Two-view geometry, stereo, and disparity
Course announcements

• Homework assignment 6 posted, due December 12.
  - Start early: Capturing structured light stereo is challenging.

• Grades for homework assignments 3 and 4 posted.
  - Photography competition winners still pending.

• Propose topics for this week’s reading group.

• Final projects.
Overview of today’s lecture

• Triangulation.
• Epipolar geometry.
• 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).
Triangulation
Triangulation

Given camera 1 with matrix $P$, camera 2 with matrix $P'$.
Triangulation

Which 3D points map to $x$?

image 1

$\mathbf{x}$

camera 1 with matrix $\mathbf{P}$

image 2

$\mathbf{x}'$

camera 2 with matrix $\mathbf{P}'$
Triangulation

How can you compute this ray?

image 1

image 2

camera 1 with matrix $P$

camera 2 with matrix $P'$
Triangulation

Create two points on the ray:
1) find the camera center; and
2) apply the pseudo-inverse of $\mathbf{P}$ on $\mathbf{x}$.
Then connect the two points.

This procedure is called backprojection.
Triangulation

How do we find the exact point on the ray? $P^+x$
Triangulation

Find 3D object point

Will the lines intersect?

image 1

image 2

$C$

$C'$

$P$

$P'$

camera 1 with matrix

camera 2 with matrix
Triangulation

Find 3D object point
(no single solution due to noise)
Triangulation

Given a set of (noisy) matched points

\[ \{x_i, x'_i\} \]

and camera matrices

\[ P, P' \]

Estimate the 3D point

\[ X \]
Can we compute $X$ from a single correspondence $x$?
This is a similarity relation because it involves homogeneous coordinates:

\[ \mathbf{x} = \mathbf{P} \mathbf{X} \]

(homogeneous coordinate)

Same ray direction but differs by a scale factor:

\[ \mathbf{x} = \alpha \mathbf{P} \mathbf{X} \]

(homogeneous coordinate)

\[
\begin{bmatrix}
x \\
y \\
z
\end{bmatrix} = \alpha \begin{bmatrix}
p_1 & p_2 & p_3 & p_4 \\
p_5 & p_6 & p_7 & p_8 \\
p_9 & p_{10} & p_{11} & p_{12}
\end{bmatrix} \begin{bmatrix}
X \\
Y \\
Z \\
1
\end{bmatrix}
\]

How do we solve for unknowns in a similarity relation?
\[ \mathbf{x} = \mathbf{P} \mathbf{X} \]

(homogeneous coordinate)

Also, this is a similarity relation because it involves homogeneous coordinates

\[ \mathbf{x} = \alpha \mathbf{P} \mathbf{X} \]

(inhomogeneous coordinate)

Same ray direction but differs by a scale factor

\[
\begin{bmatrix}
  x \\
  y \\
  z
\end{bmatrix}
= \alpha
\begin{bmatrix}
  p_1 & p_2 & p_3 & p_4 \\
  p_5 & p_6 & p_7 & p_8 \\
  p_9 & p_{10} & p_{11} & p_{12}
\end{bmatrix}
\begin{bmatrix}
  X \\
  Y \\
  Z \\
  1
\end{bmatrix}
\]

How do we solve for unknowns in a similarity relation?
Linear algebra reminder: cross product

Vector (cross) product

takes two vectors and returns a vector perpendicular to both

\[ c = a \times b \]

\[ a \times b = \begin{bmatrix} a_2b_3 - a_3b_2 \\ a_3b_1 - a_1b_3 \\ a_1b_2 - a_2b_1 \end{bmatrix} \]

cross product of two vectors in the same direction is zero vector

\[ a \times a = 0 \]

remember this!!!

\[ c \cdot a = 0 \quad c \cdot b = 0 \]
Linear algebra reminder: cross product

Cross product

\[
\mathbf{a} \times \mathbf{b} = \begin{bmatrix}
 a_2 b_3 - a_3 b_2 \\
 a_3 b_1 - a_1 b_3 \\
 a_1 b_2 - a_2 b_1
\end{bmatrix}
\]

Can also be written as a matrix multiplication

\[
\mathbf{a} \times \mathbf{b} = [\mathbf{a}] \times \mathbf{b} = \begin{bmatrix}
 0 & -a_3 & a_2 \\
 a_3 & 0 & -a_1 \\
 -a_2 & a_1 & 0
\end{bmatrix} \begin{bmatrix}
 b_1 \\
 b_2 \\
 b_3
\end{bmatrix}
\]

Skew symmetric
Compare with: dot product

\[ c = a \times b \]

\[ c \cdot a = 0 \quad c \cdot b = 0 \]

dot product of two orthogonal vectors is (scalar) zero
Back to triangulation

\[ x = \alpha PX \]

Same direction but differs by a scale factor

*How can we rewrite this using vector products?*
\[ \mathbf{x} = \alpha \mathbf{P} \mathbf{X} \]

Same direction but differs by a scale factor

\[ \mathbf{x} \times \mathbf{P} \mathbf{X} = 0 \]

Cross product of two vectors of same direction is zero

(this equality removes the scale factor)
\[
\begin{bmatrix}
  x \\
  y \\
  z
\end{bmatrix} = \alpha
\begin{bmatrix}
  p_1 & p_2 & p_3 & p_4 \\
  p_5 & p_6 & p_7 & p_8 \\
  p_9 & p_{10} & p_{11} & p_{12}
\end{bmatrix}
\begin{bmatrix}
  X \\
  Y \\
  Z \\
  1
\end{bmatrix}
\]

Do the same after first expanding out the camera matrix and points

\[
\begin{bmatrix}
  x \\
  y \\
  z
\end{bmatrix} = \alpha
\begin{bmatrix}
  \cdots & p_1^\top & \cdots \\
  \cdots & p_2^\top & \cdots \\
  \cdots & p_3^\top & \cdots
\end{bmatrix}
\begin{bmatrix}
  X
\end{bmatrix}
\]

\[
\begin{bmatrix}
  x \\
  y \\
  z
\end{bmatrix} = \alpha
\begin{bmatrix}
  p_1^\top X \\
  p_2^\top X \\
  p_3^\top X
\end{bmatrix}
\]

\[
\begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix}
\times
\begin{bmatrix}
  p_1^\top X \\
  p_2^\top X \\
  p_3^\top X
\end{bmatrix}
= \begin{bmatrix}
  yp_3^\top X - p_2^\top X \\
  p_1^\top X - xp_3^\top X \\
  xp_2^\top X - yp_1^\top X
\end{bmatrix}
= \begin{bmatrix}
  0 \\
  0 \\
  0
\end{bmatrix}
\]
Using the fact that the cross product should be zero

\[ \mathbf{x} \times \mathbf{PX} = 0 \]

\[
\begin{bmatrix}
    y p_3^\top X - p_2^\top X \\
    p_1^\top X - x p_3^\top X \\
    x p_2^\top X - y p_1^\top X
\end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}
\]

Third line is a linear combination of the first and second lines.
(x times the first line plus y times the second line)

One 2D to 3D point correspondence give you 3 equations
Using the fact that the cross product should be zero

\[ \mathbf{x} \times \mathbf{PX} = 0 \]

\[
\begin{bmatrix}
yp_3^\top X - p_2^\top X \\
p_1^\top X - xp_3^\top X \\
kp_2^\top X - yp_1^\top X
\end{bmatrix}
= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}
\]

Third line is a linear combination of the first and second lines.
(x times the first line plus y times the second line)

One 2D to 3D point correspondence give you 2 equations
Now we can make a system of linear equations (two lines for each 2D point correspondence)

\[
\begin{bmatrix}
  y_{p_3}^T X - p_2^T X \\
  p_1^T X - x p_{p_3}^T X
\end{bmatrix}
= \begin{bmatrix}
  0 \\
  0
\end{bmatrix}
\]

Remove third row, and rearrange as system on unknowns

\[
\begin{bmatrix}
  y_{p_3}^T - p_2^T \\
  p_1^T - x p_{p_3}^T
\end{bmatrix} X = \begin{bmatrix}
  0 \\
  0
\end{bmatrix}
\]

\[A_i X = 0\]
How do we solve homogeneous linear system?

Concatenate the 2D points from both images

\[
\begin{bmatrix}
yp_3^\top - p_2^\top \\
p_1^\top - xp_3^\top \\
y'p_3'^\top - p_2'^\top \\
p_1'^\top - x'p_3'^\top \\
\end{bmatrix} \quad X = \begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}
\]

sanity check! dimensions?

\[AX = 0\]

How do we solve homogeneous linear system?
How do we solve homogeneous linear system?

Concatenate the 2D points from both images

\[
\begin{bmatrix}
yp_3^\top - p_2^\top \\
p_1^\top - xp_3^\top \\
y'p_3'^\top - p'_2^\top \\
p'_1^\top - x'p'_3^\top
\end{bmatrix} X = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
\]

\[AX = 0\]

How do we solve homogeneous linear system?

SVD!
How would you reconstruct 3D points?

Left image

Right image
Epipolar geometry
Epipolar geometry

Image plane
Epipolar geometry

Image plane

Baseline

Epipole
(projection of o’ on the image plane)
Epipolar geometry

- Epipolar plane
- Baseline
- Epipole (projection of o’ on the image plane)
- Image plane
- Epipolar geometry
Epipolar geometry

Epipolar line (intersection of Epipolar plane and image plane)

Epipole (projection of o’ on the image plane)
Epipolar constraint

Backproject \( \mathbf{x} \) to a ray in 3D

Epipolar line (intersection of Epipolar plane and image plane)

Another way to construct the epipolar plane, this time given \( \mathbf{x} \)
Epipolar constraint

Potential matches for \( x \) lie on the epipolar line \( l' \)
The point $\mathbf{x}$ (left image) maps to a ___________ in the right image.

The baseline connects the ___________ and ____________.

An epipolar line (left image) maps to a ____________ in the right image.

An epipole $\mathbf{e}$ is a projection of the ______________ on the image plane.

All epipolar lines in an image intersect at the ______________.
Where is the epipole in this image?
Converging cameras

Where is the epipole in this image?

It's not always in the image
Parallel cameras

Where is the epipole?
Parallel cameras

epipole at infinity
The epipolar constraint is an important concept for stereo vision

**Task:** Match point in left image to point in right image

*How would you do it?*
Epipolar constraint

Potential matches for $x$ lie on the epipolar line $l'$.
The epipolar constraint is an important concept for stereo vision.

**Task:** Match point in left image to point in right image

Want to avoid search over entire image

Epipolar constraint reduces search to a single line
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?)
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

camera center

image plane

O

O'

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

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

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

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

**Disparity**

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

\[
d = \frac{bf}{Z}
\]
Disparity

\[ d = x - x' \]

inversely proportional to depth

\[ d = \frac{bf}{Z} \]
Real-time stereo sensing

Nomad robot searches for meteorites in Antartica

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
Teesta suspension bridge-Darjeeling, India
This is how 3D movies work
Simple stereoscope

Google cardboard

Fun patterns: random dot stereograms

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:

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

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

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

How do I refocus?

Sum all pixels in each lenslet view.
Reminder: form lens image

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

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

Sum all pixels in each lenslet view.
Understanding Refocus

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

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

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

$$i_d(x) = \int_{\Omega} l(x + dv, v) dv$$

Where did this equation come from?
Stereo view of a lightfield camera

What are the different “cameras” in the lightfield case?
Stereo view of a lightfield camera

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

- Different aperture views $L(u = u_o, v = v_o, s, t)$.

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 $L(u = u_o, v = v_o, s, t)$.

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:

\[ d = x - x' = \frac{bf}{Z} \]

- Changing baseline \( b \) corresponds to moving along the scanline.
- Projections \( x \) of \( X \) are on a line of slope inversely proportional to depth.
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.
When are correspondences difficult?

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

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

Digital Michelangelo Project
http://graphics.stanford.edu/projects/mich/
The Digital Michelangelo Project, Levoy et al.
The Digital Michelangelo Project, Levoy et al.
The Digital Michelangelo Project, Levoy et al.
The Digital Michelangelo Project, Levoy et al.
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
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.
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

Multi-stripe
Multi-frame

Single-frame

Slow, robust  Fast, fragile
Using shadows
The idea

Desk Lamp

Stick or pencil

Camera

Desk

Time t
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  \( \sim \) 0.1% error
Scanning with the sun

Accuracy: 1 cm over 2 m

~ 0.5% error
Revisiting auto-focusing
Why does this work 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.

standard in Leica cameras
Why does this work for 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”.

Why does this work for 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?
Dual-pixels: disparity with incredibly small baseline

- 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).
References

Basic reading:
• Szeliski textbook, Sections 7.1, 8.1, Chapter 11, Sections 11.1, 12.2.
• Hartley and Zisserman, Section 11.12.
  - 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.