3D Modeling with Leap Motion - KickOff Document

Description

Users will be able to construct 3D models of basic objects in an opened image using hand/finger motions and gestures. Creating 3D models of such objects is hard and complicated by using currently available methods. These methods have a high learning curve since they require the usage of complicated softwares and tools such as Blender and Maya. We aim to provide a simpler way where anyone can achieve basic 3D modelling without extensive effort. Users will be able to model objects fastly with the help of Leap Motion Controller.

Master feature list

MF-1: Receive real-time data from Leap Motion Controller

MF-2: Process Leap Motion Controller data and interpret hand movements accurately

MF-3: Have gestures addition to built-in gestures of Leap Motion to enhance user experience

MF-4: Import an image file into the 3D modeling environment.

MF-5: Extract 3D model of an basic object in opened image file with real-time input from Leap Motion Controller

MF-6: Robust 3D modelling even with occlusion (obstacles in front).

MF-7: Robust 3D modelling even with shadows near edges.

MF-8: Controllable depth information with Leap Motion while modelling

MF-9: Save extracted 3D models in a 3D printable file format

MF-10: View saved models, rotate and move a model in 3D space with hand movements/gestures

MF-11: Able to view multiple saved models together in the same environment

Workpackages

WP #	Term	WP title (this should be as short and as descriptive as possible)	Estimated number of man-months
1	491	Project planning and architecture design	3
2	491	Receive and process Leap Motion input, render simple objects like ellipses and rectangles	2
3	491	Design and implement custom gestures for modelling phase, implement functions for recognizing these custom gestures	4
4	491	Render simple 3D objects like cylinder, rectangular prism	3
5	491	Design and implement custom gestures for viewing 3D models, implement functions for recognizing these custom gestures	3
6	492	Implement functionality for importing 2D images to rendering window	1
7	492	Implement services for detecting edges of 2D images	2
8	492	Implement algorithms for detecting wide differences between finger positions in consecutive frames	2
9	492	Implement services for rendering 3D models based on object in imported image and Leap Motion input	8
10	492	Implement save functionality for rendered models, design and implement the platform for viewing saved models	2
		Total:	30

Detailed Descriptions of High-Level Workpackages

WP1 - Project planning and architecture design

In this workpackage, the following functionalities / features / work items will be implemented

1. Develop the list of master features of the project.
2. Produce project development plan in accordance with Master Feature List.
3. Design the overall architecture of the project.
4. Analyze risks and make a management plan.

WP2 - Receive and process Leap Motion input, render simple images like ellipses and rectangles

In this workpackage, the following functionalities / features / work items will be implemented.

1. Receive frames sent by Leap Motion Controller.
2. Process frames and learn finger tip positions, map these to 2D plane.
3. Serialize/deserialize frames for recording scenes/use cases.
4. Render basic 2D images like ellipses and rectangles with variable length radiuses/edges.

WP3 - Design and implement custom gestures for modelling phase, implement functions for recognizing these custom gestures

In this workpackage, the following functionalities / features / work items will be implemented. Leap Motion. These gestures will be crucial for the user experience and thus the project.

1. Design custom gestures for 3D modelling phase.
2. Define patterns of hand/finger movements for designed custom gestures. Implement custom gestures using these patterns.
3. Services for recognizing custom gestures of modelling phase.

WP4 - Render simple 3D objects like cylinder, rectangular prism

In this workpackage, the following functionalities / features / work items will be implemented.

1. Services for rendering cylindrical and rectangular 3D objects.

WP5 - Design and implement custom gestures for viewing 3D models, implement functions for recognizing these custom gestures

In this workpackage, the following functionalities / features / work items will be implemented.

1. Design custom gestures for viewing 3D models.
2. Define patterns of hand/finger movements for designed custom gestures. Implement custom gestures using these patterns.
3. Services for recognizing custom gestures of viewing phase.

WP6 - Implement functionality for importing 2D images to rendering window

In this workpackage, the following functionalities / features / work items will be implemented.

1. Implement services for importing 2D images to rendering window.
2. Implement GUI for importing 2D images from computer.

WP7 - Implement services for detecting edges of 2D images

In this workpackage, the following functionalities / features / work items will be implemented.

1. Implement edge detection algorithms
2. Add services for automatically detecting edges of objects in imported image

WP8 - Implement algorithms for detecting wide differences between finger positions in consecutive frames

In this workpackage, the following functionalities / features / work items will be implemented.

1. Services for detecting wide differences between finger positions in consecutive frames.

WP9 - Implement services for rendering 3D models based on object in imported image and Leap Motion input

In this workpackage, the following functionalities / features / work items will be implemented.

1. Mapping finger tip positions on desired object in image with proper calibration.
2. Generating ellipses while user draws along the cylindrical objects.
3. Generating rectangles while user draws along the rectangular objects.
4. Processing Leap Motion input to provide depth information for altering generated ellipses or rectangles.
5. Altering radii of ellipses generated throughout modelling process of cylindrical objects.
6. Altering dimensions of rectangles generated throughout modelling process of rectangular objects.
7. Connecting generated shapes (ellipses or rectangles) to construct the 3D model.

WP10 - Implement save functionality for rendered models, design and implement the platform for viewing saved models

In this workpackage, the following functionalities / features / work items will be implemented.

1. Services for saving rendered models, including saving in 3D printable format.
2. Design and implement GUI for viewing saved models.
3. Implement functions for opening saved models in viewing platform
4. Implement services for viewing saved models with Leap Motion Controls.

Bonus WP1 - Texture mapping

In this workpackage, the following functionalities / features / work items will be implemented.

1. Extract the texture pattern from corresponding area of object in image.
2. Map texture pattern to rendered 3D model.

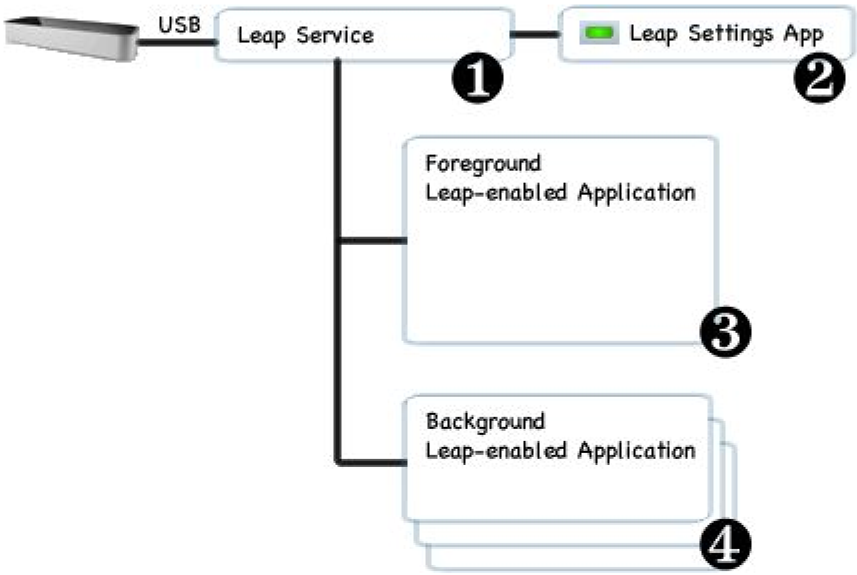
Bonus WP2 - Create a file structure for saved models where users can choose and open saved models from by tags

In this workpackage, the following functionalities / features / work items will be implemented.

1. Implement tag/label feature for saving models
2. Implement GUI where users can browse organized saved models structure by their tags/labels.

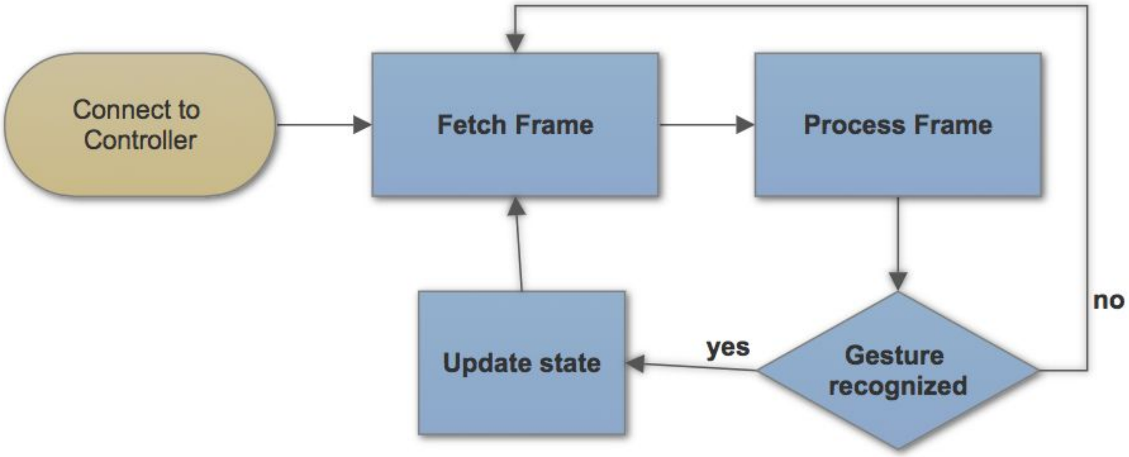
Overall Systems Architecture

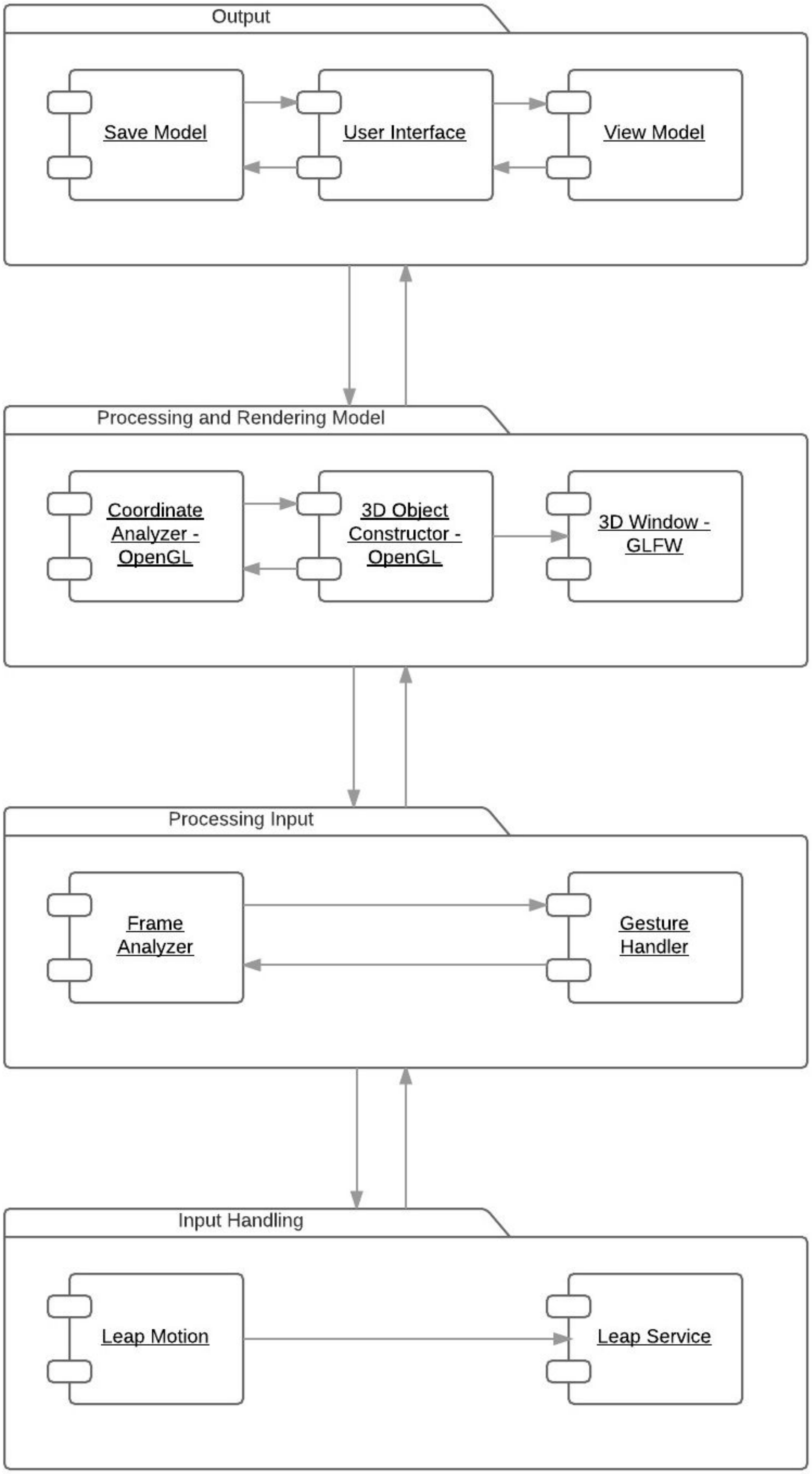
Leap Motion Controller receives data and sends it to Leap Service which processes frames captured by Controller by using Frame Analyzer and Gesture Handler. If Frame Analyzer recognizes a gesture, Gesture Handler handles necessary state changes. If no gesture is recognized, state is to set to read mode again. If a recognized gesture related with modelling, state is updated to construct model and gesture data sent to OpenGL. According to the current state and received data, 3D Object Constructor builds the model piece by piece.



3: Showing of the constructed model
4: Processing frames

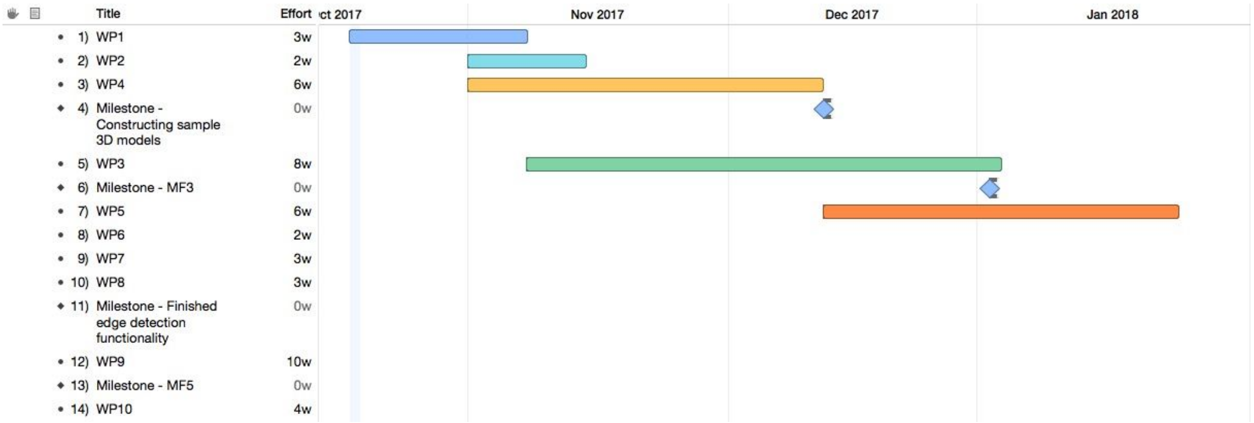
Gesture recognition basic flow chart




Output
Save Model
User Interface
View Model
Processing and Rendering Model
Coordinate Analyzer - OpenGL
3D Object Constructor - OpenGL
3D Window - GLFW
Processing Input
Frame Analyzer
Gesture Handler
Input Handling
Leap Motion
Leap Service

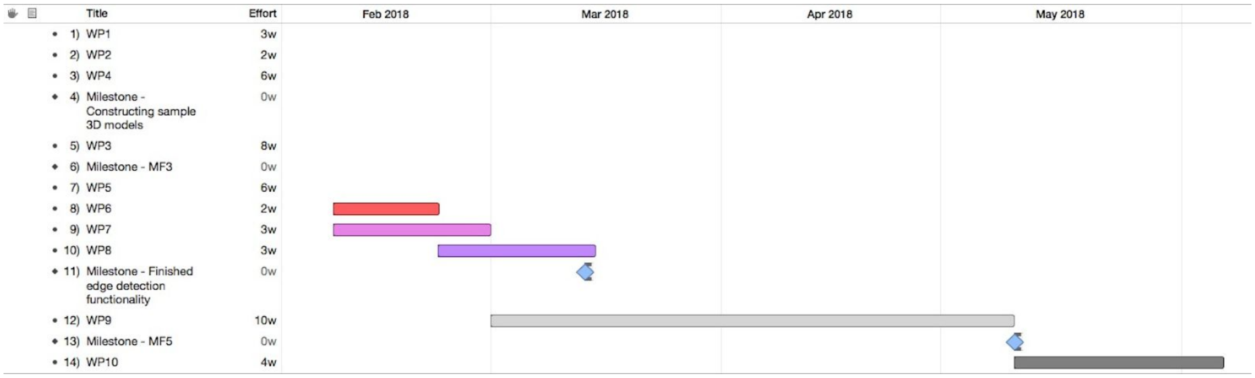
TimeLine

First Semester:



Title	Effort
1) WP1	3w
2) WP2	2w
3) WP4	6w
4) Milestone - Constructing sample 3D models	0w
5) WP3	8w
6) Milestone - MF3	0w
7) WP5	6w
8) WP6	2w
9) WP7	3w
10) WP8	3w
11) Milestone - Finished edge detection functionality	0w
12) WP9	10w
13) Milestone - MF5	0w
14) WP10	4w

Second Semester:



Title	Effort
1) WP1	3w
2) WP2	2w
3) WP4	6w
4) Milestone - Constructing sample 3D models	0w
5) WP3	8w
6) Milestone - MF3	0w
7) WP5	6w
8) WP6	2w
9) WP7	3w
10) WP8	3w
11) Milestone - Finished edge detection functionality	0w
12) WP9	10w
13) Milestone - MF5	0w
14) WP10	4w

Gantt Chart

Risk Assessment

Risk #	Description	Possible Solution(s)
1	Leap Motion can have errors in detecting finger positions.	Inform users by showing position of fingers in the modelling window to avoid misconceptions in finger positions.
2	Gestures are detected wrongly (mixed up).	Define gestures in a simple and clear way to reduce errors. Disable unused gestures according to state of the application.

3	Latency in reflecting real time input from Leap Motion	Design fast frame processing algorithms.
4	Due to possible minor hand tremors, direct lines that user try to draw may not be detected correctly.	Movements of the hand and fingers can be smoothed using error correction algorithms.