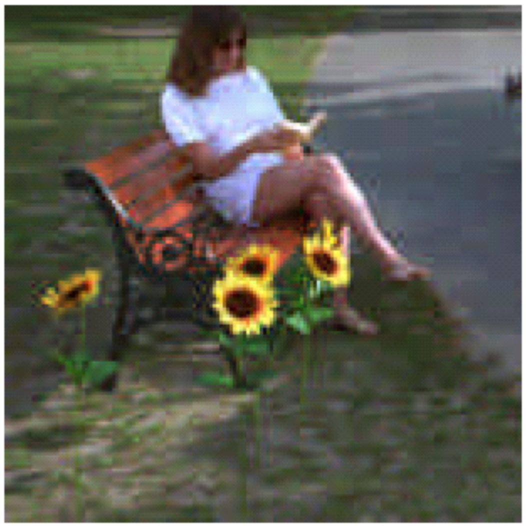
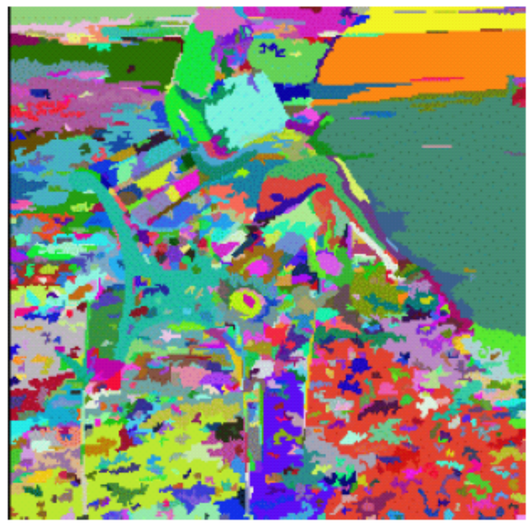
CS 3630

Frank Dellaert, Spring 14


Dense Stereo

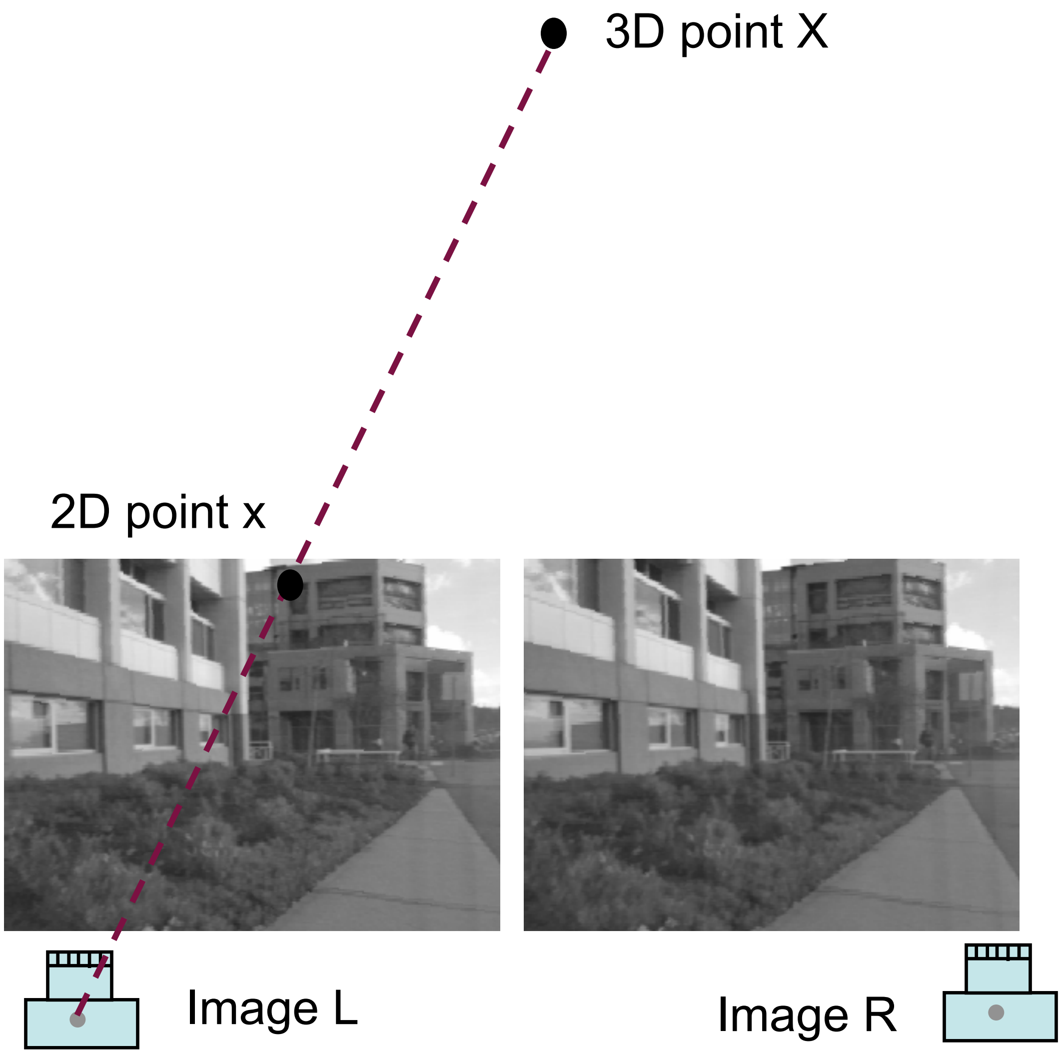
Some Slides by Forsyth & Ponce,
Jim Rehg, **Sing Bing Kang**

Etymology

Stereo comes from the Greek word for *solid* (στερεό), and the term can be applied to any system using more than one channel

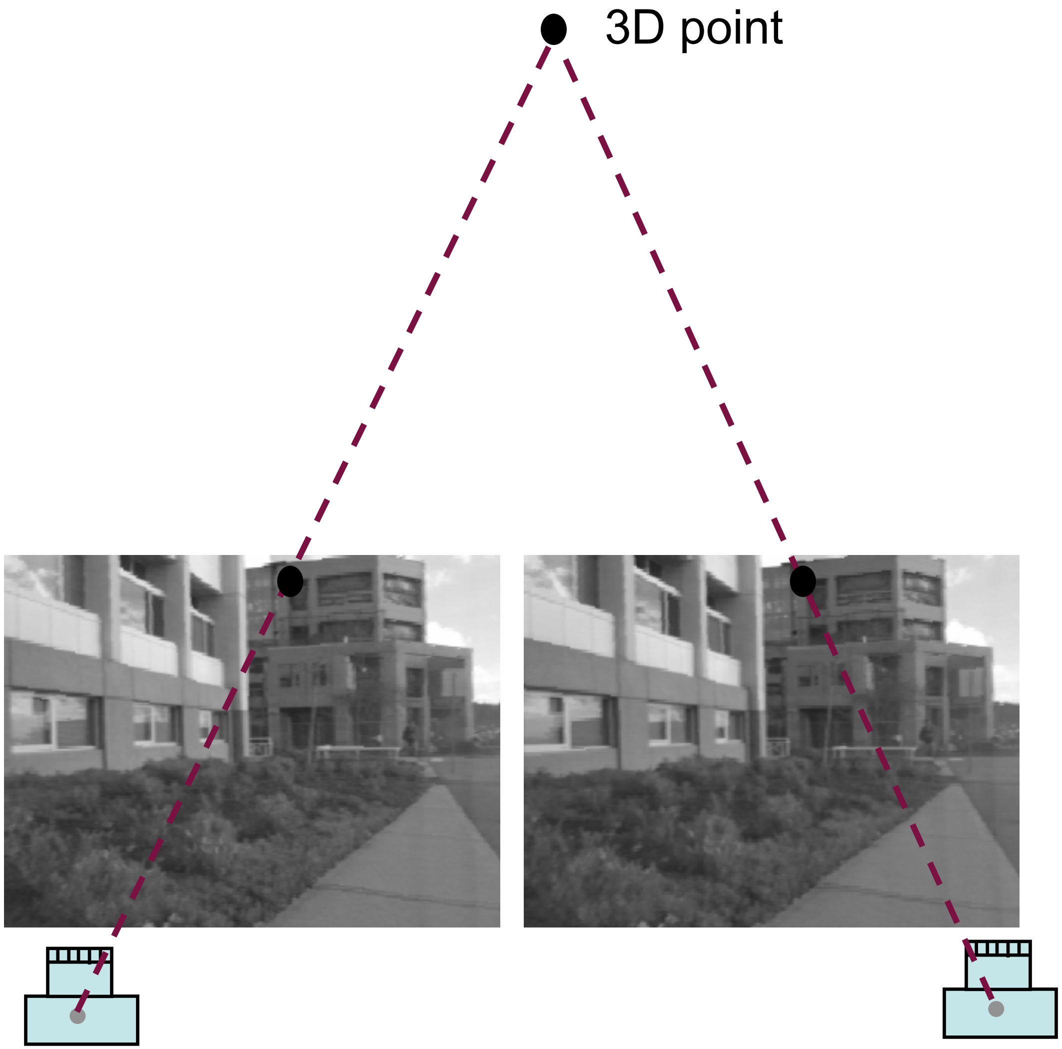
Effect of Moving Camera ?

- Given a point x in image L, where can x' appear in image R?
- Assume camera R is *exactly* to the right of camera L (stereo rig)

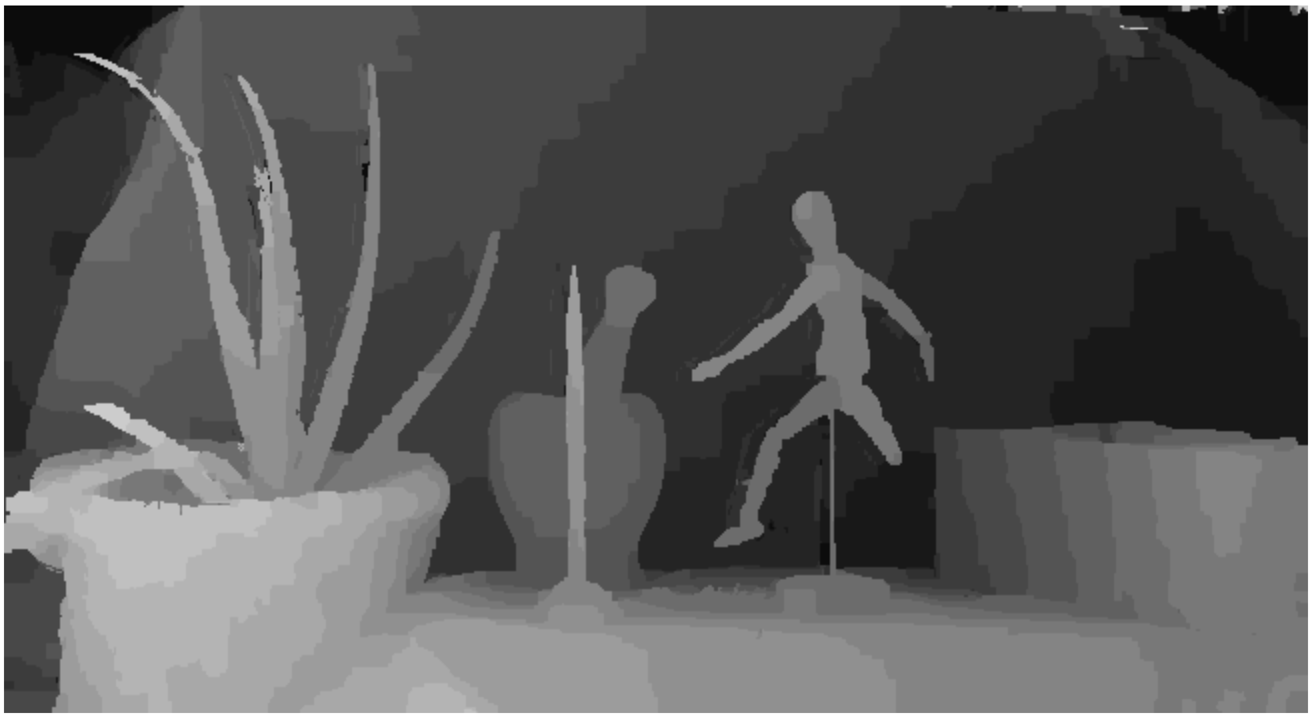


Effect of Moving Camera

- As camera is shifted (viewpoint changed):
  - 3D points are projected to different 2D locations
  - Amount of shift in projected 2D location depends on depth
- 2D shifts=Parallax



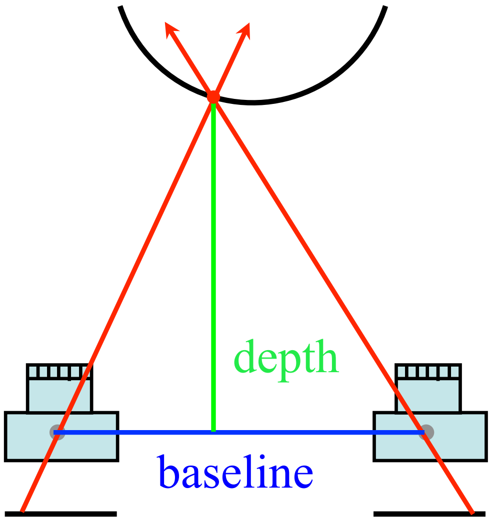
Demo



Right image
Left image
Disparity

View Interpolation

Basic Idea of Stereo



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

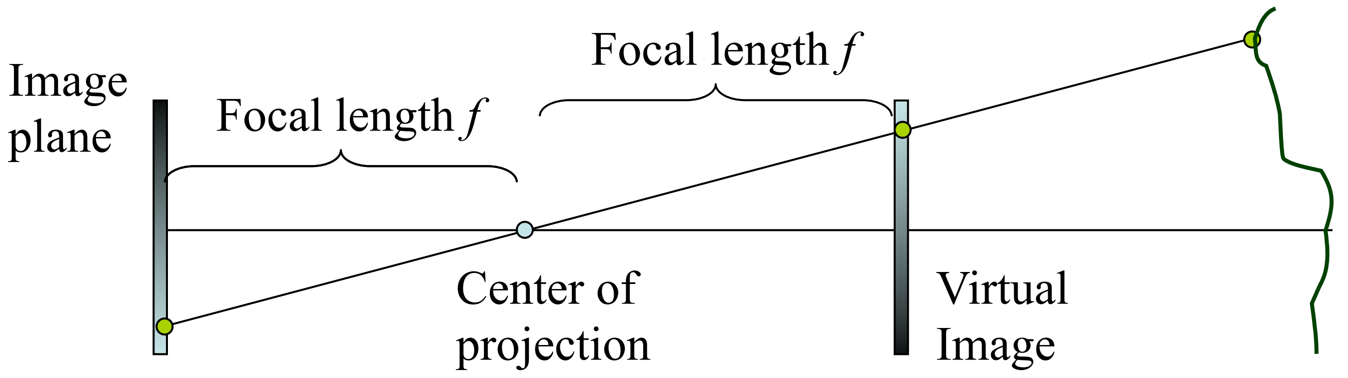
- Feature matching across views

Matching correlation windows across scan lines

Outline

- Pinhole camera model
- Basic stereo Equations
- Stereo Correspondence

Pinhole Camera Model



In actual image plane, scene appears inverted.
In virtual image, scene appears right side up.
For expediency, use virtual image for analysis.

Pinhole Camera Model

3D scene point P is projected to a 3D point Q in the virtual image plane

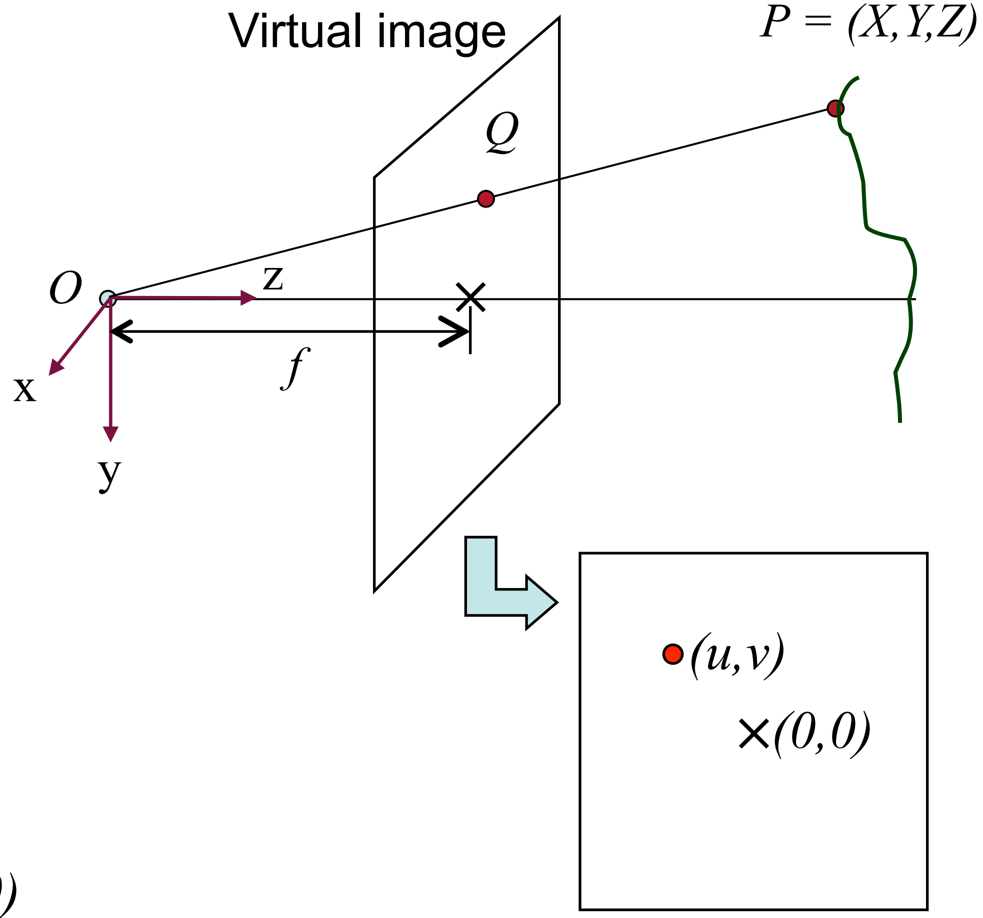
By simply rescaling:

$$Q = \left(f\frac{X}{Z}, f\frac{Y}{Z}, f\right)$$

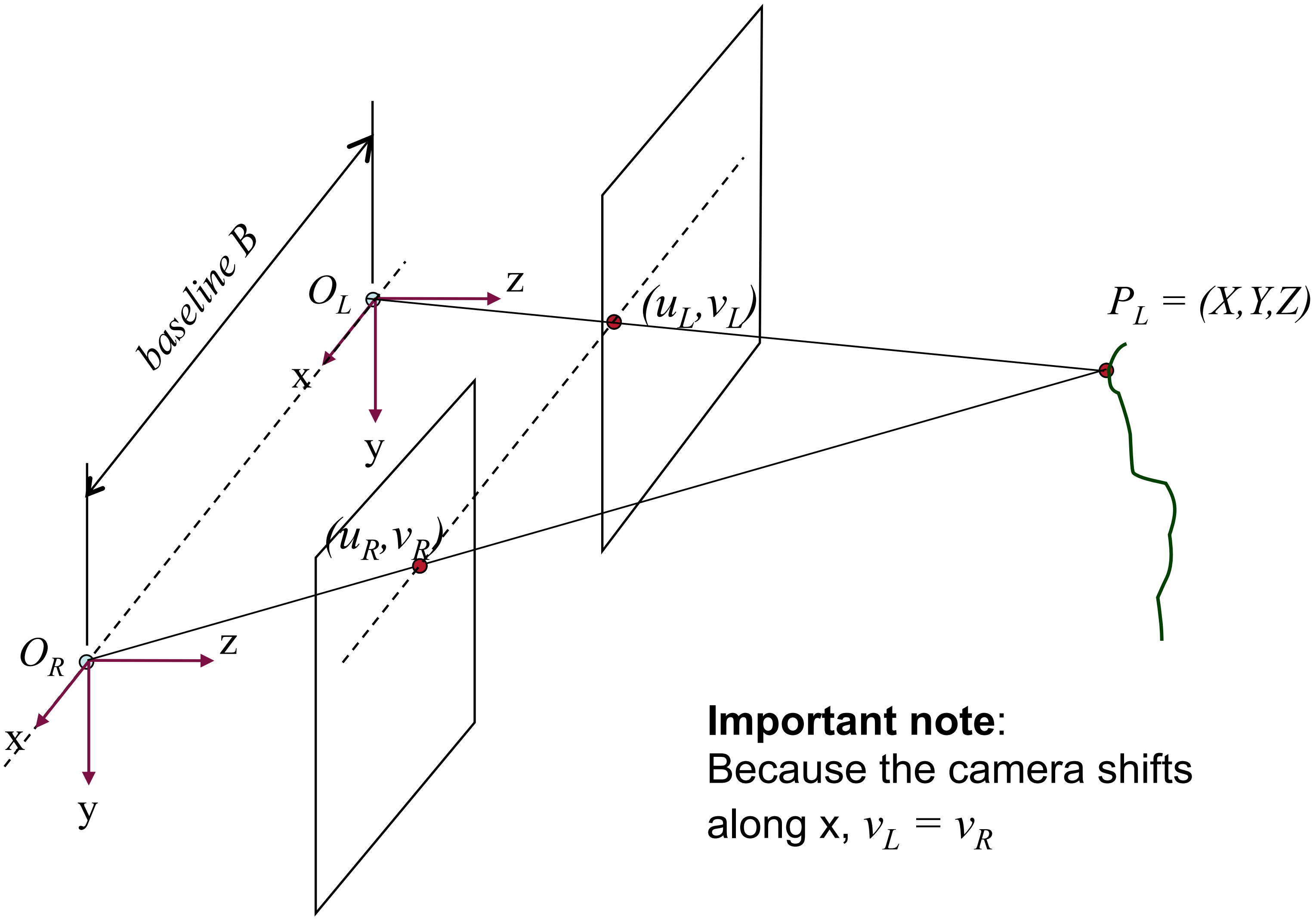
Hence, the 2D coordinates in the virtual image is given by

$$(u, v) = \left(f\frac{X}{Z}, f\frac{Y}{Z}\right)$$

Note: image center is *(0,0)*

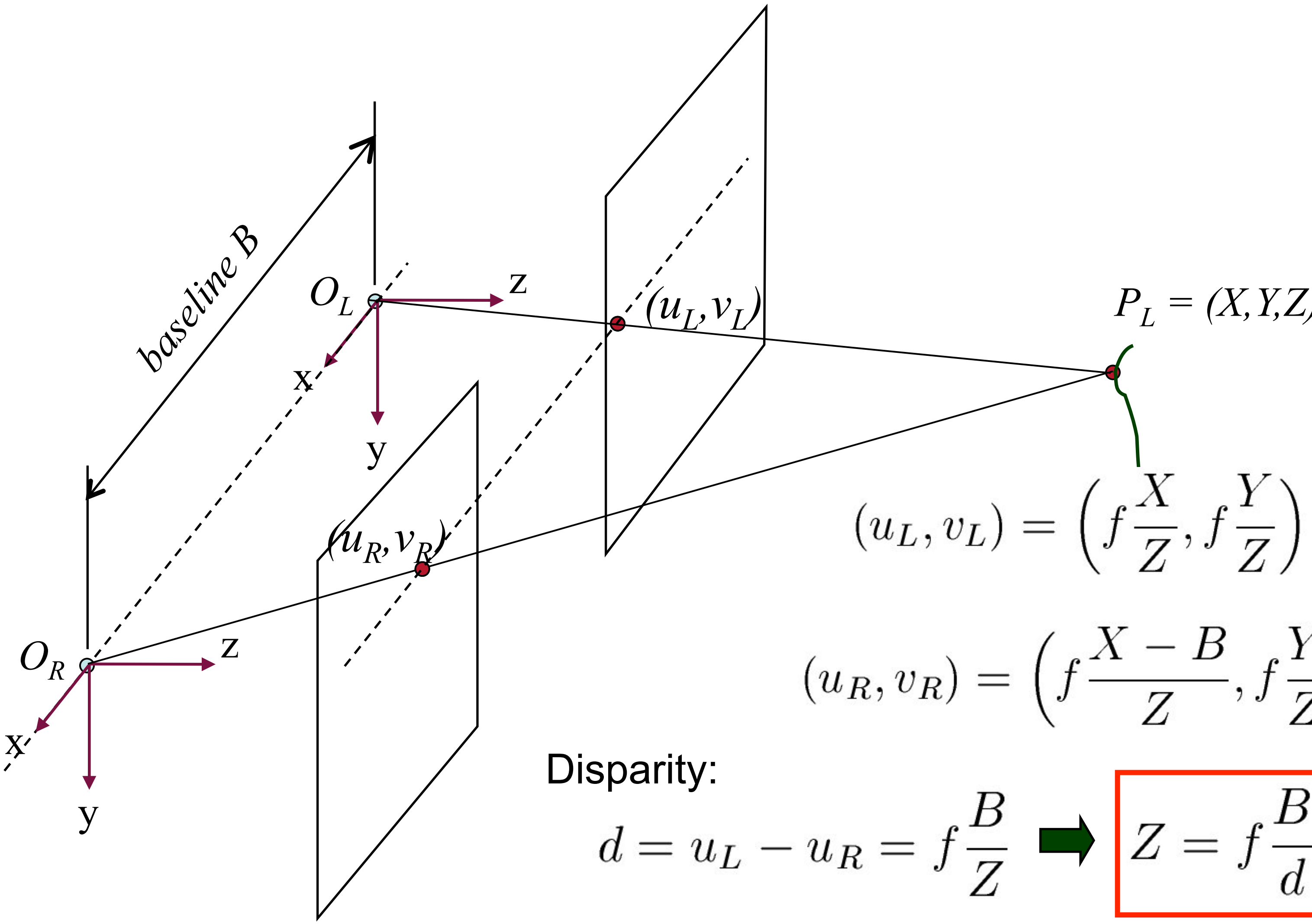


Basic Stereo Derivations



Important note:
Because the camera shifts along x, $v_L = v_R$

Basic Stereo Derivations



$$(u_L, v_L) = \left(f\frac{X}{Z}, f\frac{Y}{Z} \right)$$

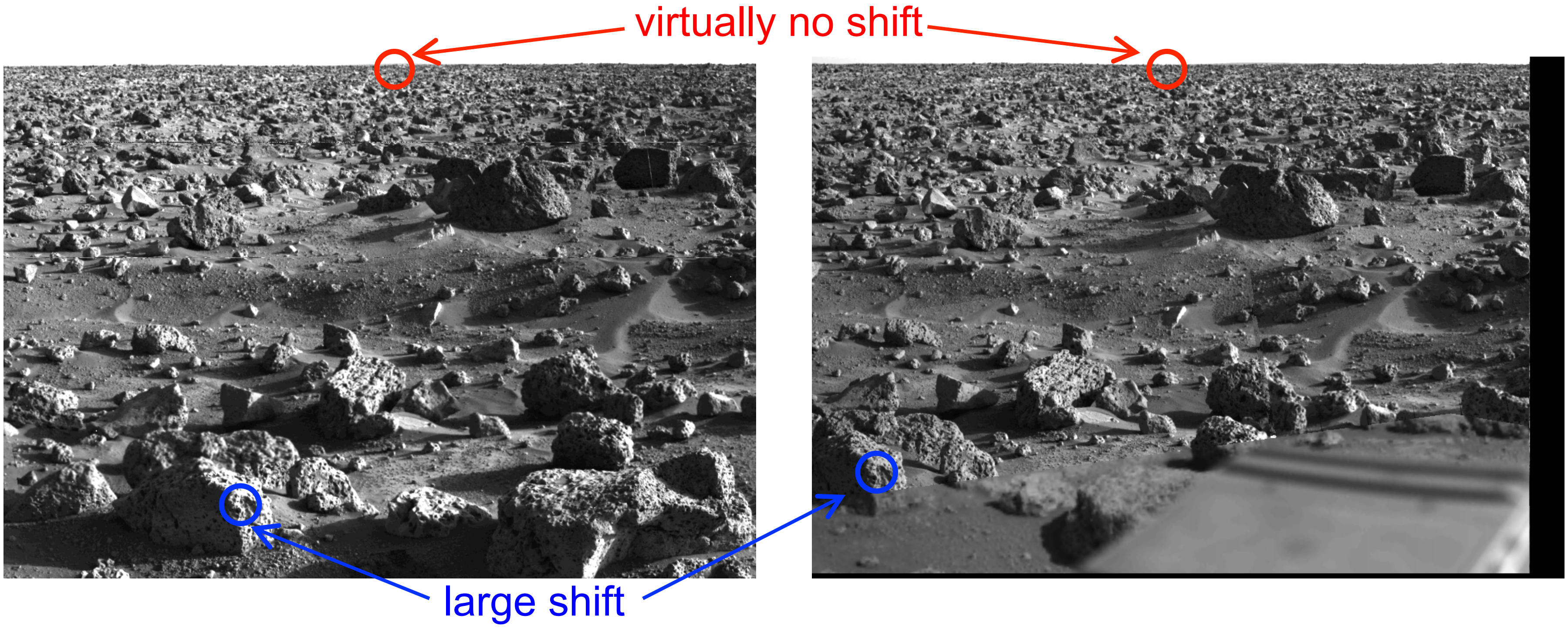
$$(u_R, v_R) = \left(f\frac{X - B}{Z}, f\frac{Y}{Z} \right)$$

Disparity:

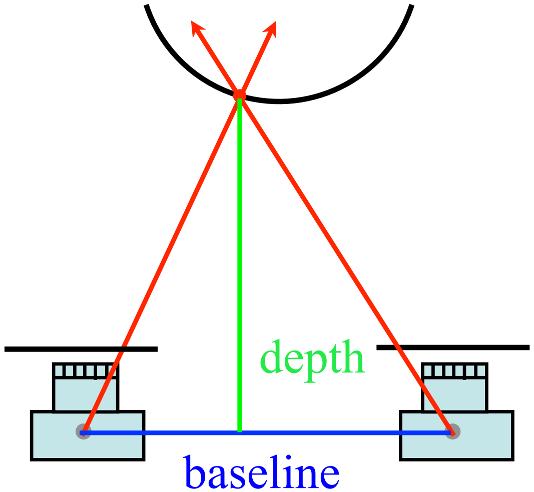
$$d = u_L - u_R = f\frac{B}{Z} \Rightarrow \boxed{Z = f\frac{B}{d}}$$

Stereo Correspondence

- Search over disparity to find correspondences
- Range of disparities can be large



Stereo Vision

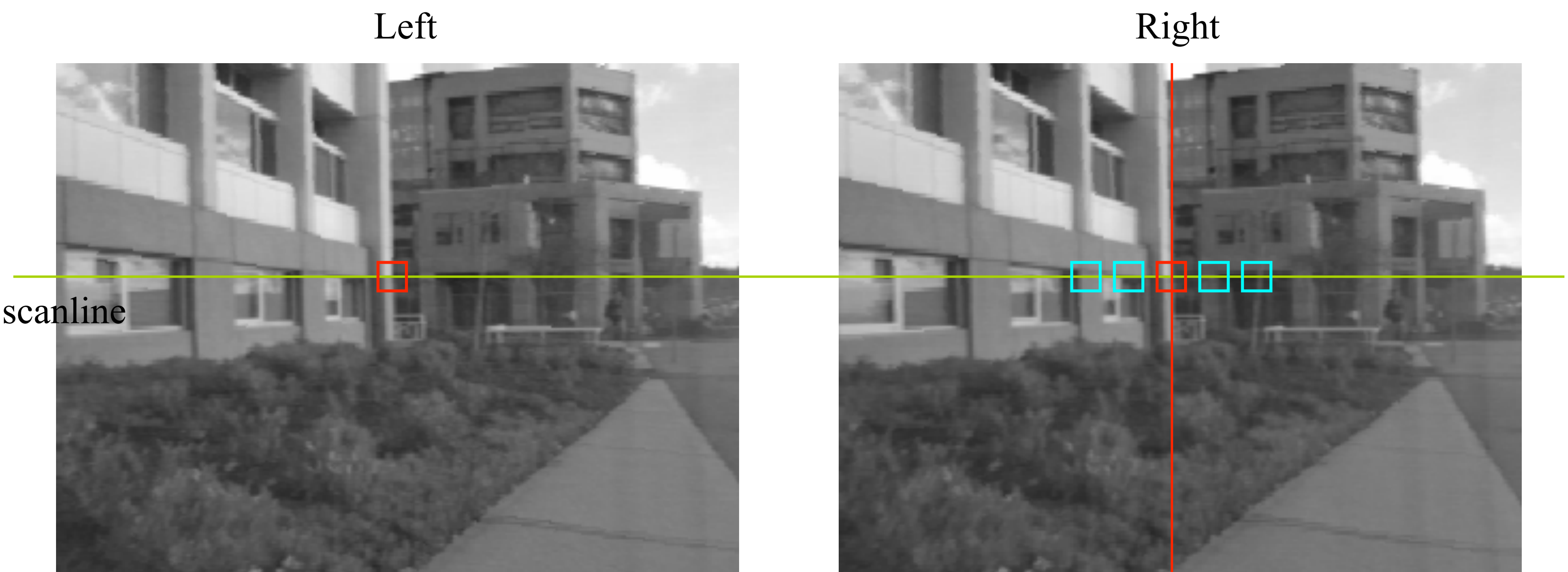


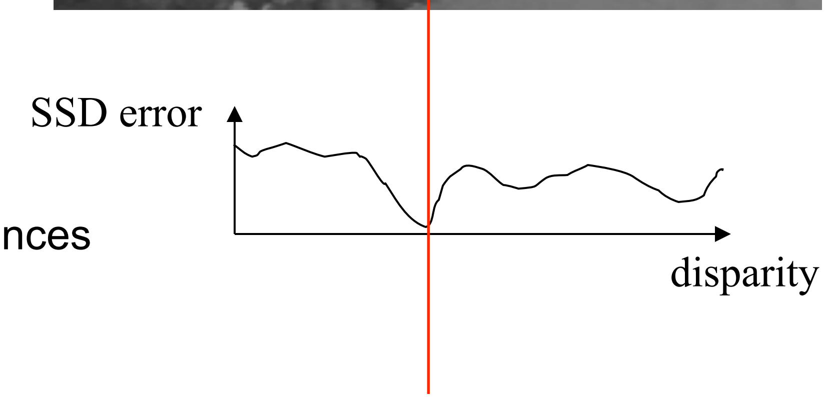
$$Z(x,y) = \frac{f\,B}{d(x,y)}$$

$Z(x, y)$ is depth at pixel (x, y)
$d(x, y)$ is disparity

Matching correlation windows across scan lines

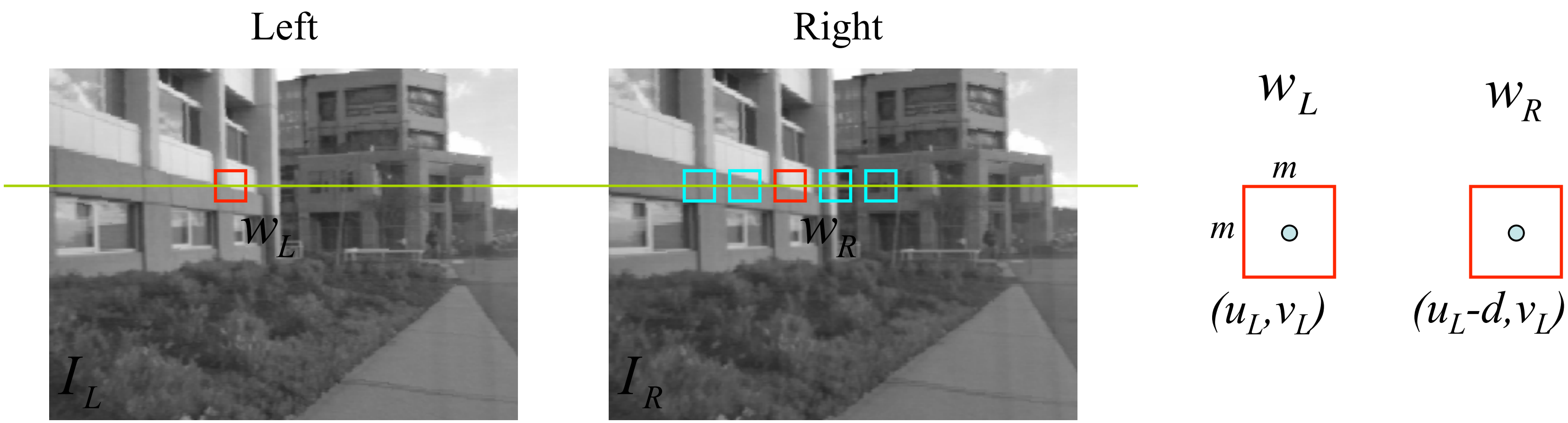
Correspondence Using Window-based Correlation





Matching criterion = Sum-of-squared differences

Sum of Squared (Intensity) Differences



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

We define the window function :

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

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

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

Correspondence Using Correlation

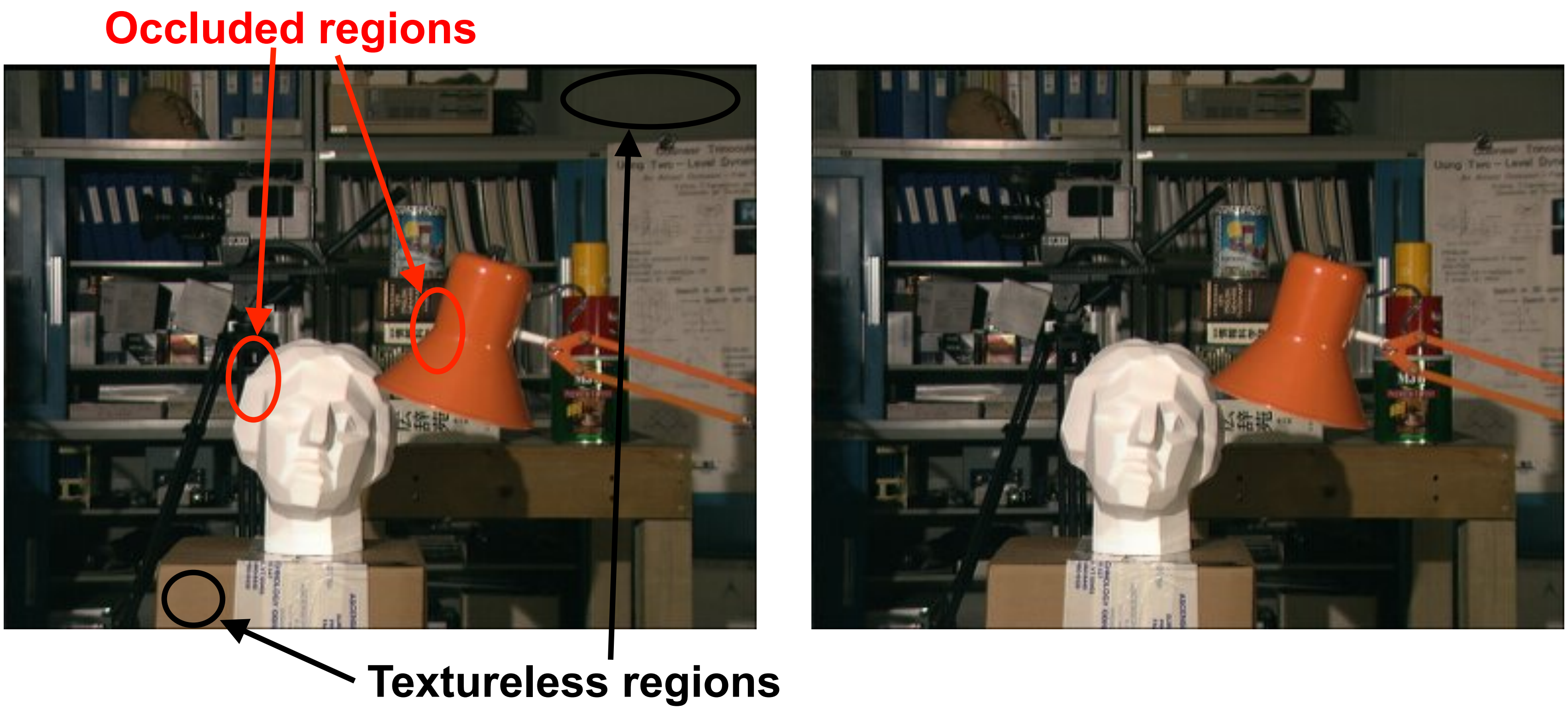
Left

Disparity Map

Images courtesy of Point Grey Research

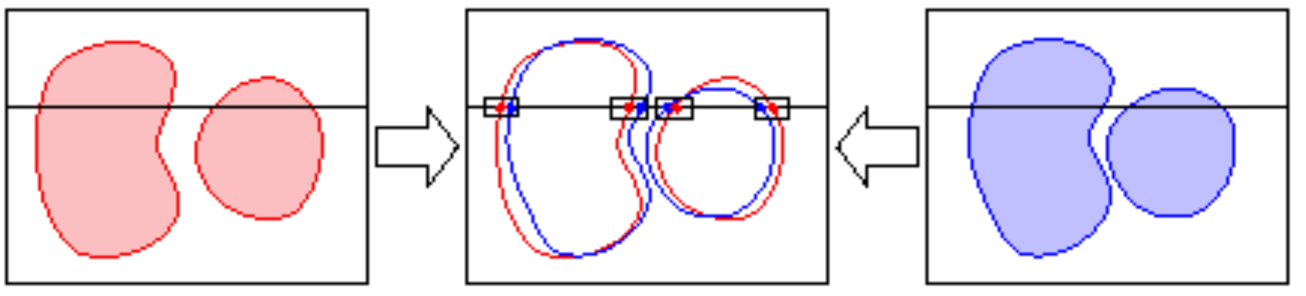
Two major roadblocks

- Texture-less regions create ambiguities
- Occlusions result in missing data



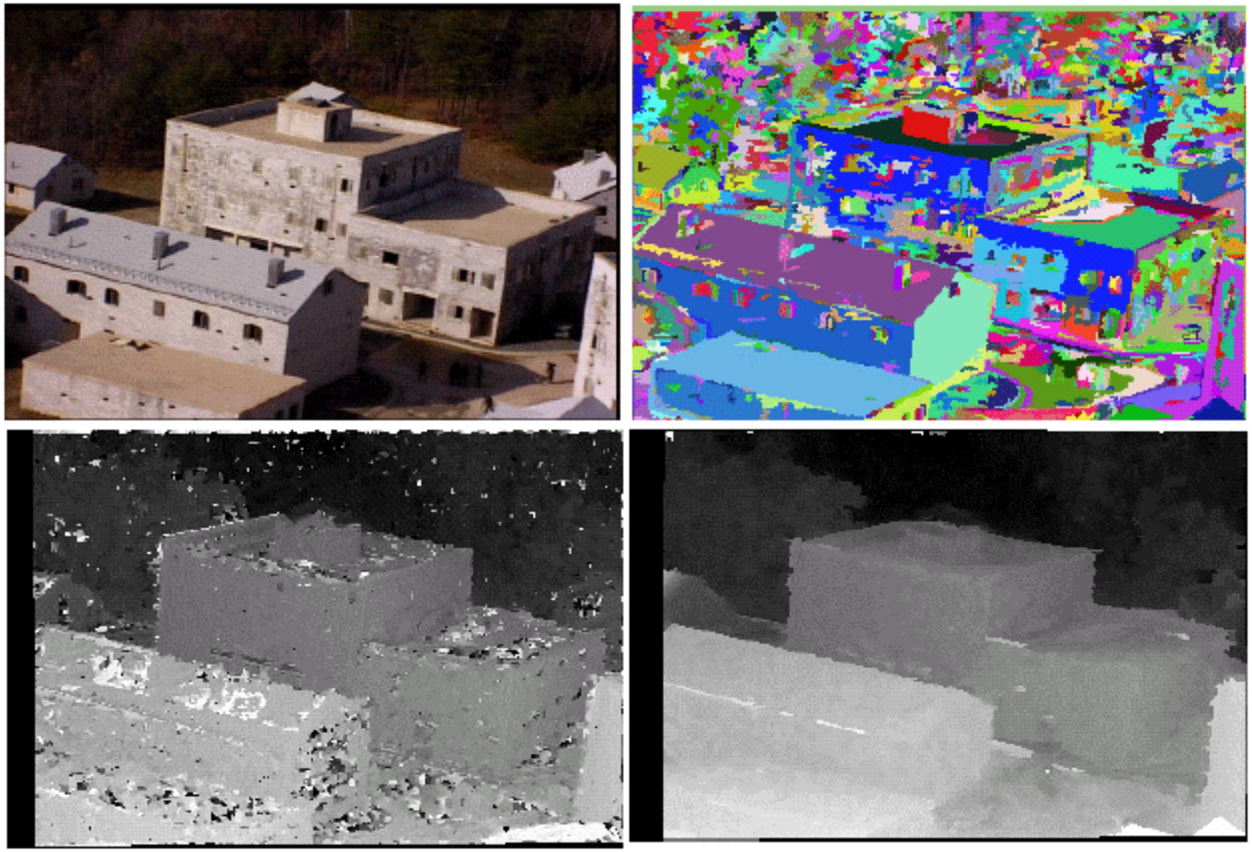
Edge-based Stereo

- Another approach is to match *edges* rather than windows of pixels:



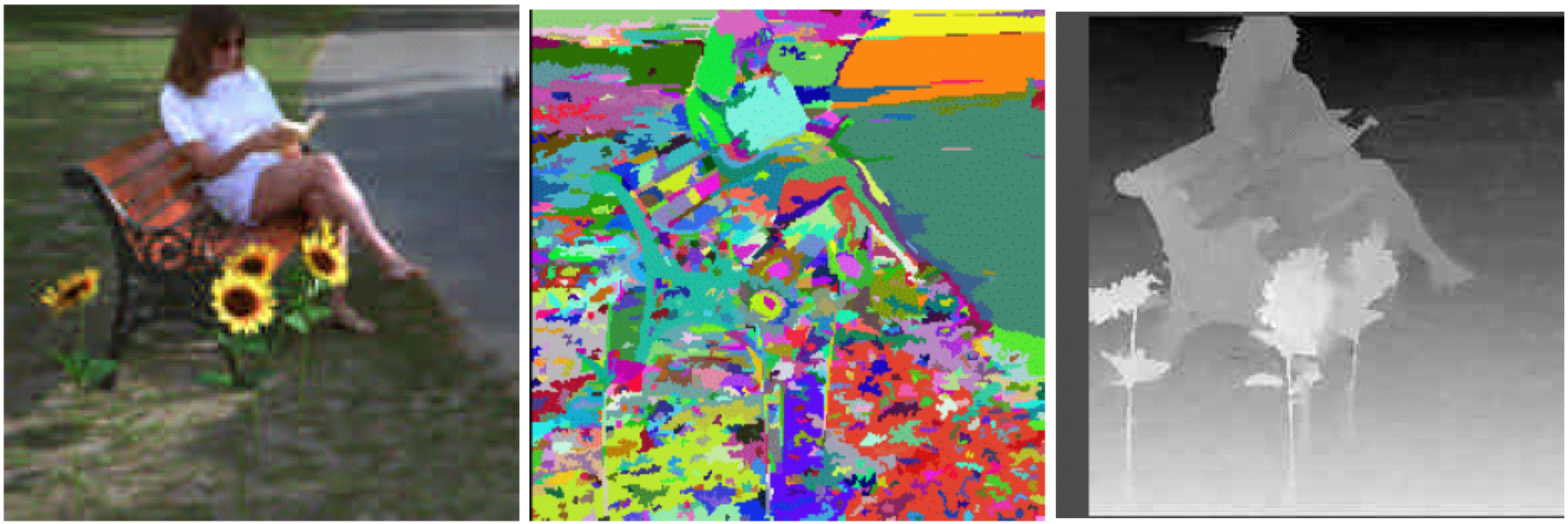
- Which method is better?
  - Edges tend to fail in dense texture (outdoors)
  - Correlation tends to fail in smooth featureless areas
  - Sparse correspondences

Segmentation-based Stereo



Hai Tao and Harpreet W. Sawhney

Another Example



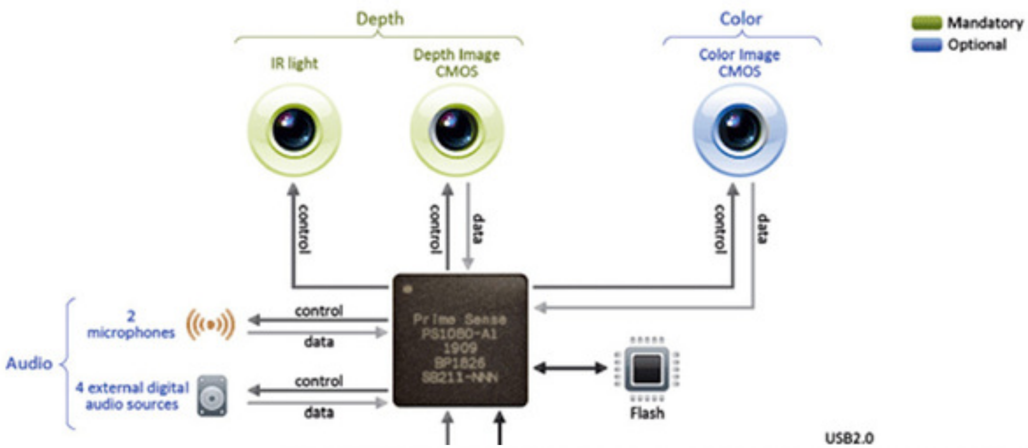
Bottom Line:
Stereo is Still Unresolved

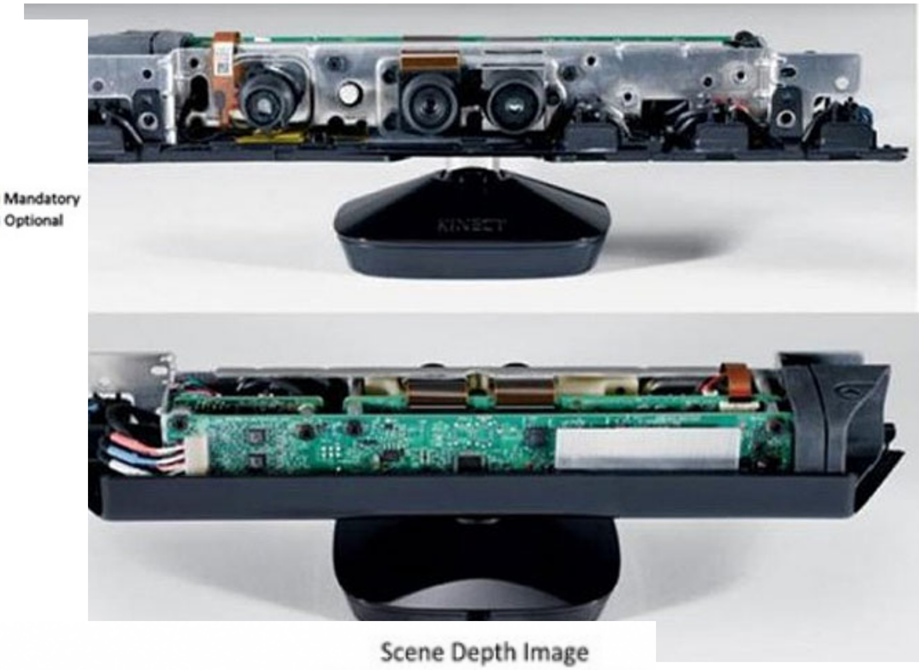
- Depth discontinuities
- Lack of texture (depth ambiguity)
- Non-rigid effects (highlights, reflection, translucency)

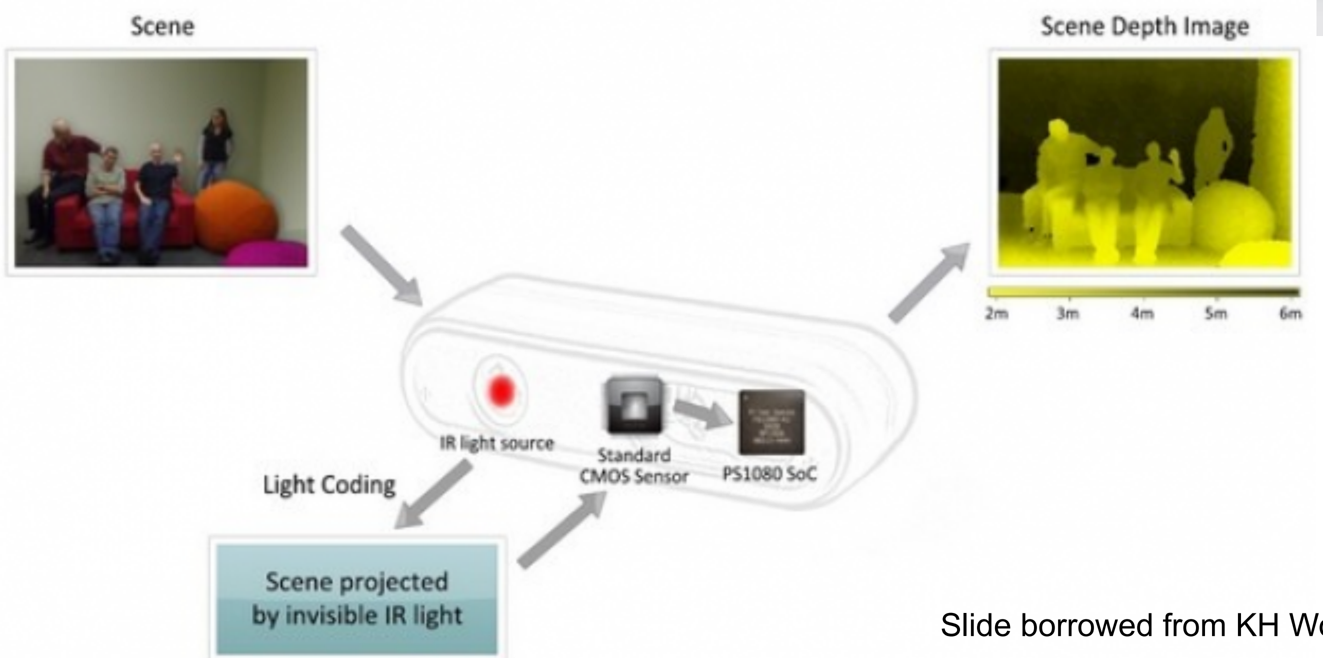
New Perspective: Kinect


KINECT
GA35610H23ABDB0249
XBOX 360

Kinect Hardware







Slide borrowed from KH Wong


Depth Image
3 PS1080 SoC
1 IR Light Source
Scene projected by invisible IR Light
2 Standard CMOS Sensor
Sensing Device
Scene

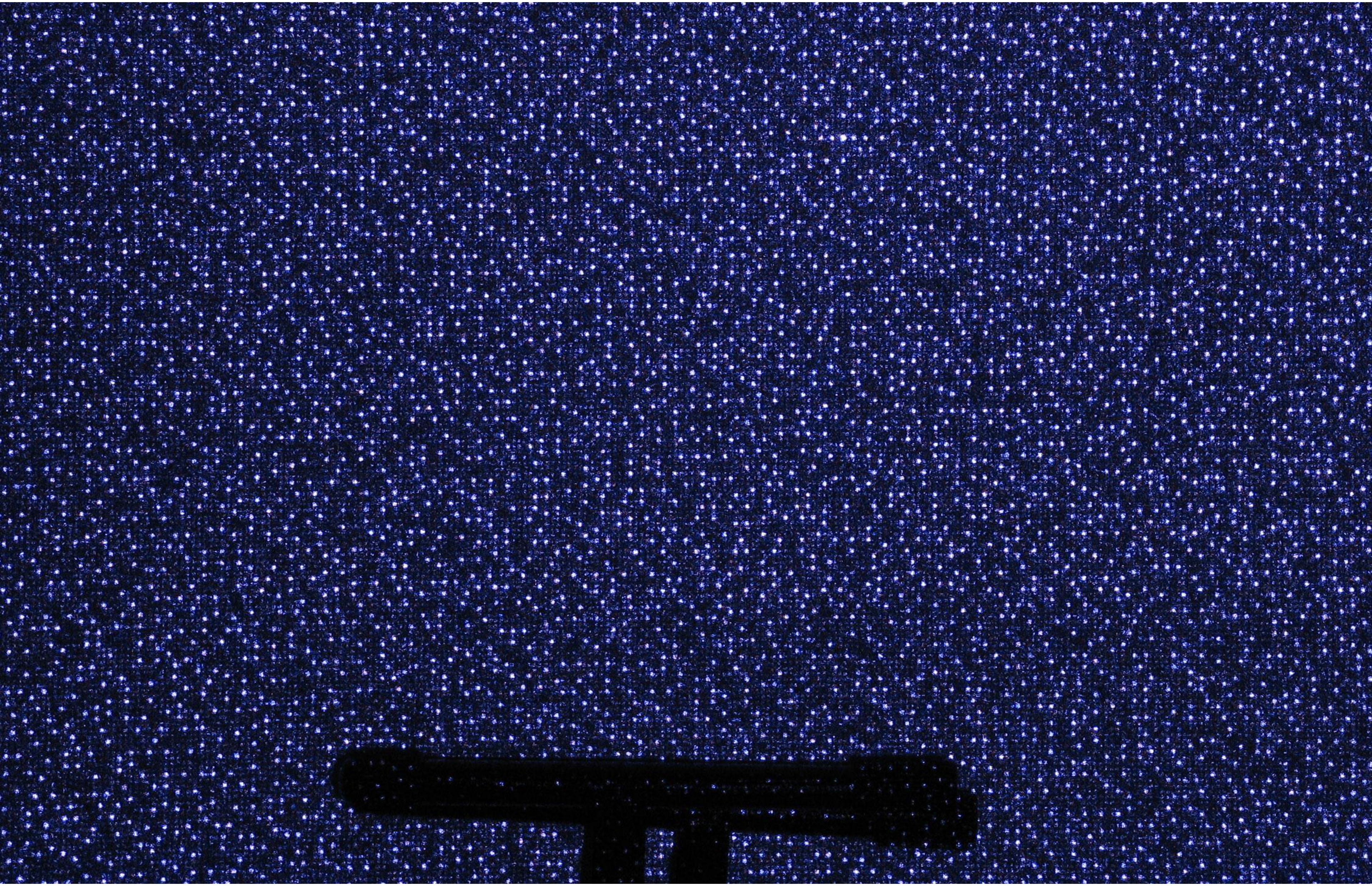
See the IR-dots emitted by KINECT

•

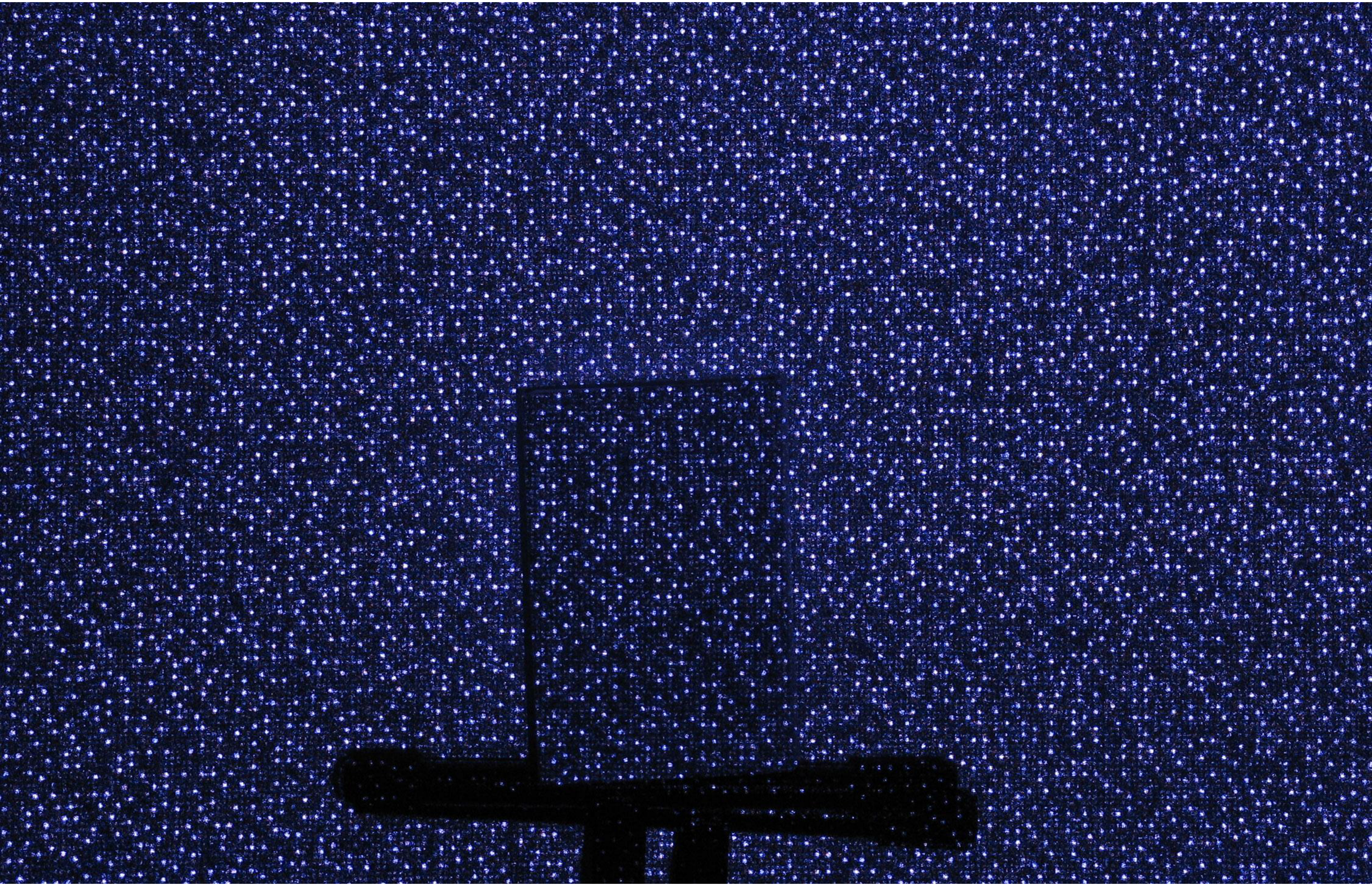
http://www.youtube.com/watch?v=-gbzXjdHfJA

http://www.youtube.com/watch?v=dTKINGSH9Po&feature=related

Slide borrowed from KH Wong



Source: http://www.futurepicture.org/?p=97



Source: http://www.futurepicture.org/?p=97

KinectFusion

- https://www.youtube.com/watch?v=quGhaggn3cQ
- http://people.csail.mit.edu/kaess/projects.html#kintinuous

From 2 views to >2 views

- More pixels voting for the right depth
- Statistically more robust
- However, occlusion reasoning is more complicated, since we have to account for *partial occlusion*:
  - Which subset of cameras sees the same 3D point?