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

Alan Lau Wing Leung

MCommp Computer Science
University of Bath
October 2015

Contents

Chapter 1

Problem Description

Kinect Fusion is a Micrsoft Research project that enables rapid and detailed 3D reconstructions, both real-time and off-line, through scanning an object and merging the depth data of many frames together, with the help of a Kinect camera [1][2].

A few research projects from Microsoft demonstrate some interesting ideas about the possibilities of Kinect Fusion. For example, SemanticPaint [3] provides a real-time augmented experience where users can label and segment through the combination of commands, Kinect Fusion and object recognition, allowing different objects to be painted in the reconstruction in real-time. Other researches include non-rigid scene reconstruction [4], dense surface mapping [5] and a Master's thesis on building-scale interior reconstruction. [6] These and other similar projects will be become useful information when attempting this project.

One issue with Kinect Fusion is that it does not store and recognise objects it has seen before. This means that a seen object will require the same reconstruction again, making it inefficient when trying to obtain reconstructions in scenes with the same or similar objects.

This project aims to solve this with the help of a classifier. For instance, many rooms at the University share the same equipment across multiple rooms. Rather than having to re-scan and reconstruct an object encountered before, a newly reconstructed item should be stored and be identified by the classifier later. When encountering a new scene with the same object in it, only a quick scan from one angle is needed to fill in the full, detailed scans of objects seen before. This shortens the time and minimise the effort required to reconstruct a new scene with identical objects, but provides a fairly accurate and detailed reconstruction at the same time.

As a classifier is used, it should should be able to locate the best match anyway. The best matched item could be put in place with a one-angled quick scan for a new scene with new but similar objects in it (e.g. a similar round table). If it is deemed good enough, the room reconstruction can be used, without the need of further scanning. If further scanning of an object is required, it can then replace an object of the reconstructed scene at a later time.

1.1 Aim

The aim is to quickly reconstruct a scene without needing to re-scan each object in detail again to obtain a well-reconstructed scene, by using classifiers or other appropriate machine-learning techniques.

1.2 Objectives

1.2.1 Main Objectives

1. Develop one or a set of classifiers that can accurately identify a seen object or able to find the next best match if it does not exist.
2. Vigorously train and test the classifier with many scans of similar objects to improve its accuracy
3. Design and implement algorithms to locate potential object positions and obtain the partial information
4. Integrate the classifier with the algorithms to create a reconstructed scene using the identified reconstructions

1.2.2 Stretched Objectives

These objectives are nice to have if they can be done on top of the main objectives, but it is expected these will not be achievable with the limited time and scope of this project.

1. Adapt and develop a real-time version where objects are being put in place right at the scene
2. Provide an augmented reality experience similar to SemanticPaint [3] - one that allows real-time recognition and colouring.

Chapter 2

Requirements Specification

1. The classifier needs to be accurate in recognising and discerning between different objects.
    - Precision and Recall needs to be high enough to be useful.
    - NYU Depth Dataset [7] can be used to in early prototype classifiers.
    - The classifier, especially the prototypes, must use a simple language (e.g. *R*, *Python*) for quick iteration.
2. Training of the classifier requires a large dataset to make it accurate.
    - Utilise the NYU Depth Dataset [7] as the basis of training the classifier, alongside with scans made of objects at the University.
3. The algorithms for locating and filling in objects need to be fairly accurate.
    - Otherwise, the objects not be fitted at the right place and makes the reconstruction inaccurate.

Chapter 3

Project Plan

3.1 Project Management

There are three main elements in this project:

1. a classifier that finds out the object in disguise;
2. an algorithms that figures out where the missing bits are; and,
3. combinding the two to obtain a reconstructed scene.

In order to complete these complicated implementations, much organisation and reading up are needed. I plan to iterate 4 times (2 weeks or so at a time) on the coding side of things for both the classifier and algorithms, starting on 1 November. This is to ensure that timely and achievable goals are set and that problems can be resolved as soon as it is discovered.

I intend not to work on the project during the three-week examination period, as realistically, most effort will be put in revision and there will be little time for other work.

A five-week period during February and March will be used to finalise the codebase. Five weeks are set apart for writing up the dissertation. This should provide enough time to handle any delay, development complications and for the disseration write-up.

See [Appendix A] for the gant chart.

Chapter 4

Resources

4.1 Technologies

1. A simple language and fast library is required for easy iteration.

    - *R* or *Python* (with the libraries such as *pyML*, *numpy* or *scikit-learn*) can be used to produce the classifier. They are languages that are easy to code with and are used for handling data.
    - Initially, the large library of Kinect data from NYU (The NYU Depth Dataset) [7] should be used while constructing the classifier. This provides a good selection of data and understanding how to interpret them.

2. Kinect Fusion and the Kinect for Windows SDK

    - These SDKs are needed to for capturing depth data. They need to be installed on a PC running Windows.
    - Other related packages such as Visual Studio may be required to allow capturing and other manipulation.

4.2 Hardware

1. A relatively powerful Windows-based computer with a (preferably, discrete) graphics card that supports DirectX 11.

    A DirectX 11-supported graphics card required for Kinect Fusion to function properly.

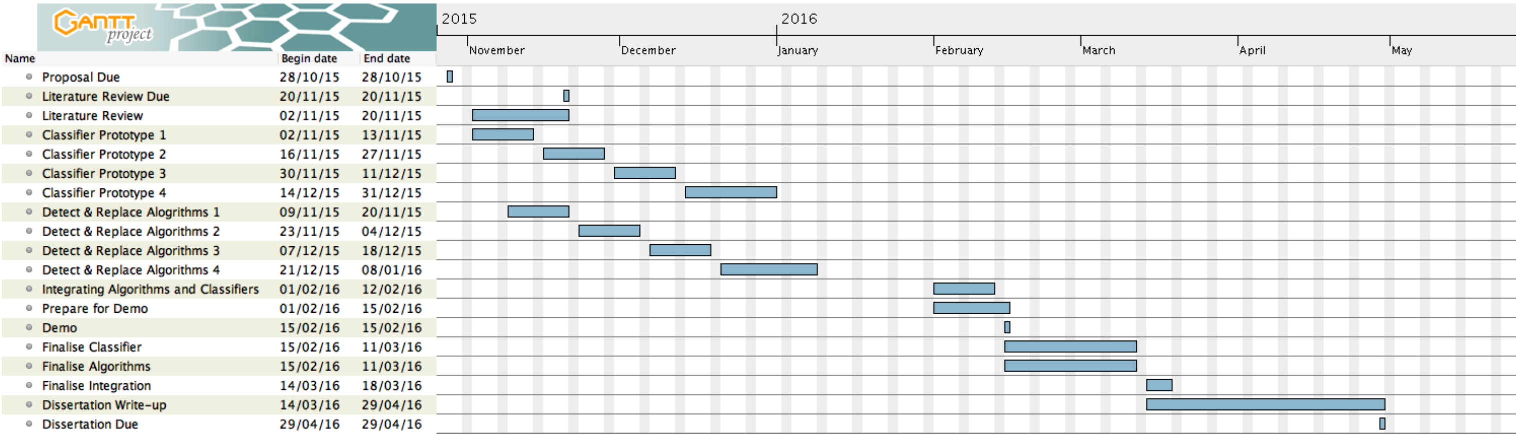
2. Kinect for Windows

    - For capturing the depth data.

Chapter 5

Appendix

5.1 Appendix A



Name	Begin date	End date
Proposal Due	28/10/15	28/10/15
Literature Review Due	20/11/15	20/11/15
Literature Review	02/11/15	20/11/15
Classifier Prototype 1	02/11/15	13/11/15
Classifier Prototype 2	16/11/15	27/11/15
Classifier Prototype 3	30/11/15	11/12/15
Classifier Prototype 4	14/12/15	31/12/15
Detect & Replace Alogrithms 1	09/11/15	20/11/15
Detect & Replace Algorithms 2	23/11/15	04/12/15
Detect & Replace Algorithms 3	07/12/15	18/12/15
Detect & Replace Algorithms 4	21/12/15	08/01/16
Integrating Algorithms and Classifiers	01/02/16	12/02/16
Prepare for Demo	01/02/16	15/02/16
Demo	15/02/16	15/02/16
Finalise Classifier	15/02/16	11/03/16
Finalise Algorithms	15/02/16	11/03/16
Finalise Integration	14/03/16	18/03/16
Dissertation Write-up	14/03/16	29/04/16
Dissertation Due	29/04/16	29/04/16

Bibliography

[1] Microsoft Corporation. *Kinect Fusion* [Online]. Place of publication: Microsoft Developer Network. Available from: `https://msdn.microsoft.com/en-us/library/dn188670.aspx` [Accessed 22 October 2015].

[2] Kinect for Windows Team, 2013. *Kinect Fusion demonstrated at Microsoft Research TechFest, coming soon to SDK* [Online]. Place of publication: Kinect for Windows Blog Available from: `http://blogs.msdn.com/b/kinectforwindows/archive/2013/03/06/kinect-fusion-demonstrated-at-microsoft-research-techfest-coming-soon-to-sdk.aspx` [Accessed 22 October 2015].

[3] Cameron, S., 2015. *Microsoft Research debuts another project, Semantic Paint* [Online]. Place of publication: WinBeta Blog Available from: `http://www.winbeta.org/news/microsoft-research-debuts-another-project-semantic-paint` [Accessed 23 October 2015].

[4] Newcombe, R. A. et al., 2015. DynamicFusion: Reconstruction and Tracking of Non-rigid Scenes in Real-Time. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7-12 June 2015 Boston, Massachusetts. Seatle: University of Washington, pp.343-352.

[5] Newcombe, R A et al., 2011. KinectFusion: Real-time dense surface mapping and tracking. *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium*, 26-29 October 2011 Basel, Switzerland. Seatle: University of Washington, pp.127-136

[6] Hambchen, N., 2014. *HouseScan: Building-scale interior 3D reconstruction with KinectFusion.* Thesis (MSc). Imperial College, London.

[7] Silberman, N. et al.. *NYU Depth Dataset V2* [Online]. Available from: `http://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html` [Accessed 23 October 2015].