Stereo

Outline:

- parallel camera axes
- convergent axes, epipolar geometry
- correspondence problem
- algorithms for stereo matching
Credits: major sources of material, including figures and slides were:

- Mallot, H.-P., Computational Vision
- Slides: Octavia Camps, Frank Dellaert, Davi Geiger, David Jacobs, Jim Rehg, Steve Seitz, Zhigang Zhu
- and various resources on the WWW
Overview

Triangulate on two images of the same point to recover depth.

- Feature matching across views
- Calibrated cameras

Matching correlation windows across scan lines
Pinhole Camera Model

Image plane | Focal length $f$

Center of projection | Virtual Image
Pinhole Camera Model

$P' = (X', Y', Z')$

$Z' = -f, \quad X' = -f \frac{X}{Z}, \quad Y' = -f \frac{Y}{Z}$

$x = -X', \quad y = -Y'$

$(X, Y, Z) \rightarrow (x, y, l) = (f \frac{X}{Z}, f \frac{Y}{Z}, 1)$
Basic Stereo Derivations

Derive expression for $Z$ as a function of $x_1, x_2, f, B$
Basic Stereo Derivations

\[ P_1 = (X, Y, Z) \]

\[ x_1 = -f \frac{X_1}{Z_1}, \quad x_2 = -f \frac{X_1 + B}{Z_1} = x_1 - f \frac{B}{Z_1} \]

\[ \Rightarrow Z_1 = \frac{fB}{x_1 - x_2} = \frac{fB}{d} \quad \text{disparity} \]
Stereo with Converging Cameras

- Stereo with Parallel Axes
  - Short baseline
    - large common FOV
    - large depth error
  - Long baseline
    - small depth error
    - small common FOV
    - More occlusion problems

- Two optical axes intersect at the Fixation Point
  - converging angle $\theta$
  - The common FOV Increases
Stereo with Converging Cameras

- Stereo with Parallel Axes
  - Short baseline
    - large common FOV
    - large depth error
  - Long baseline
    - small depth error
    - small common FOV
    - More occlusion problems

- Two optical axes intersect at the Fixation Point
  - converging angle $\theta$
  - The common FOV Increases
Stereo with Converging Cameras

- Two optical axes intersect at the Fixation Point
  - converging angle \( \theta \) (vergence)
  - The common FOV Increases

- Disparity properties
  - Disparity uses angle instead of distance
  - Zero disparity at fixation point
    - and the Zero-disparity horopter
  - Disparity increases with the distance of objects from the fixation points
    - \( >0 \) : outside of the horopter
    - \( <0 \) : inside the horopter

- Depth Accuracy vs. Depth
  - Depth Error \( \propto \) Depth\(^2\)
  - Nearer the point, better the depth estimation
Stereo with Converging Cameras

- Two optical axes intersect at the Fixation Point
  - converging angle $\theta$
  - The common FOV Increases

- Disparity properties
  - Disparity uses angle instead of distance
  - Zero disparity at fixation point
    - and the Zero-disparity horopter
  - Disparity increases with the distance of objects from the fixation points
    - $>0$: outside of the horopter
    - $<0$: inside the horopter

- Depth Accuracy vs. Depth
  - Depth Error $\propto$ Depth$^2$
  - Nearer the point, better the depth estimation
Stereo with Converging Cameras

- Two optical axes intersect at the Fixation Point
  - converging angle $\theta$
  - The common FOV Increases

- Disparity properties
  - Disparity uses angle instead of distance
  - Zero disparity at fixation point
    - and the Zero-disparity horopter
  - Disparity increases with the distance of objects from the fixation points
    - $>0$: outside of the horopter
    - $<0$: inside the horopter

- Depth Accuracy vs. Depth
  - Depth Error $\propto$ Depth$^2$
  - Nearer the point, better the depth estimation
Stereo with Converging Cameras

- Two optical axes intersect at the Fixation Point
  - converging angle $\theta$
  - The common FOV Increases

- Disparity properties
  - Disparity uses angle instead of distance
  - Zero disparity at fixation point
    - and the Zero-disparity horopter
  - Disparity increases with the distance of objects from the fixation points
    - $>0$ : outside of the horopter
    - $<0$ : inside the horopter

- Depth Accuracy vs. Depth
  - Depth Error $\propto$ Depth$^2$
  - Nearer the point, better the depth estimation

\[ \alpha_r < \alpha_l \]
\[ d\alpha < 0 \]
Stereo with Converging Cameras

- Two optical axes intersect at the Fixation Point
  - converging angle $\theta$
  - The common FOV Increases

- Disparity properties
  - Disparity uses angle instead of distance
  - Zero disparity at fixation point
    - and the Zero-disparity horopter
  - Disparity increases with the distance of objects from the fixation points
    - $>0$: outside of the horopter
    - $<0$: inside the horopter

- Depth Accuracy vs. Depth
  - Depth Error $\propto$ Depth$^2$
  - Nearer the point, better the depth estimation
Vieth Mueller Circle

Retinal disparity utilized for stereopsis

Target hits corresponding retinal points

Target hits non-corresponding retinal points

\[ x = y \]

\[ x \neq y \]
Parameters of a Stereo System

- **Intrinsic Parameters**
  - Characterize the transformation from camera to pixel coordinate systems of each camera
  - Focal length, image center, aspect ratio

- **Extrinsic parameters**
  - Describe the relative position and orientation of the two cameras
  - Rotation matrix $R$ and translation vector $T$
Epipolar Geometry

- **Motivation:** where to search correspondences?
  - Epipolar Plane
    - A plane going through point \( P \) and the centers of projection (COPs) of the two cameras
  - Conjugated Epipolar Lines
    - Lines where epipolar plane intersects the image planes
  - Epipoles
    - The image in one camera of the COP of the other
  - Epipolar Constraint
    - Corresponding points must lie on conjugated epipolar lines
Fig. 9.1. **Point correspondence geometry.** (a) The two cameras are indicated by their centres C and C' and image planes. The camera centres, 3-space point X, and its images x and x' lie in a common plane π. (b) An image point x back-projects to a ray in 3-space defined by the first camera centre, C, and x. This ray is imaged as a line l' in the second view. The 3-space point X which projects to x must lie on this ray, so the image of X in the second view must lie on l'.
Epipolar geometry

For any two fixed cameras we have one baseline.

For any 3d point $X$ we have a different epipolar plane $\pi$.

All epipolar planes intersect at the baseline.

Fig. 9.2. **Epipolar geometry.** (a) The camera baseline intersects each image plane at the epipoles $e$ and $e'$. Any plane $\pi$ containing the baseline is an epipolar plane, and intersects the image planes in corresponding epipolar lines $l$ and $l'$. (b) As the position of the 3D point $X$ varies, the epipolar planes “rotate” about the baseline. This family of planes is known as an epipolar pencil. All epipolar lines intersect at the epipole.
Epipolar line

- Suppose we know only $x$ and the baseline.
- How is the corresponding point $x'$ in the other image constrained?
- $p_i$ is defined by the baseline and the ray $Cx$.
- The epipolar line $l'$ is the image of this ray in the other image. $x'$ must lie on $l'$.
- The benefit of the epipolar line is that the correspondence search can be restricted to $l'$ instead of searching the entire image.
The epipolar constraint

- Epipolar Constraint
  - Matching points lie along corresponding epipolar lines
  - Reduces correspondence problem to 1D search along conjugate epipolar lines
  - Greatly reduces cost and ambiguity of matching
Fig. 9.3. **Converging cameras.** (a) Epipolar geometry for converging cameras. (b) and (c) A pair of images with superimposed corresponding points and their epipolar lines (in white). The motion between the views is a translation and rotation. In each image, the direction of the other camera may be inferred from the intersection of the pencil of epipolar lines. In this case, both epipoles lie outside of the visible image.
Epipolar terminology

- **Epipole:**
  - intersection of the line joining the camera centers (baseline) and the image plane.
  - the image of the other camera center in the image plane.
  - intersection of the epipolar lines.
- **Epipolar plane:**
  - a plane containing the baseline. There is a one-parameter family of epipolar planes for a fixed camera pair.
- **Epipolar line:**
  - intersection of the epipolar plane with the image plane.
  - all epipolar lines intersect in the epipole.
  - an epipolar plane intersects both image planes and defines correspondences between the lines.
Reconstruction by Triangulation

- **Assumption and Problem**
  - Under the assumption that both intrinsic and extrinsic parameters are known
  - Compute the 3-D location from their projections, pl and pr
- **Solution**
  - **Triangulation:** Two rays are known and the intersection can be computed
  - **Problem:** Two rays will not actually intersect in space due to errors in calibration and correspondences, and sampling
  - **Solution:** find a point in space with minimum distance from both rays
Simplest Case from above

assume:
- Image planes of cameras are parallel
- Focal points are at same height
- Focal lengths same

then:
- epipolar lines fall along the horizontal scan lines of the images
Rectification

- **Motivation:**
  - create case from above since it:
    - simplifies algorithms
    - improves efficiency

- **Image Reprojection**
  - reproject image planes onto common plane parallel to line between optical centers
  - Notice, only focal point of camera really matters

(Seitz)
Correspondence

- It is fundamentally ambiguous, even with stereo constraints: many 3D points could have given rise to the observed features.
What should we match?

- Objects?
- Edges?
- Pixels?
- Collections of pixels?

Visual illusions:
- double nail illusion
- line illusion
Human Stereo: Random Dot Stereogram

Julesz’s Random Dot Stereogram. The left image, a black and white image, is generated by a program that assigns black or white values at each pixel according to a random number generator.

The right image is constructed from by copying the left image, but an imaginary square inside the left image is displaced a few pixels to the left and the empty space filled with black and white values chosen at random. When the stereo pair is shown, the observers can identify/match the imaginary square on both images and consequently “see” a square in front of the background. It shows that stereo matching can occur without recognition.
Human Stereo: Illusory Contours
Correspondence: Epipolar constraint.

The epipolar constraint helps, but much ambiguity remains.
Correspondence: Photometric constraint

- Same world point has same intensity in both images.
  - True for Lambertian fronto-parallel surfaces
  - Violations:
    - Noise
    - Specularity
    - Foreshortening
Correspondence: other constraints

- ordering constraint:
- uniqueness constraint:
  - every feature in one image has only one match in the other image
  - but: transparency
Pixel matching

For each epipolar line
  For each pixel in the left image
    • compare with every pixel on same epipolar line in right image
    • pick pixel with minimum match cost

This leaves too much ambiguity, so:

Improvement: match windows  

(Seitz)
Correspondence Using Correlation

Left

Right

SSD error

disparity

scanline
Sum of Squared (Pixel) Differences

$w_L$ and $w_R$ are corresponding $m$ by $m$ windows of pixels.

We define the window function:

$$W_m(x, y) = \{u, v \mid x - \frac{m}{2} \leq u \leq x + \frac{m}{2}, y - \frac{m}{2} \leq v \leq y + \frac{m}{2}\}$$

The SSD cost measures the intensity difference as a function of disparity:

$$C_r(x, y, d) = \sum_{(u,v) \in W_m(x, y)} [I_L(u, v) - I_R(u - d, v)]^2$$
Image Normalization

- Even when the cameras are identical models, there can be differences in gain and sensitivity.
- For these reason and more, it is a good idea to normalize the pixels in each window:

\[
\bar{I} = \frac{1}{|W_m(x,y)|} \sum_{(u,v) \in W_m(x,y)} I(u,v)
\]

Average pixel

\[
\|I\|_{W_m(x,y)} = \sqrt{\sum_{(u,v) \in W_m(x,y)} [I(u,v)]^2}
\]

Window magnitude

\[
\hat{I}(x, y) = \frac{I(x, y) - \bar{I}}{\|I - \bar{I}\|_{W_m(x,y)}}
\]

Normalized pixel
Images as Vectors

“Unwrap” image to form vector, using raster scan order

Each window is a vector in an $m^2$ dimensional vector space. Normalization makes them unit length.
(Normalized) Sum of Squared Differences

$$C_{SSD}(d) = \sum_{(u,v) \in W_m(x,y)} [(\hat{I}_L(u,v) - \hat{I}_R(u-d,v))^2]$$

$$= \|w_L - w_R(d)\|^2$$

Normalized Correlation

$$C_{NC}(d) = \sum_{(u,v) \in W_m(x,y)} \hat{I}_L(u,v)\hat{I}_R(u-d,v)$$

$$= w_L \cdot w_R(d) = \cos \theta$$

$$d^* = \arg \min_d \|w_L - w_R(d)\|^2 = \arg \max_d w_L \cdot w_R(d)$$
Stereo Results

Images courtesy of Point Grey Research
Window size

- Effect of window size
- Some approaches have been developed to use an adaptive window size (try multiple sizes and select best match)

(Seitz)
Stereo testing and comparisons

True disparities

19 – Belief propagation

11 – GC + occlusions

20 – Layered stereo

10 – Graph cuts

*4 – Graph cuts

13 – Genetic algorithm

6 – Max flow

12 – Compact windows

9 – Cooperative alg.

15 – Stochastic diffusion

*2 – Dynamic progr.

14 – Realtime SAD

*3 – Scanline opt.

7 – Pixel-to-pixel stereo

*1 – SSD+MF
Results with window correlation

Window-based matching (best window size) [Image]
Ground truth (Seitz) [Image]
Results with better method

State of the art method: Graph cuts

Ground truth

(Seitz)
Stereo Correspondences

Left scanline

Right scanline

***
Stereo Correspondences

Left scanline

Right scanline

Match

Occlusion

Disocclusion

Match
Search Over Correspondences

Three cases:
- **Sequential** - add cost of match (small if intensities agree)
- **Occluded** - add cost of no match (large cost)
- **Disoccluded** - add cost of no match (large cost)
Dynamic programming yields the optimal path through grid. This is the best set of matches that satisfy the ordering constraint.

Occluded Pixels

Dis-occluded Pixels

Start

Left scanline

Right scanline

End
Dynamic Programming

- Efficient algorithm for solving sequential decision (optimal path) problems.

How many paths through this trellis? $3^T$
Dynamic Programming

Suppose cost can be decomposed into stages:

\[ \Pi_{ij} = \text{Cost of going from state } i \text{ to state } j \]
Dynamic Programming

Principle of Optimality for an n-stage assignment problem:

$$C_t(j) = \min_i (\Pi_{ij} + C_{t-1}(i))$$
### Dynamic Programming

**Equations:**

- \( C_t(j) = \min_i (\Pi_{ij} + C_{t-1}(i)) \)
- \( b_t(j) = \arg \min_i (\Pi_{ij} + C_{t-1}(i)) \)

**Diagram:**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

- \( i = 1 \)
- \( i = 2 \)
- \( i = 3 \)

\( b_t(2) = 2 \)
Stereo Matching with Dynamic Programming

Scan across grid computing optimal cost for each node given its upper-left neighbors. Backtrack from the terminal to get the optimal path.
Stereo Matching with Dynamic Programming

Scan across grid computing optimal cost for each node given its upper-left neighbors. Backtrack from the terminal to get the optimal path.
Stereo Matching with Dynamic Programming

Scan across grid computing optimal cost for each node given its upper-left neighbors. Backtrack from the terminal to get the optimal path.
Correlation Approach

• PROS
  • Easy to implement
  • Produces dense disparity map
  • Maybe slow

• CONS
  • Needs textured images to work well
  • Inadequate for matching image pairs from very different viewpoints due to illumination changes
  • Window may cover points with quite different disparities
  • Inaccurate disparities on the occluding boundaries
Feature-based Approach

• Features
  • Edge points
  • Lines (length, orientation, average contrast)
  • Corners

• Matching algorithm
  • Extract features in the stereo pair
  • Define similarity measure
  • Search correspondences using similarity measure and the epipolar geometry
• Example: match *edges* rather than windows of pixels:

![Diagram showing comparison between edges and correlation]

• Which method is better?
  • Edges tend to fail in dense texture (outdoors)
  • Correlation tends to fail in smooth featureless areas
Feature-based Approach

• PROS
  • Relatively insensitive to illumination changes
  • Good for man-made scenes with strong lines but weak texture or textureless surfaces
  • Work well on the occluding boundaries (edges)
  • Could be faster than the correlation approach

• CONS
  • Only sparse depth map
  • Feature extraction may be tricky
    • Lines (Edges) might be partially extracted in one image
    • How to measure the similarity between two lines?
Computing Correspondences

• Both methods fail for smooth surfaces

• There is currently no good solution to the correspondence problem
Some More Advanced Topics

- Sub-pixel matching to improve accuracy
  - Find the peak in the correlation curves

- Self-consistency to reduce false matches esp. for half-occlusions
  - Check the consistency of matches from L to R and from R to L

- Multiple Resolution Approach
  - From coarse to fine for efficiency in searching correspondences

- Local warping to account for perspective distortion
  - Warp from one view to the other for a small patch given an initial estimation of the (planar) surface normal

- Multi-baseline Stereo
  - Improves both correspondences and 3D estimation by using more than two cameras (images)
Segmentation-based Stereo

Hai Tao and Harpreet W. Sawhney
Another Example
Further Questions and Remarks

- recall: vision is for action. Do you need a dense depth map at all? How exactly do people use stereo?
- many other depth cues available: cue integration
- segmentation helping with stereo, but also stereo helping with segmentation: integration
- disparity tuned neurons: how do you learn them and how do you learn to control vergence?
- why do some infants have problems learning stereo vision?
• *active vision* can be utilized for stereo vision:

  Example: *zero disparity filtering* is cheap and effective

  • once you fixate on something: find features (edges or corners), try to match them in small disparity range around zero, throw out everything else

  • to track: adjust vergence angle to bring “interesting features” back to zero disparity if they start to deviate
Next time: Marty

- physiology part
  - role of foveation in stereo
  - how are local disparities estimated by the brain
  - horizontal vs. vertical disparity
  - using occlusion
  - “higher level”