Project Proposal 1

KinectFusion: Rigid 3D Scene Reconstruction for Creating Interactive Scenes in Video Games

1 Abstract

Video games have become a core part of modern entertainment with further applications in virtual and augmented reality for professional training and simulation. A key challenge in game development is the creation of immersive scenes. Typically, game scenes are manually designed, which is time consuming. One solution is to use 3D reconstruction techniques to generate basic scenes very quickly. The use of 3D reconstruction has become increasingly accessible with the release of low cost RGB-D sensors like the Microsoft Kinect in 2010. For our project, we propose the implementation of a game engine integration pipeline, which makes use of a commodity Kinect sensor and the KinectFusion model outlined by [1] and [2] for static 3D scene reconstruction and the game engine Unity3D as shown in figure 1. However, we will omit the ICP-outlier module of the original KinectFusion model. The selected ICP-algorithm will be based on the findings in [3]. To demonstrate our pipeline, we will create a virtual driving game with a simple collision model, which allows users to drive a model car around the 3D reconstructed scene. Users are rewarded for avoiding collisions and distance driven.

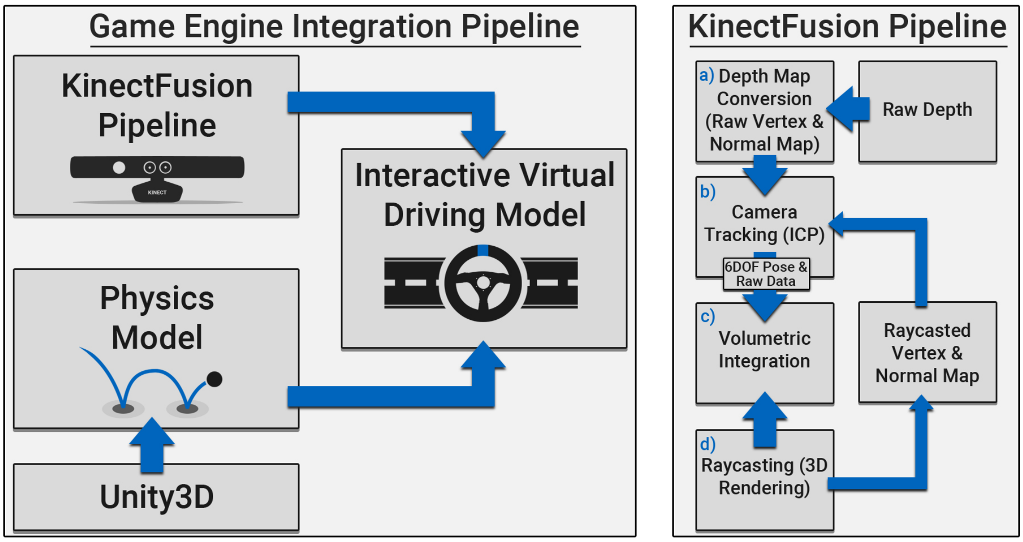

Figure 1: Overview of implementation pipeline. (Left) Game engine integration pipeline and (right) KinectFusion pipeline (adapted from [1] without the ICP-outlier module.

As optional sub-tasks, in the event of additional time, the KinectFusion pipeline will be improved in two ways: using a Kinect sensor noise model as implemented in [4] and increasing scanning and reconstruction range through extended KinectFusion as shown in [5].

2 Hardware Requirements

The main part of the project, the implementation of rigid and static 3D scene reconstruction using KinectFusion, will require the following pieces of hardware:

1. Microsoft Kinect sensor for Windows for collecting RGB-D data of the target scene.
2. Computer with commodity GPU for implementing real-time 3D scene reconstruction.

3 Team

The project team consists of four group members from various academic backgrounds:

1. Gorbachev, Gleb - Master Robotics, Cognition, Intelligence - Mtr-Nr. 03709785
2. Hernandez, Ivan - Master Robotics, Cognition, Intelligence - Mtr. Nr. 03712543
3. Rohregger, Alex - Master Robotics, Cognition, Intelligence - Mtr.-Nr. 03716431
4. Yagubbayli, Farid - Master Informatics - Mtr.-Nr. 03708842

4 Project Timeline

The implementation of the game engine integration pipeline will take around 6 weeks. As seen in the project timeline of figure 2, the first three weeks focus on the KinectFusion pipeline implementation and the last three weeks on the game engine integration.

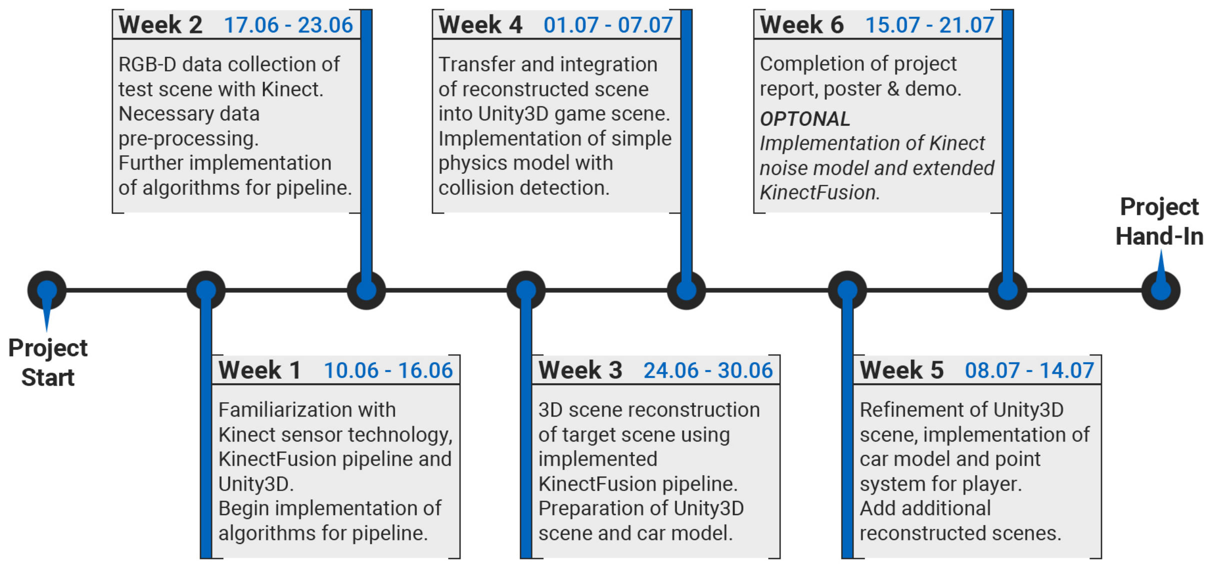

Figure 2: Project timeline broken down into weekly milestones.

References

[1] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, *et al.*, “Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera,” in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pp. 559–568, ACM, 2011.

[2] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon, “Kinectfusion: Real-time dense surface mapping and tracking.,” in *ISMAR*, vol. 11, pp. 127–136, 2011.

[3] S. Rusinkiewicz and M. Levoy, “Efficient variants of the icp algorithm.,” in *3dim*, vol. 1, pp. 145–152, 2001.

[4] C. V. Nguyen, S. Izadi, and D. Lovell, “Modeling kinect sensor noise for improved 3d reconstruction and tracking,” in *2012 second international conference on 3D imaging, modeling, processing, visualization & transmission*, pp. 524–530, IEEE, 2012.

[5] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, “Kintinuous: Spatially extended kinectfusion,” 2012.