CTSNET ROBOTICS

Bi-Weekly Report

13 February 2015

Team Information

This document contains a summary of all that has been achieved over the last three weeks by team 19 on the CTSNet Robotics project as a part of the module Systems Engineering II of the University College London Computer Science course. Team 19 is formed of three students, Edward Collins (team leader), Kirthi Muralikrishnan (website and documentation lead) and Tom Page (research and programming lead).

Client Information

Team 19 has three clients:
∼Dr Shabnam Parkar, paediatric surgeon at Great Ormond Street Hospital.
∼Dr Joel Dunning, cardiothoracic surgeon at James Cook University Hospital.
∼Dr Lourdes Agapito, computer graphics and vision expert at UCL and primary supervisor.

Summary of Progress

One of the main events of the last two weeks was that the team met with the Surgical Robot Vision group at UCL, led by Danail Stoyanov. This UCL based group is running numerous research projects on robotic surgery, and have already solved many of the problems that we are faced with. This team owns a Da Vinci surgical robot which we were able to use on a small prop to get a feel for what robotic surgery may be like. We saw here how applicable our project could be, as judging how far away tools were from the prop was extremely difficult, exactly what our project hopes to solve.

Furthermore the team has implemented the green tool identification algorithm. The idea is to use bright green markers on our props to identify them, rather than performing advanced statistical analysis of each frame in order to identify the tool. The team began by creating some props - which are simply rods of aluminium, about the same size as surgical tools, with fluorescent green tape wrapped around the end of the prop to provide the green marker.

The first algorithm designed to identify the green part of the tool was written in Java, and once this was perfected the algorithm was translated into C# so that it could be used with the Kinect. Each team member has written a version of this algorithm based on the original Java one, so that the best solutions could be combined to produce an optimal algorithm. The algorithm now works such that it can identify the green part of the prop in the Kinect's view without interference from the surroundings. Furthermore, this algorithm is fast meaning that the feed from the Kinect does not lag at all due to the algorithm running on each frame.

The next goal is to be able to monitor how far away the green prop is from the camera and present this information in a form of visual augmentation. To do this the team needs to learn how to map the Kinect's colour feed to its depth feed, as they are slightly offset from each other. Once we have worked out how to do this, we will be able to identify objects by colour and then monitor how far away they are from the camera and the surrounding area.

Summary of Team Meetings

Date	Topics Discussed	Time
4 February 2015	Meeting with Dr Agapito to receive advice on progressing with technical aspects of the project.	30 minutes
5 February 2015	Meeting with Dr Dean Mohamedally for milestone review.	30 minutes

Date	Topics Discussed	Time
6 February 2015	Discussion with PhD student Aron Monszpart about technical aspects of the project.	1 hour
9 February 2015	Meeting with Surgical Robot Vision group to discuss project and use the Da Vinci system.	1 hour
10 February 2015	Meeting with Dr Agapito to discuss meeting with Dan Stoyanov and the project.	45 minutes
13 February 2015	Discussion with PhD student Aron Monszpart about how to map Kinect colour to depth feed.	1 hour

Difficulties Encountered

By far the most difficult task over the past two weeks has been working out how to map the Kinect's colour feed to its depth feed. The reason we need to do this is so that we can identify a point by colour and then get depth information about that point. Doing that requires using the inbuilt CoordinateMapper class, as the Kinect's colour and depth cameras are slightly offset. The team has not yet worked out how to use this class, reconcile the information and then display it on the screen due to lack of documentation on how to use these particular functions.

In addition, the team's original algorithm for identifying the green part of a tool proved to be too computationally expensive to use on a frame by frame basis - the video feed lagged and ran very slowly because of this algorithm. The problem boiled down to the fact that a division of high-precision numbers had to be performed on every frame change. The team modified the algorithm, so that it now relies on subtraction to identify the bright green tool. This algorithm recognises the green portion of the prop easily and also runs fast.

Progress Target

By Friday 27 February, the team aims to have:

- Worked out how to reconcile the Kinect's colour and depth feeds, and then develop this algorithm so that the distance of the green prop away from the camera can be presented by visual augmentation of the camera feed.
- Develop the above program to give visual feedback reliant on how far away the tool is from the background, the main deliverable of our project and which in a real life situation would be the body tissue.

Our longer term goals include adding several other visual augmentations to give depth feedback such as one to give an explicit numeric measurement of how close the tool is to the body, as well as adding a form of audio feedback. The team also hopes to work on measurement of the functional part of the tool using the system and working out the tool's orientation. We will also create a Heads-Up Display (HUD) for the system and make it voice controllable.

Individual Description of Tasks

Ed Collins

I developed the Java code that identifies the green part of our props. I made several different iterations of this algorithm, testing it on real images of our props to check that the algorithm was sound. I have however spent the majority of my time working on programming using the Kinect sensor. I have analysed the provided examples and have worked out exactly how to build a Kinect enabled application. I proceeded to translate my Java algorithm into C# and applied this to the Kinect colour feed. Unfortunately the algorithm caused the Kinect to run extremely slowly, so I re-developed the algorithm to make it much faster, meaning that it can now reliably identify the green part of the prop.

Finally I have been working with the Kinect's depth feed, and spent the latter part of this week attempting to learn how to use the CoordinateMapper class so that the colour and depth feed can be reconciled.

Tom Page

Over the last couple of weeks we have managed to meet with Danail Stoyanov of the Surgical Robot Vision group to discuss some of the key ideas of the project. In doing this we managed to see some techniques his team had used in order to identify tools in the images. Although these techniques are far beyond the level we can hope to use, it gave us ideas on how to implement our solution. The key issue we have run into is how to use the mapping function with the Kinect SDK which allows us to map the depth information to the colour image (since this is difficult as there is a slight offset between the colour camera and the depth sensor) to solve this we have sought help from Aron Monszpart and we now have an idea on how to use the coordinate mapping in the Kinect SDK. Over the next couple of weeks we aim to implement this solution and to quickly progress with the project, we believe soon we will be at a stage where we can work on the user interface and be able to experiment with image overlays.

Kirthi Muralikrishnan

Over the last two weeks, I started working on the programming of the Kinect. The first idea was for us to start colour segmentation and identify the tool using it. We have decided to use fluorescent green tape to mark the tool. I have programmed the Kinect in C# and have identified the tool and the body (the green and any shades of red). Once we have identified the tool and background, the next step would be to get it mapped to the depth image. This would help us further determine the vicinity of the tool to the background. I have also done some further reading regarding Markov random fields to combine the probability using depth and colour. The next few weeks, I plan to get some structured programming and an organised approach towards handling the program.