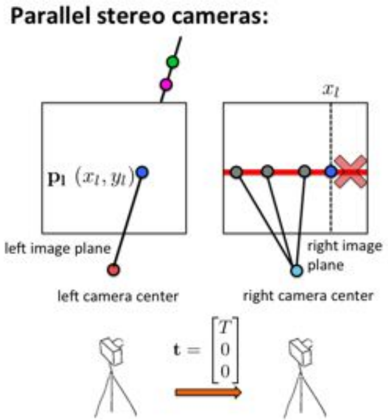
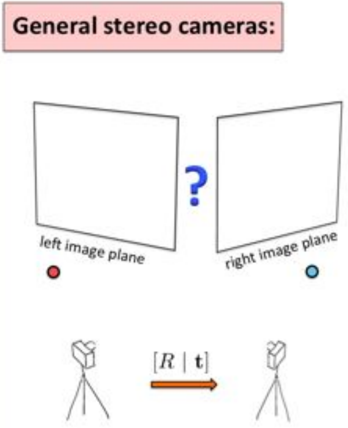
# Stereo

# Epipolar Geometry for General Cameras

**Epipolar geometry**

- Case with two cameras with parallel optical axes
- General case $\leftarrow$ **Now this**



# 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/`

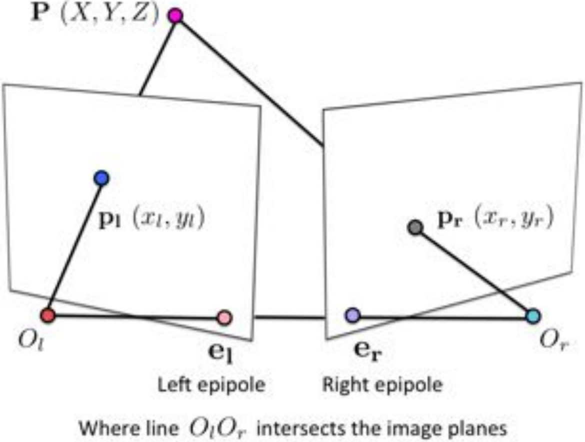
# World Cup 2014 – High Tech 3D

- 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?

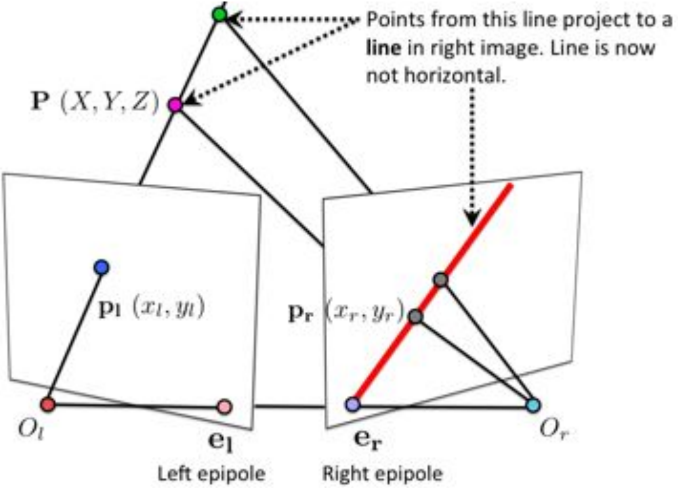
# Stereo: Parallel Calibrated Cameras

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

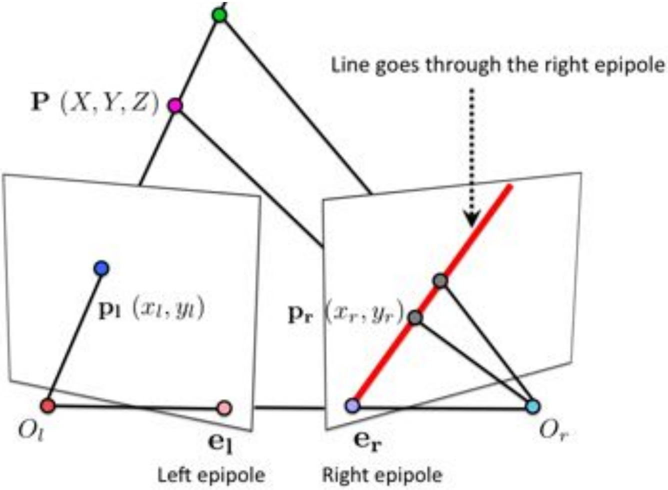


# Stereo: Parallel Calibrated Cameras

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

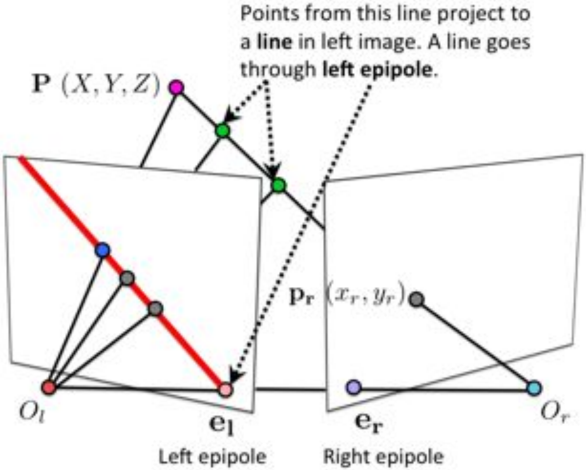


- The line goes through the right epipole.



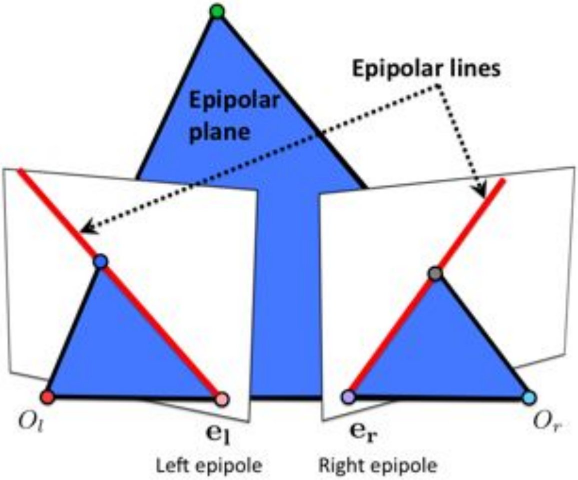
# Stereo: Parallel Calibrated Cameras

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



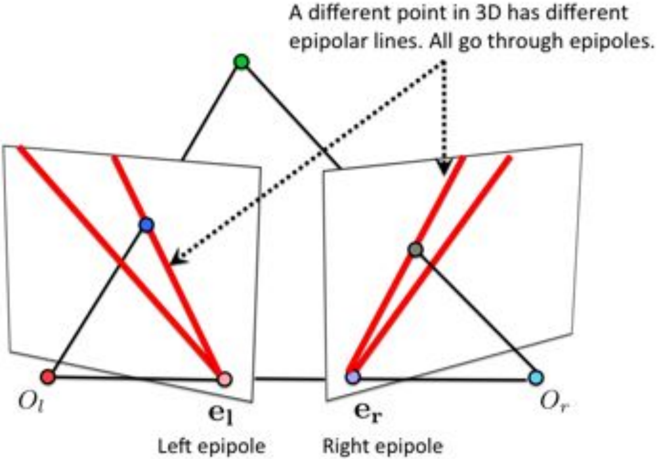
# Stereo: Parallel Calibrated Cameras

- The reason for all this is simple: points $\mathbf{O_l}$, $\mathbf{O_r}$, and a point $\mathbf{P}$ 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.

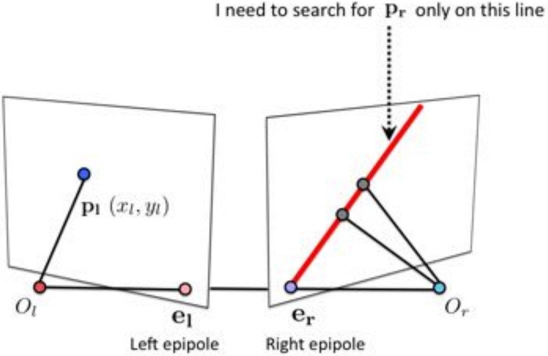


# Stereo: Parallel Calibrated Cameras

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

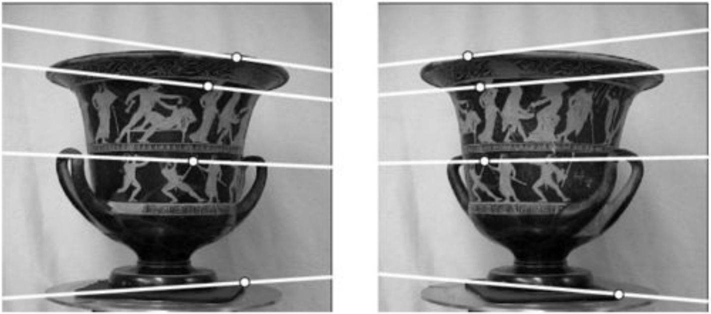
# Stereo: Parallel Calibrated Cameras

- 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]

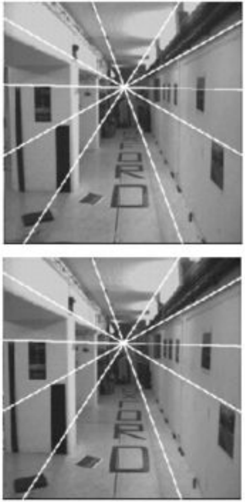
# Epipolar geometry: Examples

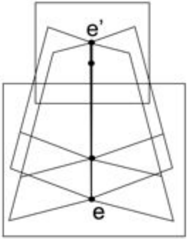
- How would epipolar lines look like if the camera moves directly forward?

[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

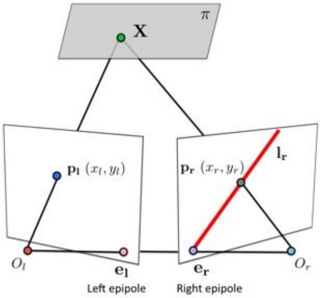
# The Fundamental Matrix

- The fundamental matrix $\mathbf{F}$ 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 F**.

# The Fundamental Matrix

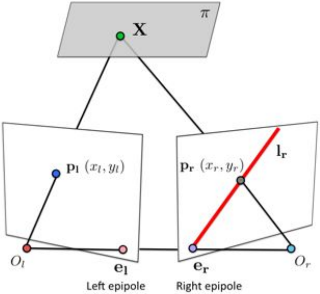
- The fundamental matrix $\mathbf{F}$ is defined as $\mathbf{l_r} = \mathbf{F}\mathbf{p_l}$, where $\mathbf{l_r}$ is the right epipolar line corresponding to $\mathbf{p_l}$.
- $\mathbf{F}$ is a $3 \times 3$ matrix
- For any point $\mathbf{p_l}$ its epipolar line is defined by the **same matrix F**.

# The Fundamental Matrix



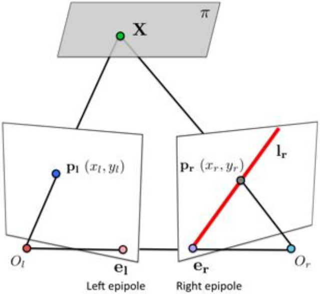

- Extend the line $\mathbf{O_l p_l}$ until you hit a plane $\pi$ (arbitrary)
- Find the image $\mathbf{p_r}$ of $\mathbf{X}$ in the right camera

# The Fundamental Matrix



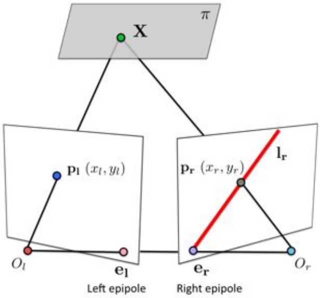

- Extend the line $\mathbf{O_l p_l}$ until you hit a plane $\pi$ (arbitrary)
- Find the image $\mathbf{p_r}$ of $\mathbf{X}$ in the right camera
- Get epipolar line $\mathbf{l_r}$ from $\mathbf{e_r}$ to $\mathbf{p_r}$: $\mathbf{l_r} = \mathbf{e_r} \times \mathbf{p_r}$

# The Fundamental Matrix



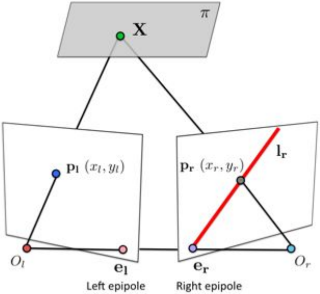

- Extend the line $\mathbf{O_l p_l}$ until you hit a plane $\pi$ (arbitrary)
- Find the image $\mathbf{p_r}$ of $\mathbf{X}$ in the right camera
- Get epipolar line $\mathbf{l_r}$ from $\mathbf{e_r}$ to $\mathbf{p_r}$: $\mathbf{l_r} = \mathbf{e_r} \times \mathbf{p_r}$
- Points $\mathbf{p_l}$ and $\mathbf{p_l}$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

# The Fundamental Matrix




- Extend the line $\mathbf{O_l p_l}$ 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 $\mathbf{p_l}$ are related via homography: $\mathbf{p_r} = H_\pi \mathbf{p_l}$
- Then: $\mathbf{l_r} = \mathbf{e_r} \times \mathbf{p_r} = \mathbf{e_r} \times H_\pi \mathbf{p_l} = \mathbf{F} \mathbf{p_l}$

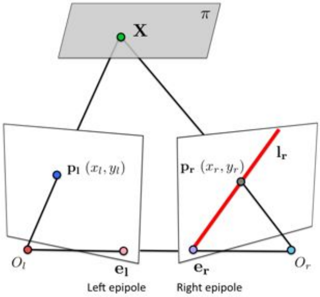
# The Fundamental Matrix




- Extend the line $\mathbf{O_l p_l}$ 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 $\mathbf{p_l}$ 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 $\mathbf{O_l p_l}$ 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 $\mathbf{p_l}$ 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 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 $\mathbf{F}$ 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 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 F**.
- So:

$$\boxed{\mathbf{p_r}^T \mathbf{F} \mathbf{p_l} = 0}$$

  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 F**.
- So:

$$\boxed{\mathbf{p_r}^T \mathbf{F} \mathbf{p_l} = 0}$$

  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 F**.
- So:

$$\boxed{\mathbf{p_r}^T \mathbf{F} \mathbf{p_l} = 0}$$

  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: $(x_{l,1}, y_{l,1}) \leftrightarrow (x_{r,1}, y_{r,1}), \ldots, (x_{l,n}, y_{l,n}) \leftrightarrow (x_{r,n}, y_{r,n})$, where $n \geq 7$
- Then you can get the parameters $\mathbf{f} := [F_{11}, F_{12}, \ldots, F_{33}]$ 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?
- **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: $(x_{l,1}, y_{l,1}) \leftrightarrow (x_{r,1}, y_{r,1})$, ..., $(x_{l,n}, y_{l,n}) \leftrightarrow (x_{r,n}, y_{r,n})$, where $n \geq 7$
- Then you can get the parameters $\mathbf{f} := [F_{11}, F_{12}, \ldots, F_{33}]$ 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?
- **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: $(x_{l,1}, y_{l,1}) \leftrightarrow (x_{r,1}, y_{r,1}), \ldots, (x_{l,n}, y_{l,n}) \leftrightarrow (x_{r,n}, y_{r,n})$, where $n \geq 7$
- Then you can get the parameters $\mathbf{f} := [F_{11}, F_{12}, \ldots, F_{33}]$ 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?
- **F** has 9 elements, but we don't care about scaling, so 8 elements.
- Turns out it really only has 7.
- We can estimate **F** with 7 correspondences. Of course, the more the better (why?).
- See Zisserman & Hartley's book for details.

# The Fundamental Matrix

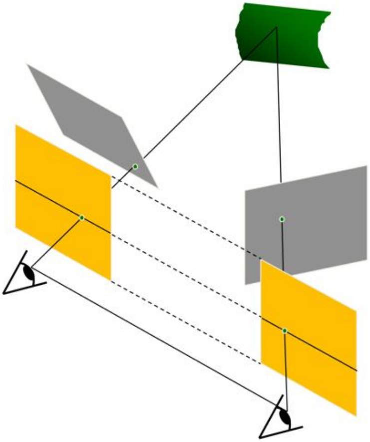
- Let's say that you found a few matching points in both images: $(x_{l,1}, y_{l,1}) \leftrightarrow (x_{r,1}, y_{r,1}), \ldots, (x_{l,n}, y_{l,n}) \leftrightarrow (x_{r,n}, y_{r,n})$, where $n \geq 7$
- Then you can get the parameters $\mathbf{f} := [F_{11}, F_{12}, \ldots, F_{33}]$ by solving:

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

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

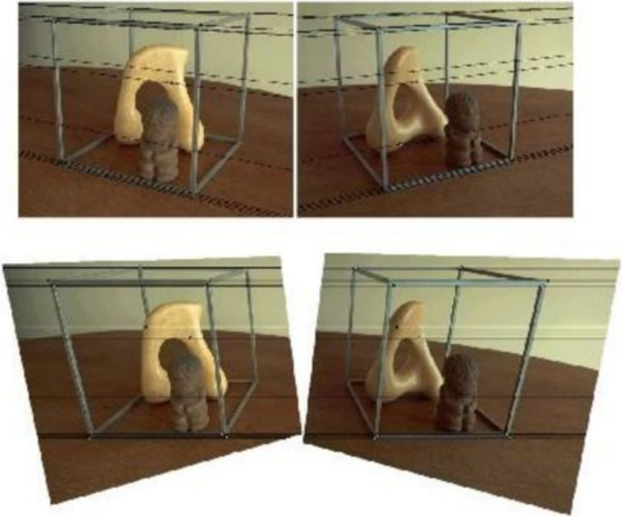
# Rectification

- Once we have **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:

$$P_{left} = [\, I_{3\times 3} \,|\, \mathbf{0} \,] \qquad P_{right} = \left[\, [\mathbf{e_r}]_x F \,|\, \mathbf{e_r} \,\right]$$

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

- **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:

$$P_{left} = [\, I_{3\times 3} \,|\, \mathbf{0} \,] \qquad P_{right} = \left[\, [\mathbf{e_r}]_x \, F \,|\, \mathbf{e_r} \,\right]$$

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

- **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: $\mathbf{e_r}^T F \mathbf{x_l} = \mathbf{0}$ for all $\mathbf{x_l}$, and therefore $\mathbf{e_r}^T F = \mathbf{0}$. 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:

$$P_{left} = [\, I_{3\times 3} \,|\, \mathbf{0} \,] \qquad P_{right} = \left[\, [\mathbf{e_r}]_x \, F \,|\, \mathbf{e_r} \,\right]$$

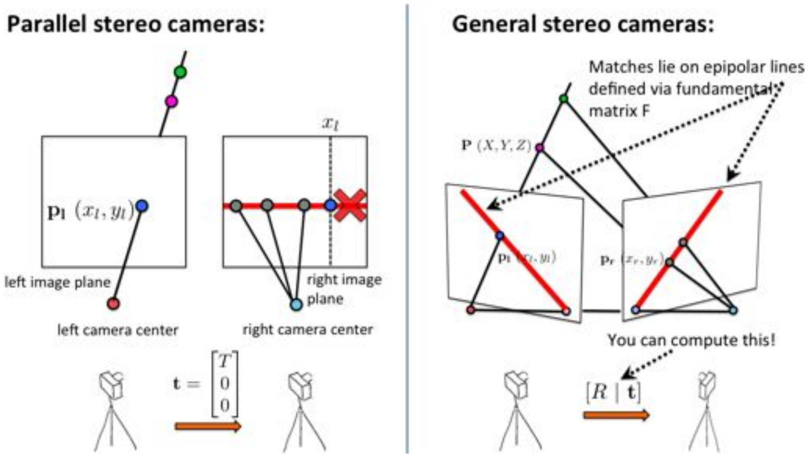
where notation $[]_x$ stands for: $[\mathbf{a}]_x = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$

- **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: $\mathbf{e_r}^T F \mathbf{x_l} = \mathbf{0}$ for all $\mathbf{x_l}$, and therefore $\mathbf{e_r}^T F = \mathbf{0}$. 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



# 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 canter 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:**

- $\textsc{disparityMap} = \textsc{disparity}(I_{left}, I_{right});$
- Function SURF is useful for plotting the point cloud

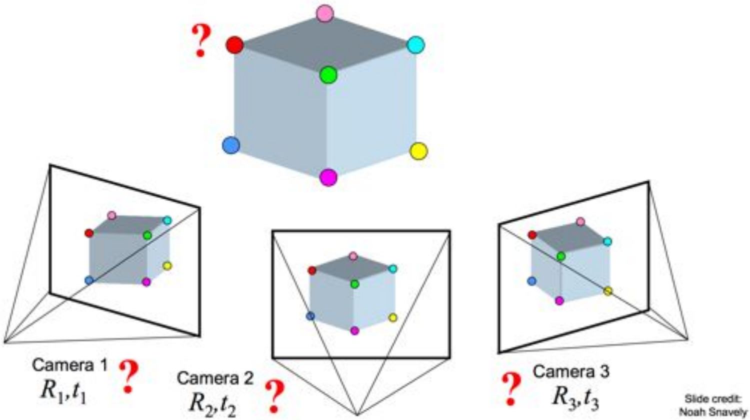
# Summary – Stuff You Need To Know

**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**



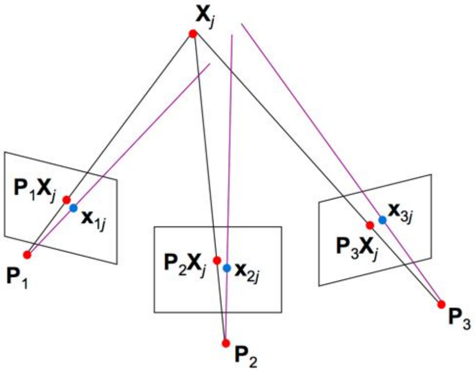

[Source: J. Hays]

# Structure From Motion

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

$$E(\mathbf{P}, \mathbf{X}) = \sum_{i=1}^{\#cameras} \sum_{j=1}^{\#points} \mathrm{dist}(\mathbf{x_{ij}}, P_i X_j)$$

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




[Source: J. Hays]

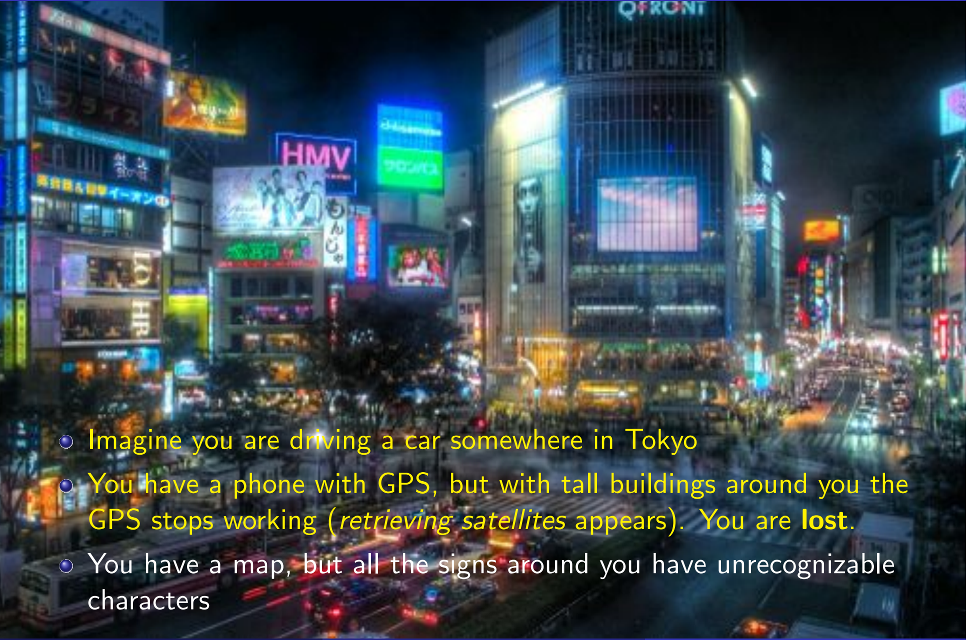
# Lost in Translation


- Imagine you are driving a car somewhere in Tokyo

# Lost in Translation

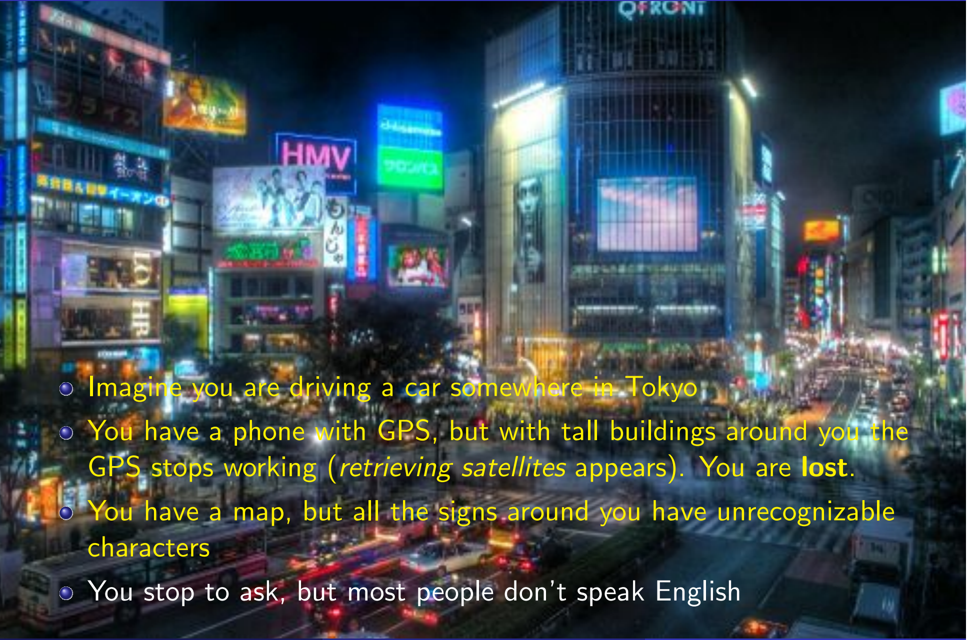

- Imagine you are driving a car somewhere in Tokyo
- You have a phone with GPS, but with tall buildings around you the GPS stops working (*retrieving satellites* appears). You are **lost**.

# 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/`

# Lost

[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

# Lost

[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

# Lost

[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

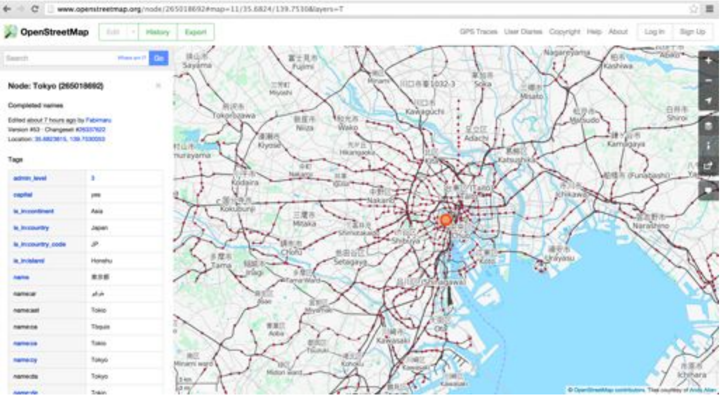



Figure: OpenStreetMap are free downloadable maps (with GPS) of the world

# Lost

[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

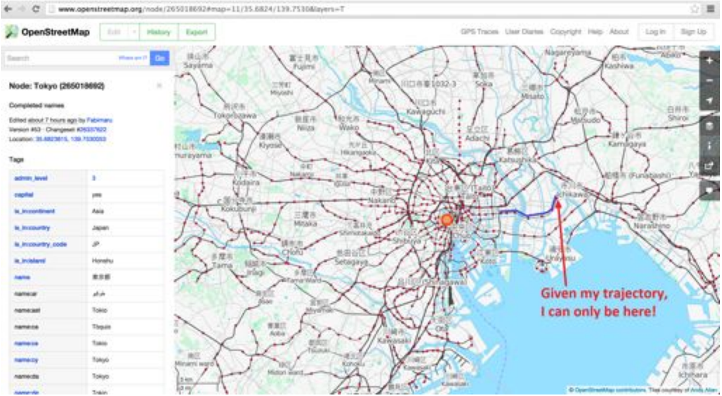
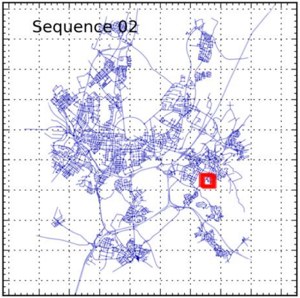



Figure: The shape of my trajectory reveals where I am

# Lost

[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



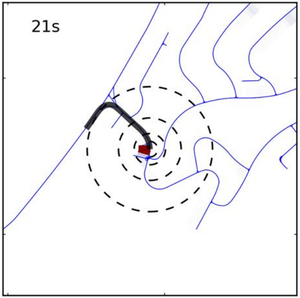




Figure: https://www.youtube.com/watch?v=4Z3shNPOdQA&feature=youtu.be

# 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?

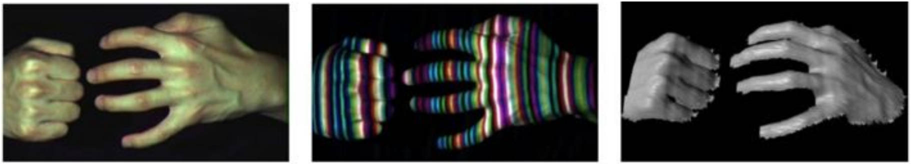
Pic from: `http://www.blogcdn.com/www.engadget.com/media/2012/05/wxzfdgrs.jpg`

# 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?

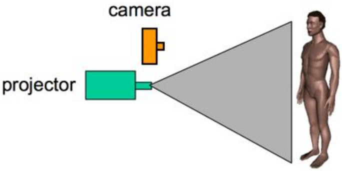
Pic from: http://www.blogcdn.com/www.engadget.com/media/2012/05/wxzfdgrs.jpg

# 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

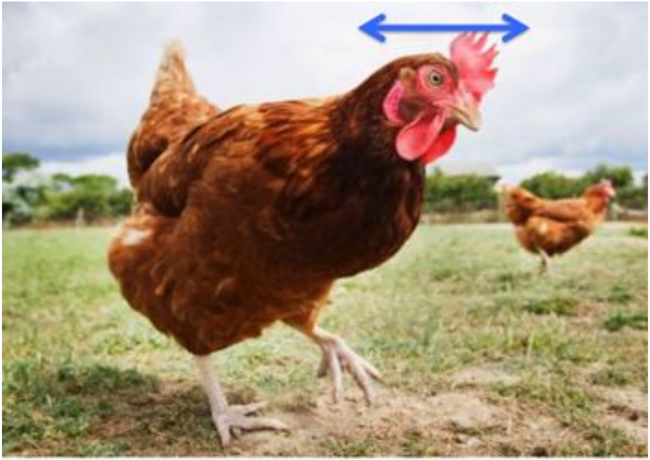
- Humans and a lot of animals (particularly cute ones) have stereoscopic vision

# 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 beak 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 beak 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

| **Problem** | **Detection** | **Description** | **Matching** |
|---|---|---|---|
| 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.

- 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.


"Dora Maar au Chat"
Pablo Picasso, 1941


"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

# It's Fine Without Depth Too

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

https://www.youtube.com/watch?v=_dPlkFPowCc

Next Time:

# Recognition