

Computer Vision Systems Programming VO

3D Vision Applications

Christopher Pramerdorfer

Computer Vision Lab, Vienna University of Technology

Topics

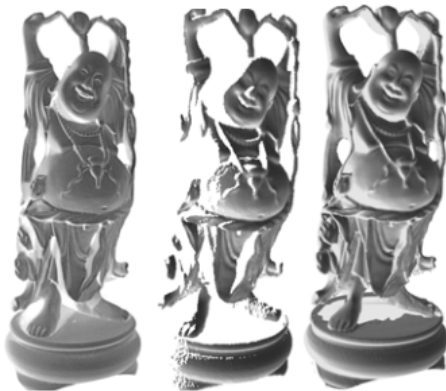
CV applications utilizing scene geometry (3D data)

- Focus on those based on Kinect



Images by Ryuzo Okada, Shotton et al. 2011, Newcombe et al. 2011

3D Reconstruction



Images from Curless and Levoy 1996

3D Reconstruction

Construction of accurate 3D models from range data

- ▶ Usually involves combining multiple point clouds

Accomplished in two steps

- ▶ Align range data (map to common coordinate system)
- ▶ Merge range data in a way that minimizes errors

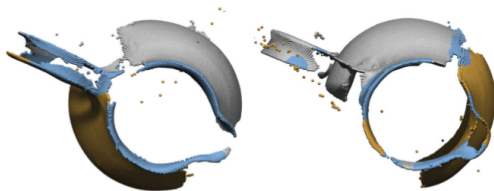
Often followed by surface reconstruction

3D Reconstruction

Range Data Alignment – Iterative Closest Points

Popular method for aligning two point clouds $\{\mathbf{r}\}$, $\{\mathbf{s}\}$

- ▶ Goal is to find parameters θ of some transformation \mathcal{T}
- ▶ Usually assuming a rigid transformation



Images from Aiger, Mitra, and Cohen-Or 2008

3D Reconstruction

Range Data Alignment – Iterative Closest Points

Algorithm iterates between

- ▶ Finding point correspondences based on distance, $\{(r_n, s_n)\}_n$
- ▶ Finding the θ that minimizes $\sum_n \|\mathbf{r}_{r_n} - \mathcal{T}(\mathbf{s}_{s_n}; \theta)\|_2^2$

Converges towards a local minimum

- ▶ Requires good initial estimate of θ

<https://www.youtube.com/watch?v=ii2vHBwlmo8>

3D Reconstruction

Range Data Merging – TSDF Fusion

Truncated signed distance functions (TSDFs)

- ▶ Similar to distance transforms in 3D (0 = surface)
- ▶ But distances are signed, measured along view rays

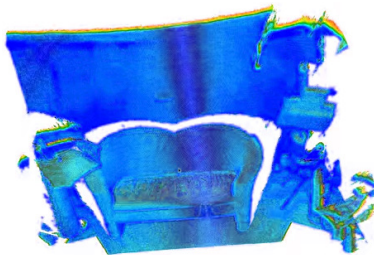


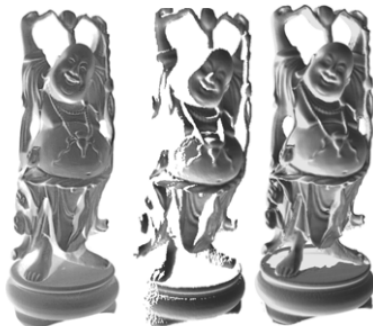
Image from <https://www.youtube.com/watch?v=AjjSZufyprU>

3D Reconstruction

Range Data Merging – TSDF Fusion

Merged data = weighted average over aligned TSDF voxels

- ▶ Weights based on e.g. object distance, angle



Images from Curless and Levoy 1996

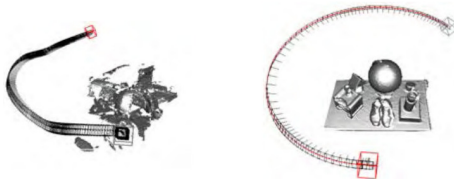
3D Reconstruction

Kinect Fusion

Temporal fusion of Kinect depth maps

Based on the above methods (ICP & TSDF fusion)

- ▶ But $\{\mathbf{r}\}$ is synthesized from merged model
- ▶ Suppresses alignment error accumulation

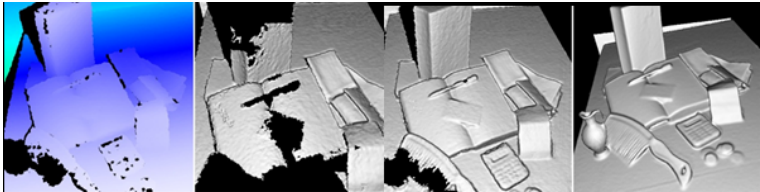


Images from Newcombe et al. 2011

3D Reconstruction

Kinect Fusion

<https://www.youtube.com/watch?v=quGhaggn3cQ>



Images from microsoft.com

3D Reconstruction

Surface Reconstruction

Reconstruction of surface mesh from point cloud

- ▶ Results in a (locally) watertight 3D model
- ▶ Allows for further processing (e.g. texturing)

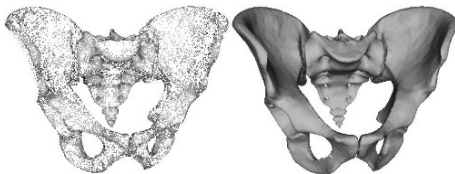


Image from Kazhdan 2005

3D Reconstruction

Surface Reconstruction

Correction of point cloud errors

- ▶ Noise, outliers, alignment errors, missing data

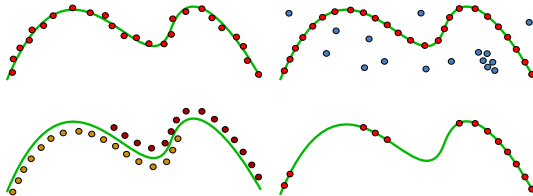


Image adapted from Berger et al. 2014

3D Reconstruction

Surface Reconstruction – Poisson Surface Reconstruction



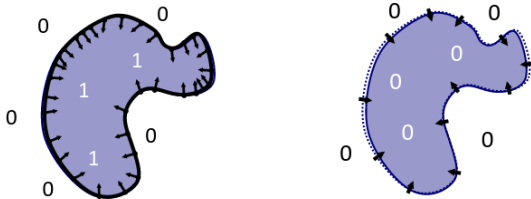
Images from Kazhdan, Bolitho, and Hoppe 2006

3D Reconstruction

Surface Reconstruction – Poisson Surface Reconstruction

Define $\chi(\mathbf{x}) = 1$ if \mathbf{x} inside the object, 0 otherwise

- ▶ Surface is at $\chi(\cdot) = 0.5$
- ▶ $\nabla\chi$ equals surface normal near surface, 0 otherwise



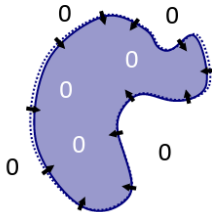
Images from Gotsman & Kazhdan's slides

3D Reconstruction

Surface Reconstruction – Poisson Surface Reconstruction

Regard oriented points $\{(\mathbf{x}, \mathbf{n})\}$ as samples from $\nabla\chi$, $\nabla\chi(\mathbf{x}) = \mathbf{n}$

- ▶ Points define vector field \mathcal{V} that corresponds to $\nabla\chi$
- ▶ Sought χ minimizes $\|\nabla\chi - \mathcal{V}\|$



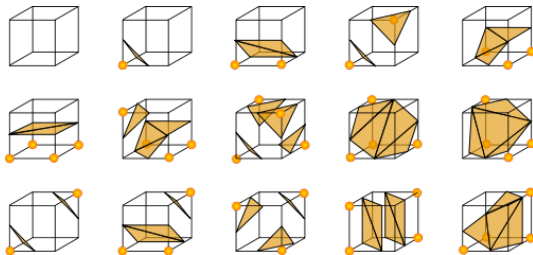
Images from Gotsman & Kazhdan's slides

3D Reconstruction

Surface Reconstruction – Poisson Surface Reconstruction

Once χ is known, the isosurface $\chi(\cdot) = 0.5$ can be extracted

- Using marching cubes, for example



Images from wikipedia.org

3D Reconstruction

Software – Point Cloud Library (PCL)

C++ open-source library for point cloud processing

Includes implementations of the above methods



Image from pointclouds.org

3D Reconstruction

Software – Point Cloud Library (PCL)

Example : align two point clouds using ICP

```
typedef pcl::PointXYZ Pt; // a point
typedef pcl::PointCloud<Pt>::Ptr Pc; // a point cloud

// assume we have two clouds r, s (above notation)
pcl::IterativeClosestPoint<Pt, Pt> icp;
icp.setInputSource(s);
icp.setInputTarget(r);
icp.align(r, theta); // theta = initial estimate
theta = icp.getFinalTransformation(); // final estimate
```

3D Reconstruction

Application Fields – Cultural Heritage

Preservation of physical artifacts



Image from Levoy et al. 2000

3D Reconstruction

Application Fields – Virtual and Augmented Reality

Project Tango

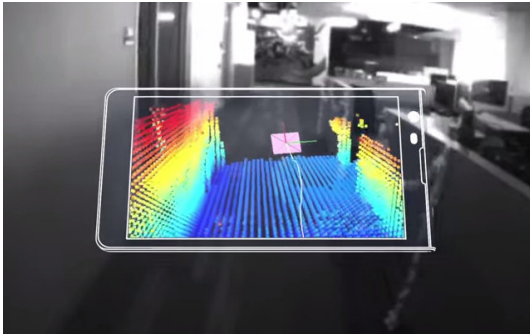
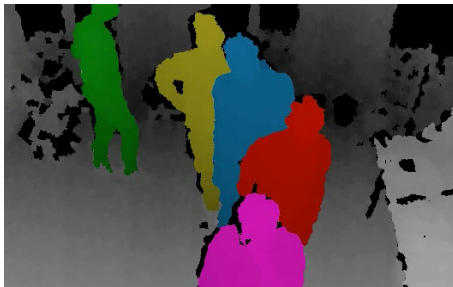


Image from <https://www.youtube.com/watch?v=Qe10ExwCqk>

Person Detection

3D data enables reliable person detection

- ▶ Robust motion detection
- ▶ Distinctive, invariant features

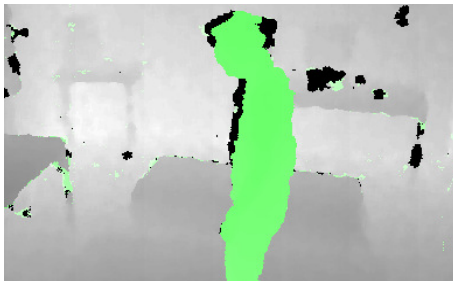


Person Detection

Motion Detection

Reliable motion detection via background subtraction

- ▶ Measurements represent object distances
- ▶ Not affected by illumination, clothing, shadows



Person Detection

Features

Scene geometry allows for distinctive, invariant features

- ▶ Object size, extent, volume, shape, ...

More on object detection later

Person Detection

Applications – Breaking Assistance

<https://www.youtube.com/watch?v=oU4XQvx010k>

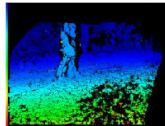


Image from Ryuzo Okada, Toyota

Person Detection

Applications – Interactive Art Installations



Image from ortlos.com

Person Detection

Applications – Fall Detection (fearless)

Fall detection system developed at CVL

- ▶ Uses data from a single Kinect sensor
- ▶ Detects falls by tracking the height of persons



Person Detection

Applications – Entertainment (Kinect Player Pose Estimation)

<https://www.youtube.com/watch?v=p2qlHoxPioM>



Kinect Player Pose Estimation

Let's take a look at how this works

Assuming we have already detected the person



Image from Shotton et al. 2011

Kinect Player Pose Estimation

Steps

Estimate body part of each pixel independently

Perform clustering to obtain joint position proposals

Fit skeleton model to joint proposals



Image from Shotton et al. 2011

Kinect Player Pose Estimation

Pixel Classification

For each pixel \mathbf{x} with depth $d(\mathbf{x})$ compute $\Pr(w|\mathbf{x})$

- ▶ With w representing the body part, $w \in \{0, \dots, 30\}$
- ▶ Note that this is a discriminative model



Image from Shotton et al. 2011

Kinect Player Pose Estimation

Pixel Classification

Classification using simple depth offset features f_{θ} ,

$$f_{\theta=(\mathbf{u},\mathbf{v})}(\mathbf{x}) = d\left(\mathbf{x} + \frac{\mathbf{u}}{d(\mathbf{x})}\right) - d\left(\mathbf{x} + \frac{\mathbf{v}}{d(\mathbf{x})}\right)$$



Image adapted from Shotton et al. 2011

Kinect Player Pose Estimation

Pixel Classification

One such feature is weak

But a strong classifier can be built by combining them

Implemented using a **random forest**

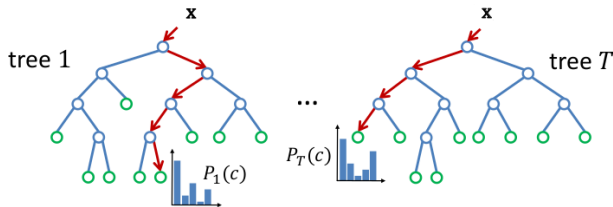


Image adapted from Shotton et al. 2011

Kinect Player Pose Estimation

Pixel Classification – Random Forests

Random forest consists of T randomized decision trees

Each tree t consist of split and leaf nodes

Each split node consists of a feature f_{θ} and a threshold τ

- ▶ \mathbf{x} branches down based on $f_{\theta_k} > \tau_k$
- ▶ Until a leaf node is reached, which stores $\text{Pr}_t(w|\mathbf{x})$

Tree trained from training samples (\mathbf{x}, w)

- ▶ Samples differ between trees
- ▶ θ_k, τ_k selected from random subset

Kinect Player Pose Estimation

Pixel Classification – Random Forests

All trees contribute to the result, $P(w|\mathbf{x}) = 1/T \sum_{t=1}^T \text{Pr}_t(w|\mathbf{x})$

- This training & inference strategy is called **bagging**

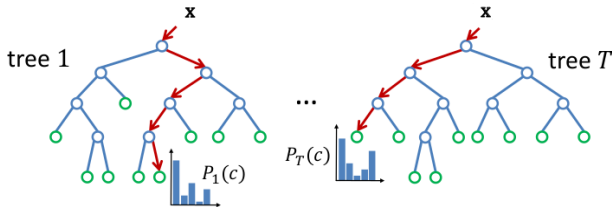


Image adapted from Shotton et al. 2011

Kinect Player Pose Estimation

Joint Proposals

Project classified pixels to 3D

Perform clustering for each label w using mean-shift

- ▶ <https://www.youtube.com/watch?v=kmaQAsotT9s>

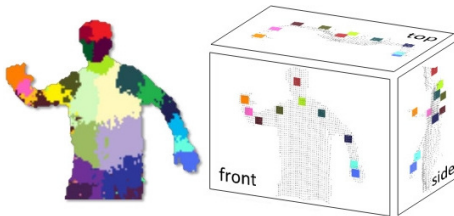


Image adapted from Shotton et al. 2011

Kinect Player Pose Estimation

Skeleton Fitting

Results in n_w joint position proposals per body part w

Goal is to find best joint configuration

Can be accomplished using a constellation model, for example

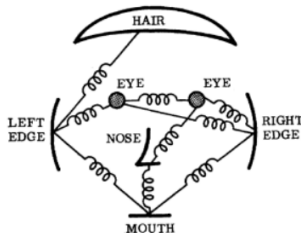


Image from Fischler and Elschlager 1973

Kinect Player Pose Estimation

Remarks

Skeleton data is available from Kinect SDK

- ▶ <http://www.microsoft.com/en-us/kinectforwindows/>

Random forests make for powerful classification/regression models

Implementations readily available

- ▶ Matlab (TreeBagger)
- ▶ OpenCV (CvRTrees)
- ▶ scikit-learn (RandomForestClassifier)

Summary

Knowledge of scene geometry enables powerful CV applications

We have covered a selection

- ▶ 3D reconstruction for virtual reality
- ▶ Person detection for breaking assistance
- ▶ Human pose estimation for gaming

Interested in 3D vision?

- ▶ There is an own VO (183.129) and UE (183.130)

Aiger, Dror, Niloy J Mitra, and Daniel Cohen-Or (2008). **4-points congruent sets for robust pairwise surface registration.**

Berger, Matthew et al. (2014). **State of the Art in Surface Reconstruction from Point Clouds.**

Curless, Brian and Marc Levoy (1996). **A volumetric method for building complex models from range images.**

Fischler, Martin A and Robert A Elschlager (1973). **The representation and matching of pictorial structures.**

Kazhdan, Michael (2005). **Reconstruction of solid models from oriented point sets.**

Kazhdan, Michael, Matthew Bolitho, and Hugues Hoppe (2006).
Poisson surface reconstruction.

Levoy, Marc et al. (2000). **The digital Michelangelo project: 3D scanning of large statues.**

Newcombe, Richard A et al. (2011). **KinectFusion: Real-time dense surface mapping and tracking.**

Shotton, Jamie et al. (2011). **Real-Time Human Pose Recognition in Parts from a Single Depth Image.**