How the Kinect Works

Computational Photography

Derek Hoiem, University of Illinois

Photo frame-grabbed from: http://www.blisteredthumbs.net/2010/11/dance-central-angry-review

Language Models and Diffusion Networks – Awesome short vids by Steve Seitz

- ~~**Text to Image: Parti, Dall-E 2, Imagen**~~

~~https://www.youtube.com/watch?v=GYyP7Ova8KA&list=PLWfDJ5nla8UpwShx-lzLJqcp575fKpsSO&index=22~~

- **Text to Image: Part 2 -- Diffusion**

https://www.youtube.com/watch?v=lyodbLwb2lY&list=PLWfDJ5nla8UpwShx-lzLJqcp575fKpsSO&index=23

How the Kinect Works

Computational Photography

Derek Hoiem, University of Illinois

Photo frame-grabbed from: https://www.youtube.com/watch?v=srk7WIvx-wQ

Kinect Device

Kinect Device

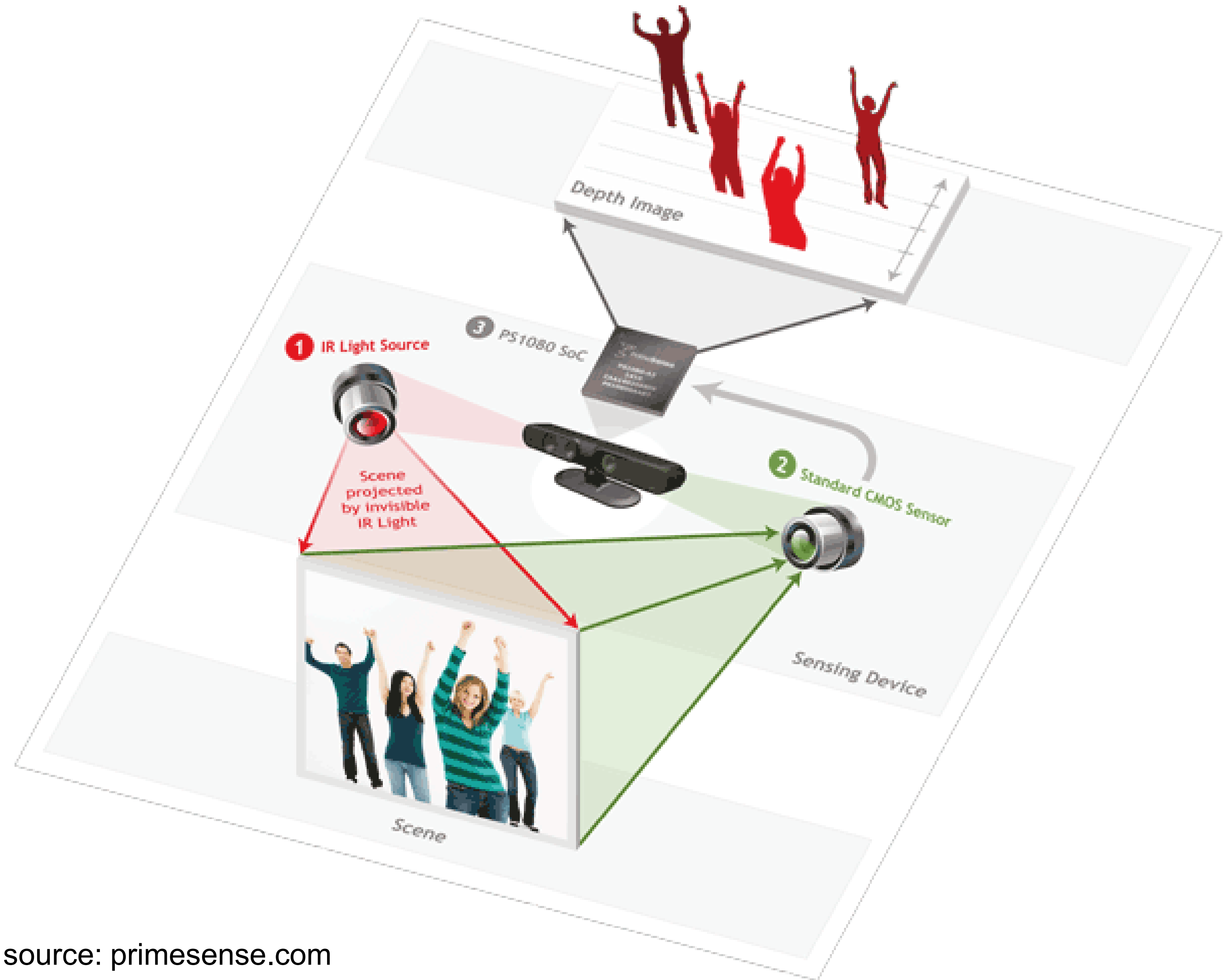
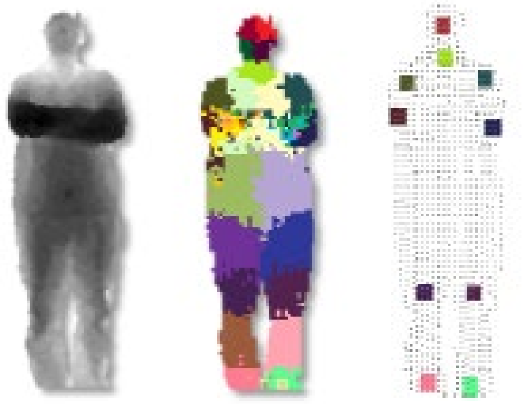


illustration source: primesense.com

What the Kinect does

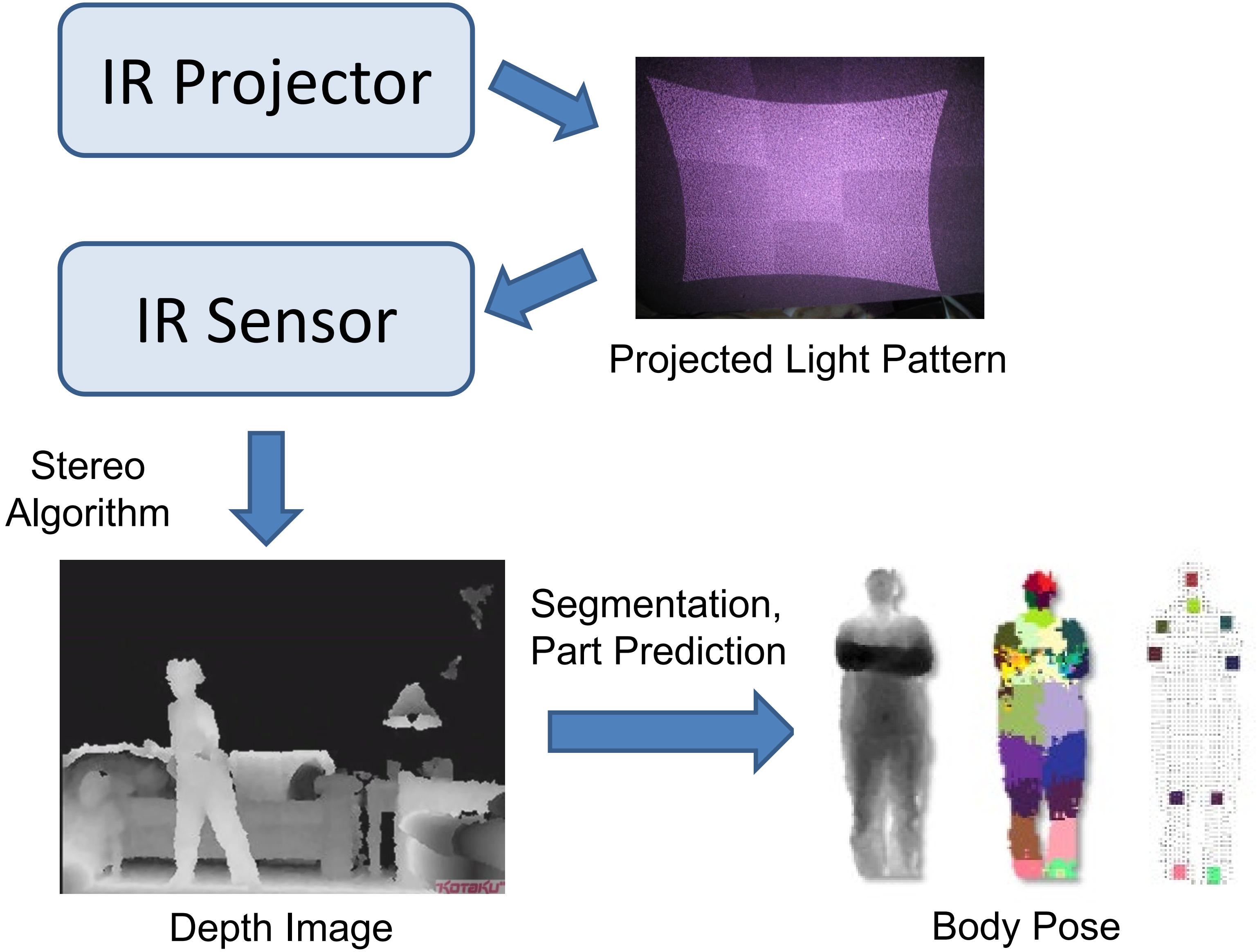
Get Depth Image



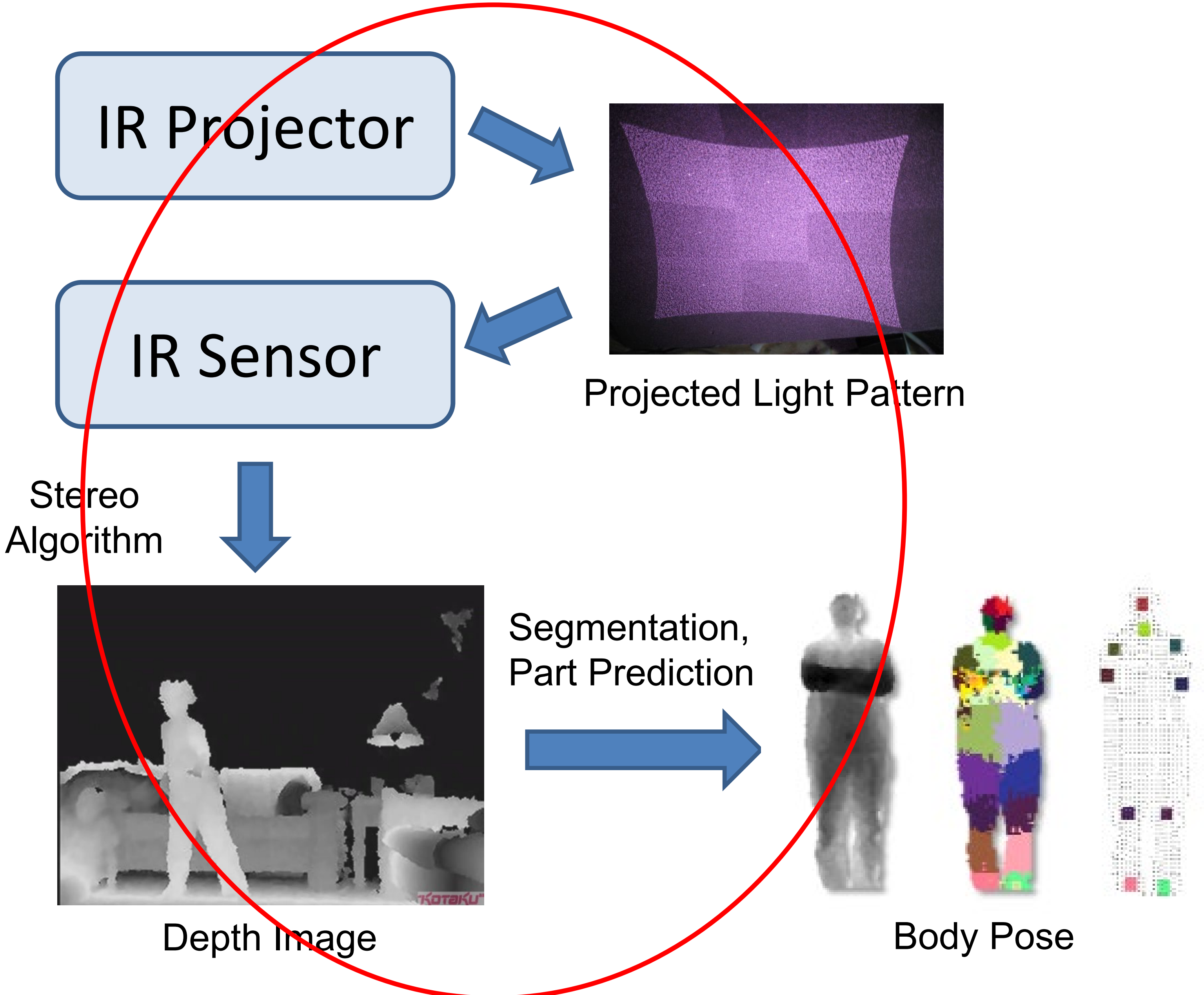
Estimate Body Pose

Application (e.g., game)

How Kinect Works: Overview



Part 1: Stereo from projected dots



Part 1: Stereo from projected dots

1. Overview of depth from stereo

2. How it works for a projector/sensor pair

3. Stereo algorithm used by Primesense (Kinect)

Depth from Stereo Images

image 1

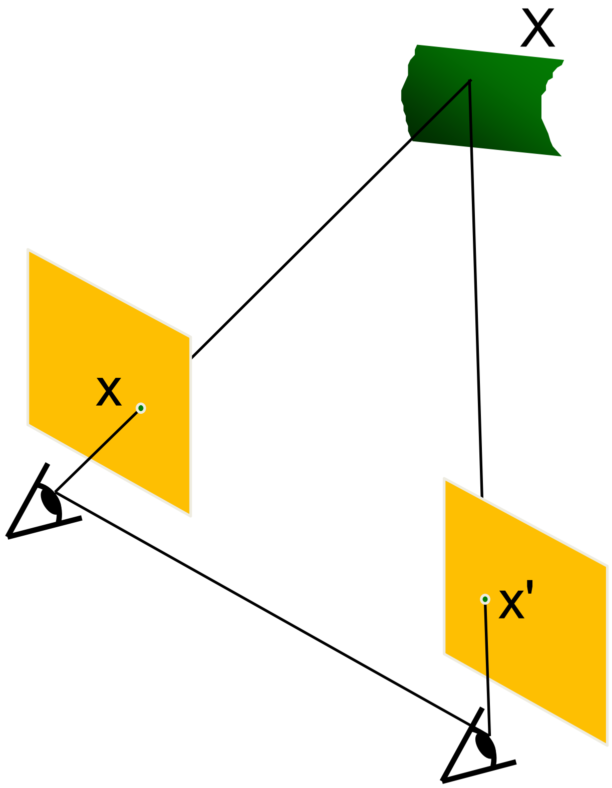
image 2

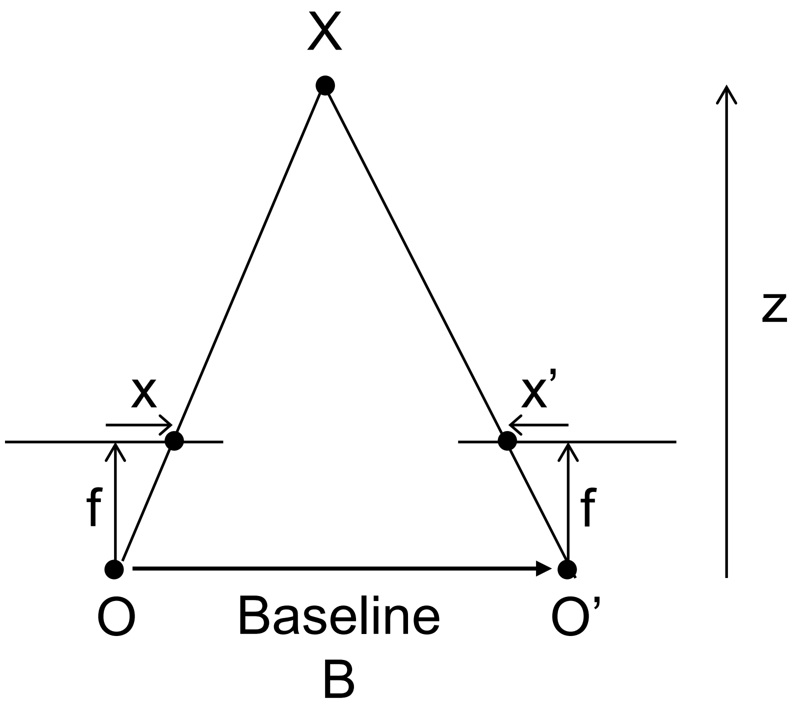
Dense depth map

Some of following slides adapted from Steve Seitz and Lana Lazebnik

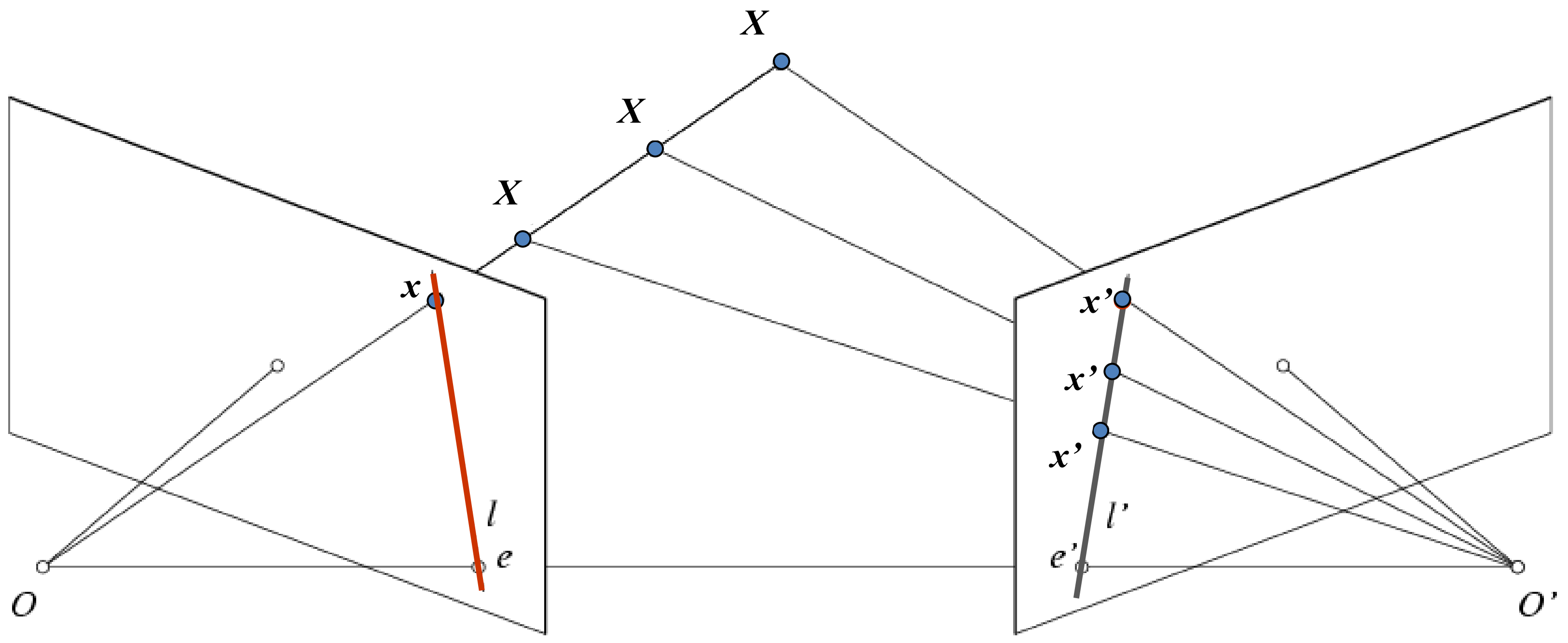
Depth from Stereo Images

- Goal: recover depth by finding image coordinate x’ that corresponds to x





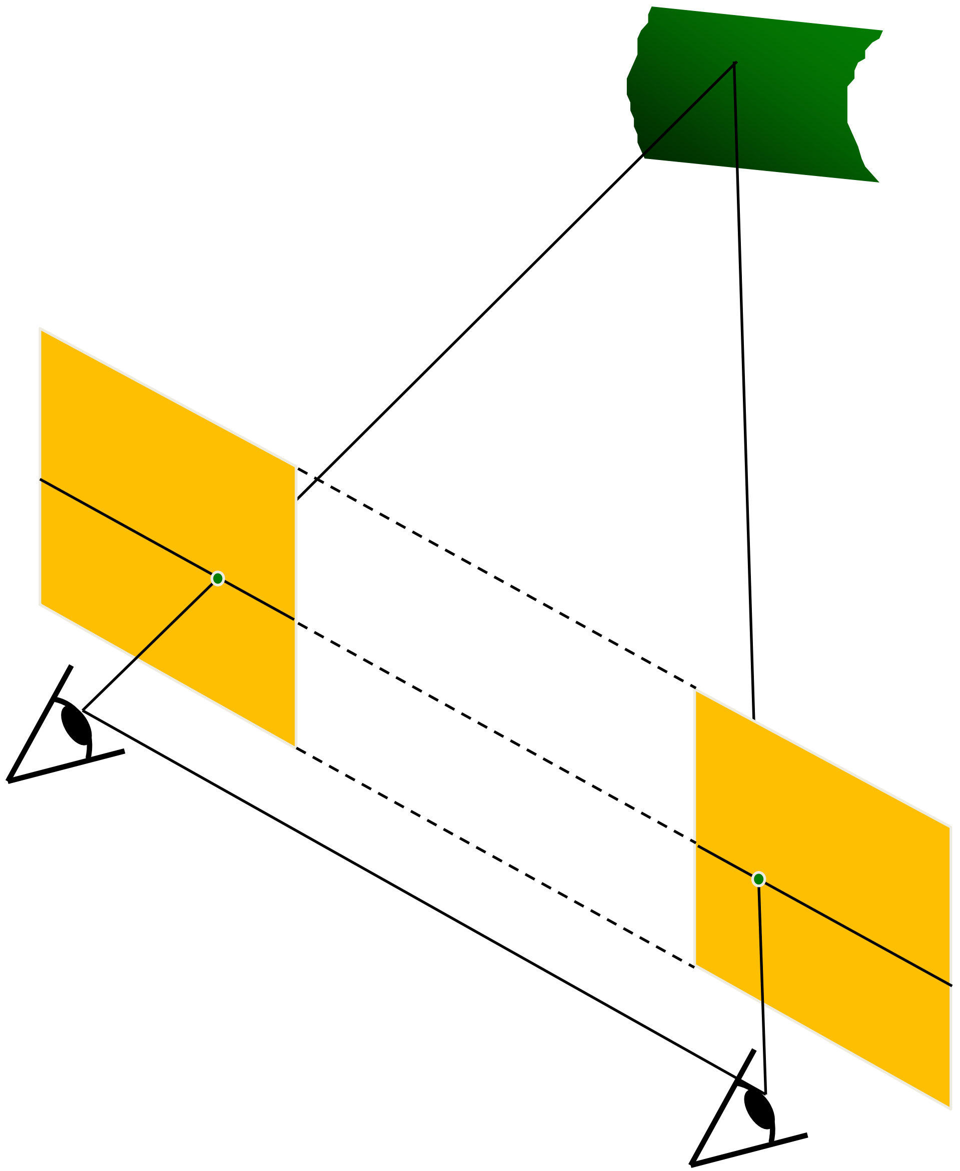
Stereo and the Epipolar constraint



Potential matches for *x* have to lie on the corresponding line *l'*.

Potential matches for *x'* have to lie on the corresponding line *l*.

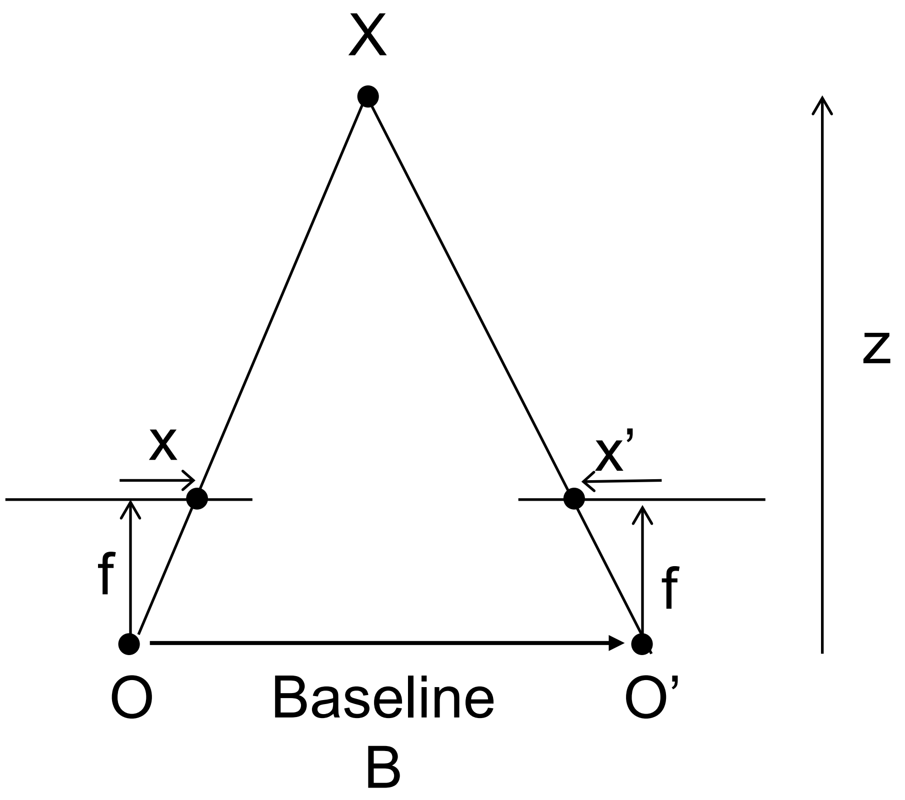
Simplest Case: Parallel images



- Image planes of cameras are parallel to each other and to the baseline
- Camera centers are at same height
- Focal lengths are the same
- Then, epipolar lines fall along the horizontal scan lines of the images

Depth from disparity

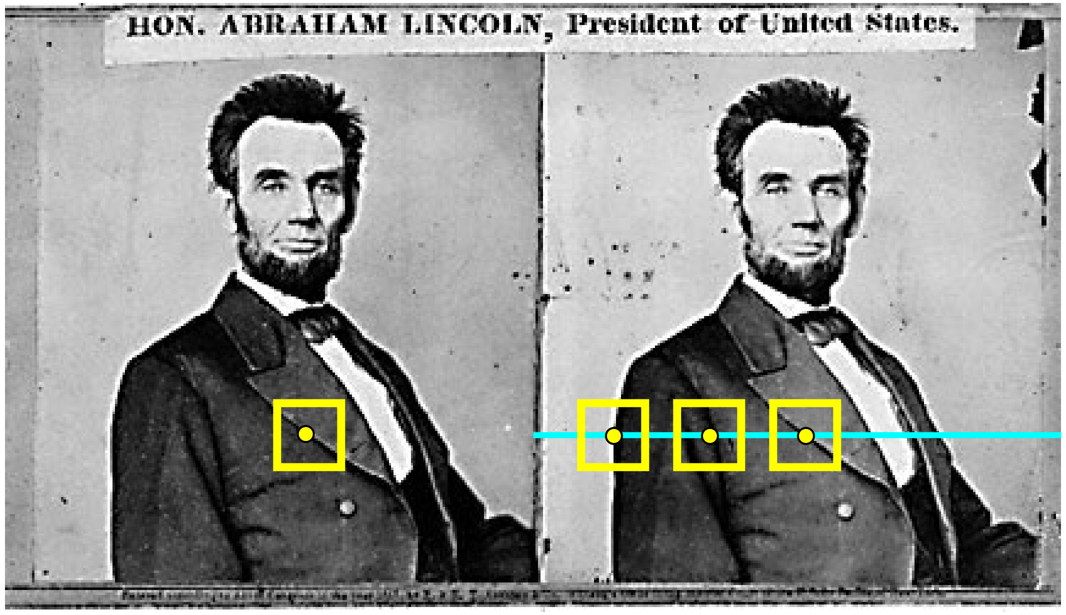
$$\frac{x - x'}{O - O'} = \frac{f}{z}$$



$$disparity = x - x' = \frac{B \cdot f}{z}$$

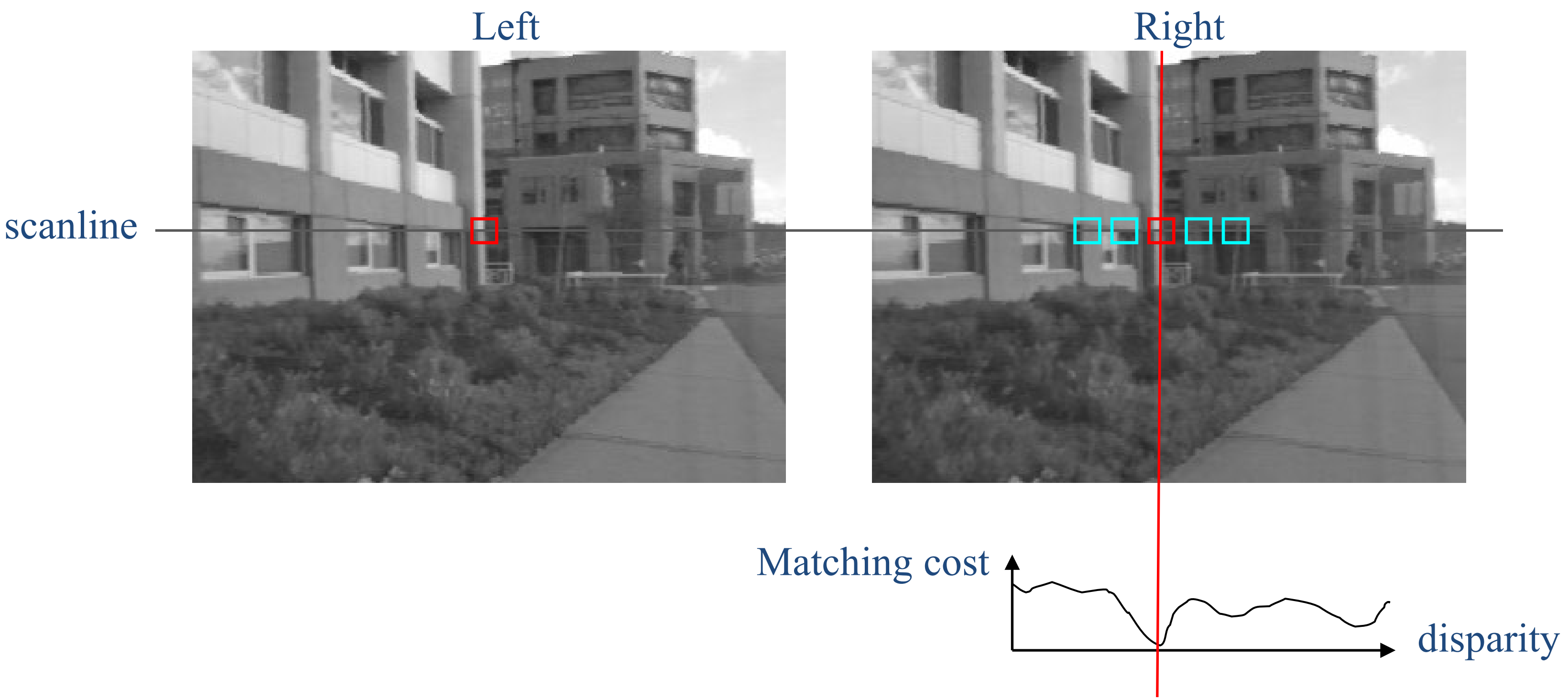
Disparity is inversely proportional to depth.

Basic stereo matching algorithm



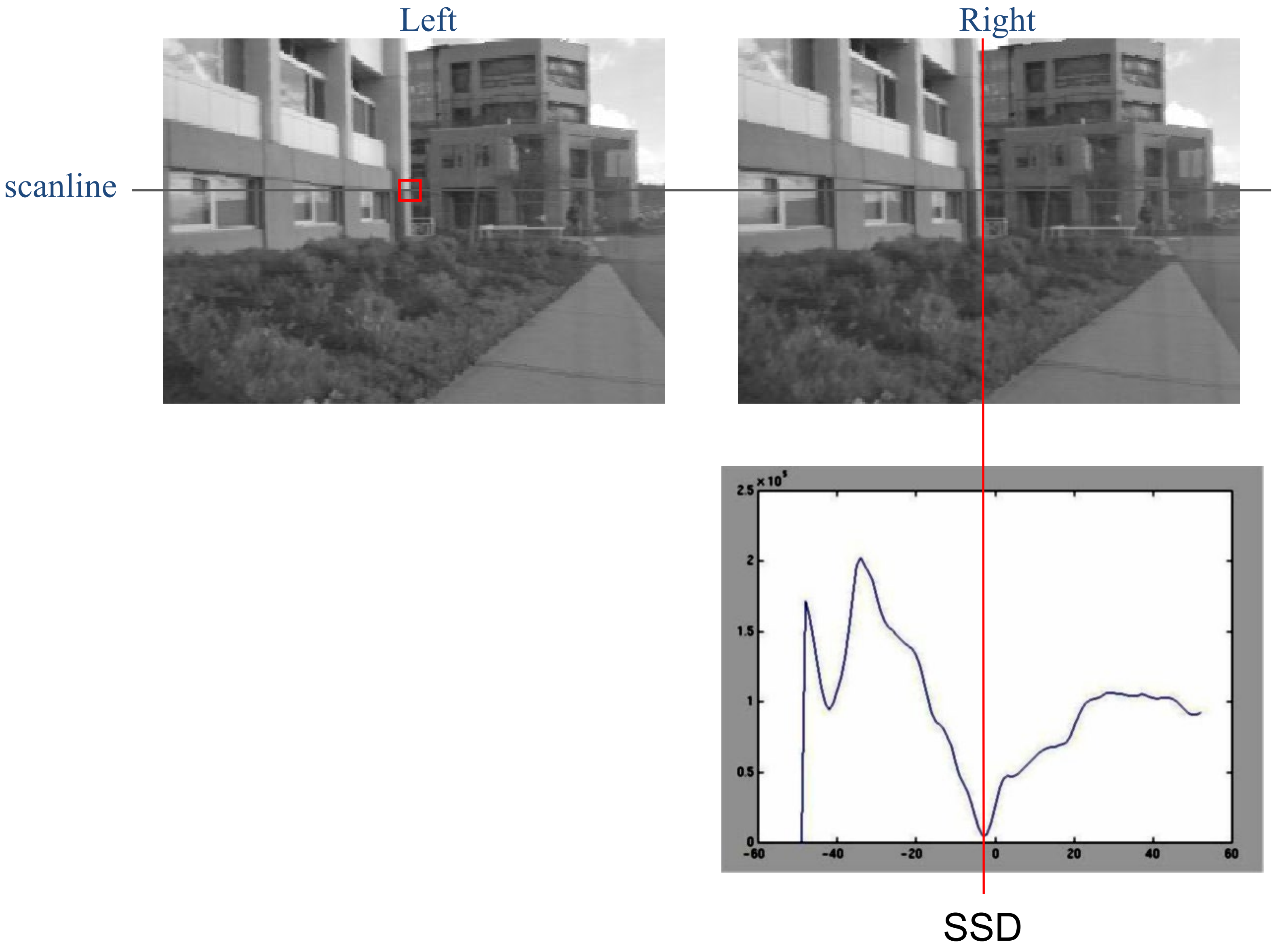
- If necessary, rectify the two stereo images to transform epipolar lines into scanlines
- For each pixel x in the first image
  - Find corresponding epipolar scanline in the right image
  - Examine all pixels on the scanline and pick the best match x’
  - Compute disparity x-x’ and set depth(x) = fB/(x-x’)

Correspondence search

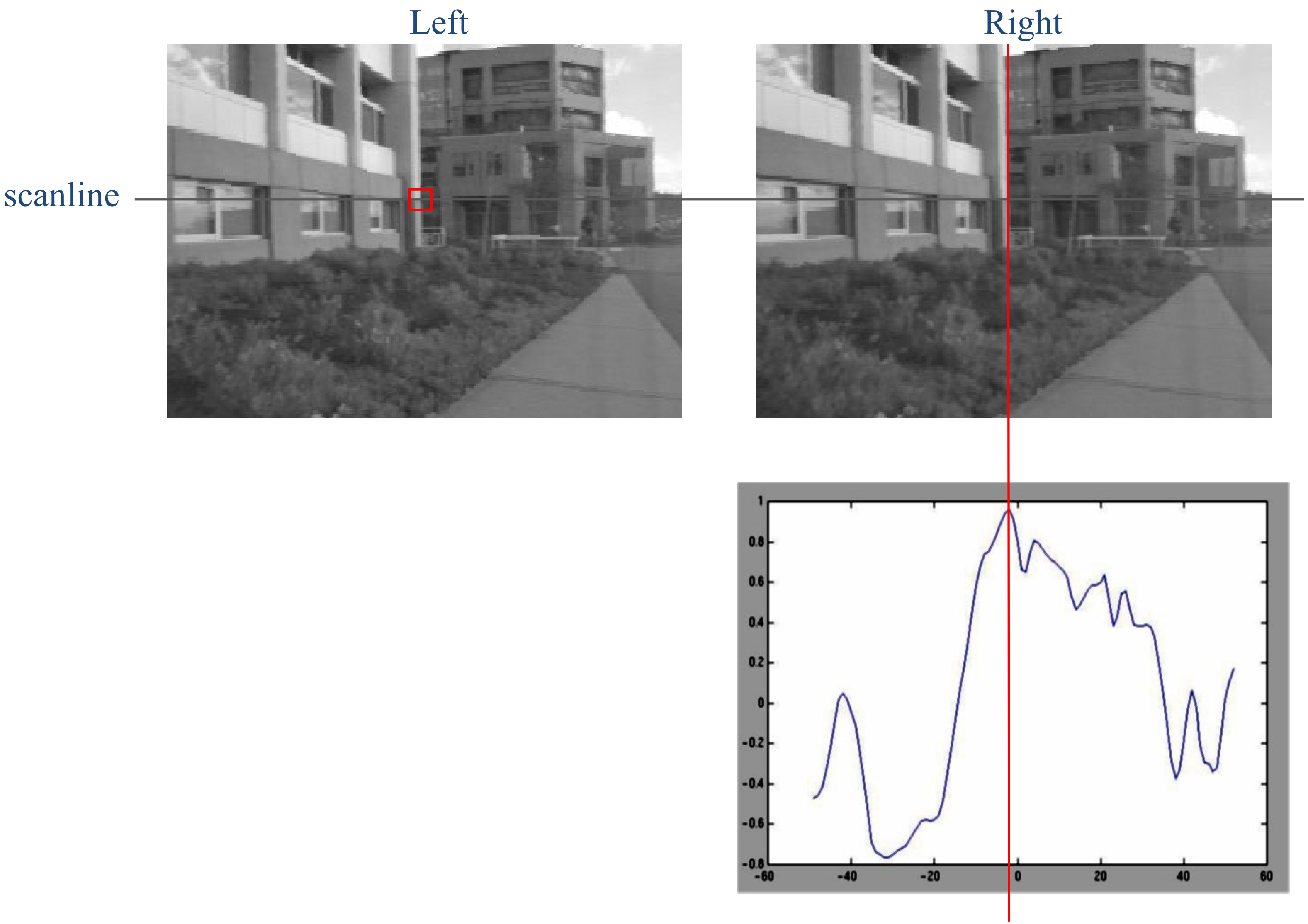


- Slide a window along the right scanline and compare contents of that window with the reference window in the left image
- Matching cost: SSD or normalized correlation

Correspondence search



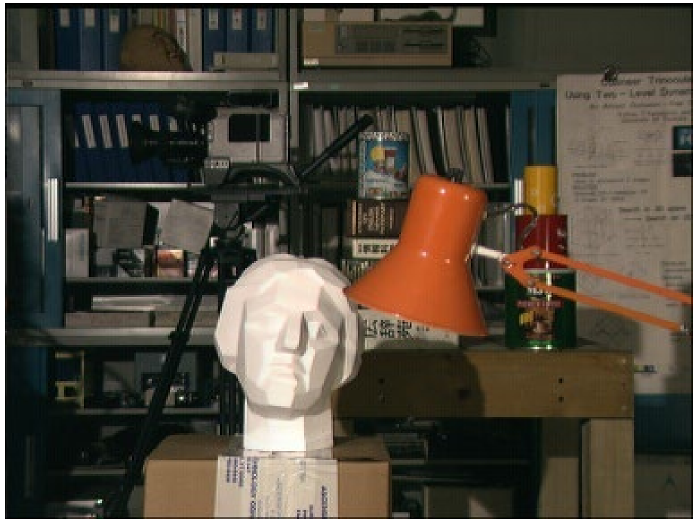
Correspondence search



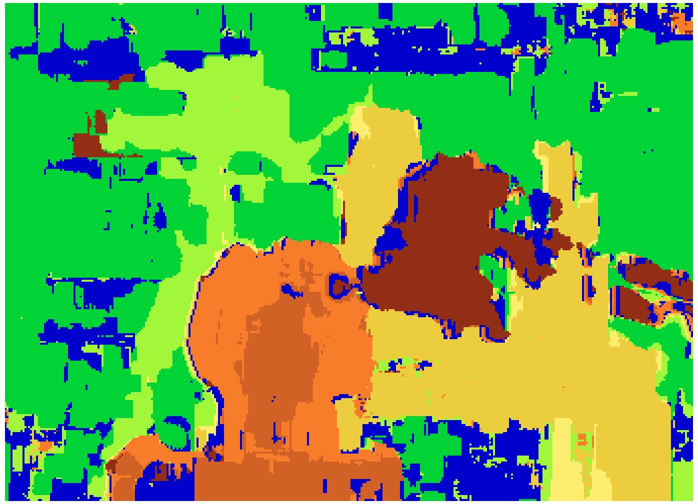
Norm. corr

Results with window search

Data



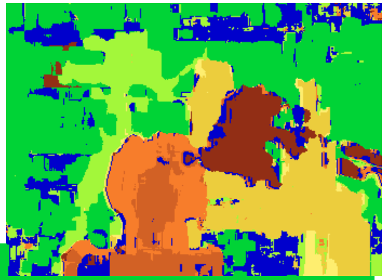
Window-based matching

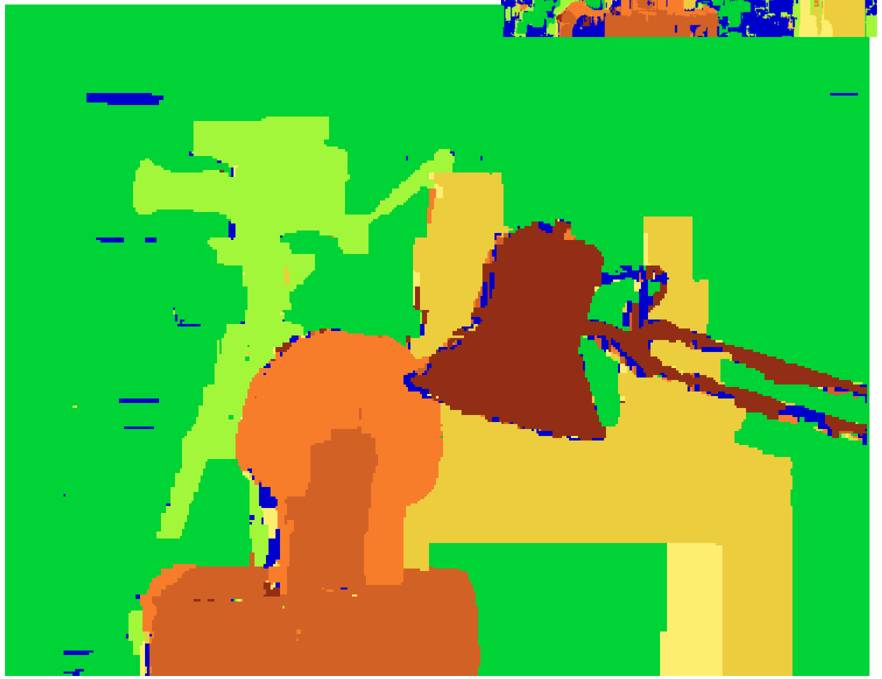


Ground truth

Add constraints and solve with graph cuts

Before





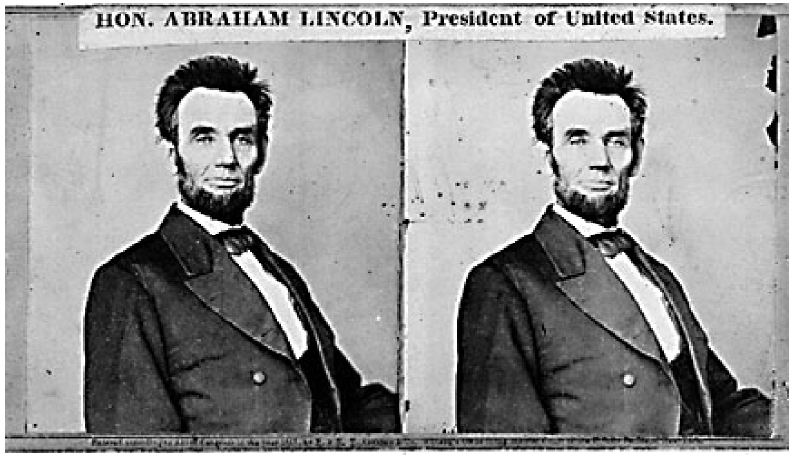
Graph cuts

Ground truth

Y. Boykov, O. Veksler, and R. Zabih, Fast Approximate Energy Minimization via Graph Cuts, PAMI 2001

For the latest and greatest: http://www.middlebury.edu/stereo/

Failures of correspondence search



Textureless surfaces

Occlusions, repetition

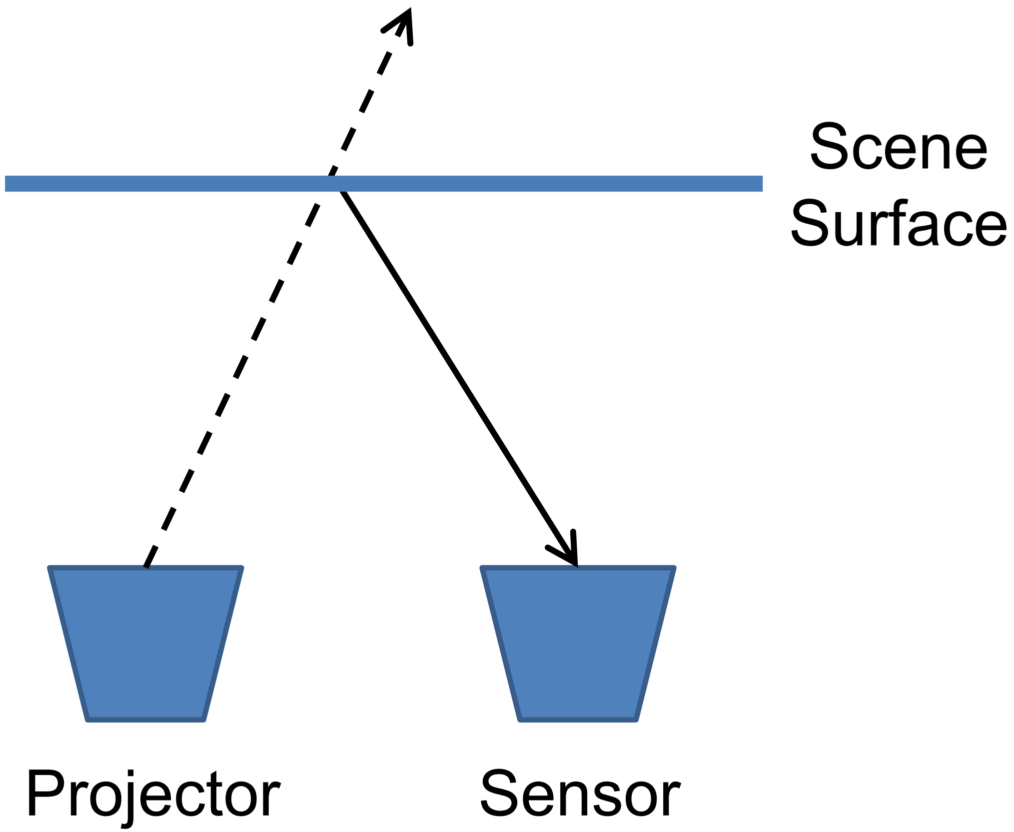
Non-Lambertian surfaces, specularities

Dot Projections

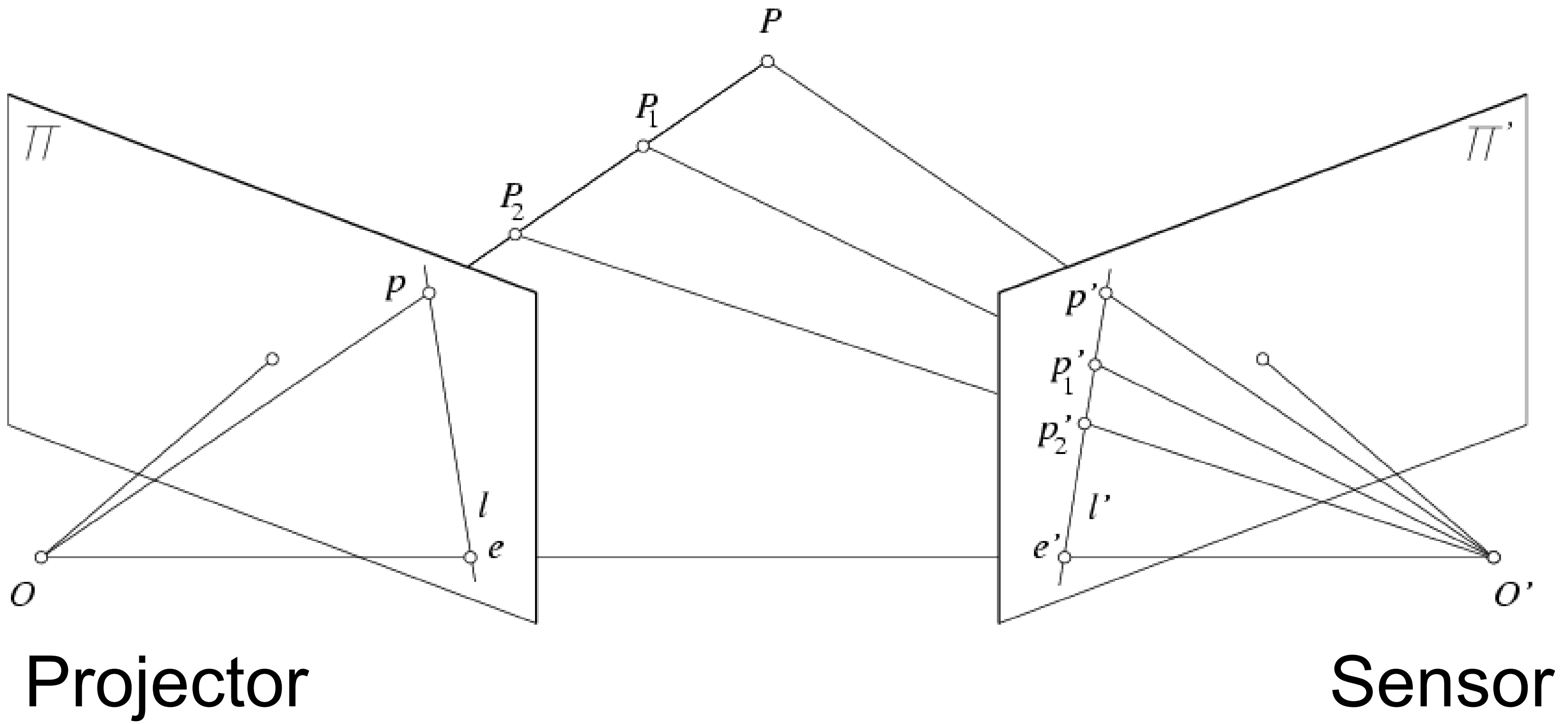
http://www.youtube.com/watch?v=28JwgxbQx8w

Depth from Projector-Sensor

Only one image: How is it possible to get depth?

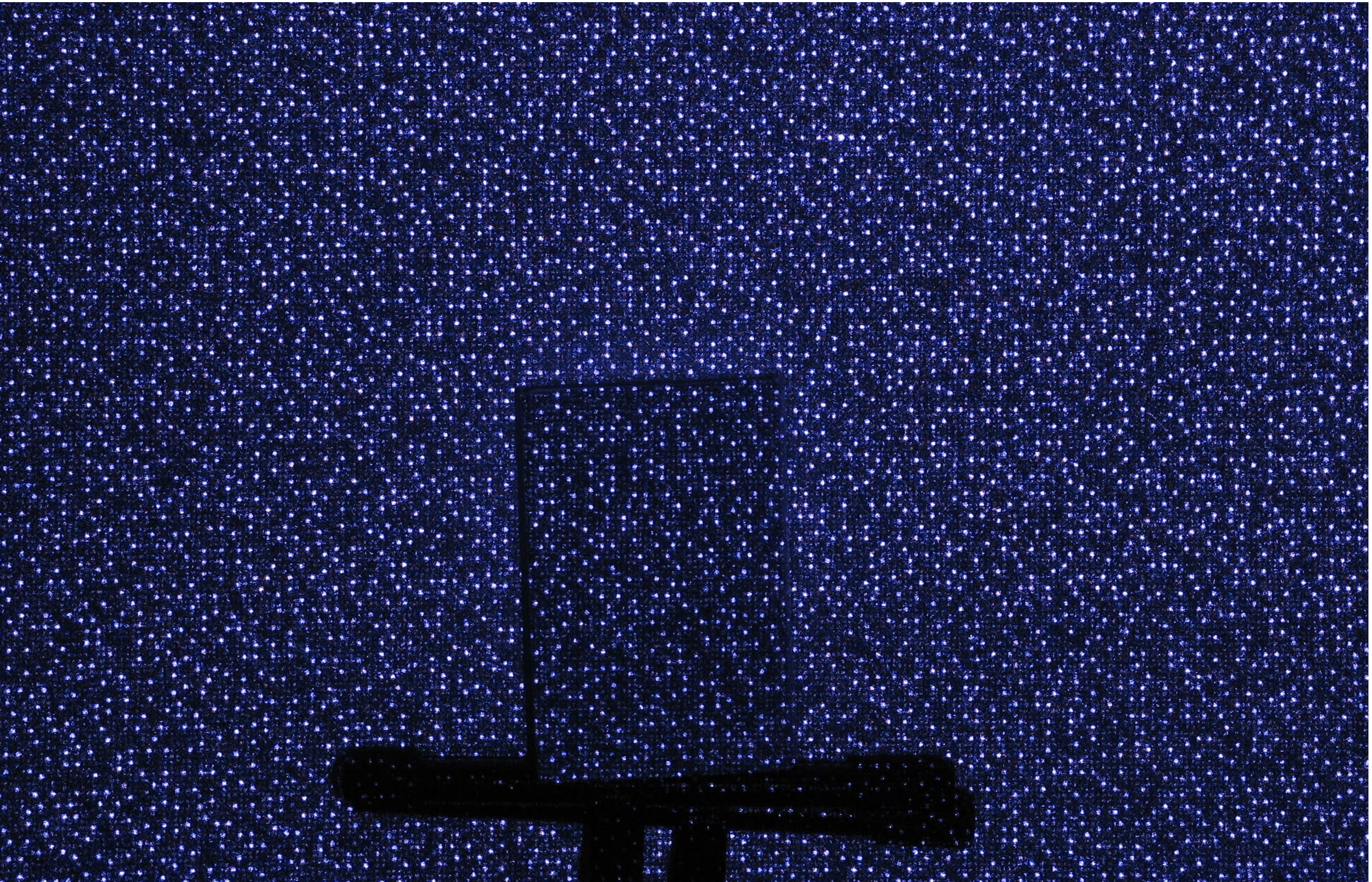


Same stereo algorithms apply



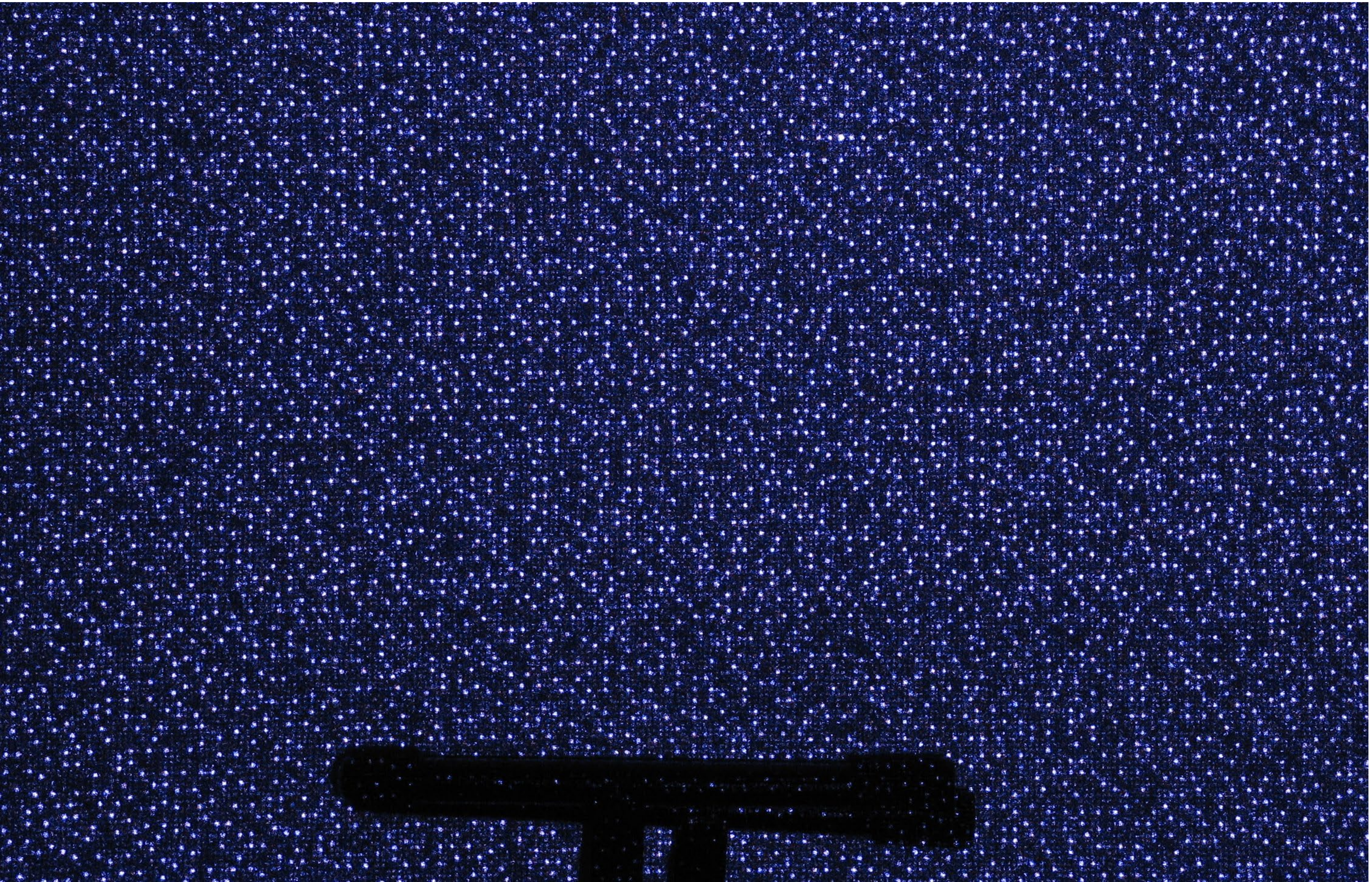
Source: http://www.futurepicture.org/?p=97

Example: Book vs. No Book



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

Example: Book vs. No Book



Region-growing Random Dot Matching

1. Detect dots ("speckles") and label them unknown
2. Randomly select a region anchor, a dot with unknown depth
   a. Windowed search via normalized cross correlation along scanline
      – Check that best match score is greater than threshold; if not, mark as "invalid" and go to 2
   b. Region growing
      1. Neighboring pixels are added to a queue
      2. For each pixel in queue, initialize by anchor's shift; then search small local neighborhood; if matched, add neighbors to queue
      3. Stop when no pixels are left in the queue
3. Repeat until all dots have known depth or are marked "invalid"

http://www.wipo.int/patentscope/search/en/WO2007043036

Projected IR vs. Natural Light Stereo

- What are the advantages of IR?
  - Works in low light conditions
  - Does not rely on having textured objects
  - Not confused by repeated scene textures
  - Can tailor algorithm to produced pattern

- What are advantages of natural light?
  - Works outside, anywhere with sufficient light
  - Uses less energy
  - Resolution limited only by sensors, not projector

- Difficulties with both
  - Very dark surfaces may not reflect enough light
  - Specular reflection in mirrors or metal causes trouble

Uses of Kinect (part 1)

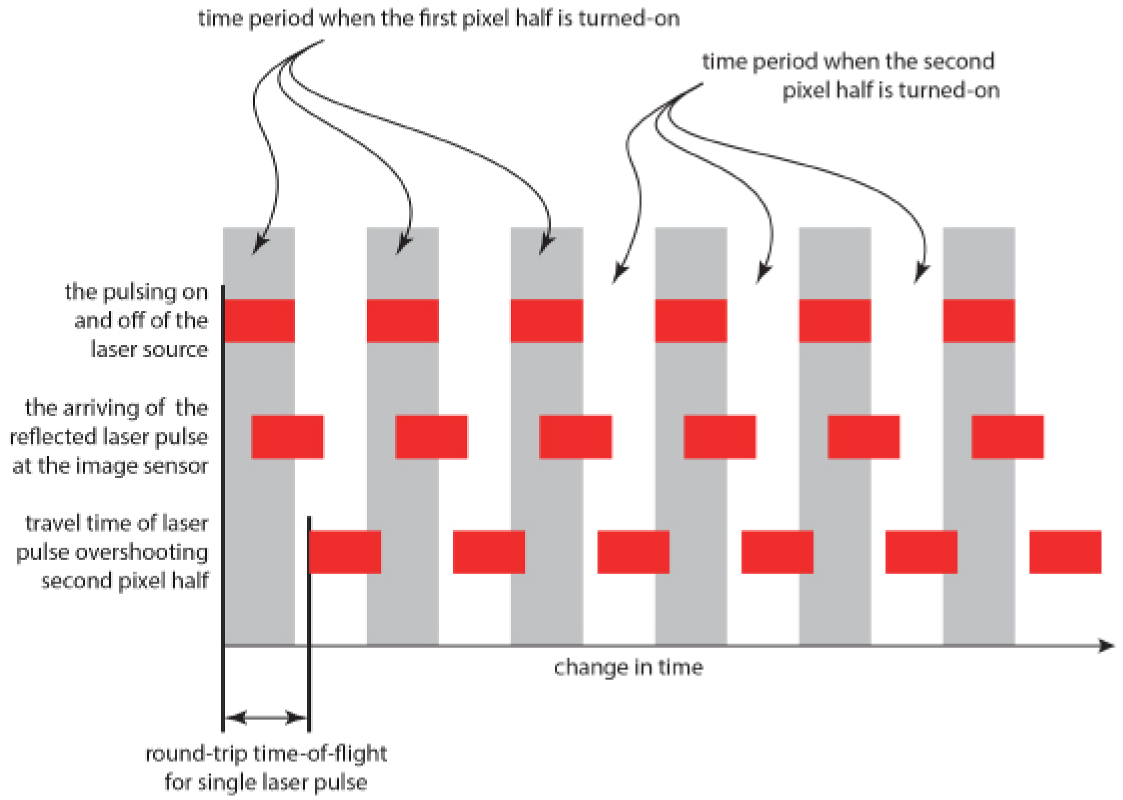
- 3D Scanner: http://www.youtube.com/watch?v=V7LthXRoESw
- IllumiRoom: http://research.microsoft.com/apps/video/default.aspx?id=191304
  https://www.youtube.com/watch?v=L2w-XqW7bF4

To learn more

- Kinect patents:
  http://www.faqs.org/patents/app/20100118123
  http://www.faqs.org/patents/app/20100020078
  http://www.faqs.org/patents/app/20100007717

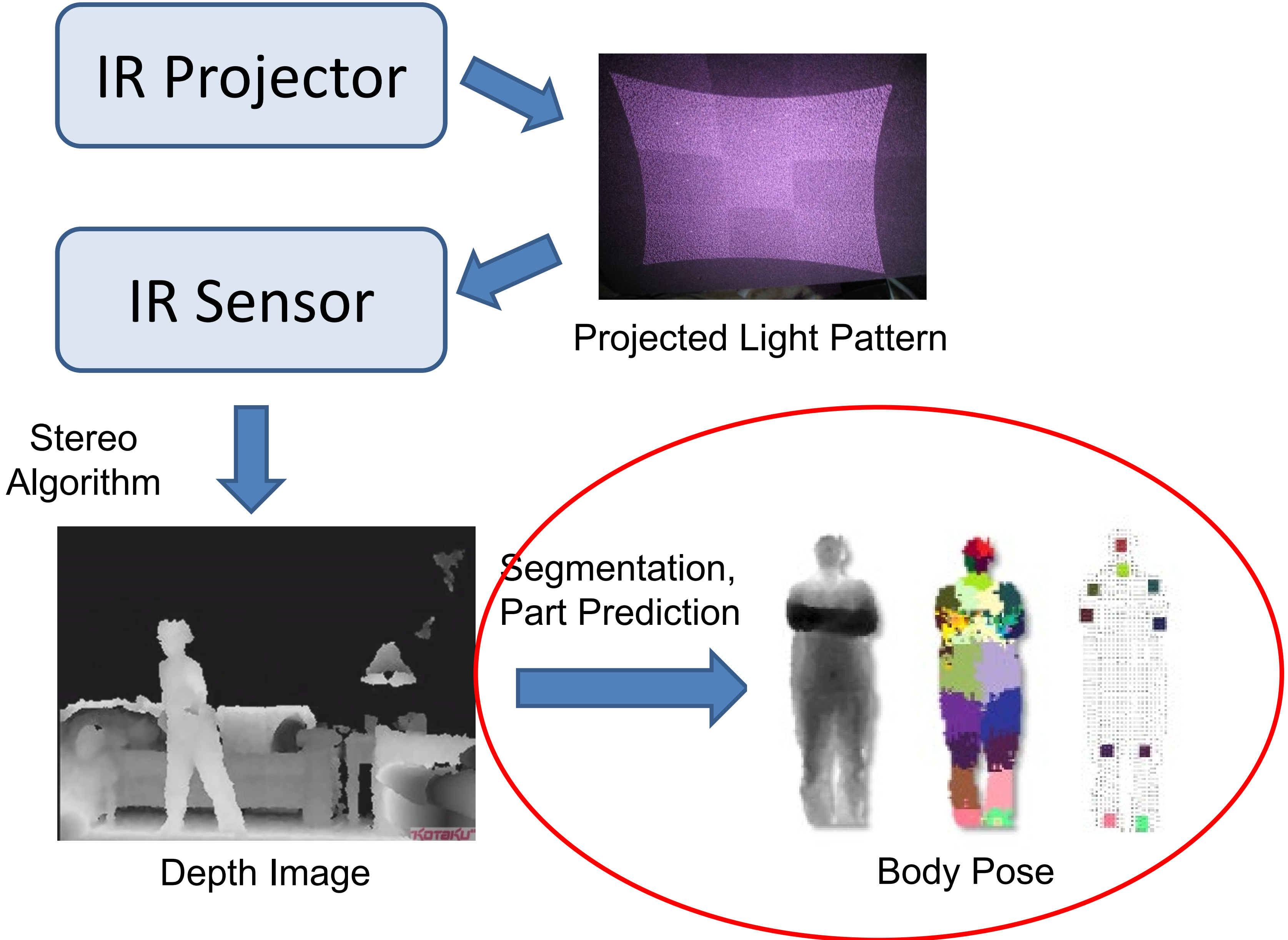
How does the Kinect v2 work?

- Time of flight sensor
  - Turn on and off two co-located pixels in alternation very quickly (e.g., 10 GHz)
  - Pulse laser just as quickly
  - Ratio of light achieved by the two pixels tells travel time (i.e., distance) of laser
  - By modifying the pulse frequency, you get a low-res and high-res estimate of travel time

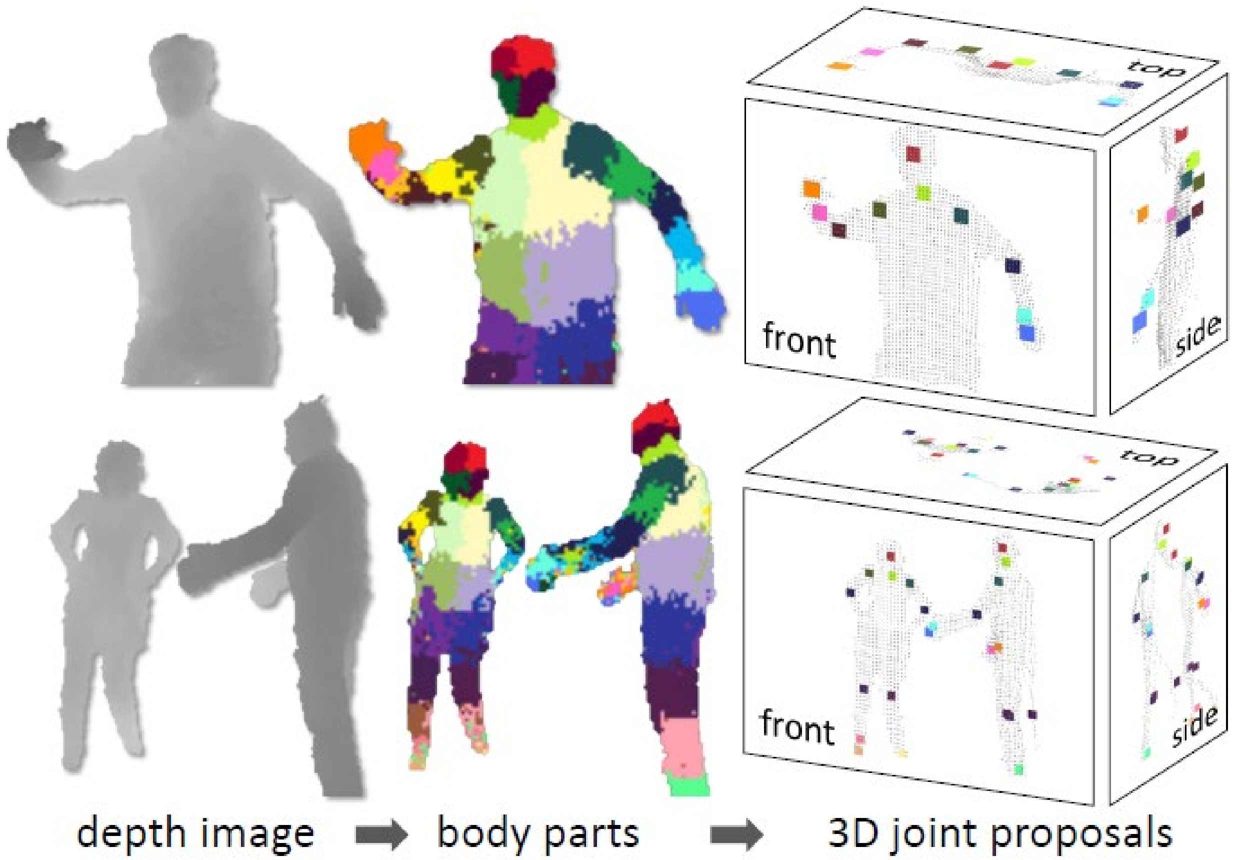


http://www.gamasutra.com/blogs/DanielLau/20131127/205820/The_Science_Behind_Kinects_or_Kinect_10_versus_20.php

Part 2: Pose from depth

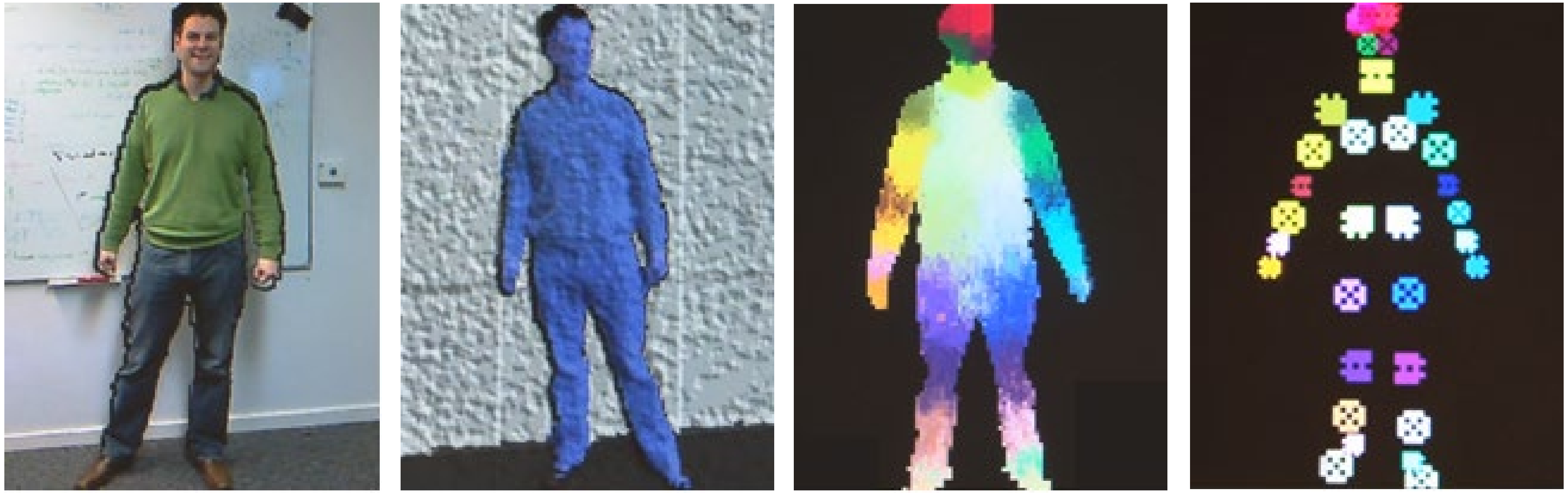


Goal: estimate pose from depth image



Real-Time Human Pose Recognition in Parts from a Single Depth Image
Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake
CVPR 2011

Goal: estimate pose from depth image



RGB Depth Part Label Map Joint Positions

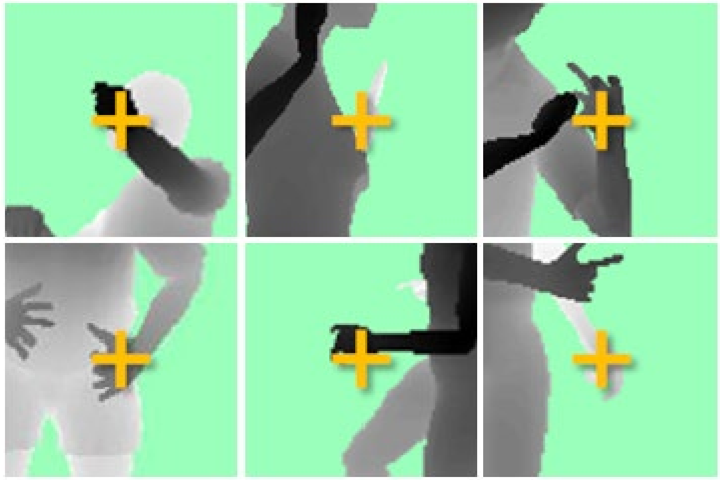
http://research.microsoft.com/apps/video/default.aspx?id=144455

Challenges

- Lots of variation in bodies, orientation, poses
- Needs to be very fast (their algorithm runs at 200 FPS on the Xbox 360 GPU)

Pose Examples

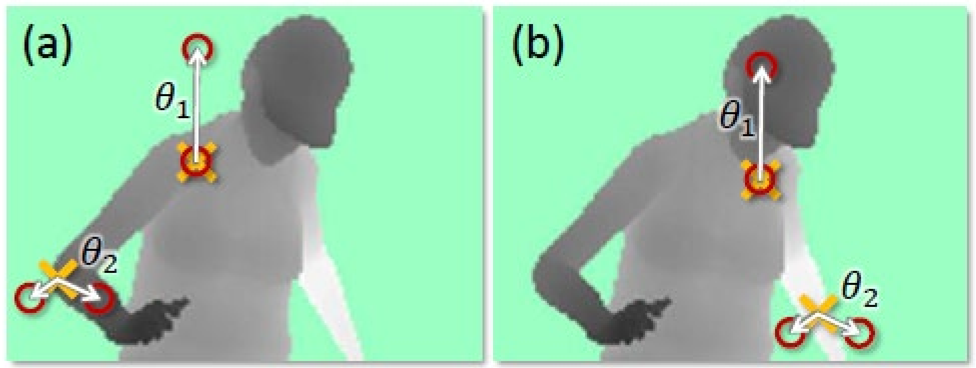
Examples of one part



Extract body pixels by thresholding depth

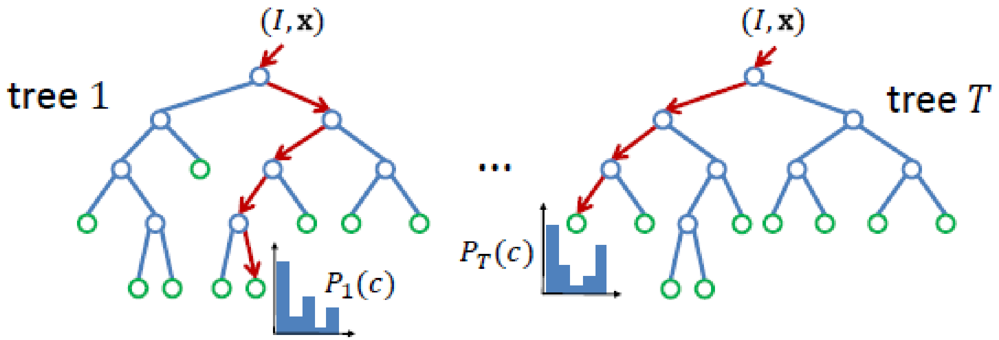
Basic learning approach

- Very simple features



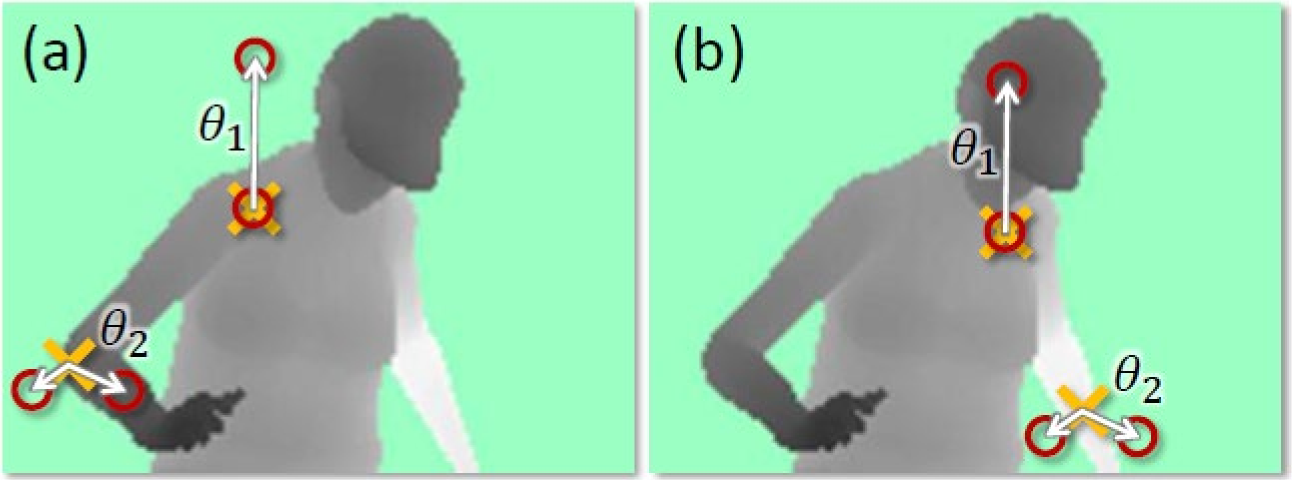
- Lots of data

- Flexible classifier



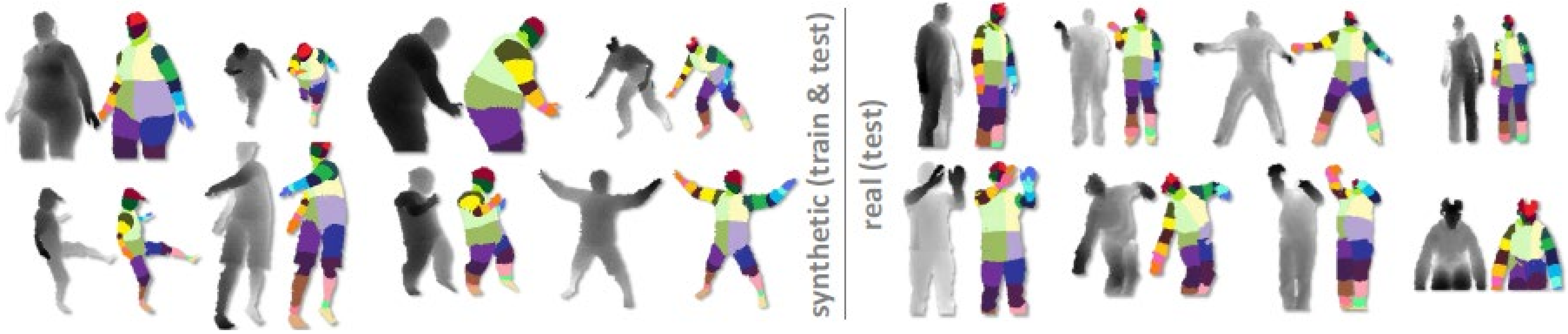
Features

- Difference of depth at two offsets
  - Offset is scaled by depth at center

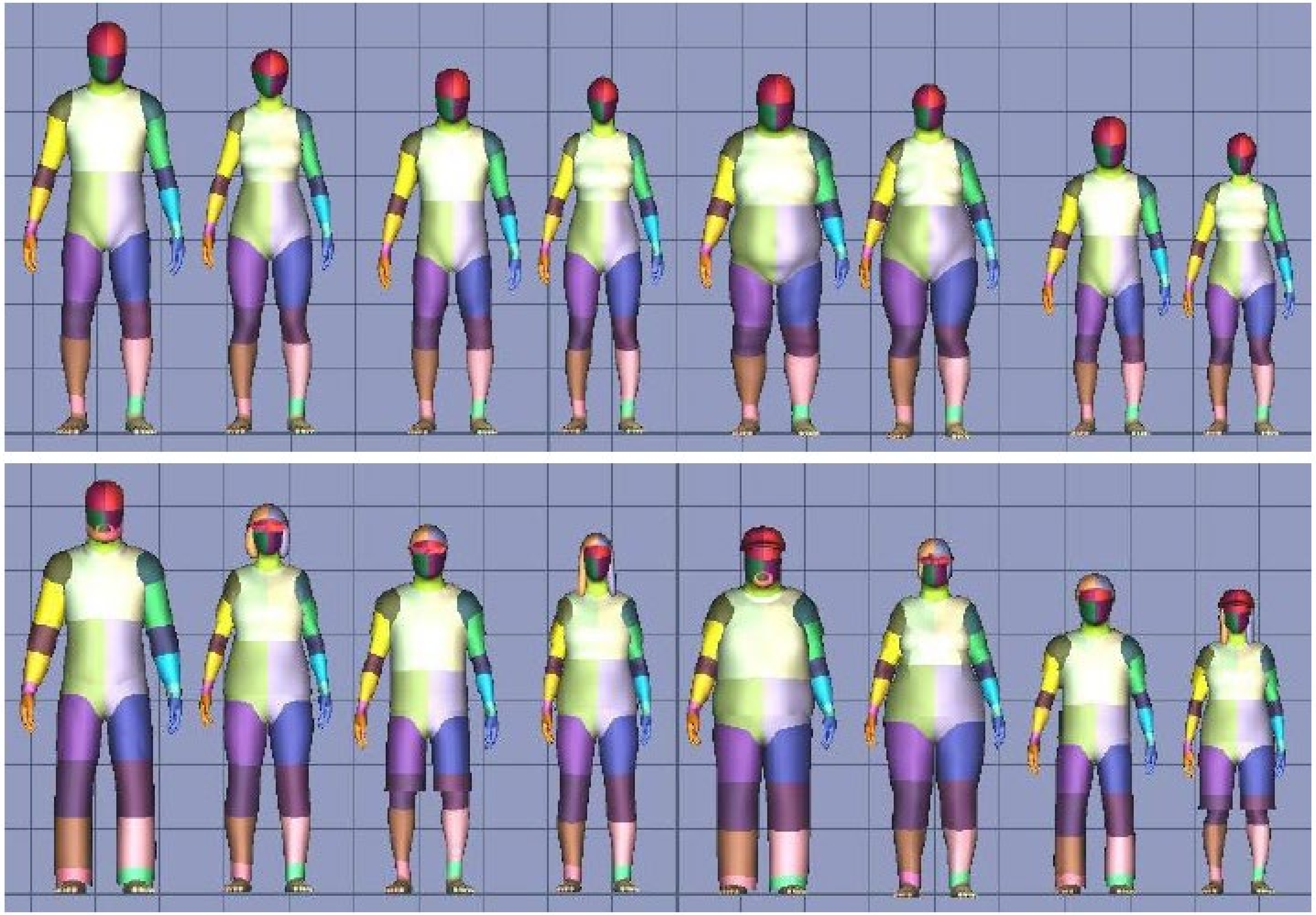


Get lots of training data

- Capture and sample 500K mocap frames of people kicking, driving, dancing, etc.
- Get 3D models for 15 bodies with a variety of weight, height, etc.
- Synthesize mocap data for all 15 body types

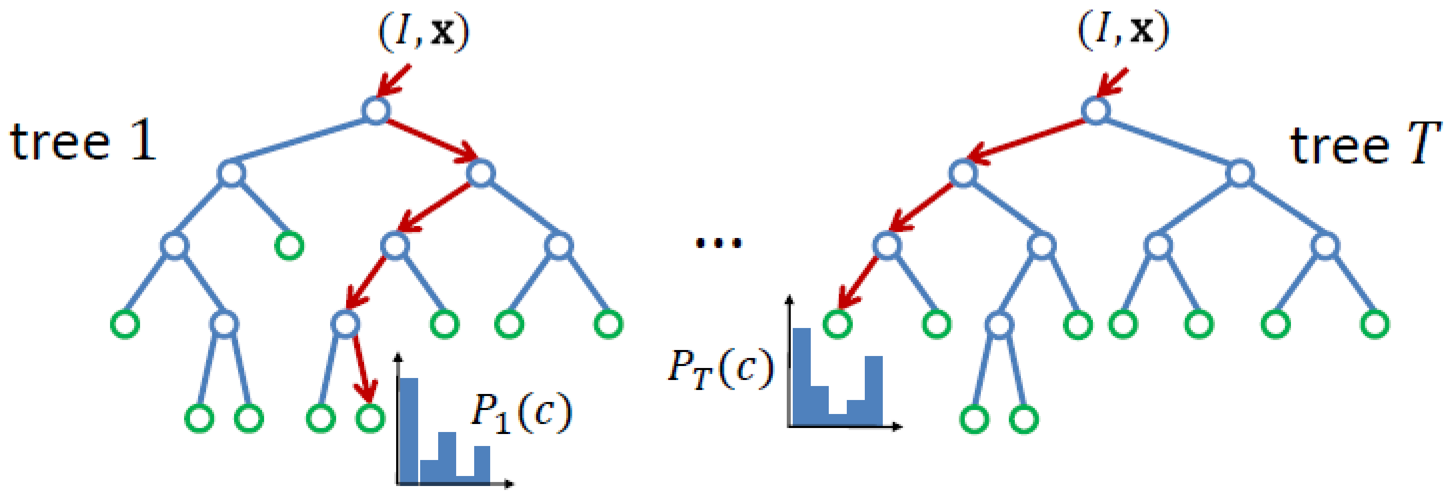


Body models



Part prediction with random forests

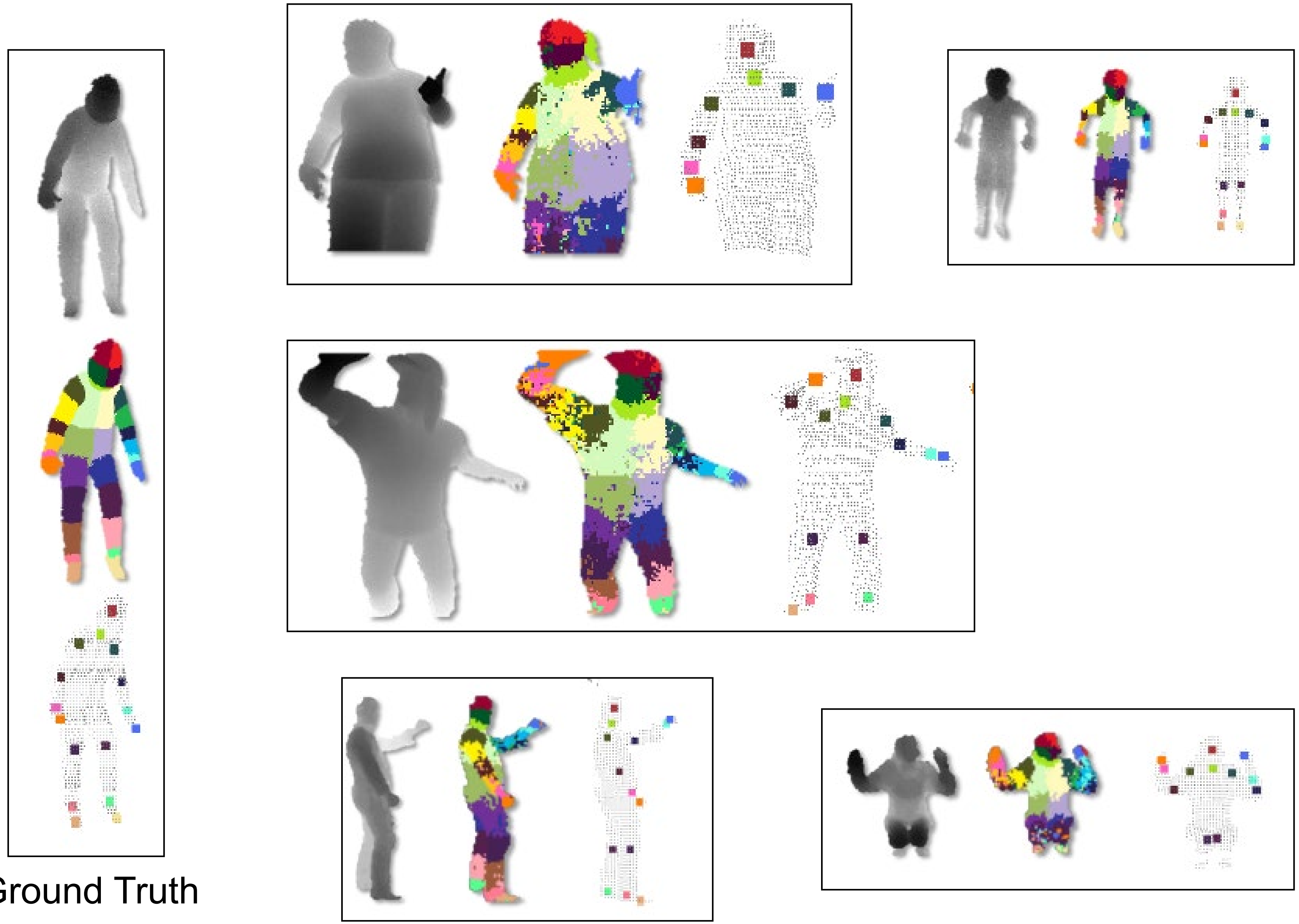
- Randomized decision forests: collection of independently trained trees
- Each tree is a classifier that predicts the likelihood of a pixel belonging to each part
  - Node corresponds to a thresholded feature
  - The leaf node that an example falls into corresponds to a conjunction of several features
  - In training, at each node, a subset of features is chosen randomly, and the most discriminative is selected



Joint estimation

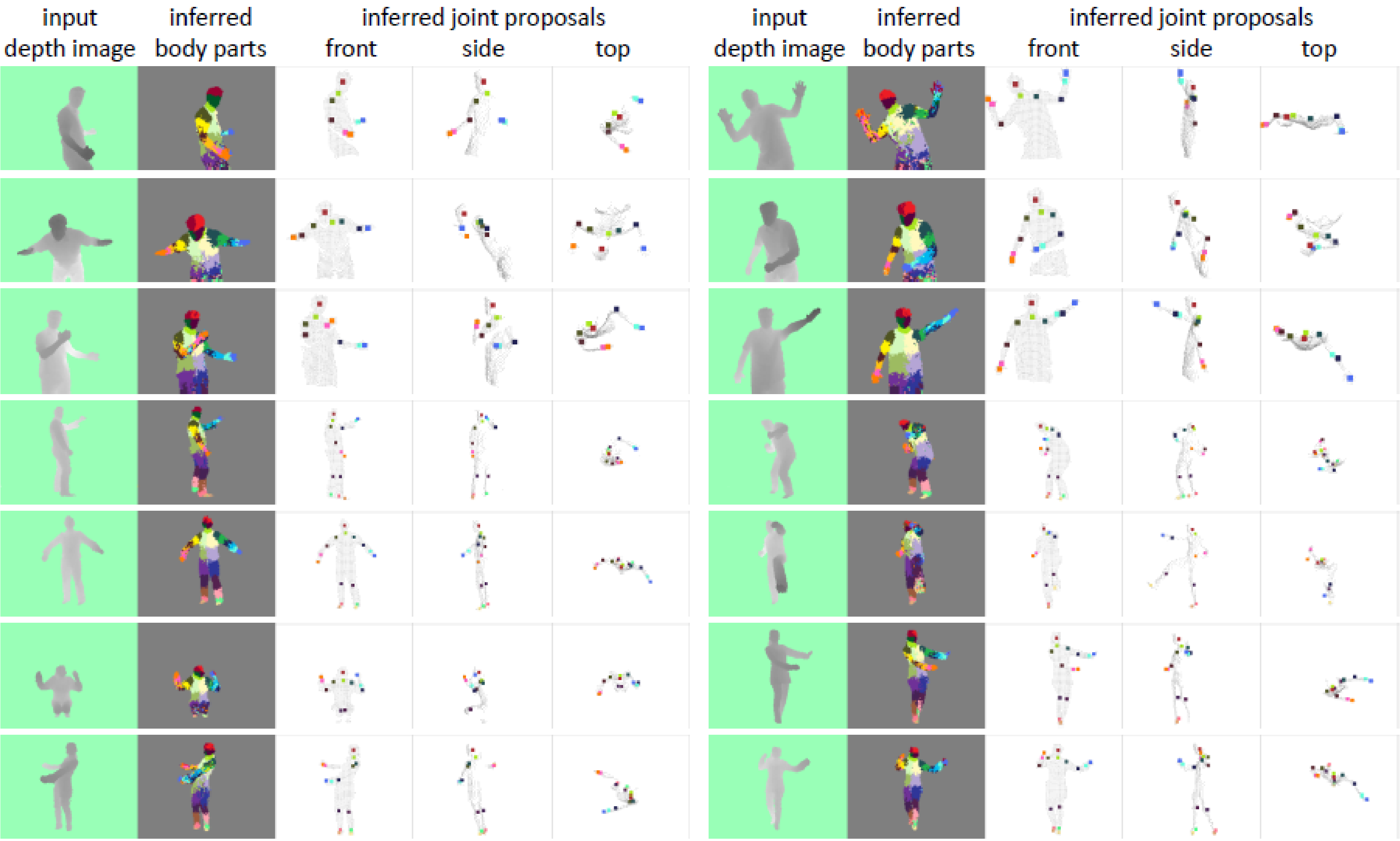
- Joints are estimated using mean-shift (a fast mode-finding algorithm)
- Observed part center is offset by pre-estimated value

Results

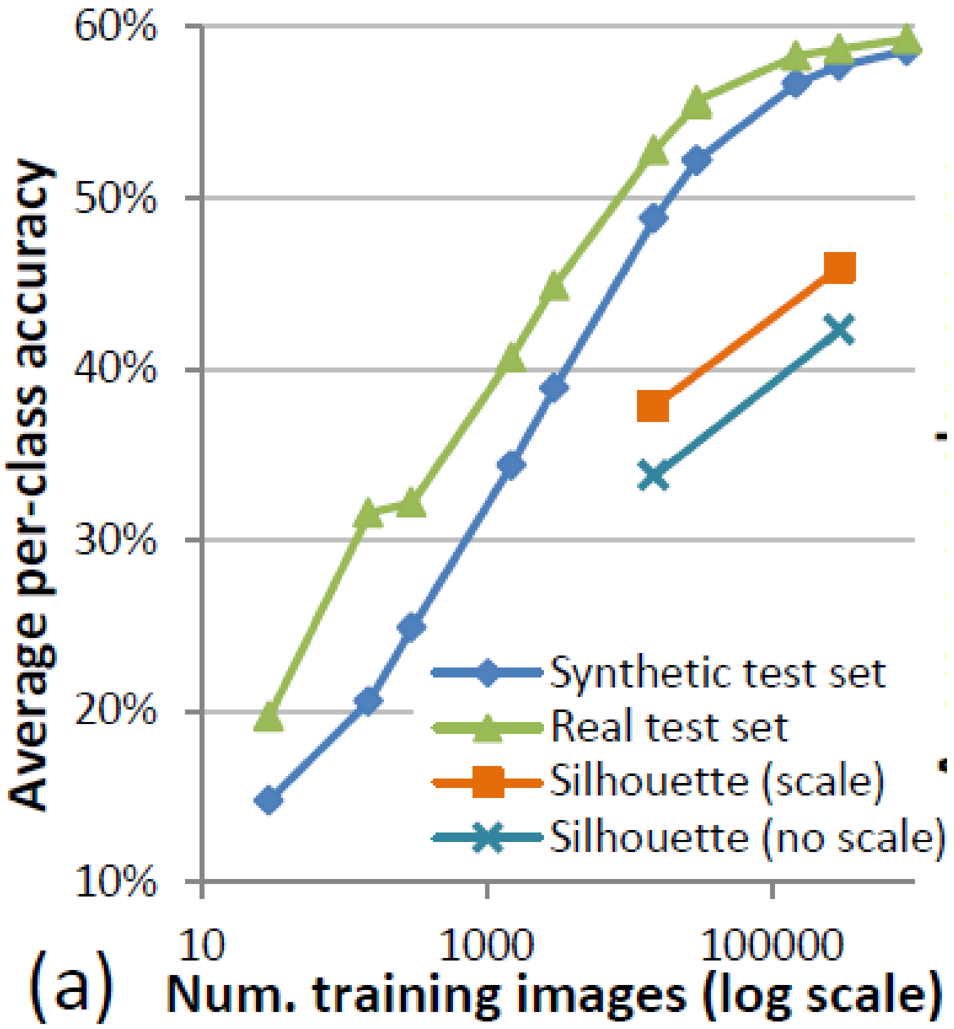


Ground Truth

More results



Accuracy vs. Number of Training Examples



Uses of Kinect (part 2)

- Mario: http://www.youtube.com/watch?v=8CTJL5lUjHg
- Robot Control: https://www.youtube.com/watch?v=7vq-1TiXi3g
- Capture for holography: http://www.youtube.com/watch?v=4LW8wgmfpTE
- Virtual dressing room: http://www.youtube.com/watch?v=1jbvnk1T4vQ
- Fly wall: http://vimeo.com/user3445108/kiwibankinteractivewall
- Sign language translator: https://www.youtube.com/watch?v=HnkQyUo3134

Note: pose from rgb video works amazingly well now

https://www.youtube.com/watch?v=pW6nZXeWlGM&t=77s

OpenPose: https://github.com/CMU-Perceptual-Computing-Lab/openpose

Coming up

- NeRF – Neural Radiance Fields