Lecture 19:
Depth Cameras

Kayvon Fatahalian
CMU 15-869: Graphics and Imaging Architectures (Fall 2011)

Continuing theme: computational photography

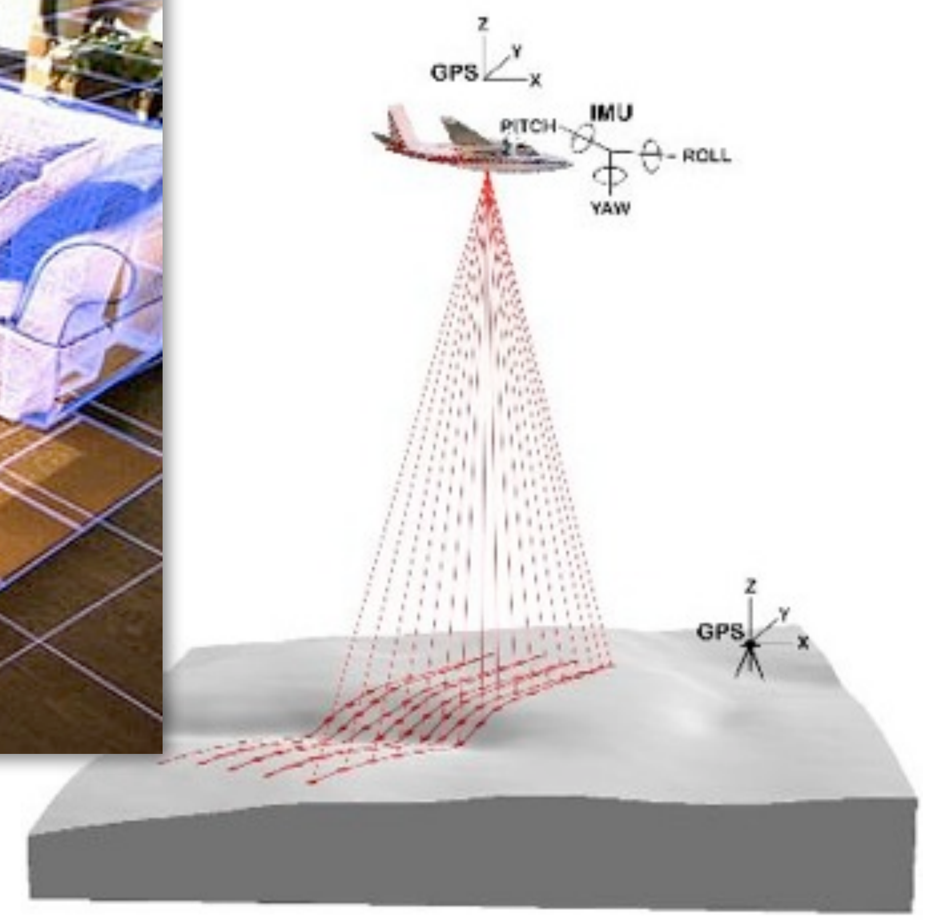
- Cheap cameras capture light, extensive processing produces desired image

- Today:
  - Capturing depth in addition to light intensity

Why might we want to know the depth of scene objects?

Scene Understanding

Navigation



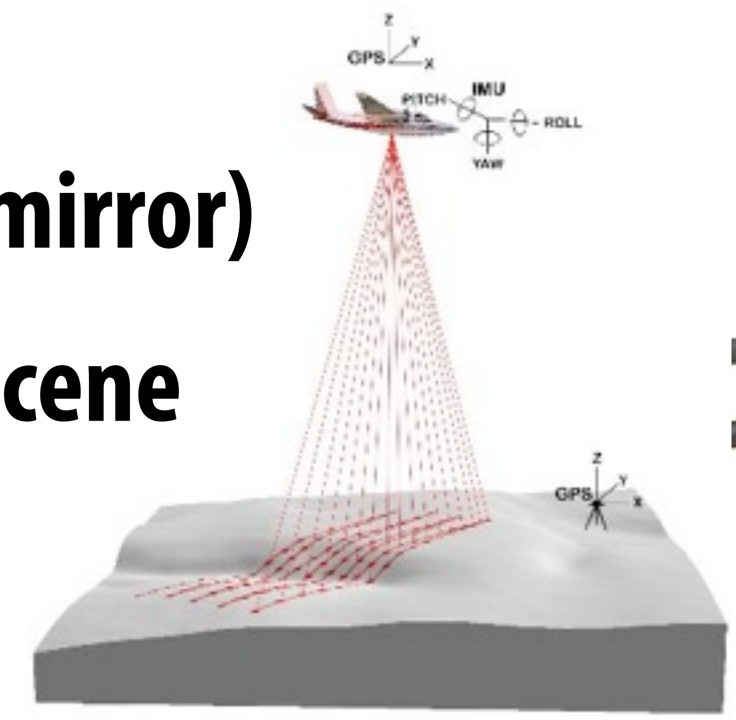
Mapping

Tracking

Segmentation

Depth from time-of-flight

- **Conventional LIDAR**
  - Laser beam scans scene (rotating mirror)
  - Low frame rate to capture entire scene



- **“Time-of-flight” cameras**
  - No moving beam, capture image of scene with each light pulse
  - Special CMOS sensor records a depth image
  - High frame rate
  - Today: still low resolution, expensive (but dropping fast)

Computing depth from images

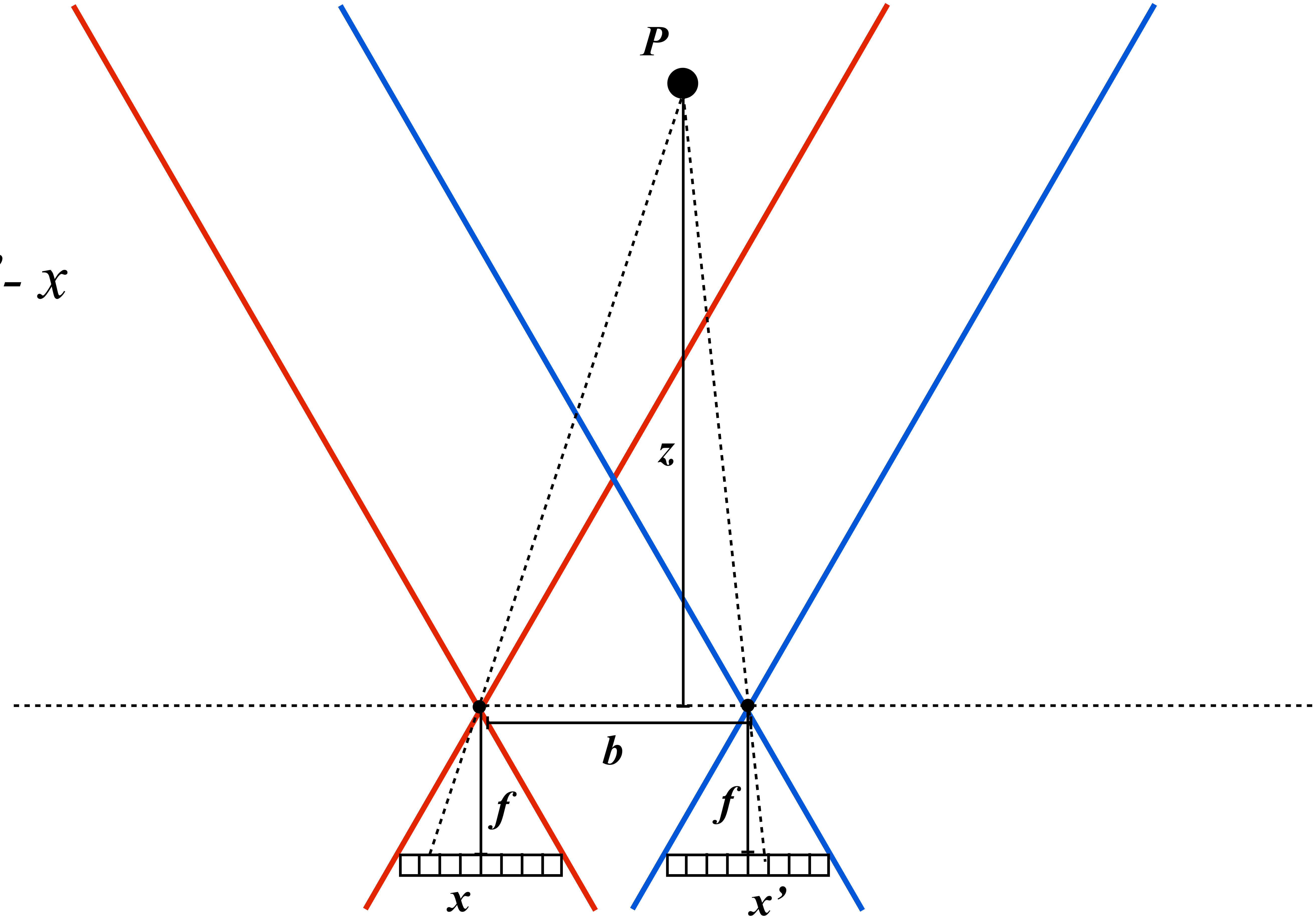
Binocular stereo 3D reconstruction of P: depth from disparity

Focal length: f

Baseline: b

Disparity: $d = x' - x$

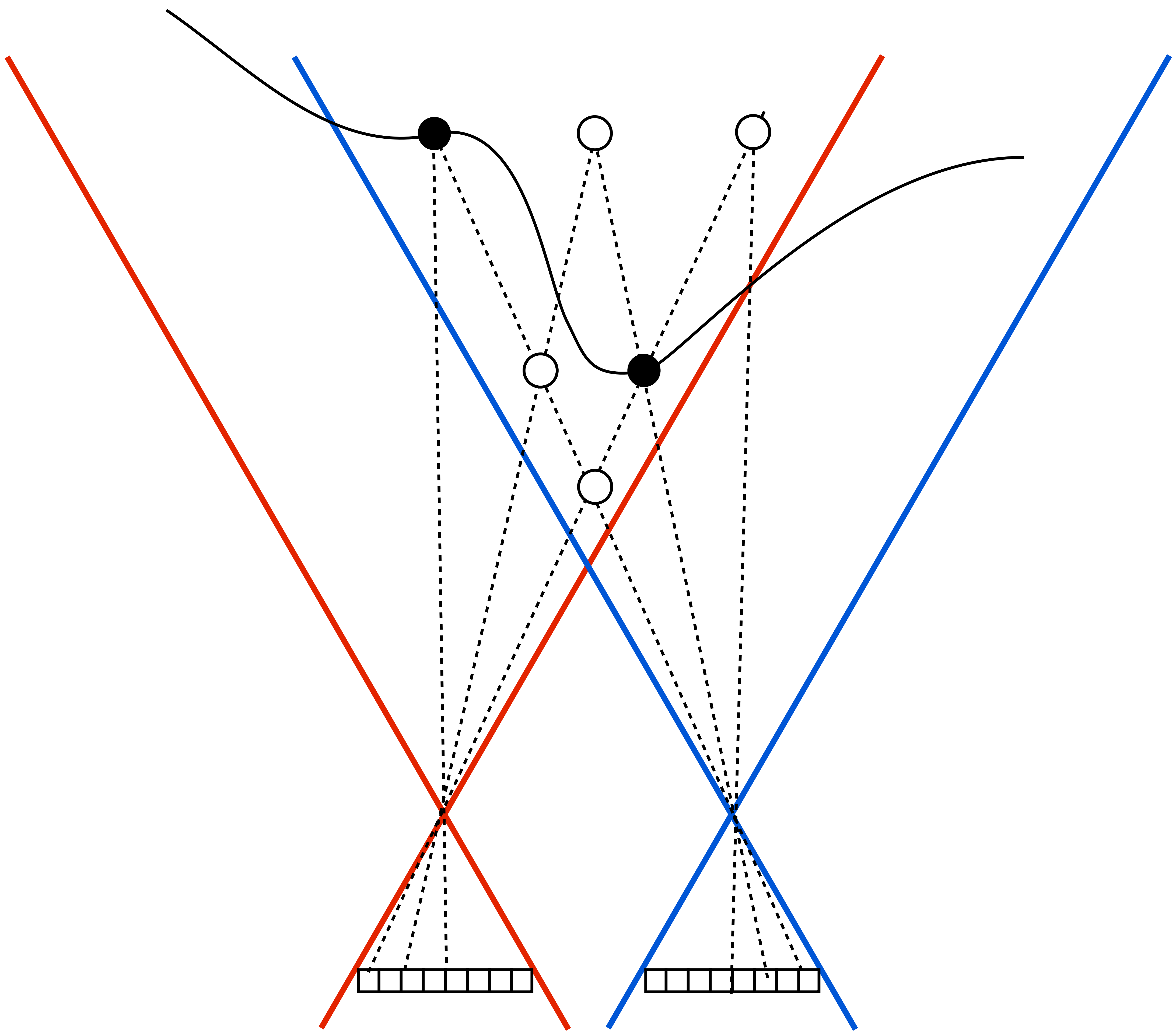
$$z = \frac{bf}{d}$$



Simple reconstruction example: cameras aligned (coplanar sensors), separated by known distance, same focal length

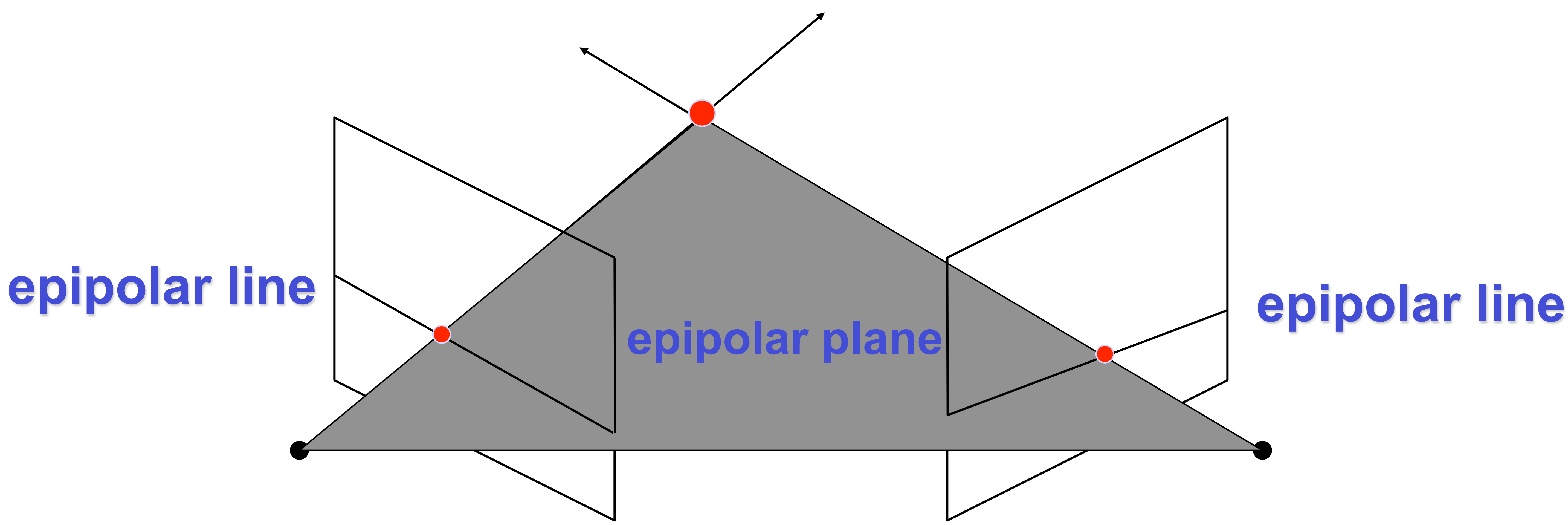
Correspondence problem

How to determine which pairs of pixels in image 1 and image 2 correspond to the same scene point?



Epipolar constraint

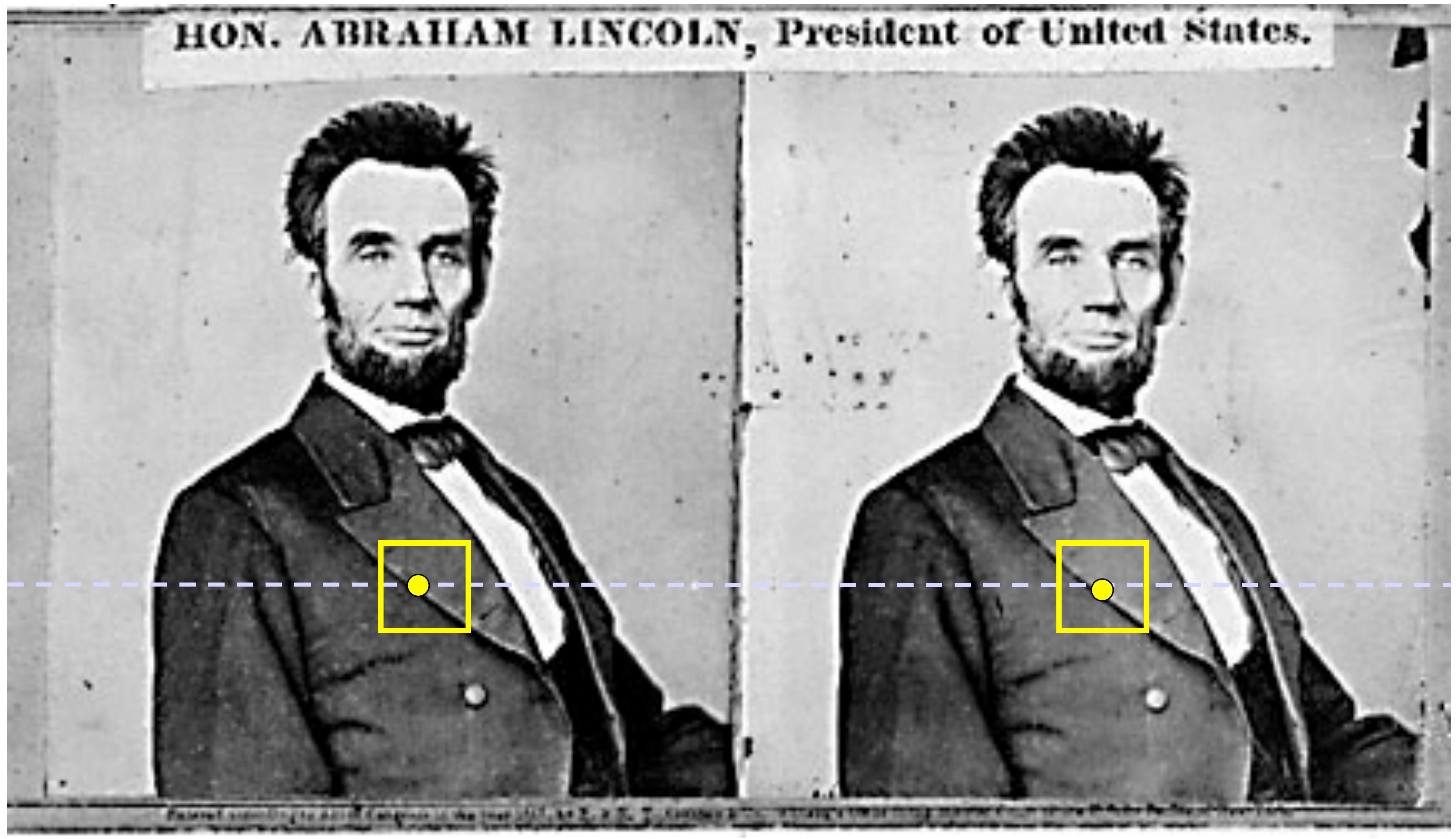
- Determine Pixel Correspondence
  - –Pairs of points that correspond to same scene point



- Epipolar Constraint
  - –Reduces correspondence problem to 1D search along *conjugate epipolar lines*

Slide credit: S. Narasimhan

Solving correspondence (basic algorithm)



For each epipolar line

For each pixel in the left image

- compare with every pixel on same epipolar line in right image
- pick pixel with minimum match cost

Improvement: match ***windows***

- This should look familiar...
- Correlation, Sum of Squared Difference (SSD), etc.

Assumptions?

Slide credit: S. Narasimhan

Correspondence: robustness challenges

- Scene with no texture (many parts of the scene look the same)
- Non-lambertian surfaces (scene appearance dependent on view)
- Pixel pairs may not be present (occlusion from one view)

Depth from defocus

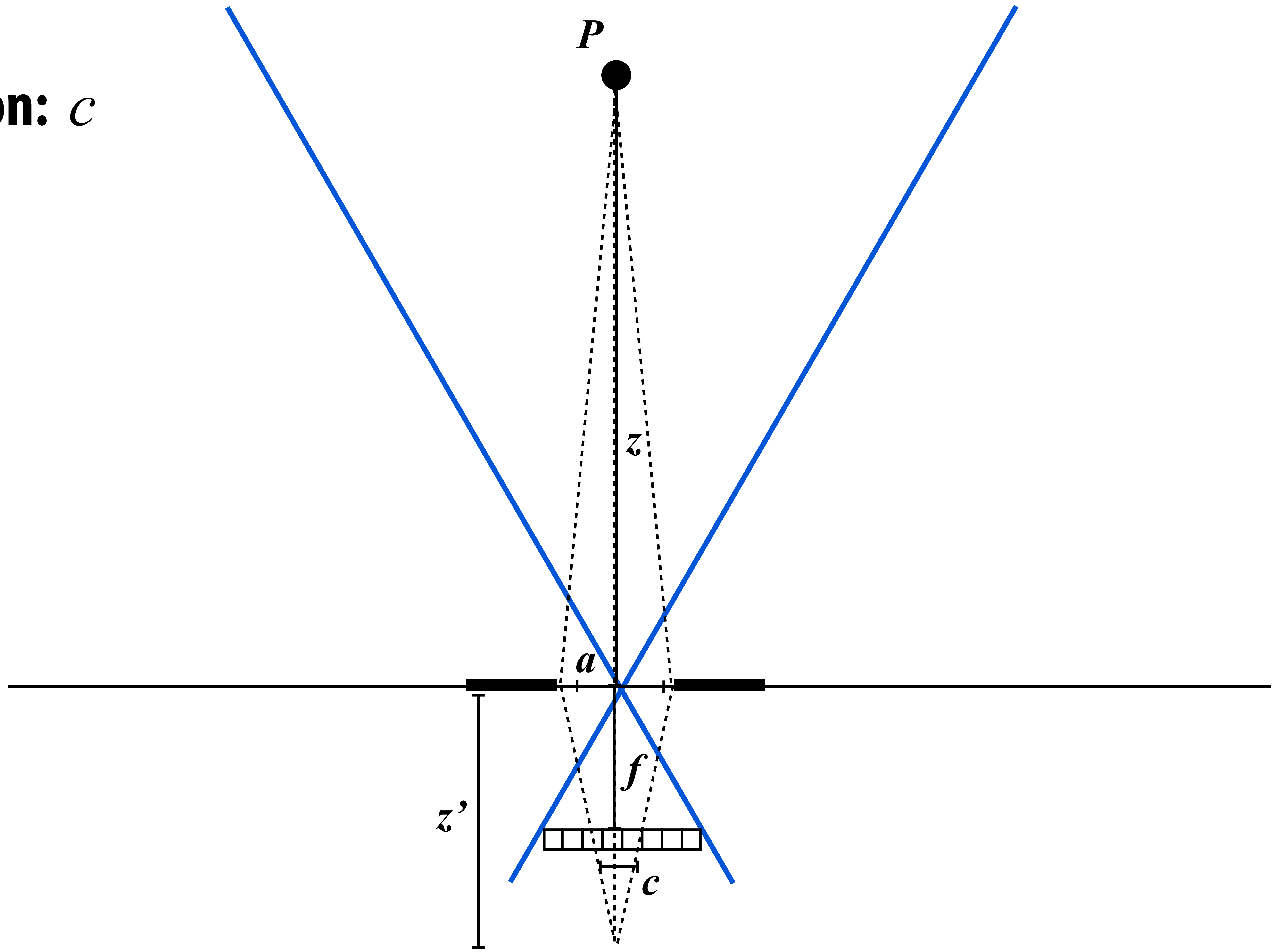
Aperture: a

Circle-of-confusion: c

$$\frac{a}{z'} = \frac{c}{z' - f}$$

Thin lens approximation:

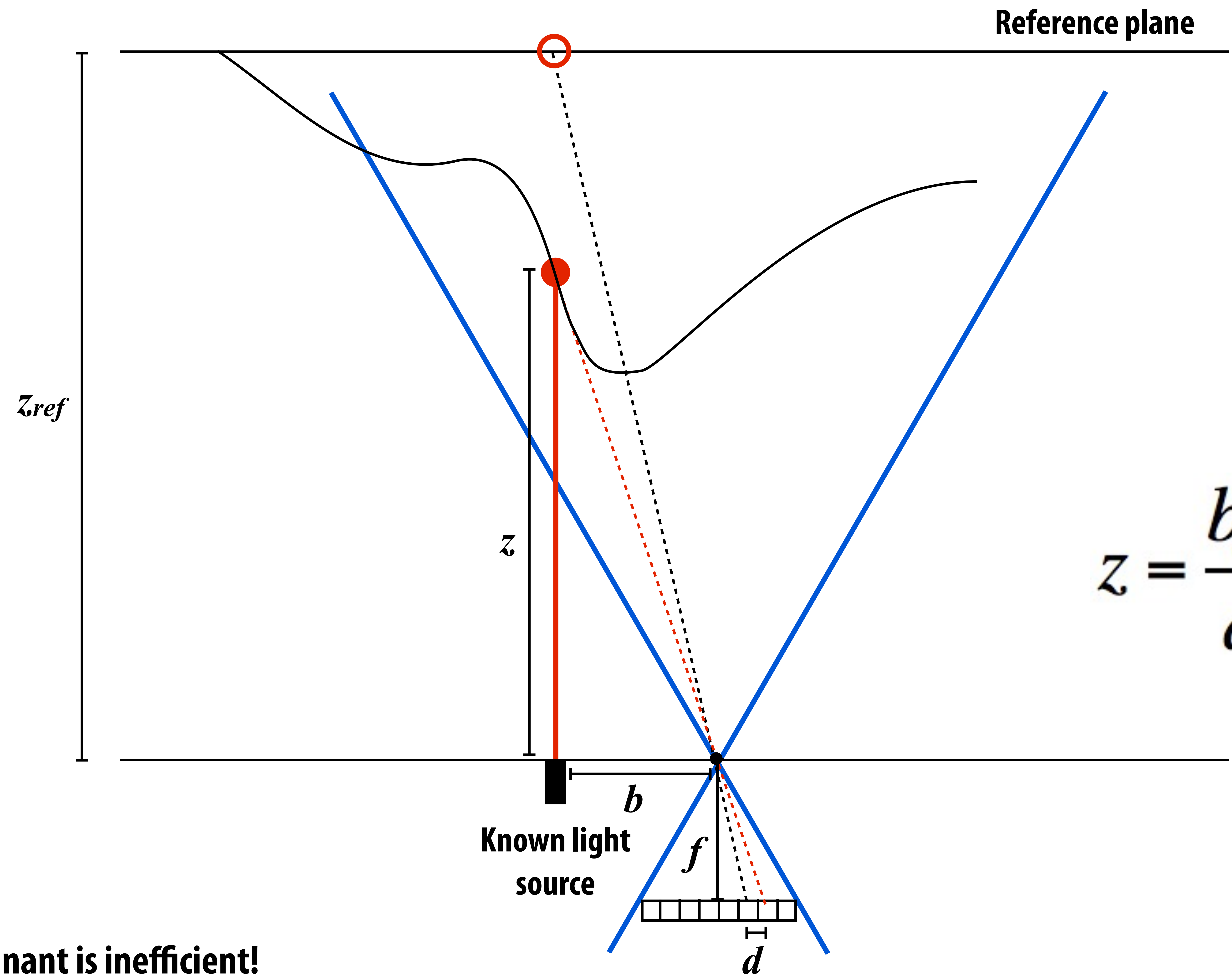
$$\frac{1}{z'} = \frac{1}{z} + \frac{1}{f}$$



Structured light

One light source emitting known beam, one camera

If the scene is at reference plane, image recorded by camera is known

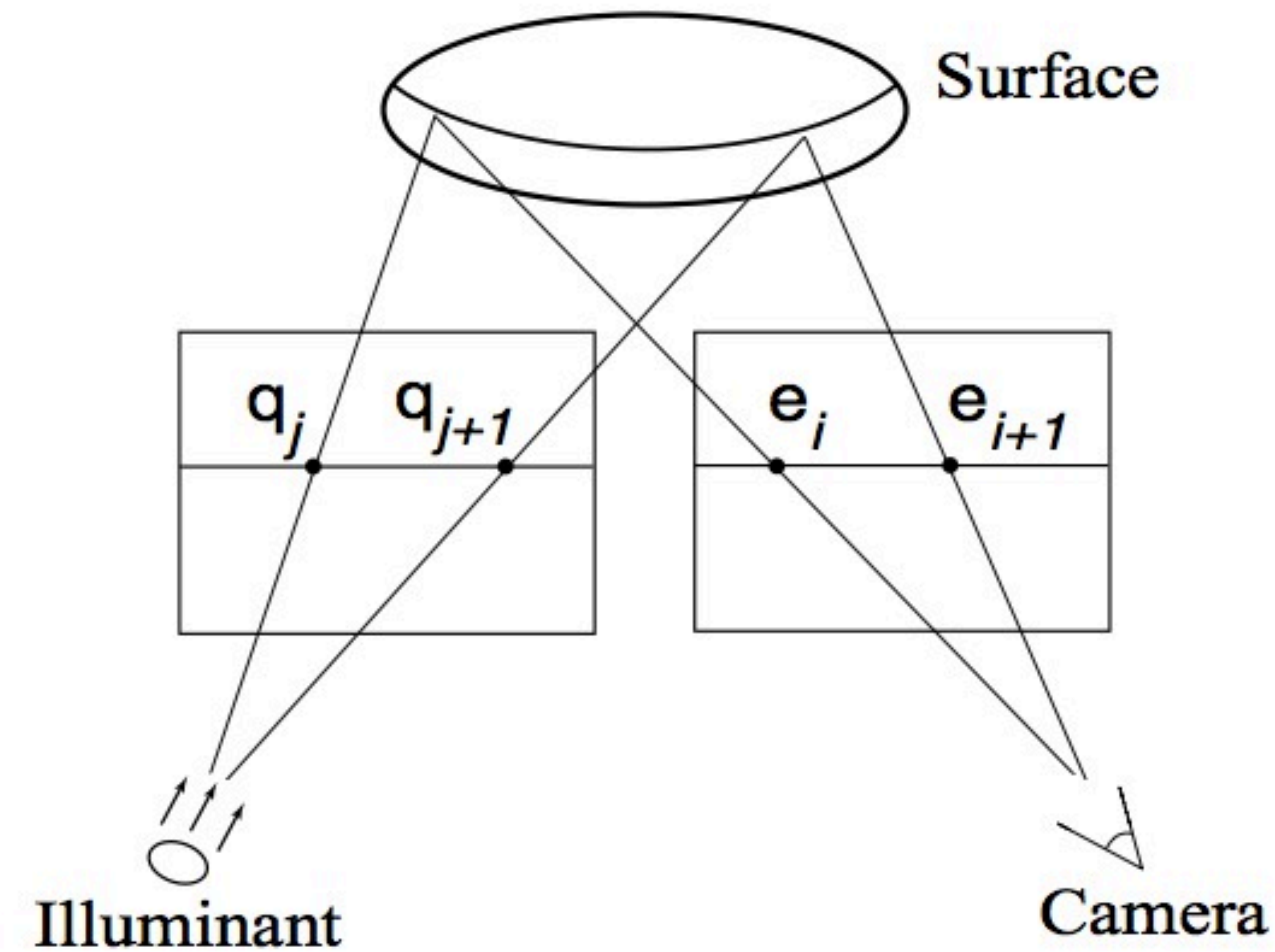


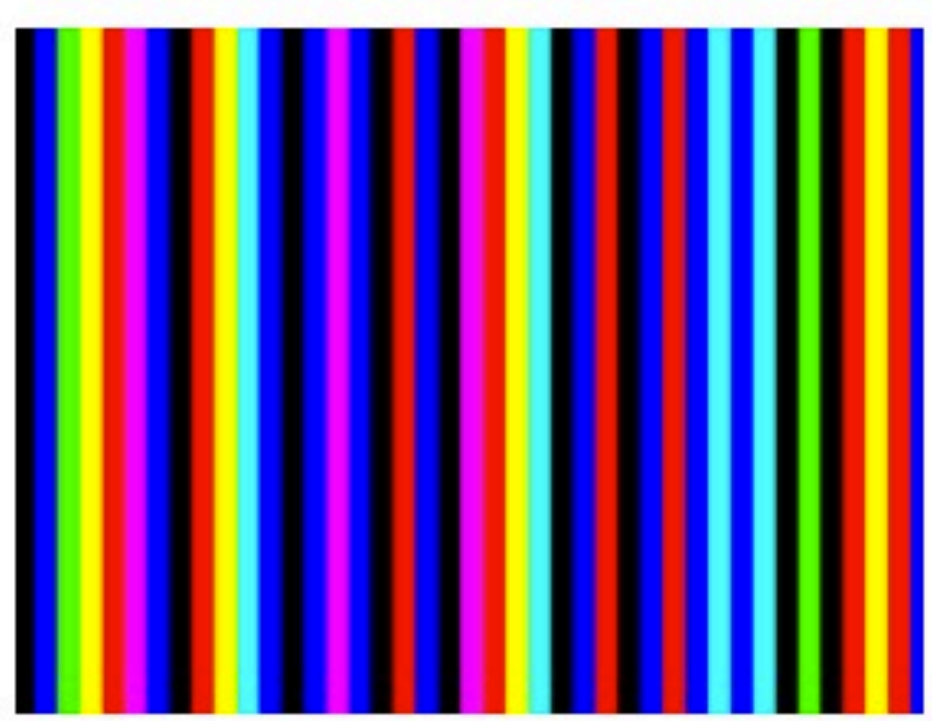
$$z = \frac{bf}{d} + z_{ref}$$

Single spot illuminant is inefficient!
(must to "scan" scene with spot to get depth)

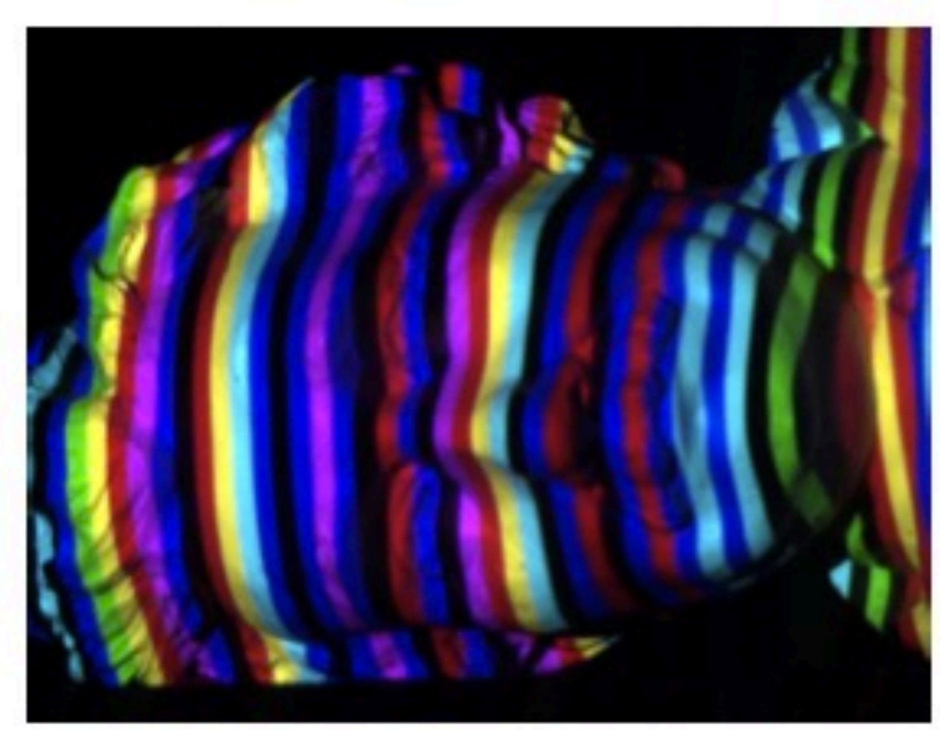
Structured light

Simplify correspondence problem by encoding spatial position in illuminant





Projected light pattern



Camera image

Image: Zhang et al.

Microsoft Kinect

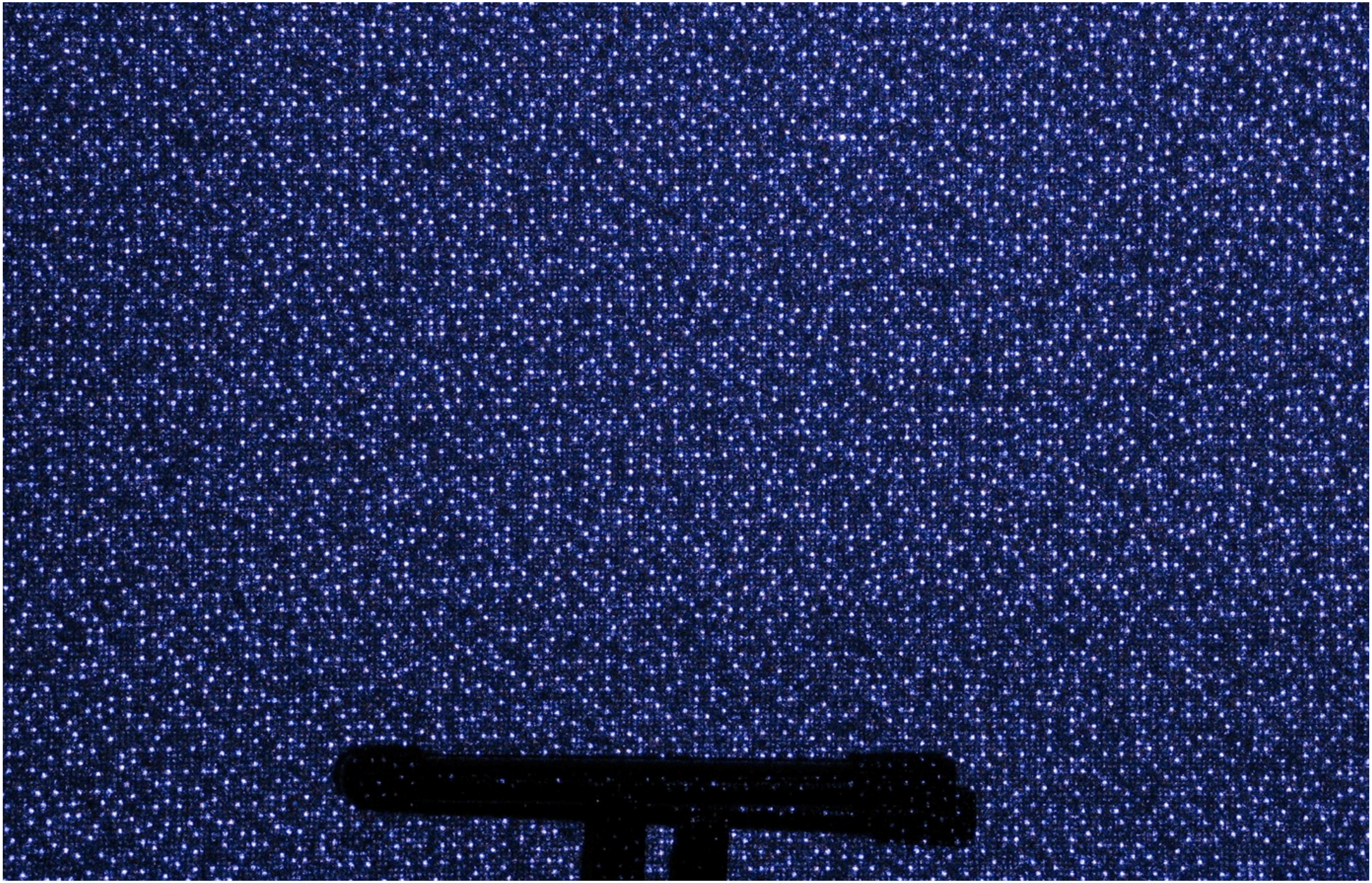
Illuminant
(Infrared Laser + diffuser)

RGB CMOS Sensor
640x480 (w/ Bayer mosaic)

Monochrome Infrared
CMOS Sensor
(Aptina MT9M001)
1280x1024 **

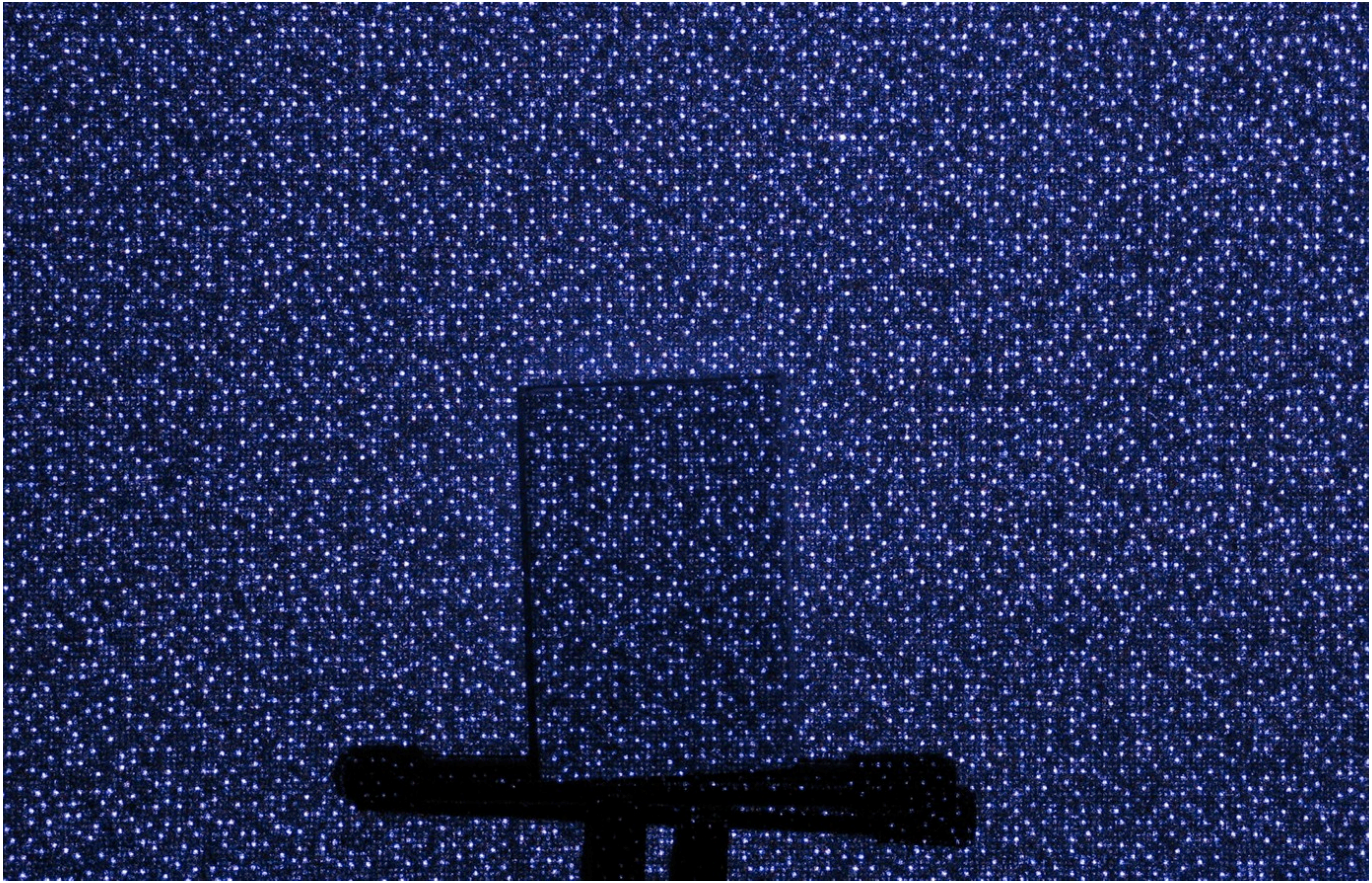
** Kinect returns 640x480 disparity image, teardowns suspect sensor configured for 2x2 binning down to 640x512, then crop

Infrared image of Kinect illuminant output



Credit: www.futurepicture.org

Infrared image of Kinect illuminant output



Credit: www.futurepicture.org

Computing disparity for scene

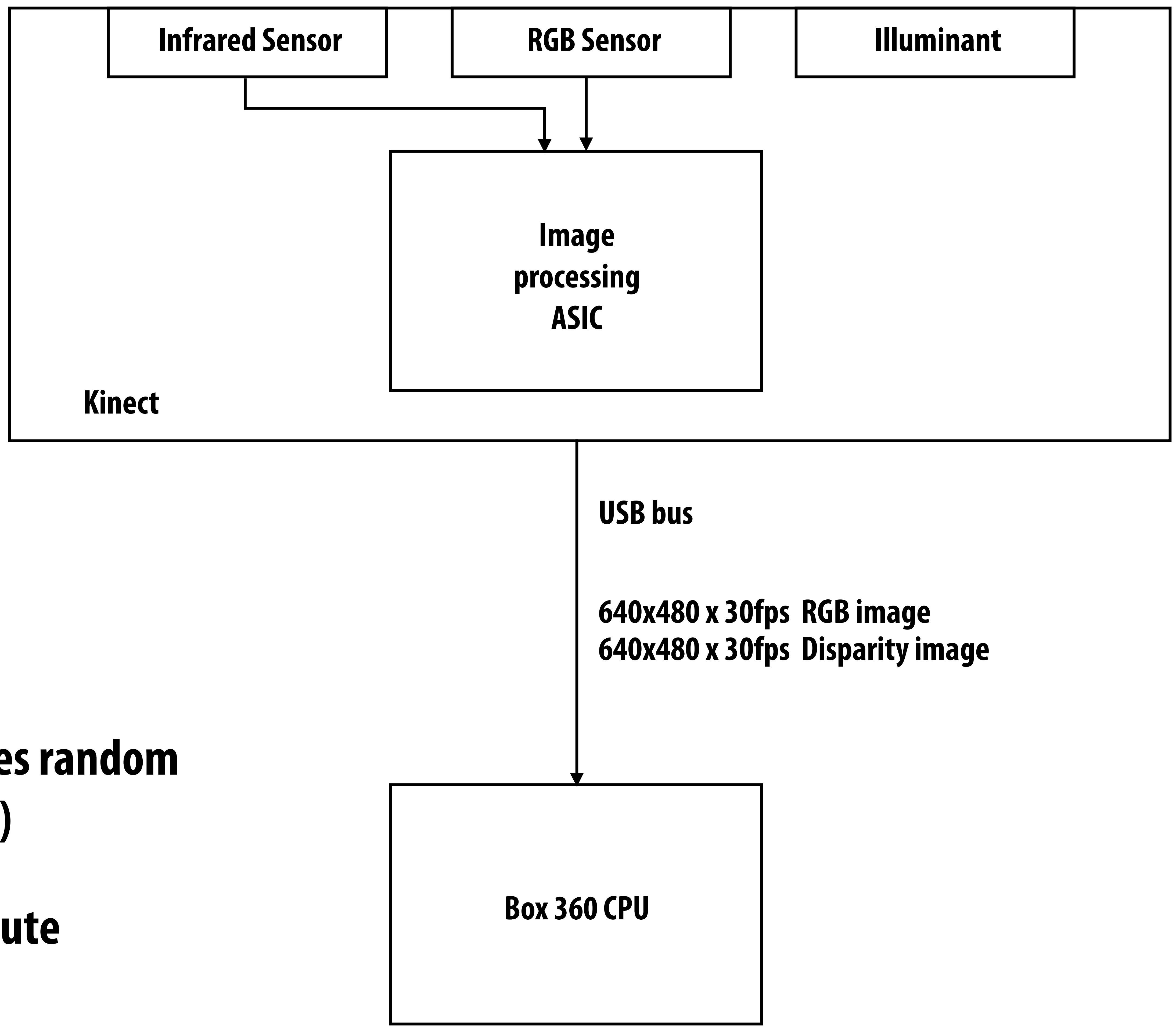
Region-growing algorithm for compute efficiency **
(Assumption: spatial locality likely implies depth locality)

1. Choose output pixels in infrared image, classify as UNKNOWN or SHADOW (based on whether speckle is found)
2. While significantly large percentage of output pixels are UNKNOWN
   - Choose an UNKNOWN pixel. Correlate surrounding NxM pixel window with reference image to compute disparity D=(dx,dy) (note: search window is a horizontal swath of image, plus some vertical slack)
   - If sufficiently good correlation is found:
     - Mark pixel as a region anchor
     - Attempt to grow region around the anchor:
       - Place region anchor in FIFO, mark as ACTIVE
       - While FIFO not empty
       - Extract pixel P from FIFO (known disparity for P is D)
       - Attempt to establish correlations for UNKOWN neighboring (left,right,top,bottom) pixels of P by searching region D + (+/-1,+/1)
       - If correlation found, mark pixel as ACTIVE, set, parent to P, add to FIFO
       - Else, mark pixel as EDGE, set depth to depth of P.

** Source: PrimeSense Patent WO 2007/043036 A1. (Likely not be actual algorithm used by Kinect)

Kinect block diagram

Disparity calculations performed by PrimeSense ASIC in Kinect, not by XBox 360 (or PC) CPU



Cheap sensors: ~ 1 MPixel

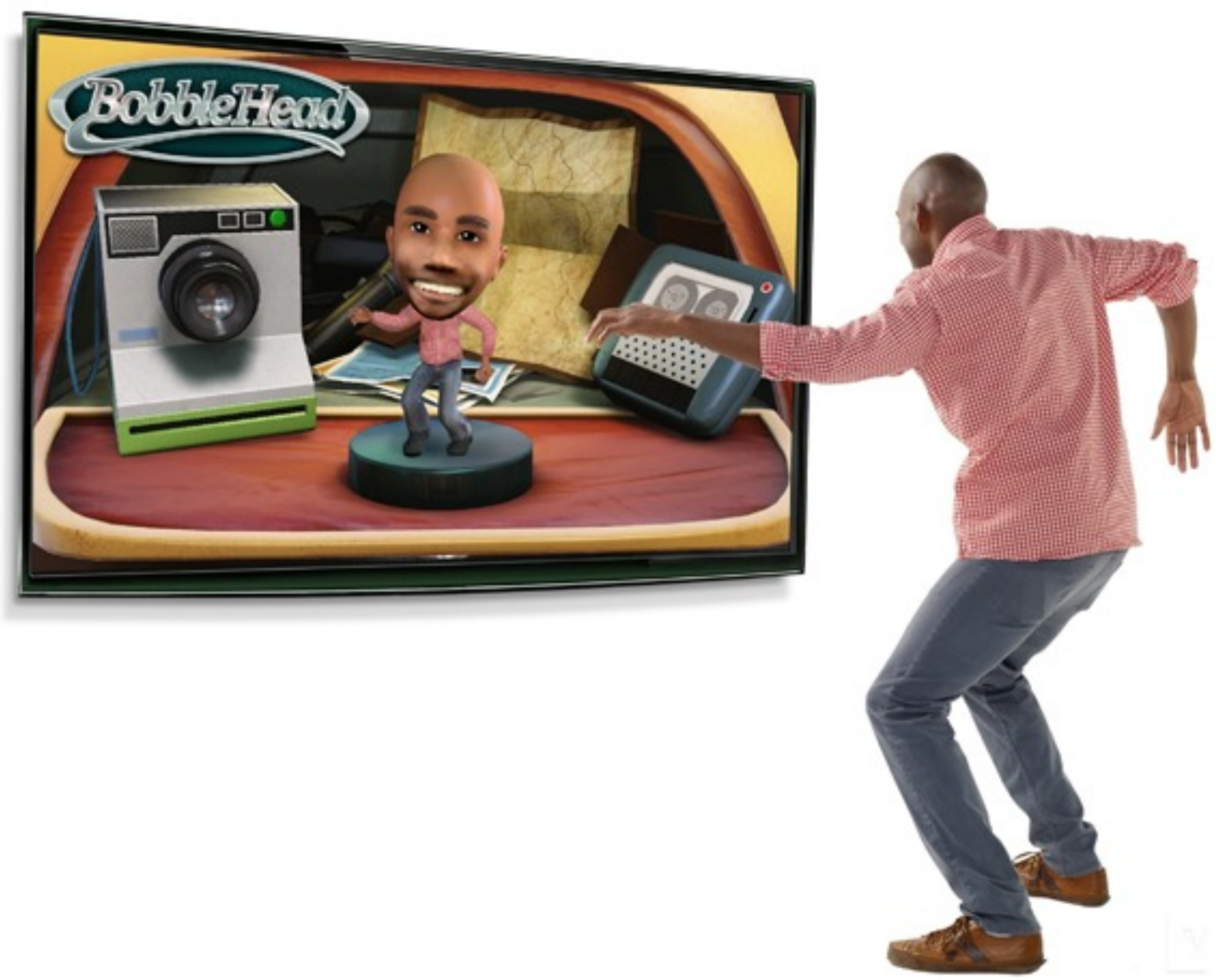
Cheap illuminant: laser + diffuser makes random dot pattern (not a traditional projector)

Custom image-processing ASIC to compute disparity image (scale-invariant region correlation involves non-trivial compute cost)

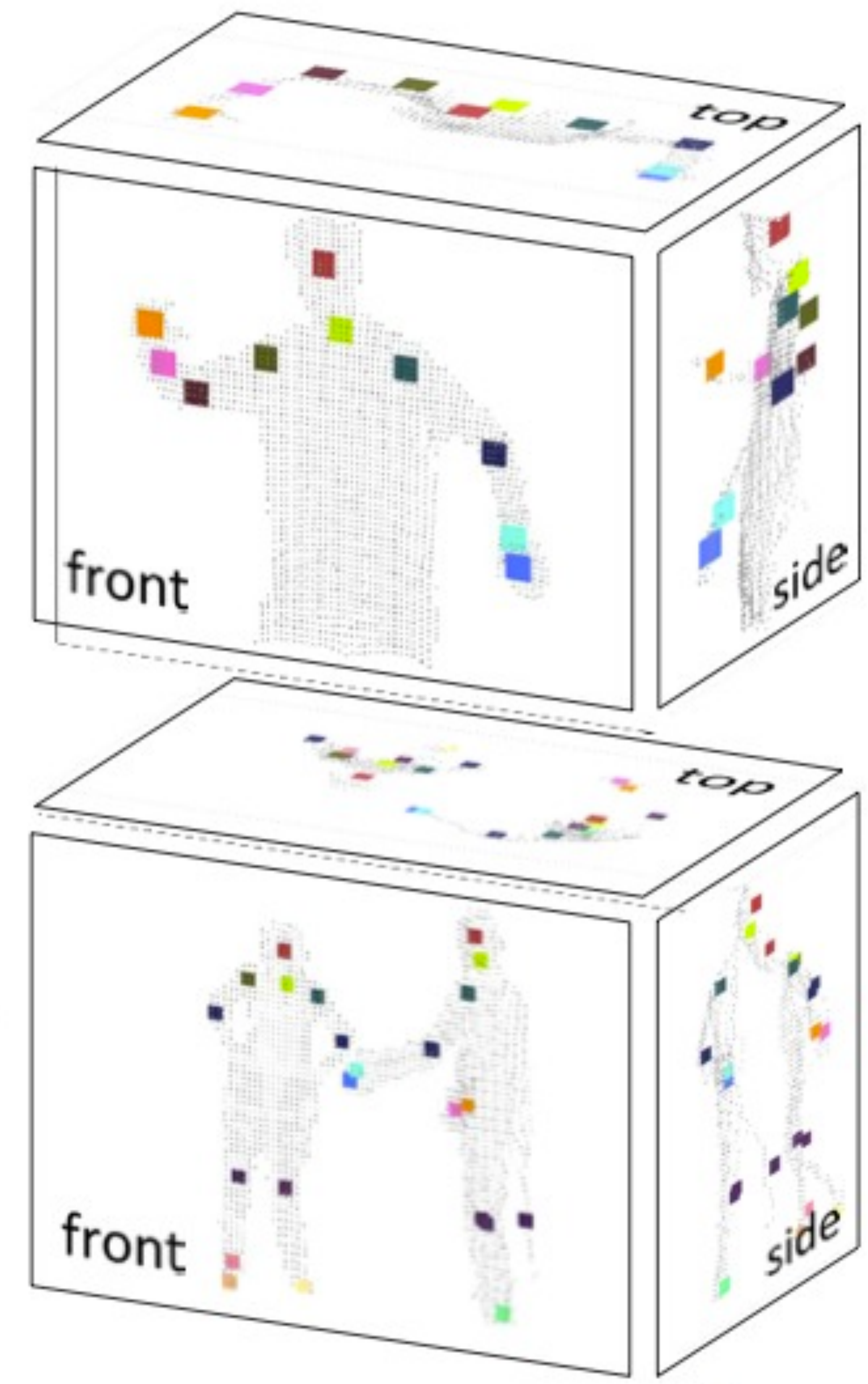
Extracting the player's skeleton

[Shotton et al. 2011]

(enabling full-body game input)



Depth Image

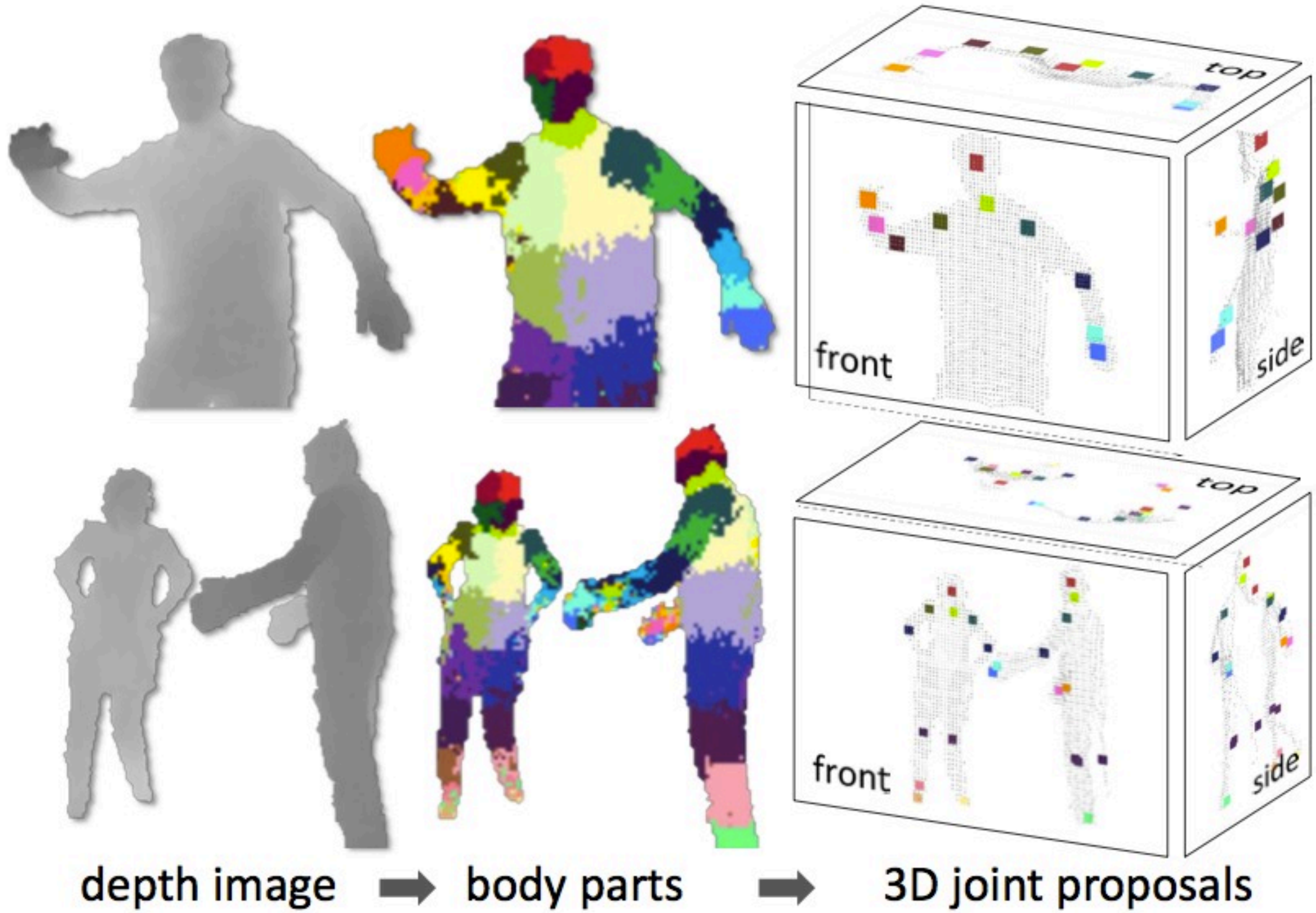


Character Joint Angles

Challenge: how to determine player's position/motion from depth images... without consuming a large fraction of the XBox 360's compute capability

Key idea: segment pixels into body regions

[Shotton et al. 2011]



Published description represents body with 31 regions

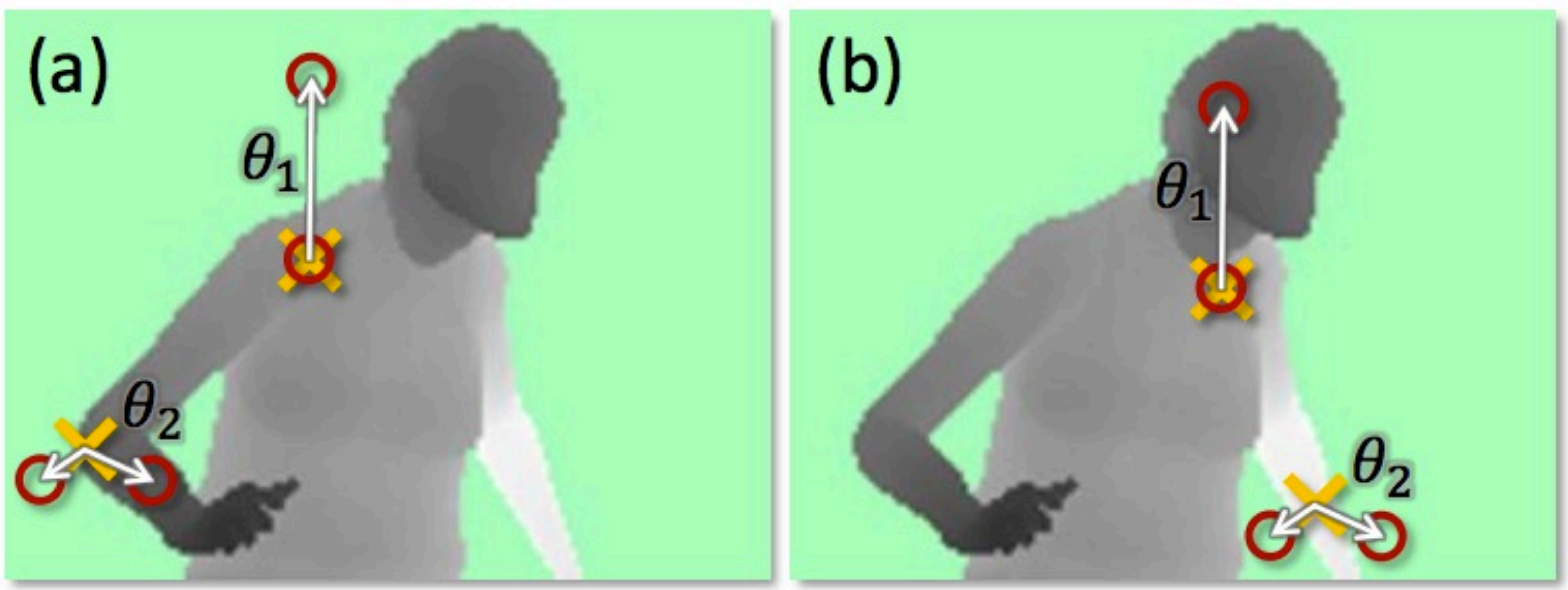
Pixel classification

[Shotton et al. 2011]

For each pixel: compute features from depth image

Pixel classifier learned from large database of motion capture data

Result: $P(c|I, \mathbf{x})$ (Prob. pixel x in depth image I is part of body part c)



Two example depth features

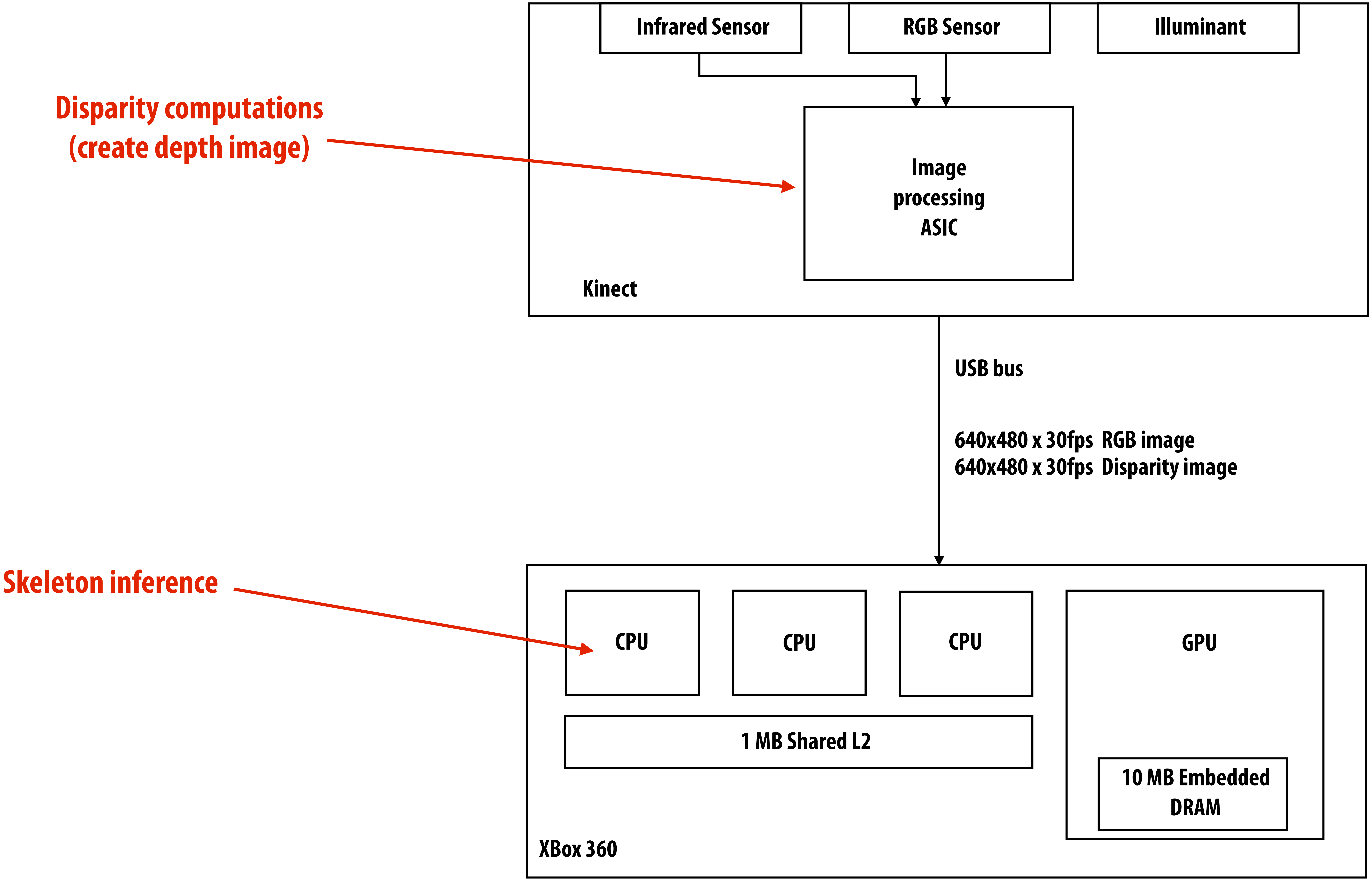
Per-pixel probabilities aggregated to compute 3D spatial density function for each body part, joint angles inferred from this density

Performance result

- Real-time skeleton estimation from depth image requires < 10% of Xbox 360 CPU

- XBox GPU-based implementation @ 200Hz (research implementation, not used in product)

XBox 360 + Kinect system



Summary

- **Kinect hardware = cheap depth sensor**
  - Structured light pattern generated by scattering infrared laser
  - Depth obtained from triangulation, not time-of-flight
  - Custom ASIC to convert infrared image into depth values

- **Interpretation of the depth values is performed on CPU**
  - Player skeleton estimation made computational feasible by machine learning approach

- **Future**
  - Calls for higher field of view, higher resolution depth