

3D Reconstruction using Time of Flight Sensors

EE/CprE 491 Final Report

Dec15-09

Week 12 (4/27/15-5/3/15)

Adviser: Tom Daniels
Client: VirtuSense Technologies
Members: Team Leader – Monica Kozbial
Team Webmaster – Sarah Files
Team Communication Leader – Kyle Williams
Team Key Concept Holder – Yee Zhian Liew

Overview of Project Status

Phase 1 Deliverables:

- Conversion of 3D model to 3D mesh of face
 - We have created 3D models using KinectFusion Explorer.
 - We have done research into how KinectFusion works and what can affect the quality of the scan.
 - We have done work with finding good parameters for the reconstruction.
 - We have looked into additional ways to create a model with the Kinect sensor and found HDFace that could be promising
 - We have looked at filtering/smoothing techniques to improve the model scan.
- Merging of multiple meshes
 - This is handled by the Kinect SDK and no additional work is needed on our part.
- Texture overlay on 3D model
 - Color data can be mapped to a model through the SDK.
- UI for capturing sensor data and generating 3D model
 - We have created screen sketches for the 3D model but have not been able to create a working application using the new architecture that was presented to us.
 - Screen Sketches can be found in the Design Document for the project.

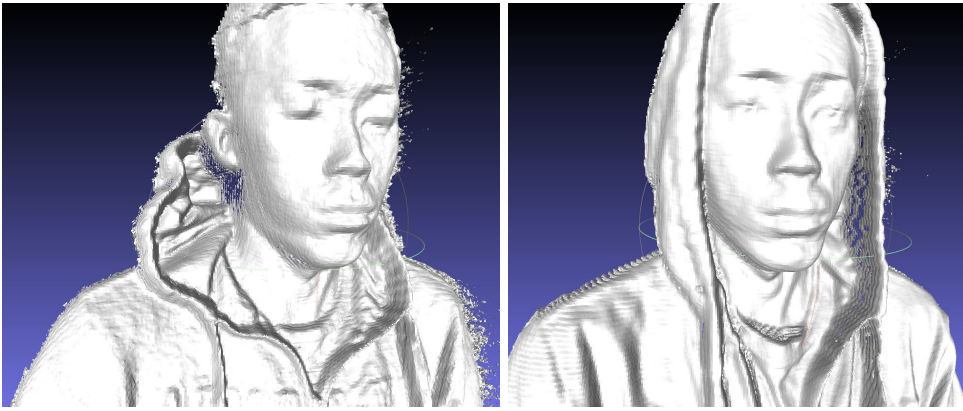
Learned So Far

Kinect Fusion

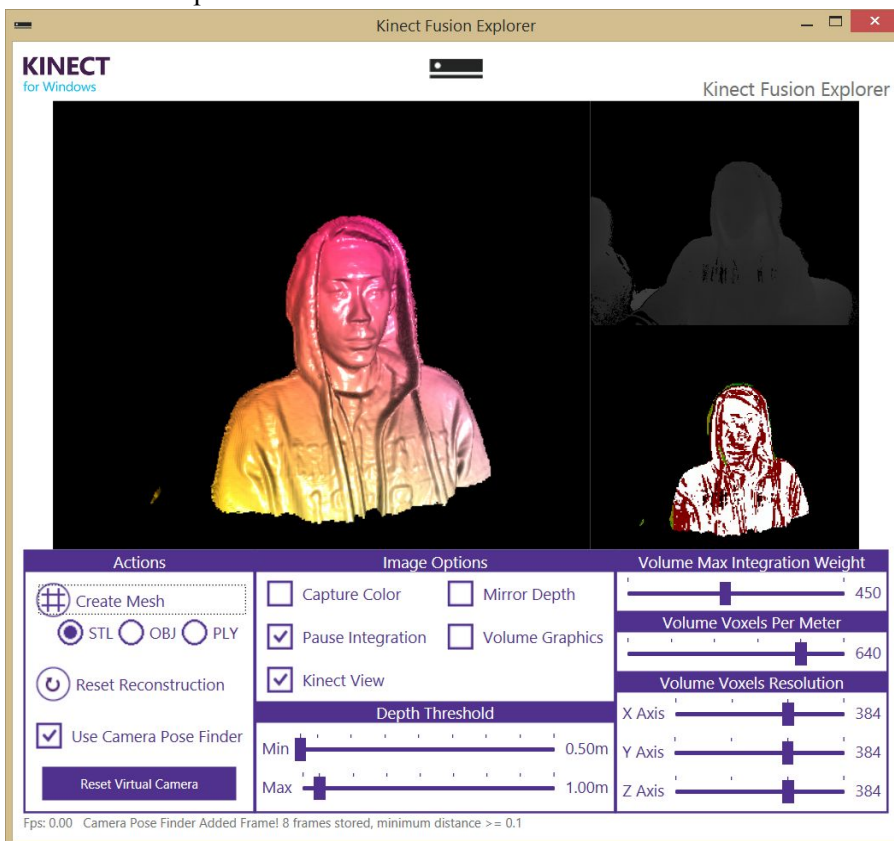
KinectFusion provides a means to collect the 3D depth data and convert it into a 3D model. In the GPU-based processing pipeline of KinectFusion, it also computes any changes in the sensor pose between frames using the Iterative Closest Point algorithm for aligning new data points to the reconstruction. Color data can also be mapped to models created through the SDK, although this is done using vertex color which is not supported for *.obj or *.stl, but is for *.ply.

Fusion Parameters

KinectFusion has several parameters that can be adjusted to work best for the type of scan being made. Below, you can see a side by side comparison of an early scans (left) and later scans (right) while we were working on finding good parameters for the reconstruction.



Adjustments to the parameters were done using the Kinect Fusion Explorer program included with the Kinect Developer Toolkit. Below is a screenshot of the scan settings that have been selected through a series of iterative testing to be used for the reconstruction. These settings have worked well for creating a reasonable quality scan at a reasonable speed.



KinectFusion creates a 3D construction with a voxel (3D pixels) representation the reconstruction (which can be converted to various file formats). An in-depth explanation of the parameters can be found at <https://msdn.microsoft.com/en-us/library/dn188670.aspx> and below is an overview of some important information.

KinectFusion parameters:

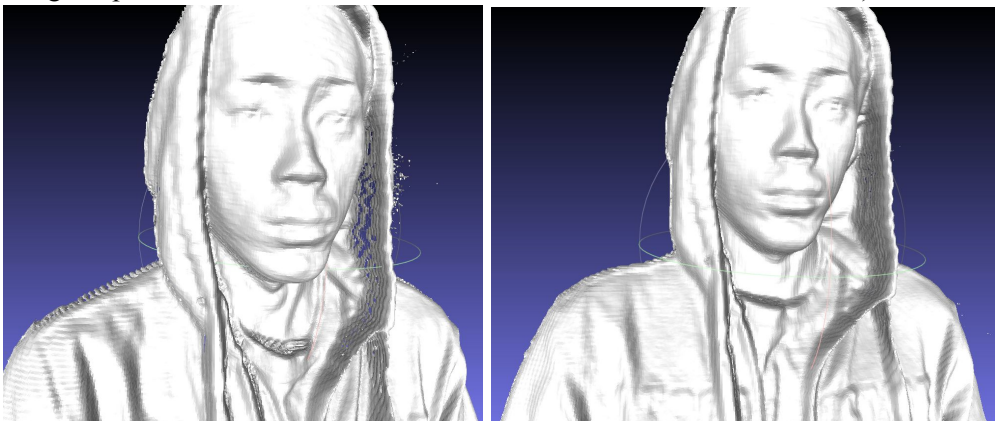
- Depth Threshold
 - This determines the region of the reconstruction volume and needs to be adjusted depending on how far away the model is in respects to the sensor.
- Maximum Integration Weight
 - controls the temporal averaging of data into the reconstruction volume. Increasing this value make a higher detailed reconstruction, but takes longer to average and does not adapt to change.

Decreasing this value makes the volume respond faster to change in depth, but is noisier overall.

- Volume Voxels Resolution
 - This parameter set determines the number of voxels allotted for the x,y, and z reconstruction of the model. Increasing these parameters will increase the number of voxels in the reconstruction, however there is a fixed number of voxels that can be created, determined by the memory available (determined by maximum contiguous memory block available for storing the reconstruction) in addition to increasing the time of the scan because more voxels are created/tracked.
- Volume Voxels Per Meter
 - The reconstruction volume is made up of small cubes in space, referred to as voxels and volume voxels per meter scales the size that 1 voxel represents in the real world. Increasing this value will clearly increase the resolution of the model. However, increasing this value will also require the resolution to be increased as well in order to have enough voxels for the representation which is limited by the memory available as explained earlier.

Graphics Card

In our data collection we have found a visible difference in the quality of the 3D reconstruction when comparing constructions made with computers of differing graphics card quality. Below you can see two reconstructions made with 2 different computers. The left model was made with a laptop with HD Graphics Family graphics card and the model scan was made with a laptop with Nvidia GeForce GT 525M. The right model is much smoother which is more apparent when looking at the shoulder of the model (Each reconstruction was made using the parameters described in the earlier section and a 30 second scan).



The graphics card quality affects the scan because KinectFusion has GPU-based pipeline and most of the processing and calculations is done on the GPU. The faster the graphics card, the more accurate the alignment of the points of the new scan since new data points are added to the reconstruction every frame.

There seems to be a way to run KinectFusion in “CPU mode” where the processing is done on the CPU rather than the GPU, but this is a very atypical implementation and so our team has not found many resources on implementing this (Here is a link referring to this

<https://social.msdn.microsoft.com/Forums/en-US/dbf88947-8084-4834-970d-262af27f8438/using-kinect-fusion-in-nonrealtime-cpu-mode?forum=kinectsdk>). It seems that CPU mode is slower in the reconstruction, but it might be a potential alternative if a high quality graphics card is not guaranteed or another mode that our program can run in.

System Requirements

The hardware and software requirements for using the Kinect v2 is best explained in the Microsoft documentation here: <https://msdn.microsoft.com/en-us/library/dn782036.aspx>

It is important to note that using a desktop computer as opposed to a laptop is highly recommended and the quality of the graphics card (as explained earlier) affects the quality of the scan.

Given the restrictions on the hardware and software of the machine that can use the Kinect, it seems to make the

most sense to target an application specifically for a desktop PC, because, although it may be possible to try to target tablets or laptops, using a machine with less processing power will result in a decrease in quality that can not be overcome through software.

Smoothing

We have investigated methods for smoothing and filtering in order to increase the quality of the model. We concluded that smoothing the model would cause undesirable decrease in the definition of features in the reconstruction after a few smoothing attempts and so we focused our time more on filtering and parameter refinement for the scan. In the Additional Resource Locations section, under alternative alignment algorithms and filtering, there are a few papers that we have found on creating a smoother model through alternative alignment algorithms. These algorithms are meant to replace the Iterative Closest Point algorithm that KinectFusion and require a very thorough understanding of the KinectFusion pipeline and we did not have time to complete an implementation.

Kinect HDFace

HDFace is a feature set of the Kinect SDK that may be a viable option to use instead of Fusion. This is a new feature with the Kinect version 2 and most of the features are related to face tracking and rigging and consequently most articles and videos are exploring those features. However, HDFace also allows for the reconstruction of a face using algorithms that read scans of different sides of the face and morphing an “average face” model to create a face reconstruction.

Some of the benefits of this feature set include that a developer can easily track whether enough data has been collected for each side of the face and would be very useful when developing the interface for collecting scan data. A downside to using this is that it would not be able to be used for the full body scan because it is only for facial reconstruction.

This route has not been explored for this project because it was discovered late into the project. There is a sample program that comes with the Kinect Developer Toolkit that uses HDFace, though it is not very intuitive and has not been confirmed that it employs the facial reconstruction feature of HDFace.

Blender3D plugin Licensing

Blender has a GNU General Public License and a plug-in made for Blender normally must follow the GNU GPL license.

According to the FAQ on the official Blender website: “Only if the plug-in doesn’t work within Blender as ‘acting as a single program’ (like using fork or pipe; by only transferring data and not using each others program code) you have the full freedom to license the plug-in as you wish.”

(<https://www.blender.org/support/faq/>)

Additional Resource Locations

(videos)

https://www.youtube.com/watch?v=_cKb3oEM47E

<https://www.youtube.com/watch?v=GKRHWBS6guM>

<https://www.youtube.com/watch?v=AnWfF0ul-RY>

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

(alternative alignment algorithms and filtering)

<http://www.cs.unc.edu/~doums/pdfs/nonrigidmatching.pdf>

http://www.researchgate.net/profile/Chuong_Nguyen6/publication/234028266_Modeling_Kinect_Sensor_Noise_for_Improved_3D_Reconstruction_and_Tracking/links/0912f50e633228e84e000000.pdf

<http://research.microsoft.com/pubs/155378/ismar2011.pdf>