Kinect
Kinect

- RGBD sensor
- Frame rate
- Active sensing.

Fig. 2. Hardware configuration of Kinect, on which we point out the location of each sensor. Additionally, two image samples captured by the RGB camera and the depth camera are provided.
Slides from Larry Zitnick, Microsoft
Why Stereo Vision?

• 2D images project 3D points into 2D:

\[ P' = Q' \]

• 3D Points on the same viewing line have the same 2D image:
  - 2D imaging results in depth information loss

(Camps)
Stereo

- Assumess (two) cameras.
- Known positions.
- Recover depth.
Recovering Depth Information:

Depth can be recovered with two images and triangulation.

(Camps)
So Stereo has two steps

- Finding matching points in the images
- Then using them to compute depth.
Epipolar Constraint

- Most powerful correspondence constraint.
- Simplifies discussion of depth recovery.
Stereo correspondence

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

• Epipolar Constraint
  - Reduces correspondence problem to 1D search along conjugate epipolar lines
    (Seitz)
Fig. 3. Illustration of Kinect depth measurement.
(Varshney)
Reconstruction

\[ \frac{T + x_i - x_r}{Z - f} = \frac{T}{Z} \]

\[ Z = f \frac{T}{x_r - x_i} \]

Disparity \[ d = x_r - x_i \]

Then given \( Z \), we can compute \( X \) and \( Y \).

\( T \) is the stereo baseline
\( d \) measures the difference in retinal position between corresponding points

(Camps)
Depth

http://www.youtube.com/watch?v=inim0x
http://www.youtube.com/watch?v=7TGF30-5KuQ&feature=related
Postprocessing

• Holes
  – Occlusion
  – Specularity
  – Transparency
  – Scattering (hair, skin)

• Filtering
  – Fill in missing data from neighbors.
  – Use RGB data to get segments
Fig. 4. Example for hole-filling based on the bilateral filter [25]. (a) Raw depth image. (b) Depth image after filtering.
Pose Recognition

Figure 1. **Overview.** From an single input depth image, a per-pixel body part distribution is inferred. (Colors indicate the most likely part labels at each pixel, and correspond in the joint proposals). Local modes of this signal are estimated to give high-quality proposals for the 3D locations of body joints, even for multiple users.
Pose Recognition

- Depth image segmented into per-pixel labels for body parts.
- Then generate 3D locations for joints.
Finding body parts

• What should we use for a feature?
  – Difference in depth

• What should we use for a classifier?
  – Random Decision Forests
Pixel Classification with Machine Learning

- Get lots of training examples.
- Extract features for each pixel
- Train classifier that maps features to class (body part).
Synthetic data
Synthetic training/testing
Real test
$f_\theta(I, x) = d_I \left( x + \frac{u}{d_I(x)} \right) - d_I \left( x + \frac{v}{d_I(x)} \right)$
Classification

Learning:

1. Randomly choose a set of thresholds and features for splits.
2. Pick the threshold and feature that provide the largest information gain.
3. Recurse until a certain accuracy is reached or depth is obtained.
Implementation details

- 3 trees (depth 20) *(why so few?)*
- 300k unique training images per tree.
- 2000 candidate features, and 50 thresholds
- One day on 1000 core cluster.
Results
Joint estimation

• Apply mean-shift clustering to the labeled pixels.
• “Push back” each mode to lie at the center of the part.
Results
Failures

• Why would the system fail?
Video


Story about the making of Kinect: