Project plan for 3D-NUI – Three dimensional Natural User Interface / Rafi Vivanti

Stages	Description	Deadline
1. Intro	• Get the Leap Motion device and play with it. • Download the LP software: SDK and Developer Kit. • Compile and run the solution: Leap_Motion\LeapDeveloperKit\Examples\MotionVisualizer	1.4. 14
	• Change one of the examples to display the distance between a finger and the screen. Compile it and run it.	15.4. 14
	• Download ITKsnap.exe (free medical image viewer) • Get CT scan of the Liver from Rafi, with a tumor segmentation image. • Display the liver in ITKsnap. It is very hard to see the tumor, since it is very similar to the liver tissue. Use windowing on the histogram (ctrl+I, then move the dots) to find a way to see the tumor. • Display the liver with the tumor segmentation. Find the slice where the tumor is the biggest in all 3 planes. • Use the 'update mesh' button to see the 3D mesh representing the tumor. Rotate it.	1.5.14
	• Download VTK (open-source medical images graphics library) • Compile it (you'll need CMake for it.) • Run an example which displays a slice. • Change the example to work with our liver image, and display the slice with the tumor.	1.6.14
2. Basic Application	Build a windows application which: • Reads one medical image. • Reads the location of the pointing finger using the leap motion. • Displays different slice for every different distance of the finger from the screen. • Draws a big cross on the presented slice-image which represent the location of the finger, • When the user hits and holds the space-bar, it draws a thick red circle on the image where the finger was. The user can change slices while drawing. • Finally it saves the 3D marking to a file.	1.9.14
3. Algorithm	Implement a medical image processing algorithm which: • Reads a 3D medical image. • Reads a robust marking of a tumor on the image. • Runs a graph-based segmentation algorithm to delineate the tumor (details will follow). • Saves the output to a file.	1.11.14
4. Extended Application	Extend the application to: • Read a segmentation image. • Apply the 'Marching cubes' algorithm to get the mesh of it boundary.	1.1.15

	• Present the mesh to the user. (With 3d glasses?) • Read the pointing finger location from the leap. • Present the location of the finger to the user (e.g. a 3d axis centered at the finger) • Let the user scribble on the mesh. He will scribble 'good segmentation' in green and 'leak' in red. Save the scribbles. • Apply a graph-cut algorithm on the mesh (Achiya's implementation) to remove the leaks. • Save the corrected segmentation in a file.	
5. Validation	• Get a database of liver scans with tumors. (from Rafi) • Get ground truth segmentation files. (from Rafi) • Use your application to mark all of the tumors. • Compare numerically the results of your algorithm to the GT, using volumetric and geometric measures.	1.3.15
6. Writing	Present your hard work in these many formats: Demanded by your track: • Written project report. • Project presentation. • Project poster. • Simulation station for the project day. Things we wish: • Upload a Leap-app to the 'airspace' – the leap motion app market. • Submit an abstract to a medical imaging conference.	Final deadline

Guidelines

- Detailed timeline for stages 2-6 will be scheduled.
- All code should be written in C++ or Python.
- All code should be controlled and backed up using Sub-Version-Control software like SVN. We have a repository at CASMIP, but source-forge is better.
- Windows is the preferred environment. If you work on Linux, please make sure it is portable to Windows.
- Compile C++ code using CMake, so it will be portable to Linux if we need it. Most open source in C++ comes in CMake anyway, including VTK.
- I recommend Visual Studio as a Windows editor.
- I suggest running a Visual log; a Word document with everything you did. It should include every step and trial, including failures. Every visual result should be added as a full-resolution print-screen. This will help you tell me what you did in our meetings, and will make your writing easier, especially the images. The log should be backed up in the SVN.