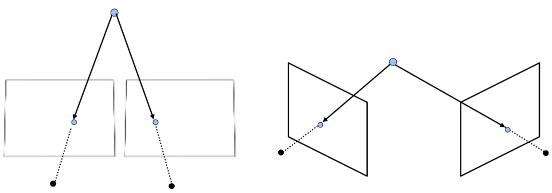
Depth From Disparity

Image $I(x,y)$ Disparity map $D(x,y)$ Image $I'(x',y')$

$$(x', y') = (x + D(x,y), y)$$

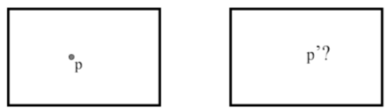
General Case With Calibrated Cameras

- The two cameras need not have parallel optical axes.



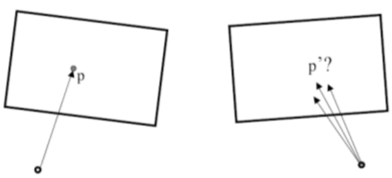
vs.

Stereo Correspondence Constraints



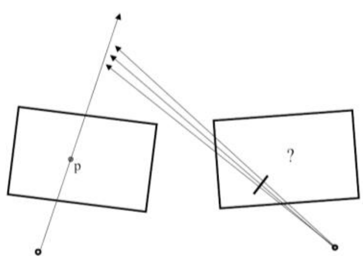
- Given p in the left image, where can the corresponding point p' in the right image be?

Stereo Correspondence Constraints



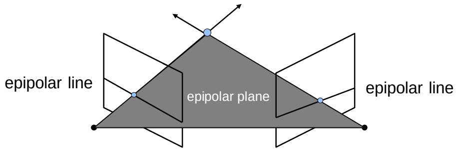
- Given p in the left image, where can the corresponding point p' in the right image be?

Stereo Correspondence Constraints



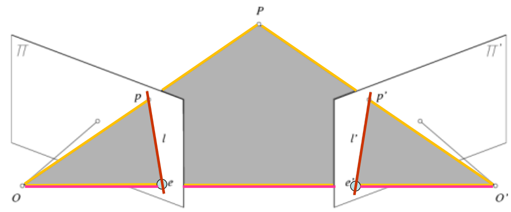
Stereo Correspondence Constraints

- Geometry of two views allows us to constrain where the corresponding pixel for some image point in the first view must occur in the second view.



- Epipolar constraint: Why is this useful?
  - Reduces correspondence problem to 1D search along conjugate epipolar lines.

Epipolar Geometry

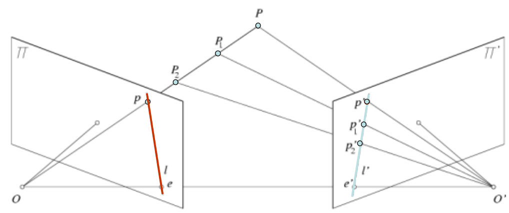


- Epipolar Plane
- Baseline
- Epipoles
- Epipolar Lines

Epipolar Geometry: Terms

- *Baseline*
  - Line joining the camera centers
- *Epipole*
  - Point of intersection of baseline with the image plane
- *Epipolar plane*
  - Plane containing baseline and world point
- *Epipolar line*
  - Intersection of epipolar plane with the image plane
- Properties
  - All epipolar lines intersect at the epipole.
  - An epipolar plane intersects the left and right image planes in epipolar lines.

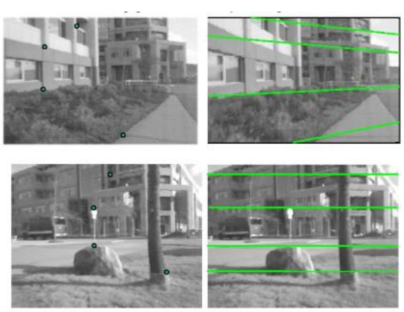
Epipolar Constraint



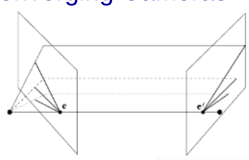
- Potential matches for p have to lie on the corresponding epipolar line l'.
- Potential matches for p' have to lie on the corresponding epipolar line l.

http://www.ai.sri.com/~luong/research/Meta3DViewer/EpipolarGeo.html

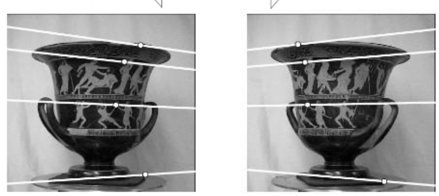
Example



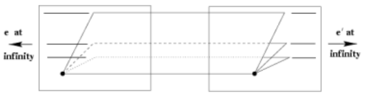
Example: Converging Cameras

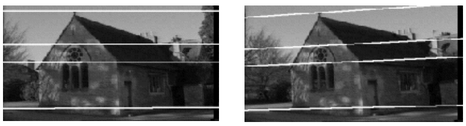


As position of 3D point varies, epipolar lines "rotate" about the baseline

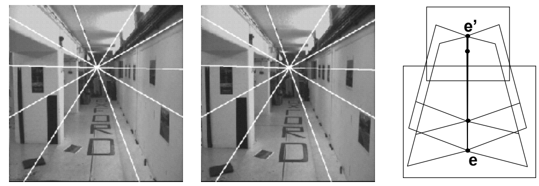


Example: Motion Parallel With Image Plane





Example: Forward Motion

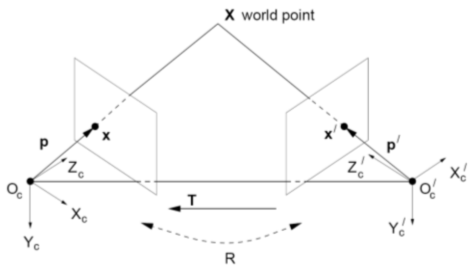


- Epipole has same coordinates in both images.
- Points move along lines radiating from e: "Focus of expansion"

Let's Formalize This!

- For a given stereo rig, how do we express the epipolar constraints algebraically?
- For this, we will need some linear algebra.
- But don't worry! We'll go through it step by step…

Stereo Geometry With Calibrated Cameras



- If the rig is calibrated, we know:
  - How to rotate and translate camera reference frame 1 to get to camera reference frame 2.
    - Rotation: 3 x 3 matrix; translation: 3 vector.

Rotation Matrix

$$\mathbf{R}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

$$\mathbf{R}_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

$$\mathbf{R}_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Express 3D rotation as series of rotations around coordinate axes by angles α, β, γ

Overall rotation is product of these elementary rotations:

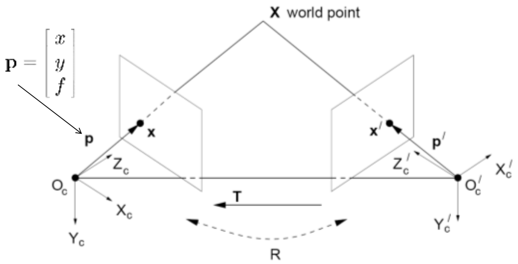
$$\mathbf{R} = \mathbf{R}_x \mathbf{R}_y \mathbf{R}_z$$

3D Rigid Transformation

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$$

$$\mathbf{X}' = \mathbf{R}\mathbf{X} + \mathbf{T}$$

Stereo Geometry With Calibrated Cameras



- Camera-centered coordinate systems are related by known rotation R and translation T:

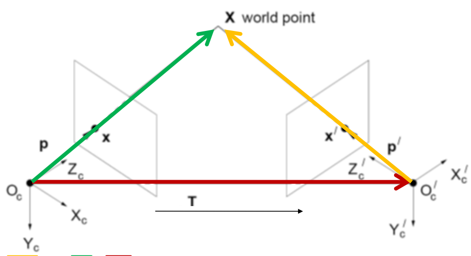
$$\mathbf{X}' = \mathbf{R}\mathbf{X} + \mathbf{T}$$

Excursion: Cross Product

$$\vec{a} \times \vec{b} = \vec{c} \qquad \begin{array}{rcl} \vec{a} \cdot \vec{c} &=& 0 \\ \vec{b} \cdot \vec{c} &=& 0 \end{array}$$

- Vector cross product takes two vectors and returns a third vector that's perpendicular to both inputs.
- So here, c is perpendicular to both a and b, which means the dot product is 0.

From Geometry to Algebra



$$\boxed{\mathbf{X}'} = \mathbf{R}\boxed{\mathbf{X}} + \boxed{\mathbf{T}}$$

$$\underbrace{\mathbf{T} \times \mathbf{X}'}_{\text{Normal to the plane}} = \mathbf{T} \times \mathbf{R}\mathbf{X} + \mathbf{T} \times \mathbf{T}$$

$$= \mathbf{T} \times \mathbf{R}\mathbf{X}$$

$$\mathbf{X}' \cdot (\mathbf{T} \times \mathbf{X}') = \mathbf{X}' \cdot (\mathbf{T} \times \mathbf{R}\mathbf{X})$$

$$0 = \mathbf{X}' \cdot (\mathbf{T} \times \mathbf{R}\mathbf{X})$$

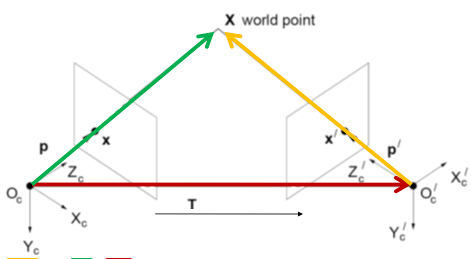
Matrix Form of Cross Product

$$\vec{a} \times \vec{b} \qquad\qquad = \vec{c} \qquad \begin{array}{rcl} \vec{a} \cdot \vec{c} &=& 0 \\ \vec{b} \cdot \vec{c} &=& 0 \end{array}$$

"skew symmetric" matrix

$$[a_\times] = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \qquad \boxed{\vec{a} \times \vec{b} = [a_\times]\,\vec{b}}$$

From Geometry to Algebra



$$\boxed{\mathbf{X}'} = \mathbf{R}\boxed{\mathbf{X}} + \boxed{\mathbf{T}}$$

$$\underbrace{\mathbf{T} \times \mathbf{X}'}_{\text{Normal to the plane}} = \mathbf{T} \times \mathbf{R}\mathbf{X} + \mathbf{T} \times \mathbf{T}$$

$$= \mathbf{T} \times \mathbf{R}\mathbf{X}$$

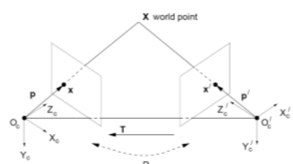
$$\mathbf{X}' \cdot (\mathbf{T} \times \mathbf{X}') = \mathbf{X}' \cdot (\mathbf{T} \times \mathbf{R}\mathbf{X})$$

$$\boxed{0 = \mathbf{X}' \cdot (\mathbf{T} \times \mathbf{R}\mathbf{X})}$$

Essential Matrix

$$\boxed{\mathbf{X}' \cdot (\mathbf{T} \times \mathbf{R}\mathbf{X}) = 0}$$

$$\mathbf{X}' \cdot (\mathbf{T}_x\ \mathbf{R}\mathbf{X}) = 0$$



Let $\mathbf{E} = \mathbf{T}_x\mathbf{R}$

$$\mathbf{X}'^T \mathbf{E}\mathbf{X} = 0$$

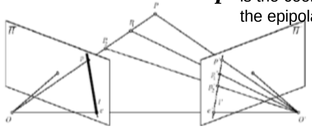
- This holds for the rays p and p' that are parallel to the camera-centered position vectors X and X', so we have: $\boxed{\mathbf{p}'^T\ \mathbf{E}\mathbf{p} = 0}$
- E is called the essential matrix, which relates corresponding image points [Longuet-Higgins 1981]

Essential Matrix and Epipolar Lines

$\boxed{\mathbf{p}'^{\mathrm{T}}\mathbf{E}\mathbf{p} = 0}$ Epipolar constraint: if we observe point p in one image, then its position p' in second image must satisfy this equation.

$l' = Ep$ is the coordinate vector representing the epipolar line for point p

(i.e., the line is given by: $l'^{\top}x = 0$)



$l = E^T p'$ is the coordinate vector representing the epipolar line for point p'

Computer Vision Summer'18

Slide credit: Kristen Grauman B. Leibe 49

Essential Matrix: Properties

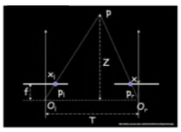
- Relates image of corresponding points in both cameras, given rotation and translation.
- Assuming intrinsic parameters are known

$$\mathbf{E} = \mathbf{T}_x\mathbf{R}$$

Computer Vision Summer'18

Slide credit: Kristen Grauman B. Leibe 50

Essential Matrix Example: Parallel Cameras



$$\mathbf{R} = \mathbf{I}$$

$$\mathbf{T} = [-d, 0, 0]^{\mathrm{T}}$$

$$\mathbf{E} = [\mathbf{T}_x]\mathbf{R} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{bmatrix}$$

$$\mathbf{p}'^{\mathrm{T}}\mathbf{E}\mathbf{p} = 0 \qquad \begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ f \end{bmatrix} = 0$$

$$\Leftrightarrow \begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 \\ df \\ -dy \end{bmatrix} = 0$$

$$\boxed{\Leftrightarrow y = y'}$$

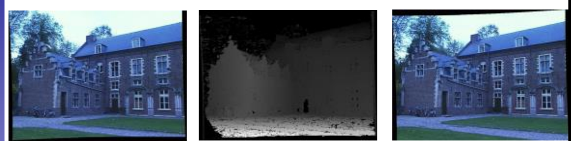
For the parallel cameras, image of any point must lie on same horizontal line in each image plane.

Computer Vision Summer'18

51 Slide credit: Kristen Grauman

More General Case

Image $I(x, y)$ Disparity map $D(x, y)$ Image $I'(x', y')$



$$(x', y') = (x + D(x, y), y)$$

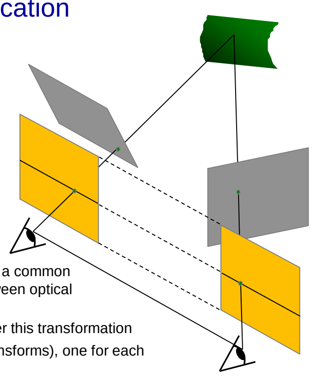
What about when cameras' optical axes are not parallel?

Computer Vision Summer'18

Slide credit: Kristen Grauman B. Leibe 52

Stereo Image Rectification

- In practice, it is convenient if image scanlines are the epipolar lines.

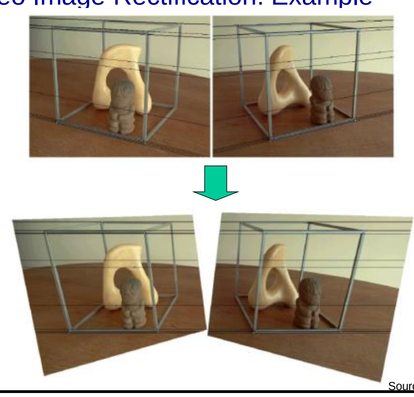


- Algorithm
  - Reproject image planes onto a common plane parallel to the line between optical centers
  - Pixel motion is horizontal after this transformation
  - Two homographies (3×3 transforms), one for each input image reprojection

Computer Vision Summer'18

Slide adapted from Li Zhang C. Loop & Z. Zhang, Computing Rectifying Homographies for Stereo Vision, CVPR'99 53

Stereo Image Rectification: Example



Computer Vision Summer'18

54 Source: Alyosha Efros

Topics of This Lecture

- Geometric vision
  - Visual cues
  - Stereo vision
- Epipolar geometry
  - Depth with stereo
  - Geometry for a simple stereo system
  - Case example with parallel optical axes
  - General case with calibrated cameras
- **Stereopsis & 3D Reconstruction**
  - Correspondence search
  - Additional correspondence constraints
  - Possible sources of error
  - Applications

Computer Vision Summer'18

B. Leibe

55

Stereo Reconstruction

- Main Steps
  - Calibrate cameras
  - Rectify images
  - Compute disparity
  - Estimate depth

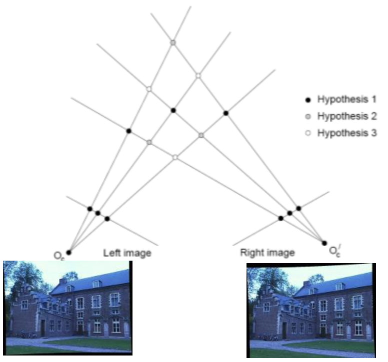
Computer Vision Summer'18

Slide credit: Kristen Grauman

B. Leibe

56

Correspondence Problem



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

Computer Vision Summer'18

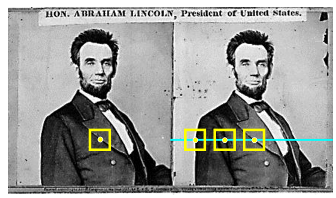
Slide credit: Kristen Grauman

B. Leibe

Figure from Gee & Cipolla 1999

57

Dense Correspondence Search



- For each pixel in the first image
  - Find corresponding epipolar line in the right image
  - Examine all pixels on the epipolar line and pick the best match (e.g. SSD, correlation)
  - Triangulate the matches to get depth information
- This is easiest when epipolar lines are scanlines
  - $\Rightarrow$ Rectify images first

Computer Vision Summer'18

adapted from Svetlana Lazebnik, Li Zhang

59

Example: Window Search

- Data from University of Tsukuba

Scene

Ground truth

Computer Vision Summer'18

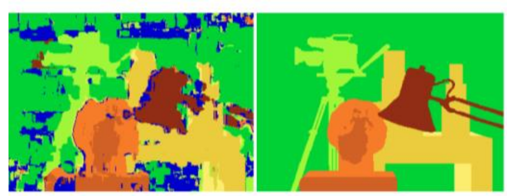
Slide credit: Kristen Grauman

B. Leibe

60

Example: Window Search

- Data from University of Tsukuba



Window-based matching (best window size)

Ground truth

Computer Vision Summer'18

Slide credit: Kristen Grauman

B. Leibe

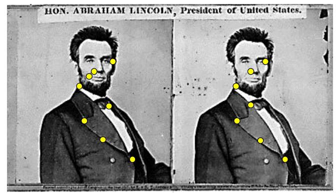
61

Effect of Window Size

$W = 3$ $W = 20$

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

Alternative: Sparse Correspondence Search



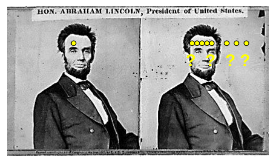
- Idea: Restrict search to sparse set of detected features
- Rather than pixel values (or lists of pixel values) use *feature descriptor* and an associated *feature distance*
- Still narrow search further by epipolar geometry

What would make good features?

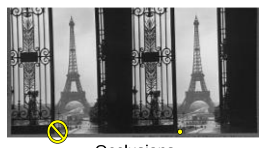
Dense vs. Sparse

- Sparse
  - Efficiency
  - Can have more reliable feature matches, less sensitive to illumination than raw pixels
  - But…
    - Have to know enough to pick good features
    - Sparse information
- Dense
  - Simple process
  - More depth estimates, can be useful for surface reconstruction
  - But…
    - Breaks down in textureless regions anyway
    - Raw pixel distances can be brittle
    - Not good with very different viewpoints

Difficulties in Similarity Constraint



Untextured surfaces



Occlusions

Possible Sources of Error?

- Low-contrast / textureless image regions
- Occlusions
- Camera calibration errors
- Violations of *brightness constancy* (e.g., specular reflections)
- Large motions

Application: View Interpolation

Right Image

Application: View Interpolation

Left Image

Slide credit: Svetlana Lazebnik

Application: View Interpolation

Disparity

Slide credit: Svetlana Lazebnik

Application: View Interpolation

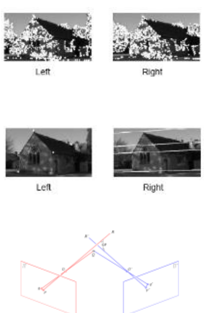
Slide credit: Svetlana Lazebnik

Application: Free-Viewpoint Video

http://www.liberovision.com

Summary: Stereo Reconstruction

- Main Steps
  - Calibrate cameras
  - Rectify images
  - Compute disparity
  - Estimate depth
- So far, we have only considered calibrated cameras…
- Next lecture
  - Uncalibrated cameras
  - Camera parameters
  - Revisiting epipolar geometry
  - Robust fitting

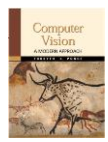


Slide credit: Kristen Grauman

References and Further Reading

- Background information on epipolar geometry and stereopsis can be found in Chapters 10.1-10.2 and 11.1-11.3 of

  D. Forsyth, J. Ponce,
  Computer Vision – A Modern Approach.
  Prentice Hall, 2003



- More detailed information (if you really want to implement 3D reconstruction algorithms) can be found in Chapters 9 and 10 of

  R. Hartley, A. Zisserman
  Multiple View Geometry in Computer Vision
  2nd Ed., Cambridge Univ. Press, 2004