A generic approach to High Performance Visualization enabled Augmented Reality

Chris J. Hughes[1a] Nigel W. John[a] Mark Riding[b]

[a]University of Wales, Bangor
[b]Manchester Computing, University of Manchester, UK

Abstract

Traditionally registration and tracking within Augmented Reality (AR) applications have been built around limited bold markers, which allow for their orientation to be estimated in real-time. All attempts to implement AR without specific markers have increased the computational requirements and some information about the environment is still needed. In this paper we describe a method that not only provides a generic platform for AR but also seamlessly deploys High Performance Computing (HPC) resources to deal with the additional computational load, as part of the distributed High Performance Visualization (HPV) pipeline used to render the virtual artifacts. Repeatable feature points are extracted from known views of a real object and then we match the best stored view to the users viewpoint using the matched feature points to estimate the objects pose. We also show how our AR framework can then be used in the real world by presenting a markerless AR interface for Transcranial Magnetic Stimulation (TMS).

Keywords: Augmented Reality (AR), High Performance Visualization (HPV), Grid

1. Introduction

Augmented Reality (AR) applications superimpose computer-generated artefacts into the user's view of the real world. These artefacts must be correctly orientated with the viewing direction of the user who typically wears a suitable Head Mounted Display (HMD). AR is a technology growing in popularity in medicine, manufacturing, architectural visualization, remote human collaboration, and the military [1, 2].

To create the illusion of a virtual artefact within the real world, it is essential that the virtual object is accurately aligned and that the computer graphics are presented in real time. Most of the existing solutions involve the use of bold markers that contain contrasting blocks of colour and shapes making them easily identifiable using computer vision techniques. To align virtual artefacts into the real world three main stages are required – see figure 1. Firstly we need to examine the user's viewpoint and identify where our virtual objects belong in the scene. Secondly we need to track the object to ensure that we have aligned the object to the correct position. Finally we use pose estimation to calculate the orientation of the object so that we can align it with the real world.

The Human Interface Technology Laboratory at the University of Washington has developed the ARToolkit, a software library providing the tools for creating marker based AR applications. The ARToolkit has provided the foundation for many of the early developments in AR and make it possible to rapidly produce AR applications using an inexpensive webcam and an average specification PC [3].

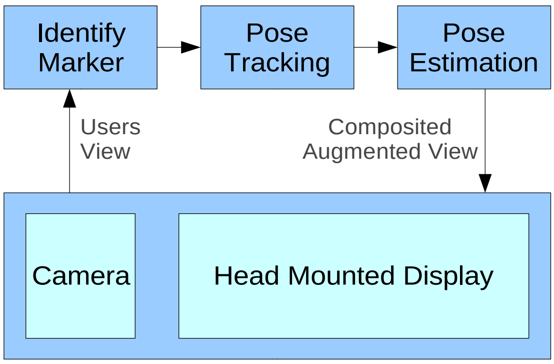

Figure 1: The three stages involved in AR

Moehring, Lessig and Bimber show that when using markers that have high contrast (for example, they use a bold type on a white background), little processing power is actually required to estimate the pose of an object even with the poor capture quality and low processing capability of a standard consumer mobile phone. [4]

Although the use of markers in optical tracking enables the pose estimation to be calculated relatively easily, having to use specific markers can limit the possible AR tools that can be made available. Therefore there are now many examples of AR solutions which do not require markers [5, 6, 7].

[1] Contact Author: chughes@informatics.bangor.ac.uk

In order to be successful, marker-less tracking is not only more computationally intensive, but also requires more information about the environment and the structure of any planes or real objects that are to be tracked. In this paper we present a generic solution that does not rely on the use of markers, but rather feature points that are intelligently extracted from the users view. We also provide a solution to the computationally intensive task of pose estimation and the rendering of complex artefacts by exploiting remote HPV resources through an advanced environment for enabling visual supercomputing – the e-Viz Project [8].

2. Robust feature point detection

In order to align our virtual object with the real world, we first need to define the object in the users view. During a calibration stage the user is given the opportunity to specify where the object exists within the viewpoint. We use a robust feature point detection algorithm to identify the points that can be repeatedly identified within the space occupied by the virtual object and use this information to estimate the objects position and orientation.

There are many existing methods for extracting feature points, most of which are based on corner detection algorithms. Since the 1970's many feature point detectors have been proposed and there is still work today to improve their accuracy and efficiency. There are three main methods for detecting feature points in images, which stem from the following methods: edge-detection, topology and autocorrelation [9, 10].

It is generally accepted that the autocorrelation methods yield the most repeatable results and they all follow the following three steps:

1. For each point in the input image a Cornerness value is calculated by the operator, and relates to how likely it is believed that that point is a corner.

2. A threshold value is used to disregard any points that are identified but are not strong enough to be true corners. The Cornerness value of these points is then typically set to zero.

3. Non-maximal suppression sets the Cornerness value for each point to zero if its cornerness value is not larger than the cornerness measure of all points within a certain distance. This ensures that we only find maximum points and so we can then assume that all non-zero points are corners.

2.1 Moravec/ Harris algorithms

A very basic algorithm was proposed by Moravec in 1977 as part of his work on machine vision enabled robotics [11, 12]. He proposed that a point could be identified as a feature point if there was a significant intensity variation in each direction from that point. Although this algorithm provides basic feature detection without being too computationally intensive, it is not repeatable as the points it finds are only repeatable when the edges are at 45º or 90º to the point being evaluated. The Harris algorithm [13] improves the Moravec algorithm but at a significant cost to the computational requirements. It becomes more robust by changing the way intensity variation is calculated between each pixel and its neighbours by allowing for edges that are not at 45º or 90º to the point being evaluated.

The Harris algorithm uses first order derivatives to measure the local autocorrelation of each point. A threshold value is then used to set all of the weaker points to zero leaving all of the non zero points to be interpreted as feature points- see figure 2.

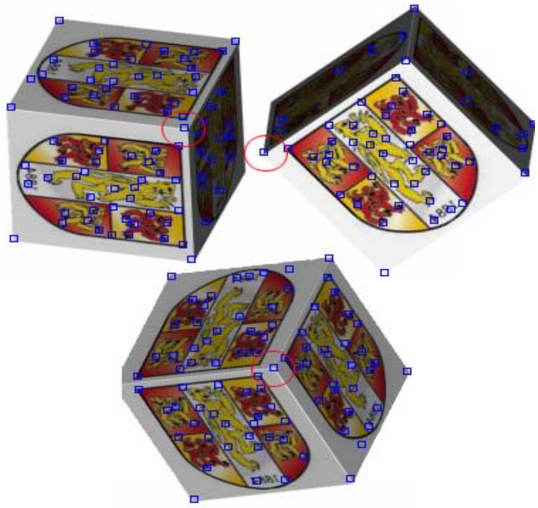

Figure 2: The Harris algorithm detecting repeatable feature points. The circles show a single point which has been accurately repeated in each movement of the cube.

3. Pose Tracking and Estimation

Our application uses the calibration information to train a Haar Classifier [15]. We have extended previous work with Haar Classifiers [16] by using multiple calibration views allowing the detector to not only be more robust but also to continue to track different sides of an object. We also maintain real-time performance even with the increased computational load by distributing the pose estimation as part of our visualization pipeline.

4. Utilizing HPV with e-Viz

The e-Viz project [8] is currently under development and aims to provide a generic flexible infrastructure for remote visualization using the Grid [17]. e-Viz address many of the issues involved in HPV [18] using a combination of intelligent scheduling for new rendering pipelines and the monitoring and optimisation of running pipelines, all based on information held in a knowledge base. This provides an adaptive visualization service that provides rendered graphics reliably without the application or user even being aware of what resources are being used. It also allows the application to render graphics in real time at a resolution that would normally be too high for the client machine.

We have followed two paths for implementing our application with e-Viz:

- **Rendering the graphics with e-Viz**

The first implementation simply uses e-Viz to render the virtual artefacts present in our AR view. The user's viewpoint is captured by the local machine and the pose estimation is calculated locally. The pose transformation is used to steer the e-Viz visualization pipeline, which in the background sends the transformation information to an available visualization server. Our client then receives the rendered image and composites it locally into the users view.

- **Distributing the pose estimation module as part of the visualization pipeline.**

In order to fully take advantage of the e-Viz resources the second implementation moves the pose estimation module onto the e-Viz visualization pipeline. In this case the e-Viz visualization is steered directly by the video stream of the users view. e-Viz distributes the pose estimation module to a suitable and available resource. The pose estimation module then steers the visualization pipeline and returns the final view back to the user after compositing the artificial rendering into the real scene.

4.1 The e-Viz API

e-Viz provides a client application that can be used to control a remote visualization pipeline as well as providing a viewer for the remotely rendered graphics to be returned to the user. It also provides an API that allows users to develop their own client applications which can utilize the e-Viz resources.

The e-Viz framework uses a web service to decide which hardware and software to make available to the client, based upon what resources are needed and what resources are available. The broker uses a knowledge base to store the status of the available servers and inventory what resources they are capable of providing. The client can interact with the Broker by the use of gSOAP calls, which will tell the Client which visualization servers to connect to.

There are generally multiple visualization servers within the e-Viz environment. Having discovered which visualization servers to use, the Client application uses a Grid middleware (such as GT2) to connect to the remote server and run the visualization task. By providing a wrapper to different visualization applications it makes it possible to execute your visualization pipeline on any visualization server regardless of what visualization software it is running.

5. Exemplar application

Transcranial Magnetic Stimulation (TMS) is the process in which electrical activity in the brain is influenced by a pulsed magnetic field. Common practice is to align an electromagnetic coil with points of interest identified on the surface of the brain, which can be stimulated helping researchers identify further information about the function of the brain. TMS has also proved to be very useful in therapy and had positive results with treating severe depression and other drug resistant mental illnesses such as epilepsy [19, 20].

In previous work we developed an AR interface for TMS using an optical tracking system to calculate the pose of the subjects head relative to user's viewpoint [21]. We are now developing a new AR interface that uses our generic framework – see figure 3. By removing the need for expensive optical tracking equipment our software will provide an inexpensive solution, making the procedure more accessible to training and further research.

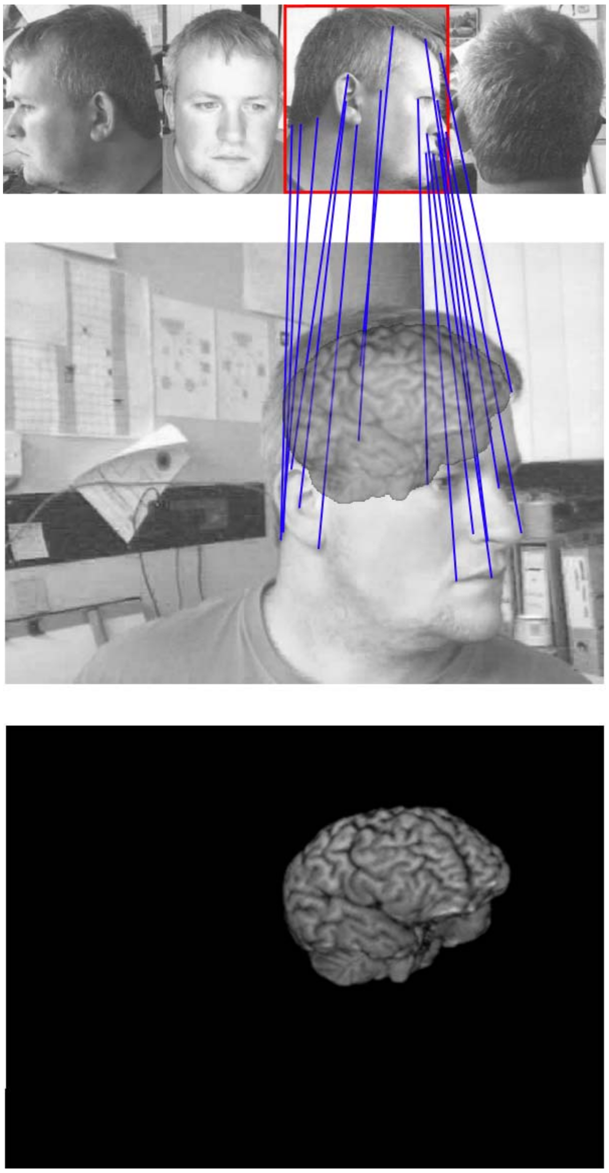

Figure 3: (a) The four images used to train the Haar classifier with the best match highlighted. (b) The real-time users view with lines illustrating some of the matched feature points. (c) The e-Viz remote rendering view.

Our research has shown that although although an average desktop PC does struggle with the pose estimation, using remote resources can ensure real-time performance. Provided the visualization server is appropriate for the rendering task the e-Viz framework is able to return the rendered artefact to the user at a reliable 15 FPS, where there is little latency. On congested networks e-Viz uses stricter compression algorithms at a cost to the image quality to try and maintain these usable frame rates.

6. Conclusions

In conclusion we have found that our approach to producing a framework for AR has been very successful, provided that optimum conditions are available. Problems occur when trying to run the pose estimation locally. It is simply too computationally intensive and so can not keep up with the real time video. Distributing this calculation to a more powerful grid resource has solved this problem.

Future work will concentrate on improving the efficiency and reliability of the feature point detection algorithms, ensuring that we have more accurate pose estimation between frames. We also need to introduce heuristics that will help predict the position of the virtual artefact, even if we are unable to calculate the pose of the object, by building up a knowledge base of previous frames.

Acknowledgements

This research is supported by the EPSRC within the scope of the project: “An Advanced Environment for Enabling Visual Supercomputing” (GR/S46567/01). Many thanks to the e-Viz team at Manchester, Leeds and Swansea for their help and support.

We would like to thank Jason Lauder and Bob Rafal from the School of Psychology, University of Wales, Bangor, for allowing us access to their TMS laboratory and for supplying data from past experiments.

References

[1] Azuma, R. T., “A survey of Augmented Reality, Presence: Teleoperators and Virtual Environments”, Vol 6, No. 4 (August), pp. 355-385, 1997

[2] Azuma, R. T., Baillot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B., "Recent Advances in Augmented Reality", IEEE Computer Graphics and Applications 21, 6 (Nov/Dec 2001), 34-47.

[3] Kato, H., Billinghurst, M., “Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conference System”, In 2nd IEEE and ACM International Workshop on Augmented Reality , San Francisco USA, 1999 pp.85-94.

[4] Mohring, M., Lessig, C., Bimber, O., “Video See-Through AR on Consumer Cell-Phones,” ismar, pp. 252-253, Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'04), 2004.

[5] Comport, A.I., Marchand, É., Chaumette, F., “A real – time tracker for marker-less augmented reality”, In proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 36-45, 2003.

[6] Genc, Y., Riedel, S., Souvannacong, F., Navab, N. “Marker-less tracking for AR: A learning-based approach”, In proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 295-304, 2002.

[7] Striker, D., Kettenbach, T., “Real-time and marker-less vision-based tracking for outdoor augmented reality applications”, In proceedings of IEEE and ACM International Symposium on Augmented Reality, pp. 189-190, 2001.

[8] Riding, M., Wood, J., Brodlie, K., Brooke, J., Chen, M., Chisnall, D., Hughes, C., John, N.W., Jones, M.W., Roard, N., "e-Viz: Towards an Integrated Framework for High Performance Visualization", Proceedings of the UK e-Science All Hands Meeting 2005, EPSRC, ISBN 1-904425-53-4, pp1026-1032

[9] Zitova, B., Flusser, J., Kautsky, J., Peters, G., “Feature point detection in mulitframe images”, Czech Pattern Recognition Workshop, February 2-4, 2000.

[10] Hodges, K. I., “Feature-Point Detection Using Distance Transforms: Application to Tracking Tropical Convective Complexes”, American Meteorological Society, March 1998.

[11] Moravec H.P, “Towards Automatic Visual Obstacle Avoidance”, Proc. 5th International Joint Conference on Artificial Intelligence, pp. 584, 1977.

[12] Moravec H.P., “Visual Mapping by a Robot Rover”, International Joint Conference on Artificial Intelligence, pp. 598-600, 1979.

[13] Harris, C., Stephens, M., “A Combined Corner and Edge Detector”, Proc. Alvey Vision Conf., Univ. Manchester, pp. 147-151, 1988.

[14] Trajkovic, M., Hedley, M., “Fast Corner Detection”, Image and Vision Computing, Vol. 16(2), pp. 75-87, 1998.

[15] Wilson, P. I. and Fernandez, J. 2006. Facial feature detection using Haar classifiers. J. Comput. Small Coll. 21, 4 (Apr. 2006), 127-133

[16] Paul Viola and Michael Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In Conference on Computer Vision and Pattern Recognition, pages 511–518, 2001.

[17] Foster, I. and Kesselman, C. (eds.). The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, 1999.

[18] Riding, M., Wood, J., Brodlie, K., Brooke, J., Chen, M., Chisnall, D., Hughes, C., John, N.W., Jones, M.W., Roard, N., “Visual Supercomputing - Technologies, Applications and Challenges”, Computer Graphics Forum, Vol. 24 Issue 2, 2005 pp217-245

[19] Ettinger, G. et al., “Experimentation with a Transcranial Magnetic Stimulation System for Functional Brain Mapping”, Medical Image Analysis, 2(2):133 - 142,1998

[20] Walsh V, Rushworth M., “A primer of magnetic stimulation as a tool for neurophysiology”, Neuropsychologia, Peergamon, 1999 Feb;37(2):125-35.

[21] Hughes, C. J., John, N.W., "A flexible infrastructure for delivering Augmented Reality enabled Transcranial Magnetic Stimulation", In Proc. Medicine Meets Virtual Reality 14, January 24-27, 2006, Long Beach, California, pp219-114