

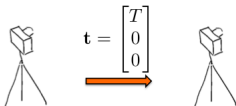
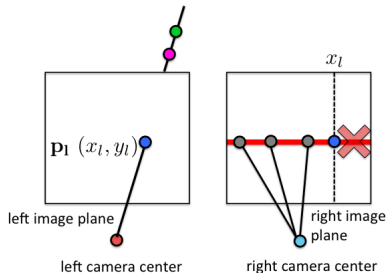
# Stereo

## Epipolar Geometry for General Cameras

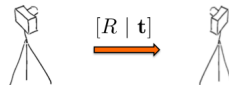
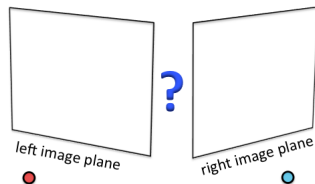
## Epipolar geometry

- Case with two cameras with parallel optical axes
- General case ← **Now this**

Parallel stereo cameras:



General stereo cameras:



# Epipolar Geometry

- If I can always mount two cameras parallel to each other, why do I need to learn math for the general case?

# Epipolar Geometry

- Let's say that you want to reconstruct a CN tower in 3D



# Epipolar Geometry

- Let's say that you want to reconstruct a CN tower in 3D
- One out of endless possibilities of why you would do that:
  - You can print it with a 3D printer to get a nice pocket or not-so-pocket edition (better than those that are sold in Chinatown)
  - Give it to your mum for Christmas (say it's a present from CSC420)



VS



# Epipolar Geometry

- Let's say that you want to reconstruct a CN tower in 3D
- You obviously can't get a good 360 shot of the CN tower with just parallel cameras. Particularly not the top of the CN tower which is very high up.

# Epipolar Geometry

- Let's say that you want to reconstruct a CN tower in 3D
- You obviously can't get a good 360 shot of the CN tower with just parallel cameras. Particularly not the top of the tower.
- But you can download great images of the tower from the web without even needing to leave the house.

# Epipolar Geometry

- But these images are not taken from parallel cameras...



# Photosynth

- You could even do part of Venice...

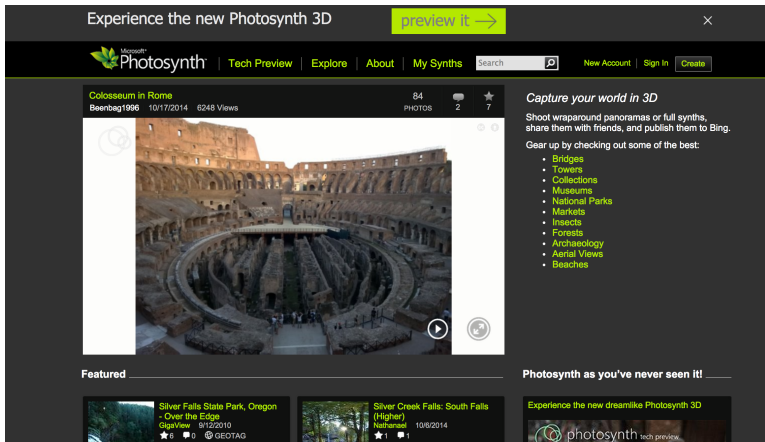


Figure: <https://www.youtube.com/watch?v=HrgHFDPJHXo>

Noah Snavely, Steven M. Seitz, Richard Szeliski, "Photo tourism: Exploring photo collections in 3D", SIGGRAPH 2006, <https://photosynth.net/>

# World Cup 2014 – High Tech 3D

- Last World Cup was monitored with 14 high-speed cameras, capturing 500 frames per second, and could accurately detect ball motion to within 5mm.
- 2,000 tests performed, all successful. By German company Goal Control.

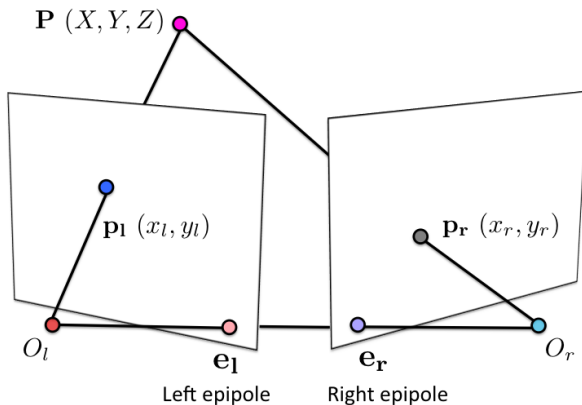


# Stereo – General Case

## Ready for the math?

# Stereo: Parallel Calibrated Cameras

- Some notation: the **left** and **right epipole**

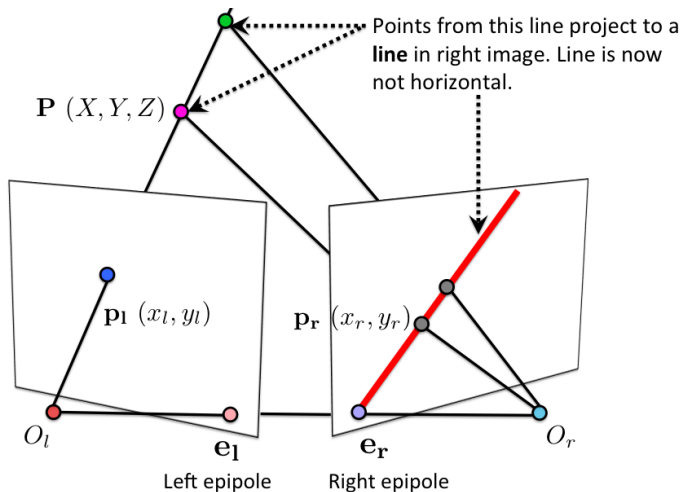


Where line  $O_l O_r$  intersects the image planes



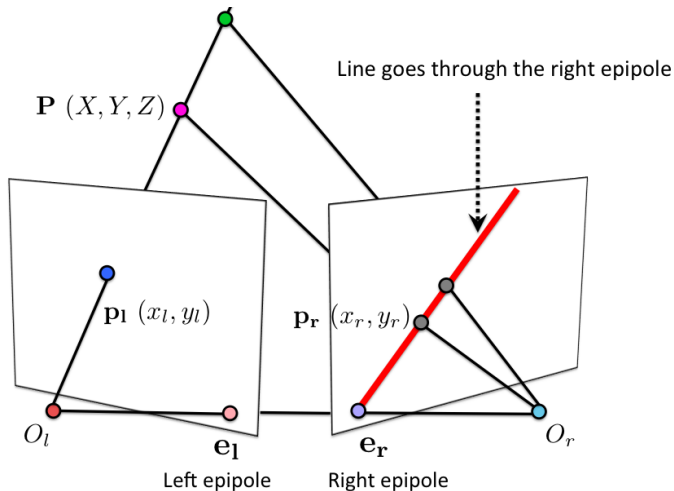
# Stereo: Parallel Calibrated Cameras

- All points from the projective line  $O_l p_l$  project to a line on the right image plane. This time the line is not (necessarily) horizontal.



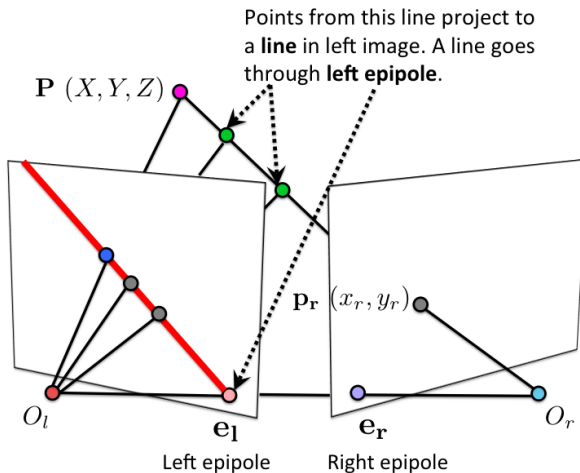
# Stereo: Parallel Calibrated Cameras

- The line goes through the right epipole.



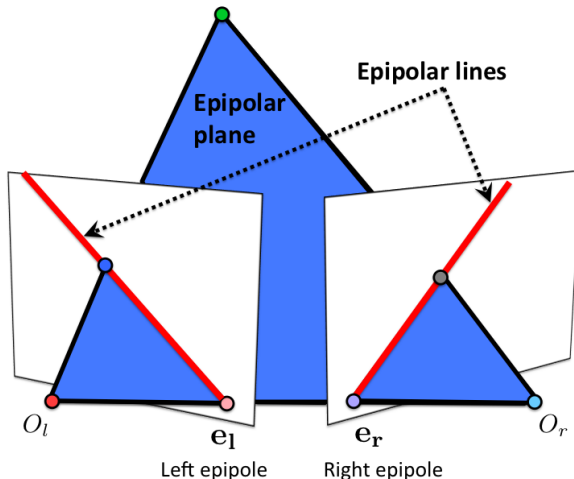
# Stereo: Parallel Calibrated Cameras

- Similarly, All points from the projective line  $\mathbf{O}_r \mathbf{p}_r$  project to a line on the left image plane. This line goes through the left epipole.



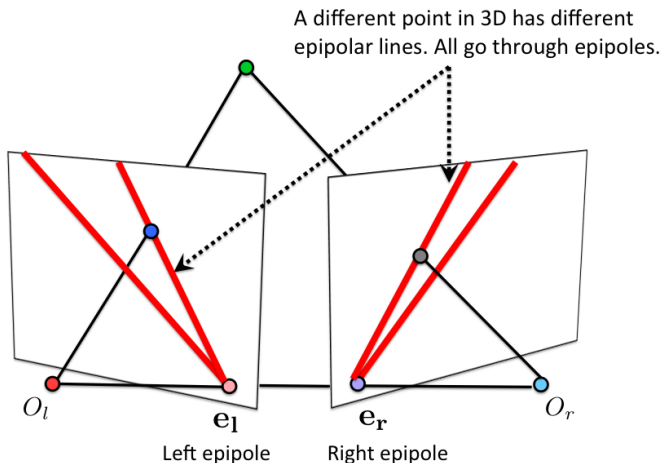
# Stereo: Parallel Calibrated Cameras

- The reason for all this is simple: points  $O_l$ ,  $O_r$ , and a point  $P$  in 3D lie on a plane. We call this the **epipolar plane**. This plane intersects each image plane in a line. We call these lines **epipolar lines**.



# Stereo: Parallel Calibrated Cameras

- Obviously a different point in 3D will form a different epipolar plane and therefore different epipolar lines. But these epipolar lines go through epipoles as well.

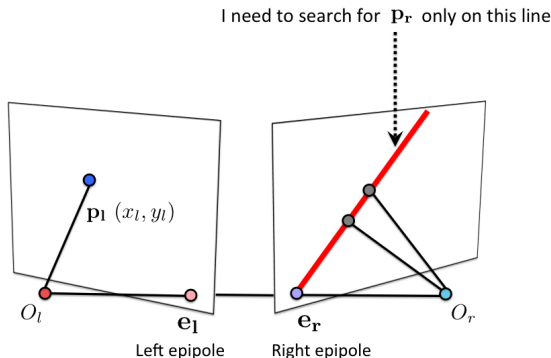


# Stereo: Parallel Calibrated Cameras

- Why are we even dumping all this notation? Are epipolar lines, epipoles, etc somehow useful?

# Stereo: Parallel Calibrated Cameras

- Remember what we did for parallel cameras? We were matching points in the left and right image, giving us a point in 3D. We want the same now.
- Epipolar geometry is useful because it constrains our search for the matches:
  - For each point  $\mathbf{p}_l$  we need to search for  $\mathbf{p}_r$  only on a epipolar line (much simpler than if I need to search in the full image)
  - All matches lie on lines that intersect in epipoles. This gives another constraint.



# Epipolar geometry: Examples

- Example of epipolar lines for converging cameras



[Source: J. Hays, pic from Hartley & Zisserman]



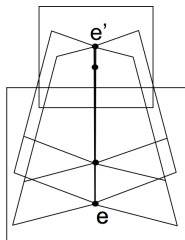
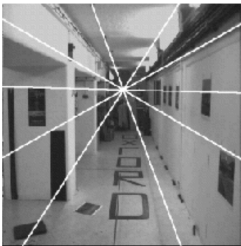
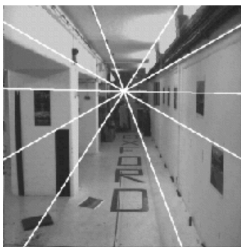
# Epipolar geometry: Examples

- How would epipolar lines look like if the camera moves directly forward?

[Source: J. Hays]

# Epipolar geometry: Examples

- Example of epipolar lines for **forward motion**



Epipole has same coordinates in both images.

Points move along lines radiating from  $e$ :  
“Focus of expansion”

[Source: J. Hays, pic from Hartley & Zisserman]

# Stereo for General Cameras

How we'll get 3D:

- We first need to figure out on which line we need to search for the matches for each  $\mathbf{p}_l$
- Each point in left image maps to a line in right image. We will see that this mapping can be described by a single  $3 \times 3$  matrix  $\mathbf{F}$ , called the **fundamental matrix**

# Stereo for General Cameras

How we'll get 3D:

- We first need to figure out on which line we need to search for the matches for each  $\mathbf{p}_l$
- Each point in left image maps to a line in right image. We will see that this mapping can be described by a single  $3 \times 3$  matrix  $\mathbf{F}$ , called the **fundamental matrix**
- Given  $\mathbf{F}$ , you can **rectify** the images such that the epipolar lines are horizontal

# Stereo for General Cameras

How we'll get 3D:

- We first need to figure out on which line we need to search for the matches for each  $\mathbf{p}_l$
- Each point in left image maps to a line in right image. We will see that this mapping can be described by a single  $3 \times 3$  matrix  $\mathbf{F}$ , called the **fundamental matrix**
- Given  $\mathbf{F}$ , you can **rectify** the images such that the epipolar lines are horizontal
- And we know how to take it from there

# Stereo for General Cameras

How we'll get 3D:

- We first need to figure out on which line we need to search for the matches for each  $\mathbf{p}_l$
- Each point in left image maps to a line in right image. We will see that this mapping can be described by a single  $3 \times 3$  matrix  $\mathbf{F}$ , called the **fundamental matrix**
- Given  $\mathbf{F}$ , you can **rectify** the images such that the epipolar lines are horizontal
- And we know how to take it from there

# The Fundamental Matrix

- The fundamental matrix  $\mathbf{F}$  is defined as  $\mathbf{l}_r = \mathbf{F}\mathbf{p}_l$ , where  $\mathbf{l}_r$  is the right epipolar line corresponding to  $\mathbf{p}_l$ .
- $\mathbf{F}$  is a  $3 \times 3$  matrix

# The Fundamental Matrix

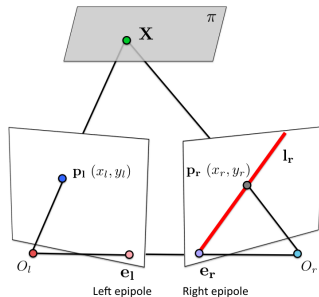
- The fundamental matrix **F** is defined as  $\mathbf{l}_r = \mathbf{F}\mathbf{p}_l$ , where  $\mathbf{l}_r$  is the right epipolar line corresponding to  $\mathbf{p}_l$ .
- **F** is a  $3 \times 3$  matrix
- For any point  $\mathbf{p}_l$  its epipolar line is defined by the **same matrix F**.



# The Fundamental Matrix

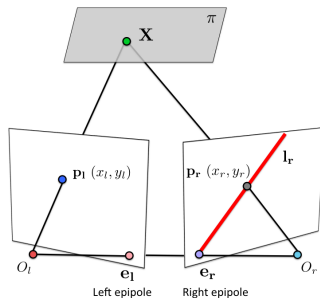
- The fundamental matrix  $\mathbf{F}$  is defined as  $\mathbf{l}_r = \mathbf{F}\mathbf{p}_l$ , where  $\mathbf{l}_r$  is the right epipolar line corresponding to  $\mathbf{p}_l$ .
- $\mathbf{F}$  is a  $3 \times 3$  matrix
- For any point  $\mathbf{p}_l$  its epipolar line is defined by the **same matrix  $\mathbf{F}$** .

# The Fundamental Matrix



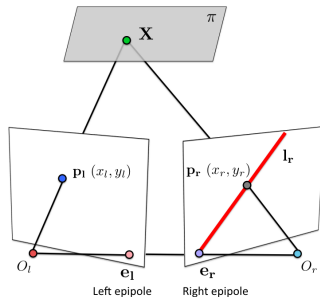
- Extend the line  $O_l p_l$  until you hit a plane  $\pi$  (arbitrary)
- Find the image  $p_r$  of  $X$  in the right camera

# The Fundamental Matrix



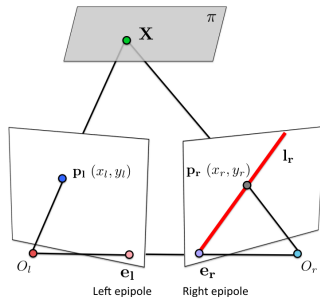
- Extend the line  $O_l p_l$  until you hit a plane  $\pi$  (arbitrary)
- Find the image  $p_r$  of  $X$  in the right camera
- Get epipolar line  $l_r$  from  $e_r$  to  $p_r$ :  $l_r = e_r \times p_r$

# The Fundamental Matrix



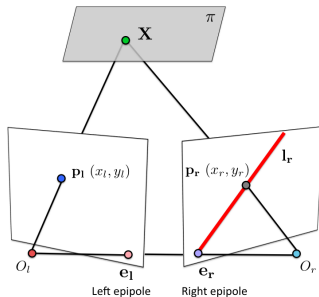
- Extend the line  $O_l p_l$  until you hit a plane  $\pi$  (arbitrary)
- Find the image  $p_r$  of  $X$  in the right camera
- Get epipolar line  $l_r$  from  $e_r$  to  $p_r$ :  $l_r = e_r \times p_r$
- Points  $p_l$  and  $p_r$  are related via homography:  $p_r = H_\pi p_l$

# The Fundamental Matrix



- Extend the line  $\mathbf{O}_l \mathbf{p}_l$  until you hit a plane  $\pi$  (arbitrary)
- Find the image  $\mathbf{p}_r$  of  $\mathbf{X}$  in the right camera
- Get epipolar line  $\mathbf{l}_r$  from  $\mathbf{e}_r$  to  $\mathbf{p}_r$ :  $\mathbf{l}_r = \mathbf{e}_r \times \mathbf{p}_r$
- Points  $\mathbf{p}_l$  and  $\mathbf{p}_r$  are related via homography:  $\mathbf{p}_r = H_\pi \mathbf{p}_l$
- Then:  $\mathbf{l}_r = \mathbf{e}_r \times \mathbf{p}_r = \mathbf{e}_r \times H_\pi \mathbf{p}_l = \mathbf{F} \mathbf{p}_l$

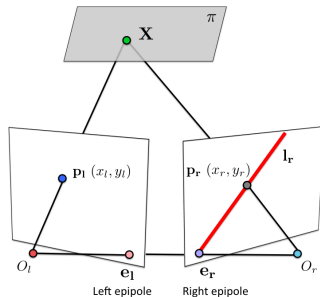
# The Fundamental Matrix



- Extend the line  $\mathbf{O}_l \mathbf{p}_l$  until you hit a plane  $\pi$  (arbitrary)
- Find the image  $\mathbf{p}_r$  of  $\mathbf{X}$  in the right camera
- Get epipolar line  $\mathbf{l}_r$  from  $\mathbf{e}_r$  to  $\mathbf{p}_r$ :  $\mathbf{l}_r = \mathbf{e}_r \times \mathbf{p}_r$
- Points  $\mathbf{p}_l$  and  $\mathbf{p}_r$  are related via homography:  $\mathbf{p}_r = H_\pi \mathbf{p}_l$
- Then:  $\mathbf{l}_r = \mathbf{e}_r \times \mathbf{p}_r = \mathbf{e}_r \times H_\pi \mathbf{p}_l = \mathbf{F} \mathbf{p}_l$
- The fundamental matrix  $\mathbf{F}$  is defined  $\mathbf{l}_r = \mathbf{F} \mathbf{p}_l$

[Adopted from: R. Urtasun]

# The Fundamental Matrix



- Extend the line  $\mathbf{O}_l \mathbf{p}_l$  until you hit a plane  $\pi$  (arbitrary)
- Find the image  $\mathbf{p}_r$  of  $\mathbf{X}$  in the right camera
- Get epipolar line  $\mathbf{l}_r$  from  $\mathbf{e}_r$  to  $\mathbf{p}_r$ :  $\mathbf{l}_r = \mathbf{e}_r \times \mathbf{p}_r$
- Points  $\mathbf{p}_l$  and  $\mathbf{p}_r$  are related via homography:  $\mathbf{p}_r = H_\pi \mathbf{p}_l$
- Then:  $\mathbf{l}_r = \mathbf{e}_r \times \mathbf{p}_r = \mathbf{e}_r \times H_\pi \mathbf{p}_l = \mathbf{F} \mathbf{p}_l$
- The fundamental matrix  $\mathbf{F}$  is defined  $\mathbf{l}_r = \mathbf{F} \mathbf{p}_l$

[Adopted from: R. Urtasun]

# The Fundamental Matrix

- The fundamental matrix  $\mathbf{F}$  is defined as  $\mathbf{l}_r = \mathbf{F}\mathbf{p}_l$ , where  $\mathbf{l}_r$  is the right epipolar line corresponding to  $\mathbf{p}_l$ .
- $\mathbf{F}$  is a  $3 \times 3$  matrix
- For any point  $\mathbf{p}_l$  its epipolar line is defined by the **same matrix  $\mathbf{F}$** .
- Do a trick:

$$\mathbf{p}_r^T \cdot \mathbf{l}_r = \mathbf{p}_r^T \mathbf{F} \mathbf{p}_l$$



# The Fundamental Matrix

- The fundamental matrix **F** is defined as  $\mathbf{l}_r = \mathbf{F}\mathbf{p}_l$ , where  $\mathbf{l}_r$  is the right epipolar line corresponding to  $\mathbf{p}_l$ .
- **F** is a  $3 \times 3$  matrix
- For any point  $\mathbf{p}_l$  its epipolar line is defined by the **same matrix F**.
- Do a trick:

$$\underbrace{\mathbf{p}_r^T \cdot \mathbf{l}_r}_{=0, \text{ because } \mathbf{p}_r \text{ lies on a line } \mathbf{l}_r} = \mathbf{p}_r^T \mathbf{F} \mathbf{p}_l$$

# The Fundamental Matrix

- The fundamental matrix  $\mathbf{F}$  is defined as  $\mathbf{l}_r = \mathbf{F}\mathbf{p}_l$ , where  $\mathbf{l}_r$  is the right epipolar line corresponding to  $\mathbf{p}_l$ .
- $\mathbf{F}$  is a  $3 \times 3$  matrix
- For any point  $\mathbf{p}_l$  its epipolar line is defined by the **same matrix  $\mathbf{F}$** .
- So:

$$\mathbf{p}_r^T \mathbf{F} \mathbf{p}_l = 0$$

for any match  $(\mathbf{p}_l, \mathbf{p}_r)$  (main thing to remember)!!

- We can compute  $\mathbf{F}$  from a few correspondences. How do we get these correspondences?

# The Fundamental Matrix

- The fundamental matrix  $\mathbf{F}$  is defined as  $\mathbf{l}_r = \mathbf{F}\mathbf{p}_l$ , where  $\mathbf{l}_r$  is the right epipolar line corresponding to  $\mathbf{p}_l$ .
- $\mathbf{F}$  is a  $3 \times 3$  matrix
- For any point  $\mathbf{p}_l$  its epipolar line is defined by the **same matrix  $\mathbf{F}$** .
- So:

$$\mathbf{p}_r^T \mathbf{F} \mathbf{p}_l = 0$$

for any match  $(\mathbf{p}_l, \mathbf{p}_r)$  (main thing to remember)!!

- We can compute  $\mathbf{F}$  from a few correspondences. How do we get these correspondences?
- By finding reliable matches across two images without any constraints. We know how to do this from our DVD matching example.
- We get a linear system.

# The Fundamental Matrix

- The fundamental matrix  $\mathbf{F}$  is defined as  $\mathbf{l}_r = \mathbf{F}\mathbf{p}_l$ , where  $\mathbf{l}_r$  is the right epipolar line corresponding to  $\mathbf{p}_l$ .
- $\mathbf{F}$  is a  $3 \times 3$  matrix
- For any point  $\mathbf{p}_l$  its epipolar line is defined by the **same matrix  $\mathbf{F}$** .
- So:

$$\mathbf{p}_r^T \mathbf{F} \mathbf{p}_l = 0$$

for any match  $(\mathbf{p}_l, \mathbf{p}_r)$  (main thing to remember)!!

- We can compute  $\mathbf{F}$  from a few correspondences. How do we get these correspondences?
- By finding reliable matches across two images without any constraints. We know how to do this from our DVD matching example.
- We get a linear system.

# The Fundamental Matrix

- Let's say that you found a few matching points in both images:  
 $(x_{l,1}, y_{l,1}) \leftrightarrow (x_{r,1}, y_{r,1}), \dots, (x_{l,n}, y_{l,n}) \leftrightarrow (x_{r,n}, y_{r,n})$ , where  $n \geq 7$
- Then you can get the parameters  $\mathbf{f} := [F_{11}, F_{12}, \dots, F_{33}]$  by solving:

$$\begin{bmatrix} x_{r,1} x_{l,1} & x_{r,1} y_{l,1} & x_{r,1} & y_{r,1} x_{l,1} & y_{r,1} y_{l,1} & y_{r,1} & x_{l,1} & y_{l,1} & 1 \\ & & & \vdots & & & & & \\ x_{r,n} x_{l,n} & x_{r,n} y_{l,n} & x_{r,n} & y_{r,n} x_{l,n} & y_{r,n} y_{l,n} & y_{r,n} & x_{l,n} & y_{l,n} & 1 \end{bmatrix} \mathbf{f} = \mathbf{0}$$

- How many correspondences do we need?
- $\mathbf{F}$  has 9 elements, but we don't care about scaling, so 8 elements.

# The Fundamental Matrix

- Let's say that you found a few matching points in both images:  
 $(x_{l,1}, y_{l,1}) \leftrightarrow (x_{r,1}, y_{r,1}), \dots, (x_{l,n}, y_{l,n}) \leftrightarrow (x_{r,n}, y_{r,n})$ , where  $n \geq 7$
- Then you can get the parameters  $\mathbf{f} := [F_{11}, F_{12}, \dots, F_{33}]$  by solving:

$$\begin{bmatrix} x_{r,1} x_{l,1} & x_{r,1} y_{l,1} & x_{r,1} & y_{r,1} x_{l,1} & y_{r,1} y_{l,1} & y_{r,1} & x_{l,1} & y_{l,1} & 1 \\ & & & \vdots & & & & & \\ x_{r,n} x_{l,n} & x_{r,n} y_{l,n} & x_{r,n} & y_{r,n} x_{l,n} & y_{r,n} y_{l,n} & y_{r,n} & x_{l,n} & y_{l,n} & 1 \end{bmatrix} \mathbf{f} = \mathbf{0}$$

- How many correspondences do we need?
- $\mathbf{F}$  has 9 elements, but we don't care about scaling, so 8 elements.
- Turns out it really only has 7.

# The Fundamental Matrix

- Let's say that you found a few matching points in both images:  
 $(x_{l,1}, y_{l,1}) \leftrightarrow (x_{r,1}, y_{r,1}), \dots, (x_{l,n}, y_{l,n}) \leftrightarrow (x_{r,n}, y_{r,n})$ , where  $n \geq 7$
- Then you can get the parameters  $\mathbf{f} := [F_{11}, F_{12}, \dots, F_{33}]$  by solving:

$$\begin{bmatrix} x_{r,1} x_{l,1} & x_{r,1} y_{l,1} & x_{r,1} & y_{r,1} x_{l,1} & y_{r,1} y_{l,1} & y_{r,1} & x_{l,1} & y_{l,1} & 1 \\ & & & \vdots & & & & & \\ x_{r,n} x_{l,n} & x_{r,n} y_{l,n} & x_{r,n} & y_{r,n} x_{l,n} & y_{r,n} y_{l,n} & y_{r,n} & x_{l,n} & y_{l,n} & 1 \end{bmatrix} \mathbf{f} = \mathbf{0}$$

- How many correspondences do we need?
- $\mathbf{F}$  has 9 elements, but we don't care about scaling, so 8 elements.
- Turns out it really only has 7.
- We can estimate  $\mathbf{F}$  with 7 correspondences. Of course, the more the better (why?).
- See Zisserman & Hartley's book for details.

# The Fundamental Matrix

- Let's say that you found a few matching points in both images:  
 $(x_{l,1}, y_{l,1}) \leftrightarrow (x_{r,1}, y_{r,1}), \dots, (x_{l,n}, y_{l,n}) \leftrightarrow (x_{r,n}, y_{r,n})$ , where  $n \geq 7$
- Then you can get the parameters  $\mathbf{f} := [F_{11}, F_{12}, \dots, F_{33}]$  by solving:

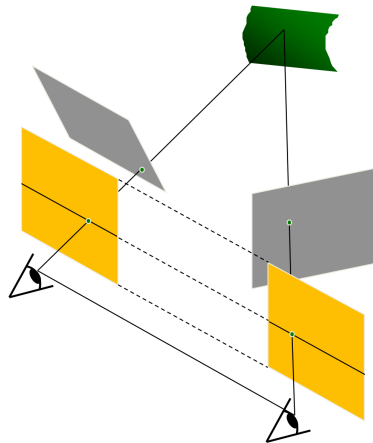
$$\begin{bmatrix} x_{r,1} x_{l,1} & x_{r,1} y_{l,1} & x_{r,1} & y_{r,1} x_{l,1} & y_{r,1} y_{l,1} & y_{r,1} & x_{l,1} & y_{l,1} & 1 \\ & & & \vdots & & & & & \\ x_{r,n} x_{l,n} & x_{r,n} y_{l,n} & x_{r,n} & y_{r,n} x_{l,n} & y_{r,n} y_{l,n} & y_{r,n} & x_{l,n} & y_{l,n} & 1 \end{bmatrix} \mathbf{f} = \mathbf{0}$$

- How many correspondences do we need?
- $\mathbf{F}$  has 9 elements, but we don't care about scaling, so 8 elements.
- Turns out it really only has 7.
- We can estimate  $\mathbf{F}$  with 7 correspondences. Of course, the more the better (why?).
- See Zisserman & Hartley's book for details.



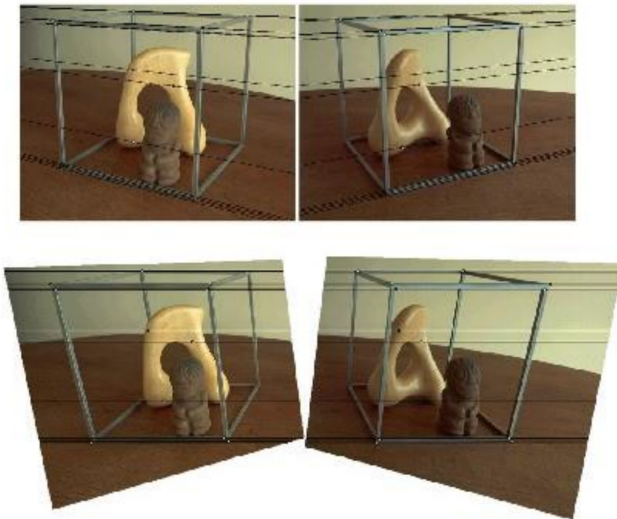
# Rectification

- Once we have  $\mathbf{F}$  we can compute homographies that transform each image plane such that they are parallel (see Zisserman & Hartley's book)
- Once they are parallel, we know how to proceed (matching, etc)



[Source: J. Hays]

# Rectification Example



[Source: J. Hays]

# The Fundamental Matrix: One Last Thing

- Once you have  $F$  you can even compute camera projection matrices  $\mathbf{P}_l$  and  $\mathbf{P}_r$  (under some ambiguity). You may choose the camera projection matrices like this:

$$P_{left} = [I_{3 \times 3} \mid \mathbf{0}] \quad P_{right} = [[\mathbf{e}_r]_{\times} F \mid \mathbf{e}_r]$$

where notation  $[\mathbf{a}]_{\times}$  stands for:  $[\mathbf{a}]_{\times} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$

- This means that I don't need the relative poses of the two cameras, I can compute it!
- This is very useful in scenarios where I just grab pictures from the web

# The Fundamental Matrix: One Last Thing

- Once you have  $F$  you can even compute camera projection matrices  $\mathbf{P}_l$  and  $\mathbf{P}_r$  (under some ambiguity). You may choose the camera projection matrices like this:

$$P_{left} = [I_{3 \times 3} \mid \mathbf{0}] \quad P_{right} = [[\mathbf{e}_r]_{\times} F \mid \mathbf{e}_r]$$

where notation  $[\mathbf{a}]_{\times}$  stands for: 
$$[\mathbf{a}]_{\times} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

- This means that I don't need the relative poses of the two cameras, I can compute it!**
- This is very useful in scenarios where I just grab pictures from the web
- We need one last thing to compute  $P_{right}$ , and that's  $\mathbf{e}_r$ . But this is easy. We know that  $\mathbf{e}_r$  lies on epipolar line  $\mathbf{l}_r$ , and so:  $\mathbf{e}_r^T \mathbf{l}_r = 0$ . We also know that  $\mathbf{l}_r = F \mathbf{x}_l$ . So:  $\mathbf{e}_r^T F \mathbf{x}_l = 0$  for all  $\mathbf{x}_l$ , and therefore  $\mathbf{e}_r^T F = 0$ . So I can find  $\mathbf{e}_r$  as the vector that maps  $F$  to  $\mathbf{0}$ .

# The Fundamental Matrix: One Last Thing

- Once you have  $F$  you can even compute camera projection matrices  $\mathbf{P}_l$  and  $\mathbf{P}_r$  (under some ambiguity). You may choose the camera projection matrices like this:

$$P_{left} = [I_{3 \times 3} \mid \mathbf{0}] \quad P_{right} = [[\mathbf{e}_r]_{\times} F \mid \mathbf{e}_r]$$

where notation  $[\mathbf{a}]_{\times}$  stands for: 
$$[\mathbf{a}]_{\times} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

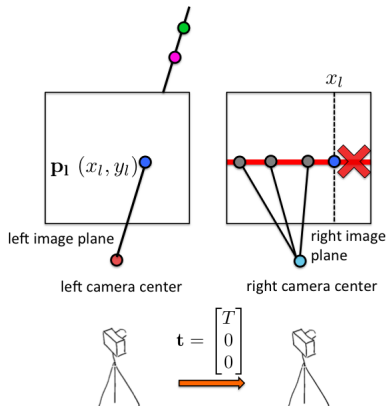
- This means that I don't need the relative poses of the two cameras, I can compute it!**
- This is very useful in scenarios where I just grab pictures from the web
- We need one last thing to compute  $P_{right}$ , and that's  $\mathbf{e}_r$ . But this is easy. We know that  $\mathbf{e}_r$  lies on epipolar line  $\mathbf{l}_r$ , and so:  $\mathbf{e}_r^T \mathbf{l}_r = 0$ . We also know that  $\mathbf{l}_r = F \mathbf{x}_l$ . So:  $\mathbf{e}_r^T F \mathbf{x}_l = 0$  for all  $\mathbf{x}_l$ , and therefore  $\mathbf{e}_r^T F = 0$ . So I can find  $\mathbf{e}_r$  as the vector that maps  $F$  to  $\mathbf{0}$ .

# Stereo: Summary

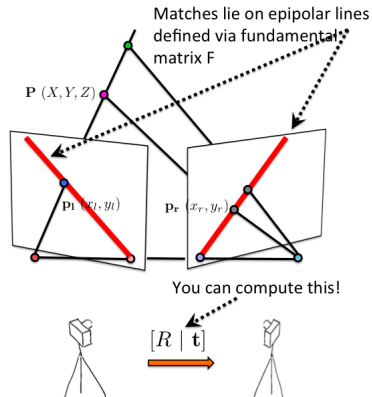
## Epipolar geometry

- Case with two cameras with parallel optical axes
- General case

### Parallel stereo cameras:



### General stereo cameras:



# Summary – Stuff You Need To Know

## Cameras with parallel optics and known intrinsics and extrinsics:

- You can search for correspondences along horizontal lines
- The difference in  $x$  direction between two correspondences is called disparity:

$$\text{disparity} = x_l - x_r$$

- Assuming you know the camera intrinsics and the baseline (distance between the left and right camera center in the world) you can compute the depth:

$$Z = \frac{f \cdot T}{\text{disparity}}$$

- Once you have  $Z$  (depth), you can also compute  $X$  and  $Y$ , giving you full 3D
- Disparity and depth are inversely proportional

### Matlab function:

- `DISPARITYMAP = DISPARITY( $I_{left}$ ,  $I_{right}$ );`
- Function `SURF` is useful for plotting the point cloud

# Summary – Stuff You Need To Know

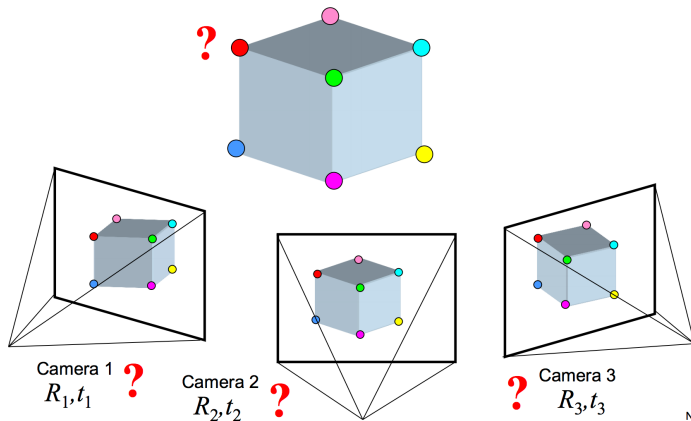
## General cameras:

- You first find matches in both images without any restriction. You need at least 7 matches, but the more (reliable) matches the better
- Solve a homogeneous linear system to get the fundamental matrix  $F$
- Given  $F$ , you can compute homographies that can rectify both images to be parallel.
- Given  $F$ , you can also compute the relative pose between cameras.



# Structure From Motion

- What if you have more than two views of the same scene?
- This problem is called **structure-from-motion**



Slide credit:  
Noah Snavely

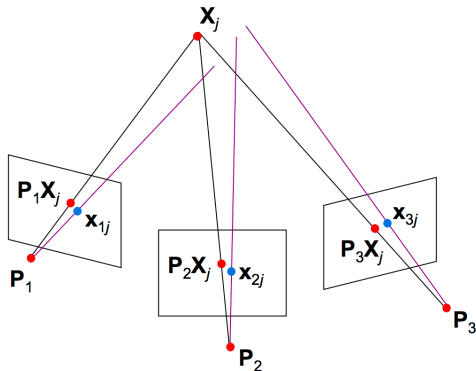
[Source: J. Hays]

# Structure From Motion

- Solve a non-linear optimization problem minimizing re-projection error:

$$E(\mathbf{P}, \mathbf{X}) = \sum_{i=1}^{\#cameras} \sum_{j=1}^{\#points} \text{dist}(\mathbf{x}_{ij}, P_i \mathbf{X}_j)$$

- This can be done via technique called **bundle adjustment**



[Source: J. Hays]

# Lost in Translation



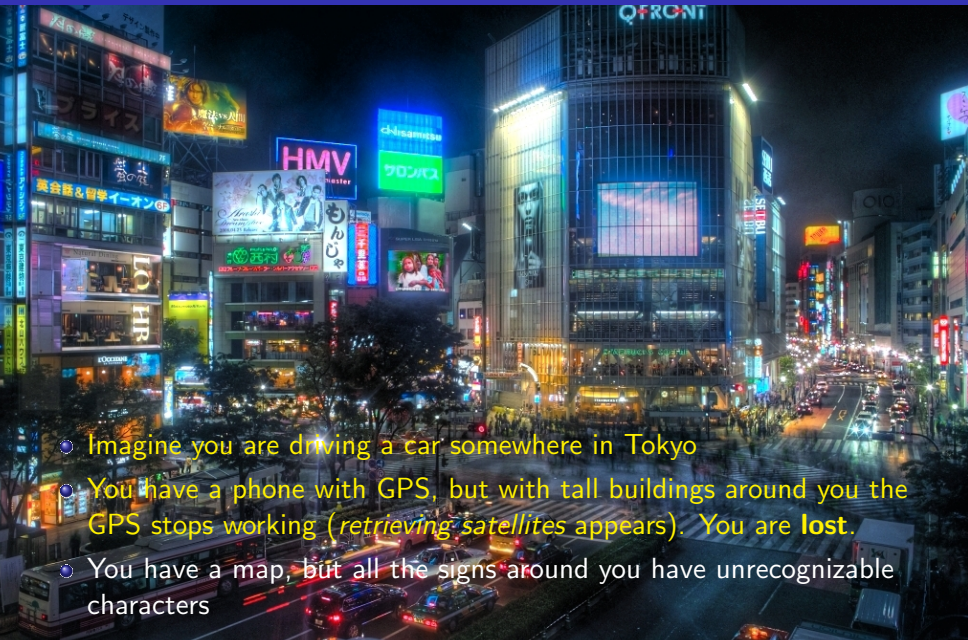
Imagine you are driving a car somewhere in Tokyo

# Lost in Translation



- Imagine you are driving a car somewhere in Tokyo
- You have a phone with GPS, but with tall buildings around you the GPS stops working (*retrieving satellites* appears). You are **lost**.

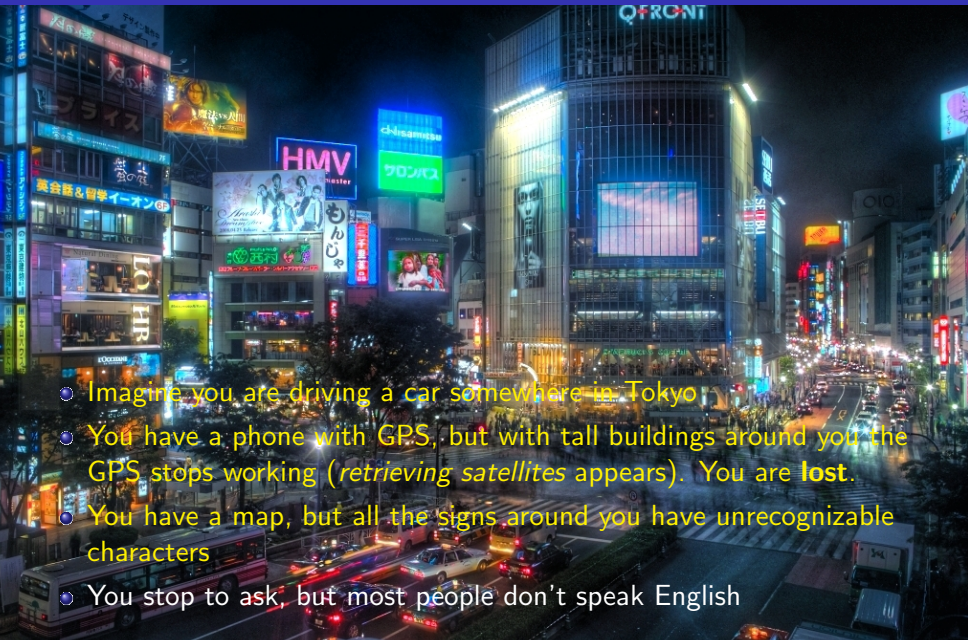
# Lost in Translation



- Imagine you are driving a car somewhere in Tokyo
- You have a phone with GPS, but with tall buildings around you the GPS stops working (*retrieving satellites* appears). You are **lost**.
- You have a map, but all the signs around you have unrecognizable characters



# Lost in Translation



- Imagine you are driving a car somewhere in Tokyo
- You have a phone with GPS, but with tall buildings around you the GPS stops working (*retrieving satellites* appears). You are **lost**.
- You have a map, but all the signs around you have unrecognizable characters
- You stop to ask, but most people don't speak English



## What can you do?

- Imagine you are driving a car somewhere in Tokyo
- You have a phone with GPS, but with tall buildings around you the GPS stops working (*retrieving satellites* appears). You are **lost**.
- You have a map, but all the signs around you have unrecognizable characters
- You stop to ask, but most people don't speak English

Take out your phone, start recording the road and

# Drive!

M. Brubaker, A. Geiger and R. Urtasun

Lost! Leveraging the Crowd for Probabilistic Visual Self-Localization  
CVPR 2013

Paper & Code: <http://www.cs.toronto.edu/~mbrubake/projects/map/>



[M. Brubaker, A. Geiger and R. Urtasun, CVPR13]

- From consecutive frames you can compute relative camera poses
- The recorded video stream therefore gives you a trajectory you are driving

[M. Brubaker, A. Geiger and R. Urtasun, CVPR13]

- From consecutive frames you can compute relative camera poses
- The recorded video stream therefore gives you a trajectory you are driving
- Probabilistic model reasons where you can be on a map given your trajectory

[M. Brubaker, A. Geiger and R. Urtasun, CVPR13]

- From consecutive frames you can compute relative camera poses
- The recorded video stream therefore gives you a trajectory you are driving
- Probabilistic model reasons where you can be on a map given your trajectory

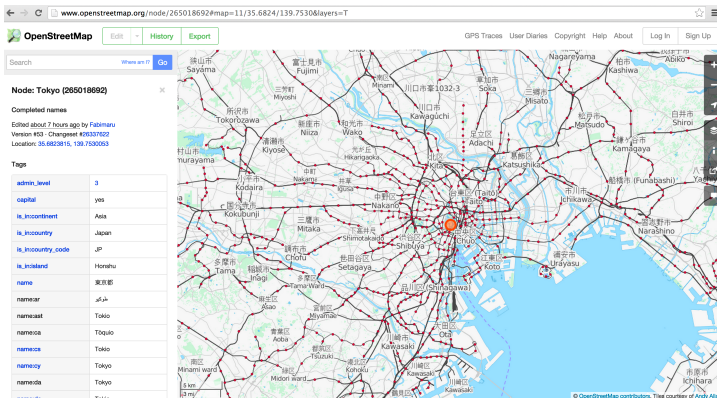


Figure: OpenStreetMap are free downloadable maps (with GPS) of the world

[M. Brubaker, A. Geiger and R. Urtasun, CVPR13]

- From consecutive frames you can compute relative camera poses
- The recorded video stream therefore gives you a trajectory you are driving
- Probabilistic model reasons where you can be on a map given your trajectory

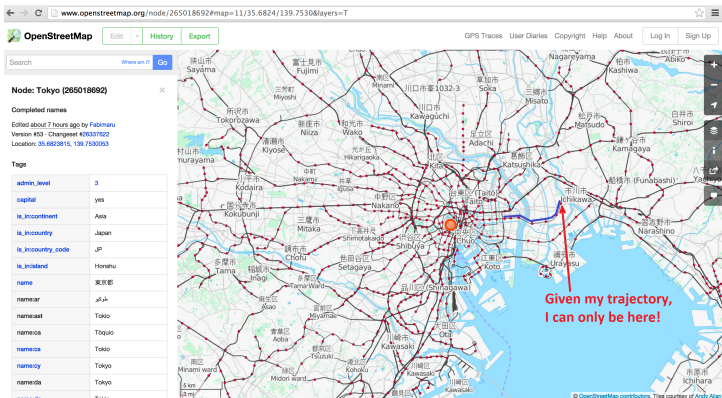


Figure: The shape of my trajectory reveals where I am

[M. Brubaker, A. Geiger and R. Urtasun, CVPR13]

- From consecutive frames you can compute relative camera poses
- The recorded video stream therefore gives you a trajectory you are driving
- Probabilistic model reasons where you can be on a map given your trajectory
- This gives you the GPS location!
- With 1 camera up to 18m accuracy, 2 cameras up to 3m accuracy

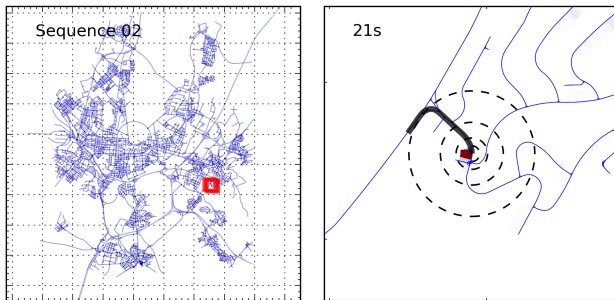


Figure: <https://www.youtube.com/watch?v=4Z3shNP0dQA&feature=youtu.be>

# Vision for Visually Impaired

- You can imagine a more complex version of the system for visually impaired



Pic from: <http://www.blogcdn.com/www.engadget.com/media/2012/05/wxzfdgrs.jpg>

# Vision for Visually Impaired

- You can imagine a more complex version of the system for visually impaired
- How else could depth / 3D help me?



Pic from: <http://www.blogcdn.com/www.engadget.com/media/2012/05/wxzfdgrs.jpg>

# Vision for Visually Impaired

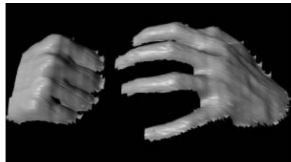
- You can imagine a more complex version of the system for visually impaired
- How else could depth / 3D help me?
- What else do we need to solve to make a vision system for visually impaired functional?



Pic from: <http://www.blogcdn.com/www.engadget.com/media/2012/05/wxzfdgrs.jpg>

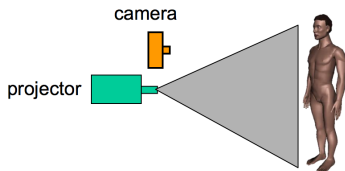


# Another Way to get Stereo: Stereo with Structured Light



Project “structured” light patterns onto the object

- Simplifies the correspondence problem
- Allows us to use only one camera



L. Zhang, B. Curless, and S. M. Seitz. Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming. 3DPVT 2002

[Source: J. Hays]

# Kinect: Structured infrared light



Figure: <https://www.youtube.com/watch?v=uq9SEJxZiUg>

[Source: J. Hays]

# Stereo Vision in the Wild

- Humans and a lot of animals (particularly cute ones) have stereoscopic vision



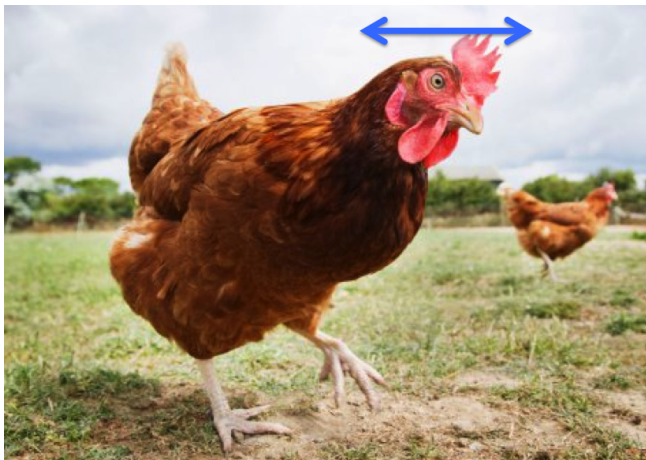
# Stereo Vision in the Wild

- Most birds don't see in stereo (each eye gets its own picture, no overlap)
- How do these animals get depth? E.g., how can a chicken peck the corn without smashing the head against the floor?



# Stereo Vision in the Wild

- Most birds don't see in stereo (each eye gets its own picture, no overlap)
- How do these animals get depth? E.g., how can a chicken peck the corn without smashing the head against the floor? **Structure-from-motion**



# Stereo Vision in the Wild

- Owls are one of the exceptions (they see stereo)



# Birdseye View on What We Learned So Far

<b>Problem</b>	<b>Detection</b>	<b>Description</b>	<b>Matching</b>
Find Planar Distinctive Objects	Scale Invariant Interest Points	Local feature: SIFT	All features to all features + Affine / Homography
Panorama Stitching	Scale Invariant Interest Points	Local feature: SIFT	All features to all features + Homography
Stereo	Compute in every point	Intensity or Gradient patch	For each point search on epipolar line

# Towards Semantics

- 3D and Projective Geometry can explain a lot of things in the image.
- However, some of the most valuable images cannot be explained by 3D at all.



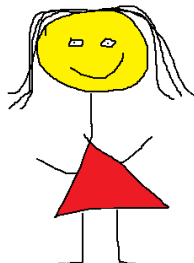
# Towards Semantics

- 3D and Projective Geometry can explain a lot of things in the image.
- However, some of the most valuable images cannot be explained by 3D at all.



100 million \$

“Dora Maar au Chat”  
Pablo Picasso, 1941



1 cent

“La Picture”  
Sanja Fidler, yesterday

[Adopted from: A. Torralba]

# Towards Semantics

- We shouldn't only look at the 3D behind the image but also at the **story** behind it.
- We need to also understand the image **semantics**.



# It's Fine Without Depth Too



[https://www.youtube.com/watch?v=\\_dPlkFPowCc](https://www.youtube.com/watch?v=_dPlkFPowCc)

# It's Fine Without Depth Too

- Chickens don't want depth, they want story ;)



[https://www.youtube.com/watch?v=\\_dPlkFPowCc](https://www.youtube.com/watch?v=_dPlkFPowCc)

# Next Time: Recognition