Microsoft Kinect

May 30, 2012
Young Min Kim
Geometric Computing Group
Kinect Effect

• **Kinect Effect**
• Everybody can have an access to 3-D data
• Real-time processing
Technology

- Motion sensor
- Skeleton tracking
- Facial recognition
- Voice recognition
Sensors

Viewing angle
Vertical tilt range
Frame rate (depth and color stream)
Audio format

43° vertical by 57° horizontal field of view
±27°
30 frames per second (FPS)
16-kHz, 24-bit mono pulse code modulation (PCM)

Audio input characteristics

A four-microphone array with 24-bit analog-to-digital converter (ADC) and Kinect-resident signal processing including acoustic echo cancellation and noise suppression
How it works

• **How Kinect works**
Noise characteristics

• Distance vs. noise
Noise characteristics

• Distance vs. noise
• Material property
• Missing & flying pixels
Noise characteristics

- Distance vs. noise
- Material property
- Missing & flying pixels
- Quantization noise
Data structure: 2.5 D

- Point cloud
- Geometry processing
  - Mesh
- Computer vision
  - Frames of 2-D grids
- Robotics
  - Ray-based model, voxel grid
RGB-D mapping [Henry et al. 2010]
RGB-D mapping [Henry et al. 2010]
RGB-D mapping [Henry et al. 2010]
ICP (Iterative Closest Point) [Besl et al., 1992]

• Given two scans $\mathbf{P}$ and $\mathbf{Q}$.

• Iterate:
  – Find some pairs of closest points $(p_i, q_i)$
  – Find rotation $\mathbf{R}$ and translation $\mathbf{t}$ to minimize

\[
\min_{\mathbf{R}, \mathbf{t}} \sum_i \| p_i - \mathbf{R}q_i - \mathbf{t} \|^2
\]
RANSAC [Fischler et al., 1981]

- **RANdom Sample Consensus**
- **Parameter estimation, robust to outliers**
- **Algorithm**
  - **Input**
    - Data
    - $k$: minimum number of samples needed for parameter assumption
    - $e$: error threshold
    - $t$: minimum number of inliers
    - $N$: number of iterations
  - **iter=1:N**
    - Sample $k$ points from data
    - Solve for parameters with sampled points
    - Count number of inliers (within $e$)
    - If the number of inliers are more than $t$, then exit
RANSAC
RGB-D mapping [Henry et al. 2010]

\[ T^* = \arg\min_T \left( \frac{1}{|A_f|} \sum_{i \in A_f} w_i |T(f^i_s) - f^i_t|^2 \right) \]

\[ T^* = \arg\min_T \left( \frac{1}{|A_f|} \sum_{i \in A_f} |\text{Proj}(T(f^i_s)) - \text{Proj}(f^i_t)|^2 \right) \]
RGB-D mapping [Henry et al. 2010]

\[
T^* = \arg\min_T \left[ \alpha \left( \frac{1}{|A_f|} \sum_{i \in A_f} w_i \left| T(f_i^j) - f_i^j \right|^2 \right) \\
+ (1 - \alpha) \left( \frac{1}{|A_d|} \sum_{j \in A_d} w_j \left| (T(p_j^i) - p_j^i) \cdot n_i^j \right|^2 \right) \right]
\]

\[
T^* = \arg\min_T \left[ \left( \frac{1}{|A_f|} \sum_{i \in A_f} \left| \text{Proj}(T(f_i^j)) - \text{Proj}(f_i^j) \right|^2 \right) \\
+ \beta \left( \frac{1}{|A_d|} \sum_{j \in A_d} w_j \left| (T(p_j^i) - p_j^i) \cdot n_i^j \right|^2 \right) \right].
\]
RGB-D mapping [Henry et al. 2010]
RGB-D mapping [Henry et al. 2010]

- Loop closure detection
  - Feature matching
- Global optimization
  - Pose graph optimization
  - Sparse bundle adjustment
Loop closure detection

- Every frame to every other frame
- Key frames
  - Every n-th frame
  - Compute visual overlap
- Filter key frames
  - Estimated global pose
  - Place recognition algorithm
Pose graph optimization [Grisetti et al., 2009]

- Vertex: pose of camera
- Edge: constraint between a pair of vertices
- Uncertainty assigned for every edge
- Use tree structure for optimization
Sparse bundle adjustment [Lourakis et al., 2009]

- Minimize re-projection error of feature points

\[
\sum_{c_i \in C} \sum_{p_j \in P} v_{ij} \left| \text{Proj}(c_i(p_j)) - (\bar{u}, \bar{v}, d) \right|^2
\]
RGB-D mapping [Henry et al. 2010]

- Map visualization
Data structure: 2.5 D

• Point cloud
• Geometry processing
  – Mesh
• Computer vision
  – Frames of 2-D grids
• Robotics
  – Ray-based model, voxel grid
Data structure: 2.5 D

- **Point cloud**
- Geometry processing
  - Mesh
- Computer vision
  - Frames of 2-D grids
- Robotics
  - Ray-based model, voxel grid
Surfels [Pfister et al. 2000]

• Display purpose

• Components
  – Location
  – Normal
  – Patch size: inferred from distance & pixel size
  – Color: choose the most direct view
  – Confidence: calculated from the histogram of accumulated normal
RGB-D mapping [Henry et al. 2010]

- Surfels
In-hand 3D object modeling [Krainin et al., 2011]

- Real-time aspect
Interactivity

[Mistry et al 2009]
Algorithm

Fetch a new frame

- Initialization
  - Pair-wise registration
  - Plane extraction
- Success
  - Global adjustment
  - Map update
- Failure
  - Left click
  - Right click
- User interaction
  - Visual feedback
  - Adjust data path
  - Select planes
  - Start a new room

Left click: Select planes
Right click: Start a new room
Registration failure
Global Adjustment

\[ y \cdot x = a \cdot x = b \cdot x = c \]

\[ \Delta_1 \quad \Delta_2 \]
Global Adjustment

\[
\begin{align*}
\Delta_1 & = 1 \\
\Delta_2 & = 36
\end{align*}
\]
Global Adjustment

\[
\begin{align*}
\Delta_1 & = a \\
\Delta_2 & = c \\
\end{align*}
\]

\[
\min_{S^x} \sum_i \left( \frac{\| \Delta_i - m_i \|_2^2}{\sigma_i^2} \right) \quad \text{s. t.} \quad c_{j1} = c_{j2}, \quad \forall (c_{j1}, c_{j2}) \in C^x
\]
Selecting components

(a) 

(b) 

(c)
Floor plan generation
Floor plan generation
Kinect Fusion [Izadi et al. 2011]
Kinect Fusion [Izadi et al. 2011]
Kinect Fusion [Izadi et al. 2011]

Use 2-D grid to estimate the normals

\[
\frac{\partial x}{\partial s} \times \frac{\partial x}{\partial t}.
\]
Kinect Fusion [Izadi et al. 2011]

Dense ICP using GPU
Projective data association
Kinect Fusion [Izadi et al. 2011]
Signed distance function [Curless et al. 1996]

\[
D(x) = \frac{\sum w_i(x) d_i(x)}{\sum w_i(x)}
\]
Signed distance function [Curless et al. 1996]
Signed distance function [Curless et al. 1996]
Data structure: 2.5 D

• Point cloud
• Geometry processing
  – Mesh
• Computer vision
  – Frames of 2-D grids
• Robotics
  – Ray-based model, voxel grid
Data structure: 2.5 D

• Point cloud
• Geometry processing
  — Mesh
• Computer vision
  — Frames of 2-D grids
• **Robotics**
  — Ray-based model, voxel grid
Kinect Fusion [Izadi et al. 2011]

Position: tri-linear interpolated grid position
Normal: $\nabla \text{sdf}(p)$
Kinect Fusion [Izadi et al. 2011]
Kinect Fusion [Izadi et al. 2011]
Kinect Fusion [Izadi et al. 2011]
Kinect Fusion [Izadi et al. 2011]
Skeleton tracking [Shotton et al., 2011]

- What the Kinect is mainly used for
- Adapts idea from object recognition with parts
Skeleton tracking [Shotton et al., 2011]
Skeleton tracking [Shotton et al., 2011]

• Independent solution
  – Per pixel classification
  – Per frame classification

• Training data
  – Synthetic depth images from motion capture data

• Deep randomized decision forest, implemented with GPU (200 fps)

• Find joint proposal
Generating synthetic training data

• Motion capture data
  – Cover variety of poses (not motion)
  – Furthest neighbor clustering
• Generating synthetic data

Base character
Skinning hair and clothing
Generating synthetic training data

- 15 base characters
- Pose from motion capture data, mirroring with prob. 0.5
- Rotation and translation of character
- Hair and clothing
- Weight and height variation
- Camera position and orientation
- Camera noise
Body part labeling

- Intermediate representation
  - Can readily be solved by efficient classification algorithms
Depth image features

• Notation
  – Depth of pixel $\mathbf{x}$ at image $I$ $d_I(\mathbf{x})$
  – Parameters $\theta = (u, v)$

• Depth image feature

$$f_\theta(I, \mathbf{x}) = d_I\left(\mathbf{x} + \frac{u}{d_I(\mathbf{x})}\right) - d_I\left(\mathbf{x} + \frac{v}{d_I(\mathbf{x})}\right)$$
Data structure: 2.5 D

• Point cloud
• Geometry processing
  – Mesh
• Computer vision
  – Frames of 2-D grids
• Robotics
  – Ray-based model, voxel grid
Data structure: 2.5 D

- Point cloud
- Geometry processing
  - Mesh
- Computer vision
  - Frames of 2-D grids
- Robotics
  - Ray-based model, voxel grid
Randomized decision forests

- An ensemble of $T$ decision trees
- Split node has feature $f_\theta$ and threshold $\tau$
- Leaf node has distribution over body part $c$

$$P(c|I,x) = \frac{1}{T} \sum_{t=1}^{T} P_t(c|I,x)$$
Training [Lepetit et al., 2005]

1. Randomly propose a set of splitting candidates $\phi = (\theta, \tau)$ (feature parameters $\theta$ and thresholds $\tau$).

2. Partition the set of examples $Q = \{(I, x)\}$ into left and right subsets by each $\phi$:

$$Q_l(\phi) = \{ (I, x) \mid f_\theta(I, x) < \tau \}$$  \hspace{1cm} (3)

$$Q_r(\phi) = Q \setminus Q_l(\phi)$$  \hspace{1cm} (4)

3. Compute the $\phi$ giving the largest gain in information:

$$\phi^* = \arg\max_{\phi} G(\phi)$$  \hspace{1cm} (5)

$$G(\phi) = H(Q) - \sum_{s \in \{l, r\}} \frac{|Q_s(\phi)|}{|Q|} H(Q_s(\phi))$$  \hspace{1cm} (6)

where Shannon entropy $H(Q)$ is computed on the normalized histogram of body part labels $l_I(x)$ for all $(I, x) \in Q$.

4. If the largest gain $G(\phi^*)$ is sufficient, and the depth in the tree is below a maximum, then recurse for left and right subsets $Q_l(\phi^*)$ and $Q_r(\phi^*)$.

Training 3 trees to depth 20 from 1 million images takes about a day on a 1000 core cluster.
Randomized decision forests
Joint position proposals

• Mean shift with a weighted Gaussian kernel

\[
f_c(\hat{x}) \propto \sum_{i=1}^{N} w_{ic} \exp \left( - \frac{||\hat{x} - \hat{x}_i||^2}{b_c} \right)
\]

\[
w_{ic} = P(c|I, x_i) \cdot d_I(x_i)^2
\]

• Pushed backwards
Figure 6. **Training parameters vs. classification accuracy.** (a) Number of training images. (b) Depth of trees. (c) Maximum probe offset.
Conclusion

• Kinect revolution
  – 3-D data is available to everyone

• Data structure: between 2-D and 3-D
  – RGB-D mapping
  – Floor plan generation
  – Kinect fusion
  – Skeleton tracking
  – ...what else?
References

- RGB-D mapping
References

• Interactive system

• Kinect fusion
  – Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon, KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera, ACM Symposium on User Interface Software and Technology, October 2011

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