Disparity and stereo
Course announcements

• Homework assignment 6 posted.
  - Due on December 5.

• Office hours this week:
  - Monday, 1:00 – 3:00 pm, Yannis.
  - Tuesday, 3:00 – 5:00 pm, Alice.
  - Wednesday, 3:00 – 5:00 pm, Jenny.

• I will post details for optional final project checkpoint meetings the week after Thanksgiving.
Overview of today’s lecture

- Revisiting triangulation.
- Disparity.
- Revisiting lightfields.
- Structured light.
- Some notes on focusing.
Many of these slides were adapted directly from:

- Srinivasa Narasimhan (16-820, Spring 2017).
- Mohit Gupta (Wisconsin).
- James Tompkin (Brown).
Revisiting triangulation
How would you reconstruct 3D points?

Left image

Right image
How would you reconstruct 3D points?

1. Select point in one image
How would you reconstruct 3D points?

1. Select point in one image
2. Form epipolar line for that point in second image (how?)
How would you reconstruct 3D points?

1. Select point in one image
2. Form epipolar line for that point in second image (how?)
3. Find matching point along line (how?)
How would you reconstruct 3D points?

1. Select point in one image
2. Form epipolar line for that point in second image (how?)
3. Find matching point along line (how?)
4. Perform triangulation (how?)
Triangulation

3D point

left image

right image

left camera with matrix $P$

right camera with matrix $P'$
Stereo rectification
What’s different between these two images?
Objects that are close move more or less?
The amount of horizontal movement is inversely proportional to ...
The amount of horizontal movement is inversely proportional to...

... the distance from the camera.

More formally...
3D point $X$

$O$ and $O'$ are camera centers.

$f$ and $x$ are coordinates on the image plane.
Important: coordinates $x$ and $x'$ are parameterized with respect to image center.
How is $X$ related to $x$?
\[ \frac{X}{Z} = \frac{x}{f} \]
\[ \frac{X}{Z} = \frac{x}{f} \]

How is X related to x'?
\[
\frac{X}{Z} = \frac{x}{f} \\
\frac{b - X}{Z} = \frac{-x'}{f}
\]
Disparity

\[ d = x - x' \quad \text{(wrt to camera origin of image plane)} \]

\[ \frac{X}{Z} = \frac{x}{f} \]

\[ \frac{b - X}{Z} = \frac{-x'}{f} \]
\[
\frac{X}{Z} = \frac{x}{f}
\]

\[
\frac{b - X}{Z} = \frac{-x'}{f}
\]

Disparity

\[
d = x - x'
\]

inversely proportional to depth
Nomad robot searches for meteorites in Antarctica

http://www.frc.ri.cmu.edu/projects/meteorobot/index.html
Subaru Eyesight system

Pre-collision braking
What other vision system uses disparity for depth sensing?
Stereoscopes: A 19th Century Pastime
HON. ABRAHAM LINCOLN, President of United States.
Mark Twain at Pool Table", no date, UCR Museum of Photography
This is how 3D movies work
Simple stereoscope

Google cardboard

Fun patterns: random dot stereograms

1. Planar (Julesz, 1960)
2. Textured Planes
3. Textureless Middle Plane (Tsirlin et al, 2010)
4. Textureless Plane

http://vision.seas.harvard.edu/stereo/
So can I compute depth using disparity from any two images of the same object?
So can I compute depth using disparity from any two images of the same object?

1. Need sufficient baseline

2. Images need to be ‘rectified’ first (make epipolar lines horizontal)
How can you make the epipolar lines horizontal?
What’s special about these two cameras?
\[ x' = R(x - t) \]
When are epipolar lines horizontal?
When are epipolar lines horizontal?

When this relationship holds:

\[
R = I \quad t = (T, 0, 0)
\]
When are epipolar lines horizontal?

When this relationship holds:

\[ R = I \quad t = (T, 0, 0) \]

Let's try this out…

\[ E = t \times R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix} \]

This always has to hold:

\[ x^T E x' = 0 \]
When are epipolar lines horizontal?

When this relationship holds:

\[ R = I \quad t = (T, 0, 0) \]

Let's try this out…

\[
E = t \times R = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & -T \\
0 & T & 0
\end{bmatrix}
\]

This always has to hold:

\[ x^T E x' = 0 \]

Write out the constraint

\[
\begin{pmatrix}
0 & 0 & 0 \\
0 & 0 & -T \\
0 & T & 0
\end{pmatrix}
\begin{pmatrix}
u' \\
v' \\
1
\end{pmatrix} = 0
\]

\[
\begin{pmatrix}
u & v & 1
\end{pmatrix}
\begin{pmatrix}
0 \\
-T \\
Tv'
\end{pmatrix} = 0
\]
When are epipolar lines horizontal?

When this relationship holds:

\[ R = I \quad t = (T, 0, 0) \]

Let's try this out…

\[ E = t \times R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix} \]

This always has to hold:

\[ x^T E x' = 0 \]

The image of a 3D point will always be on the same horizontal line.

y coordinate is always the same!
It’s hard to make the image planes exactly parallel
How can you make the epipolar lines horizontal?
Use stereo rectification
Stereo matching
Depth Estimation via Stereo Matching
1. Rectify images
   (make epipolar lines horizontal)
2. For each pixel
   a. Find epipolar line
   b. Scan line for best match
   c. Compute depth from disparity
      \[ Z = \frac{bf}{d} \]
When are correspondences difficult?
When are correspondences difficult?

- Textureless regions
- Repeated patterns
- Specularities
- Depth discontinuities
Depth discontinuities

What is the problem here?

One of two input images

Depth from disparity

Groundtruth depth
Depth discontinuities

What is the problem here?
• (Patch-wise) stereo matching blurs along the edges.
How can we fix this?

One of two input images

Depth from disparity

Groundtruth depth
Edge-aware depth denoising

\[
A_p^{(col)} = \frac{1}{k(p^{(col)})} \sum_{p' \in \Omega} g_d(|p - p'|) g_r(F_p^{(col)} - F_{p'}^{(col)}) A_{p'}^{(col)}
\]

Use joint bilateral filtering, with the input image as guide.

One of two input images

Depth from disparity

Guided filtering
Fast bilateral solver

Possible to *combine* edge-enforcement and matching in a single optimization problem, instead of just filtering in post-processing.

One of two input images  
Depth from disparity  
Bilateral stereo matching
Disparity and lightfields
Reminder: a plenoptic “image”

What are these circles?
Reminder: a plenoptic camera

reference plane \((s, t)\)  
aperture plane \((u, v)\)  
sensor plane \((s, t)\)

Lightfield \(L(u, v, s, t)\)

each lenslet corresponds to a slice \(L(u, v, s = s_o, t = t_o)\)
Reminder: form lens image

reference plane \((s, t)\)  \hspace{1cm} aperture plane \((u, v)\)  \hspace{1cm} sensor plane \((s, t)\)

Sum all pixels in each lenslet view.

How do I refocus?
How do I refocus?
- Need to move sensor plane to a different location.
Understanding Refocus

- consider light field inside camera
- synthesize image on sensor \( i_{d=0}(x) = \int_{\Omega} l(x, \nu) d\nu \)

\[
 i_d(x) = \int_{\Omega} l(x + d\nu, \nu) d\nu
\]
Understanding Refocus

- consider light field inside camera
- synthesize image on sensor $i_{d=0}(x) = \int_{\Omega} l(x, \nu) d\nu$

Where did this equation come from?

$$i_d(x) = \int_{\Omega} l(x + dv, \nu) d\nu$$
Stereo view of a lightfield camera

What are the different “cameras” in the lightfield case?
What are the different “cameras” in the lightfield case?

- Different aperture views \( L(u = u_0, v = v_0, s, t) \).

By how much do I need to shift each aperture to focus (i.e., *align*) at depth \( Z \)?
What are the different “cameras” in the lightfield case?

- Different aperture views $L(u = u_0, v = v_0, s, t)$.

By how much do I need to shift each aperture to focus (i.e., align) at depth $Z$?

- By an amount equal to the disparity relative to the center view for depth $Z$. 
Refocusing example
Refocusing example
Refocusing example
3D from lightfield

Simulate different viewpoints?
• Pick same pixel within each aperture view

Can we use different viewpoints for stereo?
3D from lightfield

Simulate different viewpoints?
• Pick same pixel within each aperture view

Can we use different viewpoints for stereo?
• Very small baseline to use disparity algorithm.
• Standard algorithm only works with two views.

Can we do something better?
3D from lightfield

Simulate different viewpoints?
- Pick same pixel within each aperture view

Can we use different viewpoints for stereo?
- Very small baseline to use disparity algorithm.
- Standard algorithm only works with two views.

Can we do something better?
- Take advantage of dense set of views.
- Use disparity to explain changes in views.
Epipolar plane images (EPIs)

Use lightfield to synthesize images for all aperture views on a horizontal line (*scanline*).
Epipolar plane images (EPIs)

Take the same row out of all images in a scanline, and stack these rows in a new 2D image.
Epipolar plane images (EPIs)

Why do we see straight lines?

Take the same row out of all images in a scanline, and stack these rows in a new 2D image.
Epipolar plane images (EPIs)

Why do we see straight lines?
- Same 3D point changes location as viewpoint changes (i.e., disparity).

What does the slope of each line correspond to?

Take the same row out of all images in a scanline, and stack these rows in a new 2D image.
Stereo view of a lightfield camera

Disparity relationship:

- Changing baseline $b$ corresponds to moving along the scanline.
- Projections $x$ of $X$ are on a line of slope inversely proportional to depth.

$$d = x - x' = \frac{bf}{Z}$$
Epipolar plane images (EPIs)

Per-pixel depth detection through line fitting and slope estimation.

Take the same row out of all images in a scanline, and stack these rows in a new 2D image.
Epipolar plane images (EPIs)

Per-pixel depth detection through line fitting and slope estimation.

Take the same row out of all images in a scanline, and stack these rows in a new 2D image.
Figure 1: Our method reconstructs accurate depth from light fields of complex scenes. The images on the left show a 2D slice of a 3D input light field, a so called epipolar-plane image (EPI), and two out of one hundred 21 megapixel images that were used to construct the light field. Our method computes 3D depth information for all visible scene points, illustrated by the depth EPI on the right. From this representation, individual depth maps or segmentation masks for any of the input views can be extracted as well as other representations like 3D point clouds. The horizontal red lines connect corresponding scanlines in the images with their respective position in the EPI.
Aside: different types of cameras

What part of the EPI is captured when we use a stereo pair of cameras?
Aside: different types of cameras

What part of the EPI is captured when we use a stereo pair of cameras?
Aside: different types of cameras

What part of the EPI is captured when we use a stereo pair of cameras?
- Two horizontal lines.

When are these two views sufficient to infer depth?
Aside: different types of cameras

What part of the EPI is captured when we use a stereo pair of cameras?
• Two horizontal lines.

When are these two views sufficient to infer depth?
• When their baseline is large enough to infer the slope of the lines in EPIs.

move along image columns

move along scanline

move along scanline

baseline
When are correspondences difficult?

- Textureless regions
- Repeated patterns
- Specularities
- Depth discontinuities
Use controlled (“structured”) light to make correspondences easier

Disparity between laser points on the same scanline in the images determines the 3-D coordinates of the laser point on object
Use controlled ("structured") light to make correspondences easier
Structured light and two cameras
Structured light and one camera

Projector acts like “reverse” camera
Structured Light

- Any spatio-temporal pattern of light projected on a surface (or volume).
- Cleverly illuminate the scene to extract scene properties (e.g., 3D).
- Avoids problems of 3D estimation in scenes with complex texture/BRDFs.
- Very popular in vision and successful in industrial applications (parts assembly, inspection, etc).
3D Scanning using structured light
Do we need to illuminate the scene point by point?
Light Stripe Scanning – Single Stripe

- Faster optical triangulation:
  - Project a single stripe of laser light
  - Scan it across the surface of the object
  - This is a very precise version of structured light scanning
  - Good for high resolution 3D, but still needs many images and takes time
Triangulation

- Project laser stripe onto object
**Triangulation**

- Depth from ray-plane triangulation:
  - Intersect camera ray with light plane

\[
\begin{align*}
  x &= \frac{x' z}{f} \\
  y &= \frac{y' z}{f} \\
  z &= \frac{-Df}{Ax' + By' + Cf}
\end{align*}
\]
Example: Laser scanner

Digital Michelangelo Project
http://graphics.stanford.edu/projects/mich/
The Digital Michelangelo Project, Levoy et al.
Binary coding
Faster Acquisition?
Faster Acquisition?

- Project multiple stripes simultaneously
- What is the problem with this?
Faster Acquisition?

• Project multiple stripes simultaneously
• Correspondence problem: which stripe is which?

• Common types of patterns:
  • Binary coded light striping
  • Gray/color coded light striping
Binary Coding

Faster:

$2^n - 1$ stripes in $n$ images.

Example:

3 binary-encoded patterns which allows the measuring surface to be divided in 8 sub-regions

Projected over time

Pattern 3
Pattern 2
Pattern 1
Binary Coding

• Assign each stripe a unique illumination code over time [Posdamer 82]
Binary Coding

Example: 7 binary patterns proposed by Posdamer & Altschuler...

Codeword of this pixel: 1010010 → identifies the corresponding pattern stripe

Projected over time
More complex patterns

Works despite complex appearances

Works in real-time and on dynamic scenes

- Need very few images (one or two).
- But needs a more complex correspondence algorithm
Continuum of Triangulation Methods

- Single-stripe: Slow, robust
- Multi-stripe Multi-frame
- Single-frame: Fast, fragile
Using shadows
The geometry

$P = (O, p) \cap \Pi$
The geometry

$\Lambda = (O, \lambda) \cap \Pi_d$

$\Pi = (S, \Lambda)$
The geometry

\[ \Lambda_1 = (O, \lambda_1) \cap \Pi_d \]

\[ \Lambda_2 = (O, \lambda_2) \cap \Pi_v \]

\[ \Pi = (\Lambda_1, \Lambda_2) \]
Angel experiment

Accuracy: 0.1mm over 10cm  ~ 0.1% error
Scanning with the sun

Accuracy: 1cm over 2m

~ 0.5% error
Some notes on (auto-)focusing
Different cameras have different focusing processes
Manual focus in rangefinder cameras

- Focusing based on triangulation: when the image is in focus, you will see the two copies aligned.
- Very accurate but very painstaking.
- Different perspective than that of the main lens.
Manual focus in (D)SLR cameras

- Same view as main lens.
- Just rotate the focusing ring until you are satisfied by the sharpness.
- Viewfinder indicators can help this process.

These arrows will tell you if the focus is in/out and the green dot lets you know that you are in focus.

Grid of points where sharpness is evaluated.
Manual focus in (D)SLR cameras

- Same view as main lens.
- Just rotate the focusing ring until you are satisfied by the sharpness.
- Viewfinder indicators can help this process.

Instead of a grid, you can also focus based on a single point.
Active auto-focus: time-of-flight sensors

- Basically how SONAR works (we’ll also see time-of-flight sensors later in class.
- Method used in Polaroid cameras, which used ultrasound waves.
- Energy inefficient.
- Limited range.
- Multi-path interference (e.g., glass surfaces back-reflected the waves).
Passive auto-focus: phase detection

- As the lens moves, ray bundles from an object converge to a different point in the camera and change in angle.
- This change in angle causes them to refocus through two lenslets to different positions on a separate AF sensor.
- A certain spacing between these double images indicates that the object is “in focus”.

Passive auto-focus: phase detection

Each yellow box indicates *two* sensors, each measuring light from different parts of the aperture.

- Which one is correct focusing?
- How do you need to move the lens or sensor to get correct focusing?
Passive auto-focus: contrast detection

• Sensors at different image distances will see the same object at high-contrast if it’s in focus, or low-contrast if it’s not.
• Move the lens until the high-contrast sub-image falls on the middle sensor, which is conjugate to the camera’s main sensor.
• Compute contrast using local differences of pixel values. Effectively the same as depth from focus.

High-end DSLRs use phase detection

- Distance between sub-images allows lens to move directly into focus, without hunting
- Many AF points corresponding to different points on imaging sensor, complicated algorithms for choosing among them: generally use closest point, but also consider position in FOV.

DSLR mirror has a translucent part that allows some light to make it to the AF sensor.
Low-end cameras (and phones) use contrast detection

- Nowadays it is mostly done using main camera sensor instead of dedicated sensors.
- Requires repeated measurements as lens moves, which are captured using the main sensor (an “autofocus stack”).
- Slow, requires hunting, suffers from overshooting.

But

- People have come up with creative uses for the autofocus stack (depth-from-focus on a cell phone, HDR+ on Android).

[Suwajanakorn et al., CVPR 2015; Hasinoff et al., SIGGRAPH Asia 2016]
Low-end cameras (and phones) use contrast detection

- Nowadays it is mostly done using main camera sensor instead of dedicated sensors.
- Requires repeated measurements as lens moves, which are captured using the main sensor (an “autofocus stack”).
- Slow, requires hunting, suffers from overshooting.

But

- People have come up with creative uses for the autofocus stack (depth-from-focus on a cell phone, HDR+ on Android).

[Suwajanakorn et al., CVPR 2015; Hasinoff et al., SIGGRAPH Asia 2016]
Modern mirror-less cameras use phase detection

- Dedicate a small number of pixels on the imaging sensor to work for phase detection.
- Do this at different parts of the sensor to be able to autofocus at different parts of the image.

Any downsides?
Dual-pixel phase detection autofocus

• Split each pixel into two independent photodiodes—like a two-view lightfield.
• Use different pixels for phase detection.
• Many other interesting opportunities (depth from stereo/lightfield with a tiny baseline).
Should you use autofocus?
Should you use autofocus?

Quick answer: Yes.

More detailed answer: Yes, except for certain special circumstances.

• You are using a lens that does not have an autofocus motor (e.g., vintage or otherwise old lenses, high-end lenses, industrial and machine vision lenses).

• You are trying to capture an image under conditions where autofocus is prone to fail (e.g., macrophotography, poorly-lit scenes, imaging through glass or occluders).

• You intentionally want some part of the scene to be out of focus (e.g., for artistic effect, or because you want a face or other featured to be obscured).

• You are in an once-in-a-lifetime opportunity to photograph something, and you cannot afford to risk autofocus failing. This additionally assumes that:
  - Your scene is static enough that you can take the time to focus manually.
  - You are experienced enough so that the probability of manual focus failing is smaller than the probability of autofocus failing.
Basic reading:

• Szeliski textbook, Sections 7.1, 11.1, 12.1.
  - This classical paper introduces EPIs, and discusses how they can be used to infer depth.
• Lanman and Taubin, “Build Your Own 3D Scanner: Optical Triangulation for Beginners,” SIGGRAPH course 2009.
  - This very comprehensive course has everything you need to know about 3D scanning using structured light, including details on how to build your own.
  - This paper introduces the idea of using shadows to do structured light 3D scanning, and shows an implementation using just a camera, desk lamp, and a stick.

Additional reading:

  - This paper has a very detailed treatment of standard patterns used for structured light, problems arising due to global illumination, and robust patterns for dealing with these patterns.
• Barron and Poole, “The fast bilateral solver,” ECCV 2016.
  - The above two papers show how to combine edge-aware filtering (and bilateral filtering in particular) with disparity matching for robust stereo. The first paper also shows how the resulting depth maps can be used to create synthetic defocus blur.
• Kim et al., “Scene reconstruction from high spatio-angular resolution light fields,” SIGGRAPH 2013.
  - These two papers show detailed systems for using EPIs to extract depth.
  - This paper uses EPIs to show how different types of imaging systems (pinhole cameras, plenoptic cameras, stereo pairs, lens-based systems, and so on) relate to each other, and analyze their pros and cons for 3D imaging.