Project Progress 1

Teo Sze Lee (HT0301340H)

Most of the work accomplished so far involved research and testing out on various effects to be showcased in the final output video. An initial investigation was carried out to explore the current available motion tracking tools, in order to incorporate CG elements into real video scene. Both Maya (Maya Live) and Adobe After Effects provide such functions, but they involved elaborate and tedious jobs of manually selecting correspondence points and require further fine-tuning.

What I have in mind is using a simple marker for real-time tracking. After conducting the investigations on Maya Live and Adobe After Effects, I am even more convinced that marker tracking could provide a simpler and accurate method of motion tracking, even with moving camera.

The two available marker-board tracking software tools I considered are ARToolkit (developed by Dr Hirokazu Kato of Hiroshima City University) and MXRToolkit (from Mixed Reality Lab, NUS and also available at http://sourceforge.net/projects/mxrtoolkit), the first in is Linux platform while the latter is in Windows. The remarkable thing about these two tools is that the CG elements can be rendered in real-time onto the live video stream. Hence I could almost create a WYSIWYG special effect video production environment. Figure 1 shows the actual video scene I have taken using MXRToolkit using a webcam with the CG element rendered in real time onto the video at more than 30 frames per second (actually since this is just a webcam, most of the frames are being reused for several frames, I think). Of course I am also leaving room for further post production stages in order to fine-tune some areas (by separating out the CG element as separate layer of image sequence)

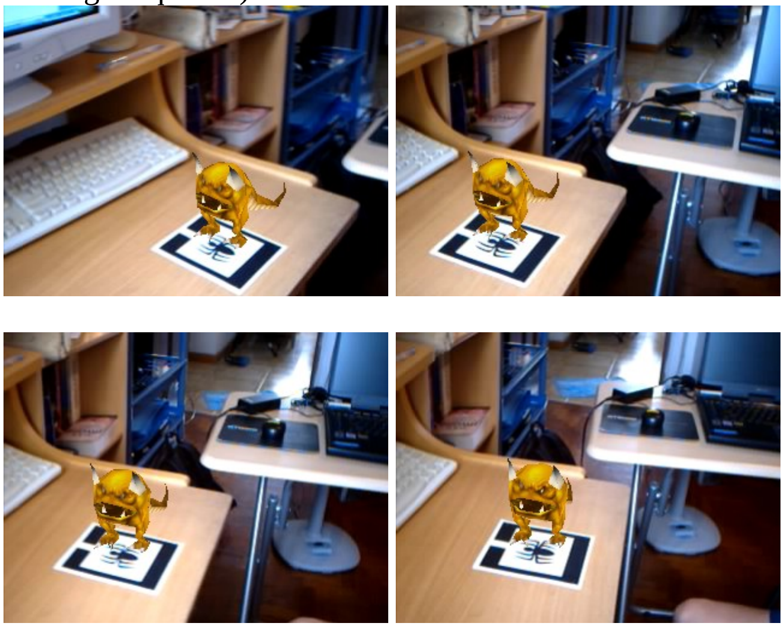

Figure 1: Selected 4 images from a video sequence taken with a moving camera

I chose to use MXRToolkit eventually as it would be more convenient to work in Windows platform, since I would still need perform other post production tasks on software such as Adobe After Effects, Adobe Premiere, Adobe Photoshop etc. Besides, the rendered CG elements are saved as image sequences as a separate layer from of the corresponding video scene, so that further filtering and compositing could be done.

One of the drawbacks of using marker-board tracking is the presence of the marker board itself, which tends to look unnatural. To overcome this, I considered two ways:

1. Drawing another graphical surface to cover the marker. On top of that we can still render other 3D graphical elements on the scene. Figure 2 shows a simple overlay of marker with a brown color plane. Of course the plane could be substituted with texture plane, image or even video. Notice that the CG element can be translated (and rotated) out of the plane, hence giving the impression that the two things are uncorrelated. This technique is useful for moving camera.

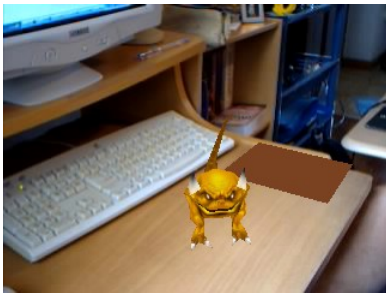

Figure 2: Hiding the marker from the scene by overlaying with other graphical plane.

2. Another method is simply to use a marker-board in the very beginning of the video scene to get a transformation matrix of the reference origin. Then all the CG elements will be based on that reference. Obviously this could only work for stationary camera, which is acceptable in most cases too.

Another problem is the occlusion problem - CG elements can only be rendered on top of the video, occluding every other real scene objects. To overcome this, the occluding real real object is shot separately and superimposed later using clean plate technique. Figure 3 shows the result.

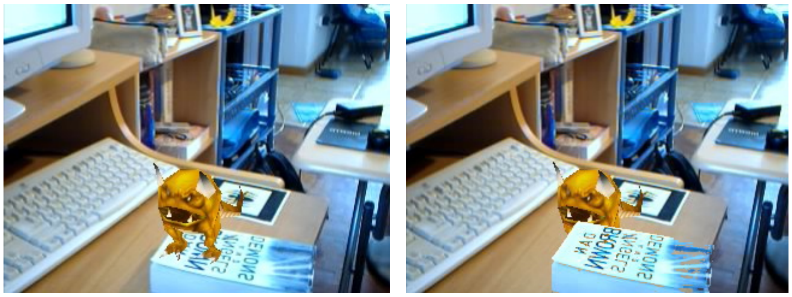

Figure 3: CG element occluding real object on the left when it is supposed to be "behind" the object. On the right, the book occluding the CG elements on the right after post production using clean plate technique.

Similarly, to create the effect of CG elements interacting with real objects, a video of the CG in the real environment is first obtained. This is followed by another same scene without CG elements, but with the presence of objects being interacted (by means of rigs which should be invisible after final composition).

To produce convincing CG that will blend into the environment, lighting and shadow are also important. Conventional method such as using Maya will produce the shadow as a stage after the CG elements are in place. I aim to produce the shadow in real-time together with the CG elements as well by using OpenGL. And again, the shadow will be rendered as separate layers as image sequence for further post production if required. MXRToolkit does not provide shadow rendering, hence I am writing this part myself. The result looks promising on OpenGL primitive polygon (Figure 4), and I am still currently trying to extend it further for other 3D model formats (which seems to take longer than expected).

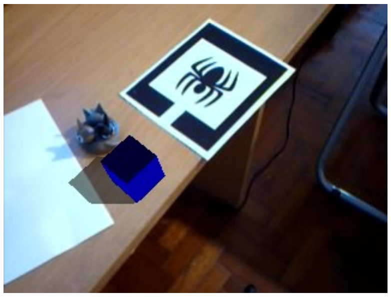

Figure 4: A comparison between the OpenGL rendered shadow (bottom) and the real shadow (top)

MXRToolkit provides a vast choice of media to be loaded, from 2D image and movies, audio such as wav and midi, and 3D models formats (with animation supported in some cases) ranging from md2, md3, obj, vrml, ase and mdl which I am concerned with for this project. The CG element shown in the previous figures is an md2 model (from Quake 2).

Nonetheless, as the MXRToolkit is still in the early stage of development, there still many bugs which I will have to debug by directly examining the source code and recompiling the code. On the other hand, the availability to access the source code also enables me to further customize the SDK for my needs for this project.

The next stage for the coming weeks will be to put everything together (marker-hiding, occlusion, shadow, lighting and CG-real object interaction) and come up with a series of video scenes, which would then be further refined with other software. Besides, I am also planning to revise the story line to make it more interesting.