Using Machine-learning Techniques to Increase the Efficiency of Kinect Fusion Reconstructions

Literature Review

Alan Lau

MCommp Computer Science
University of Bath

October 2015

Contents

1 Introduction

This project aims to quicken the time required to create a 3-diemensional reconstruction of objects in a room with just a simple 2-dimentional scan from one angle, using a Microsoft Kinect Camera. To enable such goal, classification, a form of machine-learning, is proposed to be used. There are many different statistical classification algorithms that can be used for labelling objects through training with similar objects and recognising the category and object belongs when put in use.

Being an RGBD camera, not only does the Kinect Camera captures a colour (RGB) image, it also captures depth information via an infrared laser combined with a monochrome camera [2]. This information enables a detailed 3-dimentional reconstruction.

A group of New York University researchers have created a depth dataset called the *NYU Depth Dataset* [1], which is freely available online.[1] A variety of objects are scanned, labelled or segmented from a scene. The dataset is freely available for various applications in different formats. The number of scans for different classes of objects makes it ideal to be used as training data for the proposed classifier.

This Review attempts to explore in greater detail about the NYU Dataset and its applications, how depth maps are useful to 3-dimensional reconstruction and how the depth dataset is useful for training. We also attempt to justify the algorithms that will potentially fit to label objects in a reconstructed scene.

[1] The datasets are available for download at `http://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html`

2 Kinect Fusion

The Kinect Camera was first released for the company's gaming console, XBox 360. It was designed to recognise gestures, faces and voices, providing a more physical way and new dimension to interact with the interface and games than a conventional controller. A Windows-compatible version of the Camera, an SDK and Kinect Fusion were released later, enabling researches and the development of commercial products. [KIN1]

Augmented reality (AR) and real-time reconstructions are some of the most popular research applications of Kinect Fusion. SemanticPaint [SP] demonstrates the posibility of Kinect Fusion in AR in real-time scenes. It allows parts of the scene to be 'painted' by the user. It also uses segmentation and object recognition to paint similar objects in the same colour. Although this project does not invovle real-time processes, it provides

2.1 Kinect Camera Technologies

The two generations of Kinect Camera use different 3-dimensional camera technologies to obtain depth information about a scene. Each of these technologies has its pros and cons, which is discussed below. However, the common problem of these technologies is that it does not deal with very bright light. It will cause detail to be lost. [books?]

2.1.1 Structured Light

Structured light is used in the first generation Kinect Camera (2010). A sequence of known infrared patterns are projected onto the scene. A deformed pattern is formed when objects are 'in the way' of the patterns. The object is then observed from another angle by the monocrhome camera. Through analysing the deformed patterns and obervations, the depth information about the scene can be obtained. [3] [4]

2.1.2 Time-of-Flight

Rather than looking at the deformed pattern, the second generation Kinect Camera (2013) estimates depth information based on the time the infrared beam takes to travel back and forth. The difference between the reference signal and returned signal allows the calculation of a tiem difference, which helps compute the required depth information. [chi-book]

2.1.3 Comparing the Two Systems

2.2 Data Representation

2.2.1 Point Cloud

A point cloud stores data points in a cooridinate system. [chi-book] [wiki] In a real-world case (as well as the captured images), a 3-dimensional coordinate system is used. A point cloud can be used to generate a mesh so that it can be used to render a visual image of the reconstruction volume.

2.3 Reconstruction

2.3.1 Pipeline

A single raw capture with the Camera does not provide a detailed reconstruction. Combining the depth information of many images together enables a super-resolution [3d-paper] reconstruction, making a detailed and high quality reconstruction possible. The following is the full pipeline of how a reconstruction is created from raw depth data:

1. Raw Input Conversion
    - The raw *depth map* is captured by the infrared-monochrome subsystem of the Camera. This information is not very detailed, as shown in [fig x].
    - It needs to be combined with the *normal map*, which is the surface normals associated with each vertex [spe-book], to provide a more detailed but noisy reconstruction at this stage. [image 0:35 of MSR video].
    - Further conversion and reconstruction is needed to retain detail, remove noise, and fill in holes missed by the Camera, possibly due to bright light.
    - This information is stored as a point cloud and will be combined into one representation later on.
2. Camera Pose Tracking
    - The location and orientation (world pose) of each frame are tracked as the Camera moves around.
    - This alignment is constantly traced, allowing all the point clouds to be aligned together. [website]
    - The frames captured from different poses, even the smallest movement (e.g. caused by a hand-shake), will allow further quality improvements to the scene, achieving more than what a single raw capture is capable of [ms-3d-paper].
3. Fusing
    - The depth data converted from the raw input is combined into a single 3-dimensional space per frame.
    - A running average of depth is kept. This reduces noise, and creates a refined reconstruction by combinding all the information in one place. [website] [3d-paper]
4. Resultant Reconstruction Volume
    - The resultant point cloud will be of a highly detailed reconstruction of the scene. For example, grills of a millimetre, as shown in [fig x] can be reconstructed properly.
    - A rendered image of the 3D reconstruction volume is possible by using methods such as raycasting, providing a visual feedback of the scene, and allows for many possibilities, such as augmented reality applications.

2.3.2 Volumetric (3D) Reconstruction

By averaging the surface models and depth data from multiple viewpoints into one volumetric voxel volume, the scene appears to live in a 3-dimensional 'box'. With more data,

2.3.3 ? (2D) Reconstruction

2.4 Segmentation

In order to identify an object in a scene with many objects, there needs to be a way to single them out individually so that something such as labelling can be done on them. In computer vision, this is called *segementation.* There are many segmentation algorithms available. Some popular ones include *k-means*, *Gaussian*, *mean-shift*, *normalised cuts*, *similarity graph-based segmentation* and *binary Markov random fields using graph cuts.*

K-means

2.4.1 Segmentation in NYU Dataset V2

As Kinect was born to support gesture recognition, there is limited support to segmentation of a scene.

3 NYU Dataset V2

In order for the classifier to produce accurate results, it needs to be trained with an extensive and quality data before it can be used to classifiying objects into classes. The second version of the NYU Depth Dataset provides a wide range of objects and different scenes

4 Classification

The idea of classification is to identify of which group an object belongs to. This relies heavily on a good algorithm that is able to group similar objects together in the first place. In more formal words, a classifier groups objects that has some semantic similarity enough to be classed as the same type. Each class has a label in which these objects are identified as. For example, round objects that can hold liquid can be classed as "bowls", despite being different in colour or and of slightly different shapes.

There are two types of classifiers - *supervised* and *unsupervised.*

Supervised

There is a trade-off between speed and precision and recall. We also talk about precision and recall.

4.1 Classification Projects

There are many

4.1.1 `scikit-learn`

`scikit-learn` is a machine-learning library for Python, which is actively in development. It has quality implementations of popular machine-learning algorithms for multiple machine-learning problems. It is built on `numpy` and `scipy`, making it easy for data manipulation [5]

4.2 Training

4.2.1 NYU Depth Dataset V2 [1]

In order for the classifier to produce accurate results, it needs to be trained with an extensive and quality data before it can be used to classifiying objects into classes. The second version of the NYU Depth Dataset provides a wide range of objects and different scenes

4.3 Classification Algorithms

There are many popular and useful classification algorithms.

4.3.1 Comparing Algorithms

In this project, it is reckoned that the performance at classifying a scene is more important than the time required at training. The trade-off between time and precision and recall is little, so it is worth trying both SVM and Random Forest to try to obtain the highest accuracy and low false positive/negative rates.

5 Summary

Bibliography

[1] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor Segmentation and Support Inference from RGBD Images. In *ECCV*, 2012.

[2] Microsoft Coporation. Kinect Fusion MSDN Documentation.

[3] Ling Shao, Jungong Han, Pushmeet Kohli, and Zhengyou Zhang. *Computer Vision and Machine Learning with RGB-D Sensors*. Springer, 2014.

[4] Hamed Sarbolandi, Damien Lefloch, and Andreas Kolb. Kinect range sensing: Structured-light versus time-of-flight kinect. *CoRR*, abs/1505.05459, 2015.

[5] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.*, 12:2825–2830, November 2011.