Computational Issues Appendix Stuff

* Accurately projecting 3D objects into an augmented reality space.

The issue in projecting objects into reality is firstly how to display the object. Augmented reality uses overlapping techniques to give the illusion that something virtual is occupying a real space. This starts off as images being placed over a live camera feed, however this ruins the illusion as small movements quickly reveal the fact that it’s just an image and nothing more. To get around this, several techniques are implemented to continue the illusion such as overlaying the camera with a virtual space, so when you place the object, you actually place it in a 3D space, which you can move around. This allows you to be able to correctly capture camera angles of a 3D object. This virtual space will attempt to match the angle and depth of your reality.

We explored several types of AR, and boiled it down to two. Projection based [REF] and Recognition Based[REF].

* Coding the ability to virtually paint walls which will include image segmentation, colour distance calculating and more.

As this is an uncommon computation problem, there aren’t many code extracts we could use to aid us. However we have managed to gather a lot of the computer vision topics we would need to be able to colour walls in virtually. Firstly, we would have to find a way to separate what is ‘wall’ and what isn’t. For this we would need image segmentation, a way of segmenting an image into like categories. From this, we can identify everything that’s similar, and apply an overlay where necessary. In this case, it would be a selected colour.  We could calculate colour distance  to also pick out specific colours in the camera at one time and change or overlap them with another colour. The problem as per our functional prototype, is that there are a lot of similar colours with an image, and colours constantly change throughout a camera’s feed. This causes the colouration to change as well as not cover the area correctly.