



Augmented Reality Project

In collaboration with Sheba Hospital

Presented by: Zahi Cohen, Noy Gini, Silvan Marti and Adriana Ester Dolgin

Mentors: Yaron Honen, Amit Bracha and Ron Slosberg

Spring 2021

Table of Contents

<u>TABLE OF CONTENTS</u>	<u>2</u>
<u>INTRODUCTION AND ABSTRACT</u>	<u>3</u>
<u>SYSTEM DESCRIPTION</u>	<u>4</u>
VISUAL SYSTEM	4
DEPTH SYSTEM.....	4
INERTIAL UNIT OF MEASUREMENT	4
<u>DESCRIPTION OF THE REGISTRATION PROCESS.....</u>	<u>5</u>
PRECONFIGURE THE 3D MODEL	5
LOADING THE MODEL	5
SELECTING POINTS ON THE MODEL AND ON THE PATIENT'S BODY.....	5
ADJUSTING THE MODEL ON THE PATIENT'S LOWER BACK.....	6
ALGORITHM DESCRIPTION	7
MANUAL ADJUSTMENT	9
MATCH LOCK.....	10
<u>RESULTS</u>	<u>12</u>
<u>CONCLUSION.....</u>	<u>13</u>
<u>RECOMMENDATIONS.....</u>	<u>14</u>
<u>LITERATURE SURVEY</u>	<u>15</u>

Introduction and Abstract

The purpose of this project is to use the augmented reality capabilities of the HoloLens headset to improve a medical procedure. The main task is to adjust a three-dimensional (pre-made) CT model on a patient's lower back, by finding the location and angle at which the model should be allocated, so that it fits exactly to the patient's body.

In this way, it is possible to insert a needle into the right place on the patient's body in a more accurate and simple way than before.

The project extensively uses voice commands, spatial mapping construction, and receiving and processing data from the sensors on the HoloLens headset. It also uses an SVD algorithm for the purpose of fitting the model onto the patient.

System description

In this project Microsoft's HoloLens 2 headset was used which includes advanced sensors providing accurate information about objects in space.

These sensors can be divided into three systems:

Visual system:

The visual system includes four grayscale cameras, a colour front camera with a resolution of eight megapixels and two infrared cameras for eye tracking.

The grayscale cameras are designed for head tracking. Two of these cameras are designed to capture the area in front of the camera. The other two cameras are designed to capture the side views. One for each side to give a wider dimension. These cameras allow video recording at up to 30 frames per second.

Depth system:

The HoloLens headset includes a depth sensor with a megapixel resolution that supports two modes – Capturing for the short range and recording at a high frame rate - 45 frames per second and capturing at a low frame rate designed for the long range (over a meter).

Inertial Unit of Measurement:

This unit includes an accelerometer that allows to get information about linear velocity according to the desired axis. Further, a gyroscope and magnetometer to determine rotation and orientation of the device.

In this project, we make use of all the mentioned sensing components: The visual system is used to identify the user's hand position, the depth system for making a qualitative adjustment of the model to the patient's body and the inertial measurement unit to acquire the orientation and position of the headset in space.

Description of the registration process

The system can be divided into several stages:

1. Preconfigure the 3D model
2. Loading the model
3. Selecting points on the model and on the patient's body
4. Adjusting the model on top of the patient – which includes:
 - a) Using an SVD algorithm – with two following adjustments
 - b) Optional manual adjustment
5. Match lock

Preconfigure the 3D model

There are a few stages which need to be done before loading the 3D pre-made model into the HoloLens: First, split 3D model .stl file into two parts skin and bones. We used blender to perform this operation and exported the model as a .fbx file. Next, we imported the file into a unity project to create a loadable asset bundle. Finally, we uploaded the asset bundle to a cloud and shared the URL.

Loading the model

The asset bundle is loaded into the HoloLens during runtime using the provided URL. To load the model the user must say "load model" or press "L" on an external keyboard. The model will appear at the user's finger location after that.



Figure 1 - selecting points on the 3D model

Selecting points on the model and on the patient's body

To make a qualitative match between the patient's body and the model the user must sample four points on each surface - each point selected on the model having a counterpart on the patient. These points correspond to the locations of the ECG stickers which are pre-placed on the patient's back. Fortunately, these stickers also appear on the CT scan. To select points on the model, the user must use the voice command "model

points", or alternatively press "M" on the connected keyboard. After receiving the desired signal, a window will be displayed, indicating readiness to receive the points. The window location is adjusted to the orientation and distance of the current field of view, using calculations obtained from the headset's sensors. The user must say "A, B, C, D", "1,2,3,4" or "Alpha, Bravo, Charlie, Delta" or alternatively press the "1, 2, 3, 4"-key on the connected external keyboard to select the points. The voice command for selecting points on the patient's body is "body points" or by pressing "B" on the keyboard. Following, the same process is necessary for placing points on the patient.



Figure 2 - selecting points on the 3D model

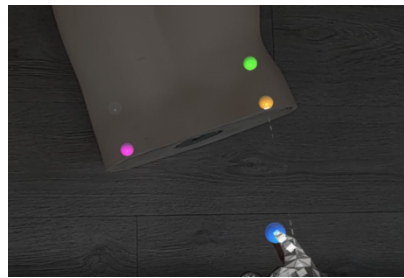


Figure 3 - selecting points on patient's lower back

Adjusting the model on the patient's lower back

The match of the model to the patient is performed by minimizing the error between the two sets of points. The translation and rotation are obtained using an SVD-approach. This process is repeated once with a new grid generated set of points to increase the quality of the fit further. In addition, we enable an option to manually improve the fit.



Figure 4 - model registration on patient's lower back

Algorithm description

Suppose we have two collections of points:

$$X = \{x_1, \dots, x_n\}$$

$$P = \{p_1, \dots, p_n\}$$

We need to find the rotation matrix and the translation vector that satisfy for each point:

$$x_i = Rp_i + t$$

Thus, this problem can be described as a minimization problem according to:

$$\min_{R,t} J = \frac{1}{2} \sum_{i=1}^n \|(x_i - (Rp_i + t))\|^2$$

For convenience, we will define the following variables that express the centre of each point collection:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

$$\bar{p} = \frac{1}{n} \sum_{i=1}^n p_i$$

We describe the minimization problem with the help of those variables:

$$\begin{aligned} \frac{1}{2} \sum_{i=1}^n \|(x_i - (Rp_i + t))\|^2 &= \frac{1}{2} \sum_{i=1}^n \|(x_i - Rp_i - t + (\bar{x} - R\bar{p})) - (\bar{x} - R\bar{p})\|^2 \\ &= \frac{1}{2} \sum_{i=1}^n \| (x_i - \bar{x} - R(p_i - \bar{p})) + (\bar{x} - R\bar{p} - t) \|^2 \\ &= \frac{1}{2} \sum_{i=1}^n (\|x_i - \bar{x} - R(p_i - \bar{p})\|^2 + \|\bar{x} - R\bar{p} - t\|^2 + \\ &\quad 2(x_i - \bar{x} - R(p_i - \bar{p}))^T (\bar{x} - R\bar{p} - t)) \end{aligned}$$

Note that the marked expression resets after the summing up is executed. Therefore, the minimization problem can be addressed as follows:

$$\min_{R,t} J = \frac{1}{2} \sum_{i=1}^n (\|x_i - \bar{x} - R(p_i - \bar{p})\|^2 + \|\bar{x} - (R\bar{p} + t)\|^2)$$

The first expression depends only on the rotation matrix. In order to obtain the translation vector, we will compare the second expression with zero with the resulting rotation matrix.

We mark:

$$x'_i = x_i - \bar{x}$$

$$p'_i = p_i - \bar{p}$$

We find the rotation matrix by minimizing the left expression in the minimization problem, i.e.

$$R^* = \arg \min_R \frac{1}{2} \sum_{i=1}^n \|x'_i - R p'_i\|^2$$

We simplify the expression:

$$\frac{1}{2} \sum_{i=1}^n \|x'_i - R p'_i\|^2 = \frac{1}{2} \sum_{i=1}^n (x_i'^T x'_i + p_i'^T R^T R p'_i - 2 x_i'^T R p'_i)$$

A rotation matrix has a norm equal to 1 and the third term is related to R.

$$\sum_{i=1}^n (-x_i'^T R p'_i) = \sum_{i=1}^n -tr(R p'_i x_i'^T) = -tr(R \sum_{i=1}^n p'_i x_i'^T)$$

We minimize this expression using SVD decomposition.

We define the following matrix:

$$W = \sum_{i=1}^n p'_i x_i'^T$$

In our case W is a 3X3 matrix which can be disassembled into SVD form. We obtain:

$$W = U \Sigma V^T$$

Σ Is a diagonal matrix composed of singular values, when W is full rank, R:

$$R = UV^T$$

We compare the right-hand side expression to zero and get the translation vector:

$$t^* = \bar{x} - R \bar{p}$$

First adjustment

We use the previously mentioned SVD based algorithm. We treat the source collection as the four points selected on the three-dimensional model and the target collection as the four points selected on the patient's lower back. To move the model and adjust it on the patient, the user can use the voice command "move model" or alternatively, press a key "V" on the connected keyboard.

The Second adjustment

To improve the quality of the fit we generate two new sets of points. We select some vertices from the model using a grid and extract the corresponding normal vectors. Using the location of the vertices and the normal vectors we create rays which we analyse if they intersect the surface generated by the HoloLens spatial map. If they do intersect, we use the points of intersection as one and the location of the vertices as the other points. Non intersecting point pairs are discarded. We assemble the point pairs to two sets and perform once again a search for the rotation and translation using the mentioned algorithm.

Manual adjustment

In order to start the manual adjustment the user must use the voice command "manual move", or alternatively, press the key "U" on the connected keyboard. For convenience, it is possible to display the axis system on the model with the help of a voice command "show axis" or alternatively, press the key "I". To hide the axis the user must say "hide axis" or press "J" on the keyboard. The model can be moved and rotated according to each of the three axis X/ Y/ Z. The translation is performed by using the following voice commands: "Move Plus X / Y / Z" ("F1", "F2", "F3" in the keyboard) and "Move Minus X / Y / Z" ("F4", "F5", "F6" in the keyboard). The rotation is performed by "Rotate Plus X / Y / Z" ("F7", "F8", "F9" in the keyboard) and "Rotate Minus X / Y / Z" ("F10", "F11", "F12" in the keyboard).

Match lock

For convenient activity during the surgery, it is possible to lock the model position after the registration. In this mode, the model remains stable on the patient and the points on the patient and everything except the bones is hidden.

The app contains also a help screen that displays the voice commands supported in the system. Entering this screen is done using the voice command “Help”, or by pressing the key "H" on the attached keyboard.

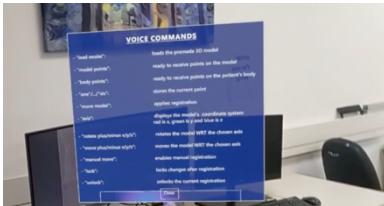


Figure 2 - help screen

Summary for voice commands and their suitable key for an external keyboard:

Key	Command
L	Load model
M	Model points
B	Body points
1,2,3,4	1, 2, 3, 4 or A, B, C, D or Alpha, Bravo, Charlie, Delta
V	Move model
U	Manual move
I	Show axis
J	Hide axis
F1, F2, F3	Move plus X / Y / Z
F4, F5, F6	Move minus X / Y / Z
F7, F8, F9	Rotate plus X / Y / Z
F10, F11, F12	Rotate minus X / Y / Z
H	Help

Demonstration video:

<https://drive.google.com/file/d/1zS7KGK91bKfCvb-Exy7ggh9Wxbjolwlr/view>

Results

Most of the user interaction processes was done by recognizing the user's finger location which is working properly in any light condition.

The voice command system works properly and recognizes the commands even in a noisy environment.

The spatial mapping is done with sufficient resolution supplied by the HoloLens's depth sensor. This helps getting accurate location of the points used for our algorithm, giving us good registration results – the fitted model was in the specified position and rotation required for performing the surgery.

Conclusion

The HoloLens 2 sensor system was found suitable to our project needs. Image recognition of the low back area was found difficult to perform since this area has no special features. Those features are essential for using computer vision algorithms. Also, we found that those algorithms are computationally heavy weight and not suitable for the HoloLens's CPU. Thus, to create a stand-alone app which does not require runtime external computer, we implemented computationally light-weight algorithm.

Recommendations

Currently our algorithm is not fully automatic since it requires user input to define the ECG stickers on the pre-made 3D model and on the patient's low back. To turn it fully automatic, we recommend collecting HoloLens's sensor data during real surgeries and use that data with AI techniques for recognizing the stickers on the pre-made 3D model and on the patient's low back.

Also, we recommend on finding better methods to determine the fit quality.

Last, we suggest finding a more elegant and robust way to load the 3D model.

Literature survey

1. HoloLens 2 API documentation
<https://docs.microsoft.com/en-us/hololens/>
2. SVD & Point Cloud registration:
<https://www.youtube.com/watch?v=dhzLQfDBx2Q>
3. Custom Hand Tracking in Unity for HoloLens
<https://www.youtube.com/watch?v=BKJ6sjJ9oao/>
4. <https://www.youtube.com/watch?v=32gN2SqB66Y/>
5. Mixed reality tool kit unity: <https://github.com/microsoft/MixedRealityToolkit-Unity/>