3D Reconstruction Using Time of Flight Sensors

Design Document

By:
Kyle Williams
Monica Kozbial
Yee Zhian Liew
Sarah Files

Advisor: Professor Daniels
Client: VirtuSense Technologies

Design Document for 3D Reconstruction project

Dec15-09

Sarah Files

Kyle Williams

Monica Kozbial

Yee Zhian Liew

Version	Date	Author	Change
1.0	3/8/2015	SF, KW, MK, YL	Initial Design Document
2.0	4/23/2015	SF, KW, MK, YL	Updates

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this design document is to provide documentation for our design and source code implementation. Implementation details are shown in this document to make sure our project is complete. With this design document, we can provide an appropriate solution for the problem beforehand. In other word, the focus mainly on the estimated final product (block diagram), and a full plan (assumption and justification) and an implementation (step by step list).

1.2 Scope

This document will explain the details on each phase as well as provide a general overview of each.

1.3 Definitions, Acronyms, Abbreviations

Term	Description
SDK	Software development kit
V1-V2	Version 1 or Version 2
UI	User interface
OpenGL	Open Graphics Library

1.4 Design Goals

Design goals

The project is split into three phases, which will take place over the course of the Spring and Fall semesters (January to December, 2015)

Phase One

- Conversion of 3D model to 3D mesh of face using time-of-flight sensor data
- Merging of multiple meshes
- Smoothing algorithm for 3D mesh
- Texture overlay on 3D model
- UI for capturing sensor data and generating 3D model

Phase 2

- Algorithms for 3D morphing on selected regions on the face
- UI for selecting individual regions on a 3D model
- UI for manipulating 3D models

Phase 3

- Geometry Calculations for locating sensors for whole body capture
- Algorithms for 3D morphing on selected regions on the whole body

1.5 Deliverables

Phase One

- Conversion of 3D model to 3D mesh of face using time-of-flight sensor data
- Merging of multiple meshes
- Smoothing algorithm for 3D mesh
- Texture overlay on 3D model
- UI for capturing sensor data and generating 3D model

Phase 2

- Algorithms for 3D morphing on selected regions on the face
- UI for selecting individual regions on a 3D model
- UI for manipulating 3D models

Phase 3 (Stretch Goal)

- Geometry Calculations for locating sensors for whole body capture
- Algorithms for 3D morphing on selected regions on the whole body

2. Design Requirements

2.1 Functional requirements

A. Create 3D model by the use of the Kinect that will collect depth data and overlaying light on objects
   a. import mesh into Blender to manipulate the data through the UI

B. 3D reconstruction for post-plastic surgery.
   a. Smoothing textures and morphing 3D models by the use of algorithms and force fields
   b. Edit meshes to produce desired shape of the model

C. User can rotate, expand and edit model through interface design tools

D. Python plugin is compatible with Blender

E. User can view original model on separate screen after modifying it

2.2 Non-functional requirements

UI Design

A. After picture is captured, model renders to Blender in less than 20 seconds

B. Size of rendered model before editing is less than 50 MB

C. Commercially licensable

D. Software has no network requirements

2.3 System Requirements

2.3.1 Hardware/Software Specifications

Hardware Requirements

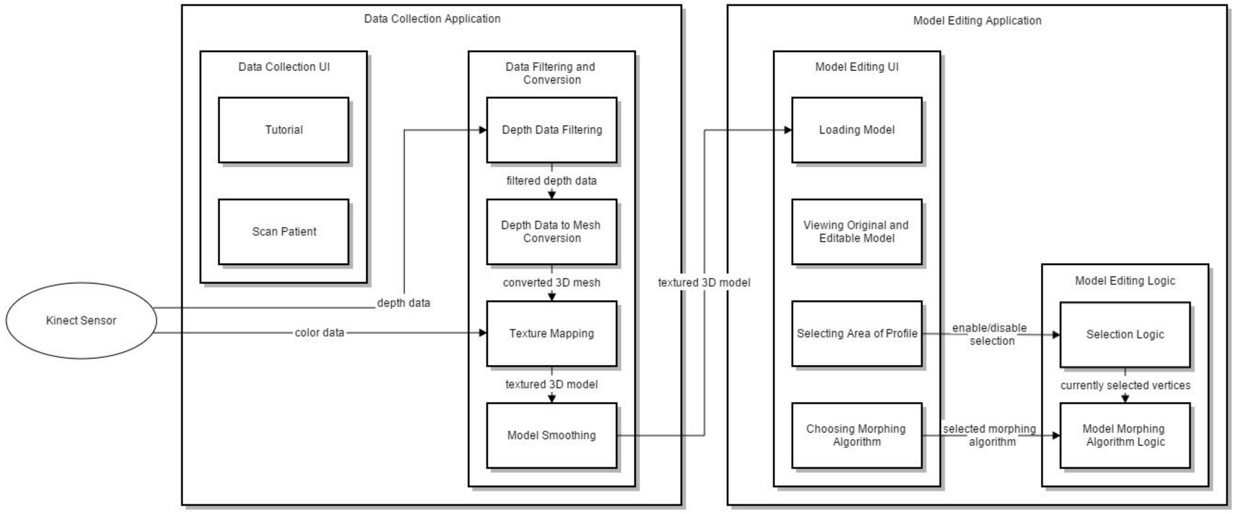
- Computer Specifications
    - 64-bit (x64) processor
    - Physical dual-core 3.1 GHz (2 logical cores per physical) or faster processor
    - USB 3.0 controller dedicated to the Kinect for Windows v2 sensor
    - 4 GB of RAM
    - Graphics card
        - DirectX 11 support
        - OpenGL-compatible
        - 256 MB RAM
    - Mouse or trackpad
    - 24 bits 1280×768 display
- Kinect for Windows v2 sensor

Software Requirements

- Windows 8 or 8.1
- Blender 2.73a

3. Project Design

3.1 Module Description


Data Collection Application
Data Collection UI
Tutorial
Scan Patient
Data Filtering and Conversion
Depth Data Filtering
filtered depth data
Depth Data to Mesh Conversion
converted 3D mesh
Texture Mapping
textured 3D model
Model Smoothing
Kinect Sensor
depth data
color data
textured 3D model
Model Editing Application
Model Editing UI
Loading Model
Viewing Original and Editable Model
Selecting Area of Profile
Choosing Morphing Algorithm
enable/disable selection
selected morphing algorithm
Model Editing Logic
Selection Logic
currently selected vertices
Model Morphing Algorithm Logic

3.1.1 Data Collection Application

3.1.1.1 Data Collection UI

The UI for the data acquisition is displayed using Windows Presentation Format. The users will select whether to view a tutorial on how to use the Kinect sensor to scan a patient to acquire good data or whether to start scanning a patient. Choosing to scan a patient will result in a few screens with reminders on how to position the Kinect sensor and will send the acquired depth data to the data filtering and conversion Module that will output a textured 3D model that the user will save to the system and can be loaded into the model editing application.

3.1.1.2 Data Filtering and Conversion

The data filtering and conversion module will take in depth data information of a scanned patient and the color data of the scan and output a textured 3D model interpretation of the data. The data will initially be filtered to remove walls and noise. Then the filtered depth data will be converted to a 3D model and the color data will be used to create a texture for the model. Finally, the 3D model will be smoothed for a better quality model and the user will be given the option of where to save the converted model.

3.1.2 Model Editing Application

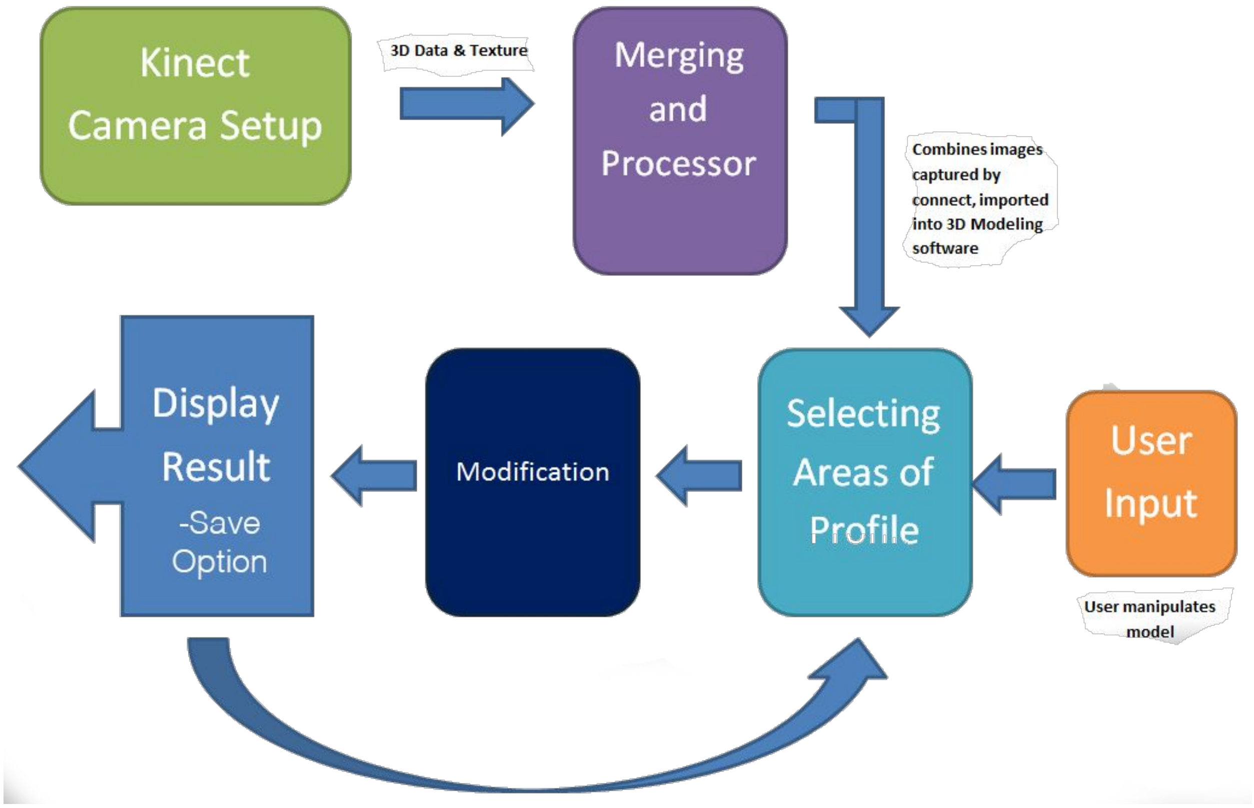
3.1.2.1 Model Editing UI

The UI for the model editing will allow users to load a 3D model created from the data collection application and interactively edit the model while viewing both the unedited and edited model side-by-side. The user will be able to select areas of the profile as well as select and apply various options of applying morphing algorithms on the selected area.

3.1.2.2 Model Editing Logic

The selection logic will calculate the selected vertices of the model and communicate the selected vertices to the morphing algorithm logic. The details of the morphing algorithm logic will be decided upon in phase 3 of the project, after consulting with plastic surgeons to get useful feedback on what type of functionality is desired.

3.2 Block Diagram



3.2.1 Kinect Camera Setup

The Kinect will capture the images of the patient in 3D mesh form.

3.2.2 Merging and Processor

Multiple images are to be captured and then combined into one 3D model. It will then be smoothed and any background will be removed. It will then be imported into Blender.

3.2.3 User input

User will manipulate XYZ modeling plane to become identified with the newly created model and apply any input on changes to the model

3.2.4 Selecting areas of profile

User will then highlight areas of the facial profile

3.2.5 Modification

The modification will be applied at this step after the user has decided which changes to make to the model.

3.2.6 Display Result

The model will update in real time as modifications are applied and then re-rendered. User will have the option at any point to save the model to a chosen format and render out to an image (jpeg, tiff, png, etc) file type.

3.3 Use Cases

3.3.1 Kinect Application

1. User starts up the Kinect and choses the "Capture" option to begin.
2. Patient is in designated area approximately 1.5 meters away from the Kinect.
3. Six profile images are captured of patient
    a. left side view
    b. 3/4 side view
    c. front view
    d. 1/4 side view
    e. right side view
4. Kinect will either verify or deny validity of captured images
    a. Upon valid images user will click export and Blender will open
    b. Upon invalid images, Kinect will guide user to recapture images
        i. Repeated until Kinect accepts all facial profile pictures

3.3.2 Blender Application

1. Blender opens with exported 3D model data. Unaltered model is displayed at all times for comparison with editable 3D model.
2. User has many editing options including
    a. Smooth
    b. Inflate
    c. Cut
3. User can save at any point in time
4. User defines when model is ready to be outputted to an image file
    a. 3D model is edited until user decides to terminate program

3.4 Testing Specifications

All user-end tests will be performed on Windows 8/8.1. UI design will be tested for clarity and ease of use; this will be tested by users outside of the project scope. Each programming milestone will be tested and examined by the client. By the team, the code will undergo extensive unit testing and be tested on multiple units.

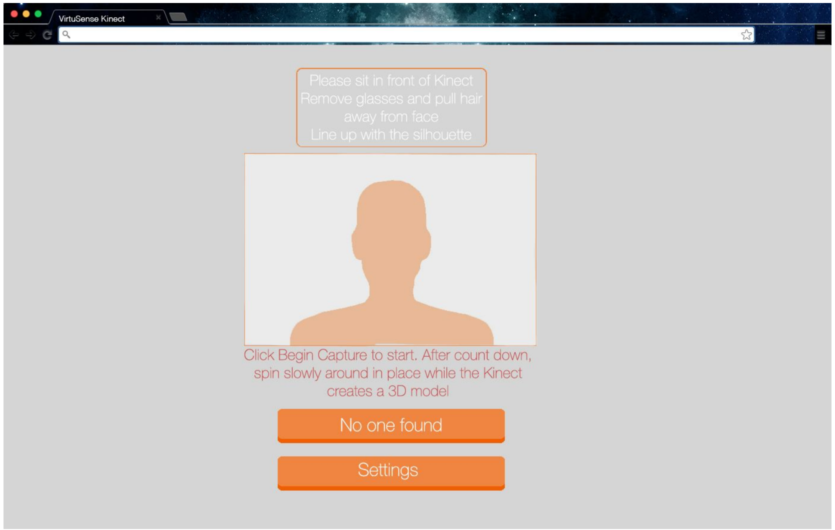
4. Interface Description

4.1 Kinect Interface

The following images depict the Kinet UI web application that will allow users to interact with the camera responsively. The Kinect for Windows SDK JavaScript APIs give HTML5 applications access to Kinect data for interactions and visualization. The web application will be saved locally and require no internet connection. The Kinect needs to be plugged in and functional for the web app to work; the application will not be able to diagnose problems. Ideally, it will work with most web browsers and is depicted as running in Google Chrome.

Screen Sketch 1:
The web application will detect when a person is in front of the Kinect and if they are approximately 1.5 meters away. The web app will not allow for capturing if a person is not present. The silhouette creates an easy guide to show how far away the person needs to be within the correct range of the Kinect.

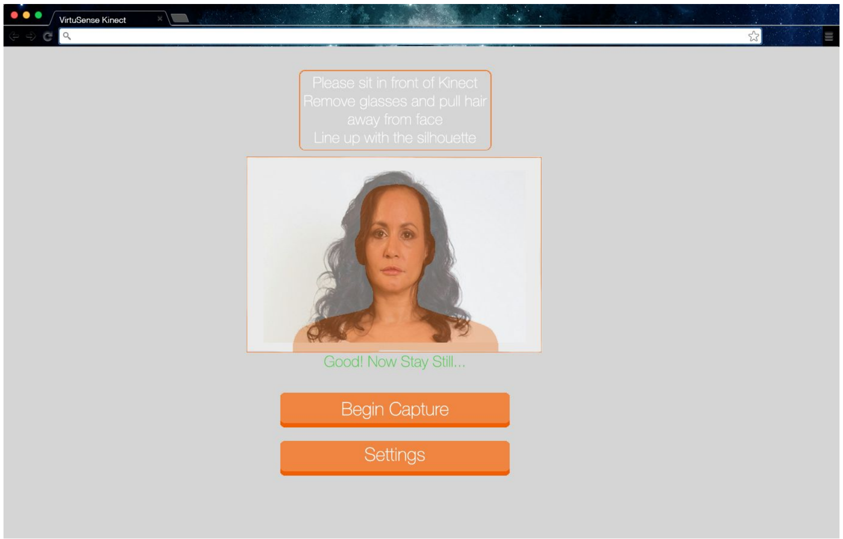
The "Settings" button reserves space for any later features that need to be added. At the moment, the settings will be preset for the Kinect, and the user will not be able to change them.



Please sit in front of Kinect
Remove glasses and pull hair away from face
Line up with the silhouette

Click Begin Capture to start. After count down, spin slowly around in place while the Kinect creates a 3D model

Screen Sketch 2:
Once a person is found and in the correct range, the "Begin Capture" button will appear and the initial instruction will change and turn green to confirm that it sees someone. For best results, glasses need to be removed and hair should be pulled back and away from the face.



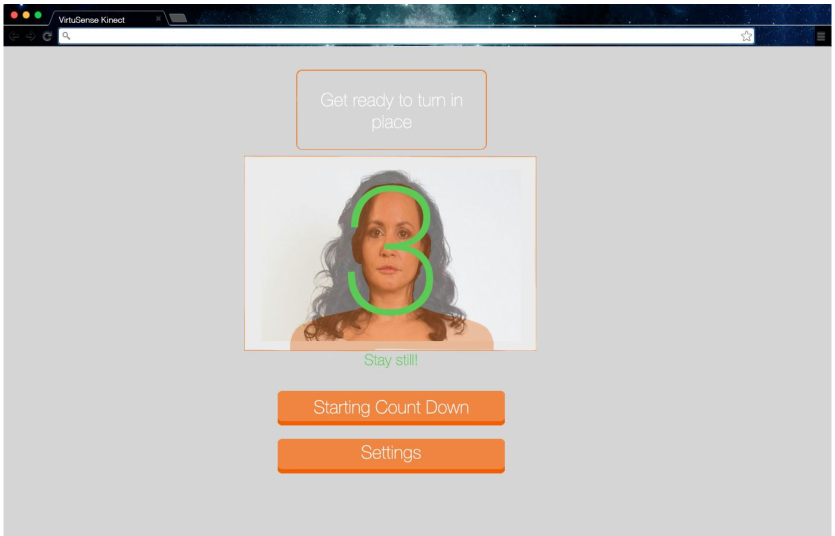
Please sit in front of Kinect
Remove glasses and pull hair
away from face
Line up with the silhouette

Good! Now Stay Still...

Screen Sketch 3:

Begin Capture: A three second countdown will begin, warning the person they will have to begin moving. At this point, buttons can no longer be pressed and the Kinect will scan.

Note: If a tutorial can be worked in, a .gif will be added to demonstrate how slowly a person needs to turn in place so the Kinect can get an accurate reading.



Get ready to turn in
place

Stay still!

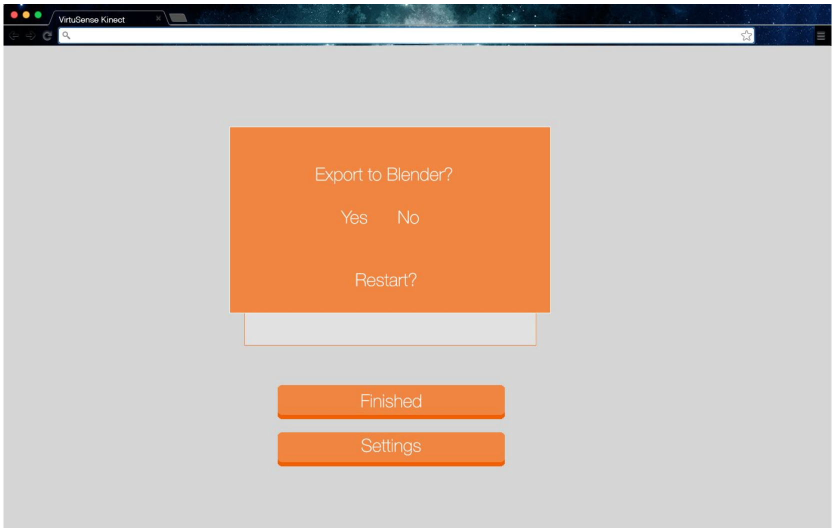
Screen Sketch 4:
After the Kinect has finished, a pop up screen will prompt the user:

Yes:
If they wish to export to Blender, click “Yes” and the web app will revert to the first screen, and Blender will automatically open with the newly created model in a .stl format.

No:
Clicking “No” will not open blender, but the model will still be saved.

Restart:
Clicking “Restart” will not open Blender and will revert the user back to the start screen without saving.

The 3D model will be saved locally to an undetermined place.



4.2 Blender Interface

- Blender will also require a UI to allow user interaction. It will entail a simplified version of the Blender.
- Basic tutorials can also be developed to create ease of use for any first time user when interacting with Blender.

5. Design Rationale

5.1 Licensing Issues

5.1.1 Problem

The Kinect SDK is EU LA and Blender software is GNU GPL, which means whatever software is developed must comply with both licenses. The client wishes to use this project to sell as a product, which will conflict with licenses.

5.1.2 Factors affecting Issue

The way the project is developed is affected by these licenses.

5.1.3 Alternatives and their pros and cons

One solution is simply just moving forward and worrying about the licenses later, which can result in the need to redo techniques or software being used. The other alternative is finding different software to use, which in itself could present difficulties in meeting the requirements of the project.

5.1.4 Resolution of Issue

The client has resources to overcome this issue, and as the design team, we plan to comply with their suggested fixes.

6. Conclusion

In conclusion, this report has discussed about our development on phase 1, 2 and 3 for this project.
Our design for this project splits up the process into two steps, data collection and manipulation, and may evolve as we research more into how to integrate these processes. In this project, we also have a potential licensing issue with using Blender to implement phase 2 of our project and so our design may change to accommodate the solution that we find for this.