

Haptic Navigation and Exploration of High Quality Pre-rendered Environments

R.G. Laycock^{†,1} S.D. Laycock¹ and A.M. Day¹

¹School of Computing Sciences, University of East Anglia,
Norwich, NR4 7TJ, UK

Abstract

Visualising in real-time high quality virtual environments, which are suitable for cultural heritage and virtual tourism, is often a challenging endeavour. This is primarily, due to the cost of rendering complex architectural structures. Incorporating the sense of touch into real time environments enhances a user's experience, leading towards a higher level of immersion. However, the confining requirement of the haptic feedback loop to cycle at 1000Hz has led to many applications employing lower quality scenes. In this paper a technique is presented which permits high quality pre-rendered animations of dynamic environments to be both visualised and navigated at high interactive rates. In particular, the approach provides the user with the ability to touch the content of the animations and to freely orientate themselves in any direction, leading to haptically aware movies. Consequently, this permits an exploration of an otherwise prohibitively complex scene consisting of large volumes of geometry and texture maps, combined with realistic lighting models. This component is illustrated within a virtual tour framework enabling users to navigate semi-prescribed routes using haptic feedback both in a spatial and temporal context.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Interaction Techniques

1. Introduction

Virtual environments provide an excellent domain in which to visualise archaeological artefacts. Unlike the traditional museum environment, where an artefact is frequently displayed out of context in a glass cabinet, the virtual environment enables the object to be digitally reconstructed and reunited with additional objects that may be stored at different geographical locations. Furthermore these objects may be visualised in the correct context and augmented with additional media to enhance the user's experience. Besides the visual impact obtained from displaying a set of objects within a geographically and temporally accurate virtual environment, additional information may be perceived via the incorporation of a haptic feedback device. A haptic feedback device provides the user with the ability to touch three-dimensional virtual objects and to feel their shape and con-

tours. This is particularly important in obtaining a better understanding of an object's three-dimensional form.

To maintain a smooth and continuous interaction between the haptic feedback device and the virtual objects the feedback loop is required to cycle at 1000Hz. Consequently the graphical rendering is typically compromised, reducing the scene's fidelity to permit the desired interactive rate to be achieved. In this paper an animation is adopted to represent the virtual environment. It thus permits large three dimensional environments, consisting of millions of vertices and many textures, to be rendered offline under computationally expensive illumination effects. By employing animation in this way the user is provided with high quality imagery representing the objects within a virtual environment. To gain further insight into an object's structure techniques are developed to enable the user to touch the objects displayed within the environment via a haptic feedback device. By giving the user the ability to touch the virtual environment an increased level of interactivity will be realised.

[†] rgl@cmp.uea.ac.uk

The main contribution of this paper is a novel technique that enables users to interactively navigate and explore a high quality pre-rendered environment with haptic feedback. This is particularly pertinent for virtual tours of cultural heritage locations, where obtaining high fidelity images at the desired interactive frame rates is difficult to achieve.

2. Previous work

There are several published works on the concept of representing a virtual museum, [BBF*02] [OBP05] [Bre01]. In particular Bergamasco et al. in [BBF*02] describe the Pure Form project, which is motivated by the requirement to include haptic feedback and therefore enable the full appreciation of three-dimensional artefacts. McLaughlin et al., in [Bre01], agrees with this claim adding that by creating programs with haptic feedback a user is able to touch fragile objects without adversely affecting their conservation. However, these approaches require a computer generated image to be rendered at interactive frame rates. Even with the progression of advanced graphics hardware technology simpler virtual environments are inevitable. These comprise of smaller texture maps and models lit with lower quality lighting calculations, [LWR*03] and this leads to a restriction when attempting to interact with large high fidelity scenes in real time.

In [MJ06], Morris et al., targeted this issue by combining a single ray-traced image with haptic feedback for an abstract scene. Their technique was driven by the concept that in a haptic feedback environment the viewpoint rarely alters, suggesting that views from static locations from within a scene are acceptable. However, in a virtual tour of a cultural scene, it is a requirement that the user is able to traverse the scene and stop to touch interesting artefacts along the way. For instance, it would be beneficial to observe a particularly ornate sculpture, in context, attached to the façade of a building. During the traversal of the environment further insight into the three dimensional work undertaken into the decorative sculpture could be gained through haptic interaction.

The functionality provided for the programmer to record the haptic device's movements has been utilised by the haptics community for many applications including virtual training of surgeons [WSJH04] and rehabilitation for stroke patients [RMJ*05]. The ability to record the device's movements and to play them back has been exploited by [GMS06] for enabling users to receive pre-recorded haptic feedback whilst watching a movie. The haptic feedback paths were manually constructed to match an action in the movie. The system did not allow direct contact between the device and the content of the movie.

3. An approach to exploring and navigating a pre-rendered scene

This section provides details on exactly how the methods have been implemented. Initially a computer graphics (CG) representation of the area to be modelled must be created. The CG representation may then be used to render movies with different paths throughout a scene. The focus of the method is to allow the user to interact with some of the objects displayed in the movie utilising a haptic feedback device. The next three sections in this paper discuss how a navigable movie can be created, how the touchable content is stored and how the haptics may be incorporated. In addition to touching the objects within the movie the user may use the haptic device to orientate themselves and to also modify the speed at which the movie is playing. It has been shown that adding these physical methods of interaction assists the user with media control [SMS*01]. Snibbe et al. [SMS*01] implied that since humans are used to interacting with visual media via physical systems techniques should be developed which replaces the physical sensations that are lost when interacting with the generic keyboard and mouse. In this section techniques for using the Phantom Omni device for control of head orientation and the movie play speed are discussed. In addition to this the haptic device is utilised to manipulate a user interface designed specifically for cultural heritage applications.

3.1. Navigable movie creation

Computer generated scenes are advantageous for cultural heritage since they provide the ability to reconstruct environments that no longer exist. In this paper three-dimensional virtual environments are constructed from archive data using 3ds Max. The interactive modelling package is sufficiently powerful to enable the construction of large environments. Furthermore, it has the capability of importing a wide range of common model formats that may be used to store the large volume of digitised artefacts that exist today. To provide the context in which the cultural heritage site may be visualised an automatic urban modelling technique is adopted, [LD03]. This approach utilises building footprints and road layout information in conjunction with height data to automatically construct the terrain and the buildings. Each building's cross section is specified by its footprint and it is converted into a three dimensional volume by incorporating the height data. Various roof modelling techniques are employed to provide each building volume with a roof model based on its cross section and height. The resulting geometric model is subsequently improved via the integration of ground level images, which were previously compiled into a texture library.

A significant factor contributing to the fidelity of a virtual environment is the illumination that is directed upon it. Many different light sources are required ranging from the light induced by the sun and sky for exteriors to interior spot lighting. Many surfaces are required to accurately represent

the interactions between these light sources and the objects in the three dimensional models. Consequently the geometry of the scene increases. A direct result of increasing the light sources and polygon count of the scenes is an increase in rendering time. Therefore the proposed solution is to maintain the visual quality obtained using an offline renderer and display the results in real time using an image based rendering technique.

Chen's paper on QuickTimeVR, [Che95], describes a number of techniques that enable the visualisation of offline rendered movies in real time. A particularly interesting theoretical concept presented involves the capture of many cube maps at small intervals along a fixed path. The cube maps are consequently switched in and out to enable the user to perceive a continuous motion. This approach is well suited to the application of a virtual tour since it enables a user to follow a fixed guided tour path leaving them to concentrate on interacting with their surroundings.

Using 3ds Max a spline is constructed and six cameras are attached to follow it. Each of the six cameras has a field of view equal to 90 degrees and is orientated to point along the six axes of a three dimensional coordinate system. To promote the easy generation of the six respective animations MaxScript is written to enable the automatic construction of the six cameras and the rendering of the corresponding movies.

The real time rendering component is implemented using OpenGL. It consists of a cube, with each face being textured with a frame from the corresponding movie. During rendering the camera is positioned in the centre of the cube and as time passes the appropriate frames from the movie are extracted and applied to the faces. In accordance with Chen's theoretical concept this simulates the effect of a virtual tour on a vehicle, where a user travels along a predefined path but is free to obtain a new view from any orientation. As the user changes their orientation view frustum culling is employed to ensure only the visible faces of the cube are updated. This reduces the time required to decode the necessary frames from the movie files.

3.2. Creating touchable content

By incorporating a force feedback device the user is able to physically interact with the virtual world. It enables the user to explore the scene and appreciate the three dimensional structures contained within the movie. This section discusses how the haptic representation is constructed leading to *haptically aware movies*.

A fundamental component of the haptic feedback system is the knowledge concerning the location and topology of the touchable objects. Therefore it is desirable to obtain information from 3ds Max concerning the position of both the camera used to acquire the movie and the touchable objects. This information is compiled using MaxScript to acquire the

six degrees of freedom translation and rotation parameters over the entire timeline of the animation.

First the virtual camera is selected and the MaxScript is run to convert the coordinates from the right handed coordinate system of 3ds Max into the left handed coordinate system of OpenGL. The converted coordinates are exported into a file for use by the real time haptic rendering component. This procedure is repeated for each touchable object in the scene. In addition each touchable object is exported in .3ds format to enable the structure of the objects to be known. The objects which will be touched are not required to be the same as the objects which are visualised. Section 3.3.1 provides more information on the permissible representations for the touchable objects.

The haptic rendering system imports the three dimensional objects along with the camera and object transformation parameters. These are used in the following way to determine how each object is related to the camera. A transformation matrix, C_{tm} , is stored for the camera in world coordinates. A transformation matrix, O_{tm} , for every object is also stored in world coordinates. Figure 1(a) illustrates these matrices in respect of one object and the camera. The origin of the scene used for the visualisation and the haptics requires the viewer to be positioned at point(0,0,0). For this reason the camera is set at point(0,0,0) and the objects in the scene are transformed relative to the position of the camera. To achieve this an object can be transformed by $local_{tm}$, which is calculated as follows. Figure 1(b) depicts the object in relation to the camera.

$$local_{tm} = C_{tm}^{-1} O_{tm} \quad (1)$$

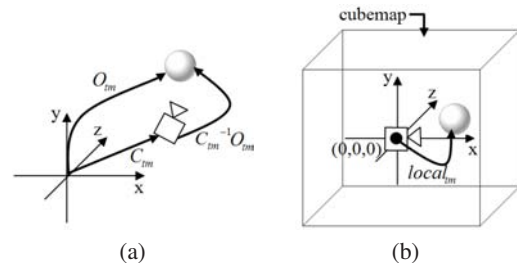


Figure 1: (a) depicts an object and a camera in world coordinates. The matrices that can be used to transform them are illustrated. (b) illustrates the same objects positioned relative to each other in a coordinate frame that is local to the camera. The cubemap, which will display the six images from the movies, is depicted in figure (b) positioned at point (0,0,0).

This is only part of the solution since the user is permitted to rotate their head as they navigate. The angles of rotation must be taken into account. The transformation matrix for the orientation of the head can be multiplied by $local_{tm}$

to obtain a new matrix, $ObjLocal_{tm}$, which can be used to transform all the objects in the scene. Once the objects have been transformed relative to the camera, methods can be employed to determine appropriate forces resulting from user interactions. The next section discusses how the haptic feedback is incorporated.

3.3. Incorporating haptic feedback

Exploiting the sense of touch in virtual reality simulations strives to provide a more immersive user experience and has been shown to improve interaction tasks [AKB01] [Che99] [CBG00]. In this paper haptic feedback is introduced into two areas; primarily for the navigation and exploration of the three dimensional scene and secondly to interface with the maps and the 2.5D user interface. This section focuses on the primary with the secondary being discussed in Section 3.6.

3.3.1. Touching the virtual objects

For many years users have been able to touch polygonal objects using a single point [ZS95] [RKK97]. This is akin to touching the objects with the tip of one finger. However, this is not particularly natural, since users often require touching an environment with an object of finite size, such as a tool. To achieve this a method similar to the Voxmap-PointShell approach may be exploited [MPT99]. Enabling a tool to be used in the environment significantly enhances the variety of applications. The three dimensional object manipulated by the user via the haptic feedback device will be referred to as the haptic tool. The process of determining the appropriate forces for a given position of the haptic tool is known as haptic rendering. A typical haptic rendering approach requires the following stages: collision detection, contact determination and restoring force calculation.

All the touchable virtual objects incorporated into the scene are constructed from triangles and are typically exported from 3ds Max, as discussed in Section 3.2. A test point approach has been developed for the haptic rendering. A simplified version of the haptic tool is created where the vertices are strategically placed. The location of the test points can be determined either by using a spatial partitioning approach or by obtaining vertices on the convex hull. To improve the efficiency of the collision detection each object in the scene is divided up into a three dimensional grid of cells in a precomputation phase. A list of triangles that pass through each cell are recorded with each cell. This information can be stored in an array, *Spatial_Array*. To perform the collision detection between a test point, tp , and a virtual object several queries are employed. The first query is devised to trivially reject a possible intersection by determining if the test point is inside the extents of the virtual object. Since the calculations for collision can be performed local to the virtual object its extents can be calculated once. The variables

```

0  Begin Procedure: Test Point Collision Detection
1      If test point,  $tp$ , is outside the extents of the Object THEN
2          return NO_COLLISION
3       $x := (tp_x - ObjectMinX)/ObjectCell\_Width$ 
4       $y := (tp_y - ObjectMinY)/ObjectCell\_Height$ 
5       $z := (tp_z - ObjectMinZ)/ObjectCell\_Depth$ 
6       $index := x*NoCells*NoCells + y*NoCells + z$ 
7      If Spatial_Array[index]  $\neq$  NULL THEN
8          Determine the closest distances from the
9           $tp$  to each triangle in the cell.
10         If closest distance  $< \epsilon$  THEN
11             return COLLISION & SCP
12         return NO_COLLISION
13 End Procedure: Test Point Collision Detection

```

Figure 2: Collision Detection procedure for a test point colliding with a polygonal object

$ObjectMinX$, $ObjectMinY$ and $ObjectMinZ$ store the coordinates of the minimum extent. Similar variables are required for the maximum coordinate. If the test point is inside the extents of the object the cell in the *Spatial_Array* that encloses the test point can be calculated. These stages can be computed efficiently, using simple formulae, and are independent of the number of triangles in the object. To obtain the point of collision between the test point and the object the closest points are determined between the test point and each triangle stored in the selected object cell. A surface contact point, SCP, is returned if the closest point to the test point is within a small threshold. The complete procedure is described formally in Figure 2. Once the collision has been detected and the surface contact point and features obtained, a constraint based haptic rendering algorithm [HBS97] is performed on a test point basis. The resultant force is obtained by adding all the forces acting on each test point in collision. To obtain high quality touchable objects a graphical representation of the virtual objects imported from 3ds files will not be illustrated to the user. Instead the virtual objects will be transformed so that they appear at the same location as they are in the movie. This is achieved by multiplying the geometry by the transformation matrix, $ObjLocal_{tm}$. The way in which this matrix is constructed was discussed in Section 3.2. This matrix will then be used to transform the test points of the haptic tool into the local coordinate frame of each virtual object in turn. By keeping the collision detection test in the virtual object's local coordinate frame the precomputation phase which created the grid cells can be utilised.

Since the virtual objects are not rendered graphically the user can touch the objects illustrated in the movie and thus touch high quality rendered objects. Additionally, this has the advantage of being able to use a different geometric representation for the touchable objects compared to those

that were utilised in 3ds Max at the point of rendering the movies. However, the geometric representation's silhouette must be preserved in comparison to the silhouette of the objects used in the movie. The next section will make the reasons for this clear. The separation between touchable geometry and graphical geometry provides flexibility which can be exploited for haptically rendering complex virtual objects. Furthermore, by not rendering the touchable objects the high quality pre-rendered movie is not compromised by the appearance of lower quality objects rendered in standard OpenGL. The next section discusses some additional factors that are taken into account to strive for seamless integration of the haptic tool within the pre-rendered environment.

3.3.2. Integrating the haptic tool

In this section techniques to integrate the OpenGL rendered graphical representation of the haptic tool with the pre-rendered movie will be presented. If the representation of the haptic tool looks out of place when visualised against the movie then the sense of immersion will be lost and a significant aim of the technique will be hampered. Two areas for seamless integration will be investigated namely, lighting and occlusion. These are motivated from Morris et al. [MJ06] where the haptic cursor was integrated into a single ray-traced point-based rendered image. To obtain a close match to the illumination, the lights used in the 3ds max scene can be exported with the cameras. Although the effects of these lights within a 3ds Max rendered scene are calculated differently to OpenGL lighting at least the positions can be obtained to ensure the specular highlights appear appropriate on the graphical rendering of the haptic tool. More importantly, is the ability to enable the haptic tool to be occluded by objects in the movie, since the haptic tool must be able to disappear behind the touchable objects within the scene.

During a prescribed route a selection of objects move in and out of the area that can be reached by the haptic tool. It is only necessary to consider objects for occlusion which are within range of the haptic tool. Section 3.4 describes how the list of touchable objects can be obtained. Although the geometric representation of the object will not be displayed it can be rendered into the depth buffer by disabling the OpenGL color buffer for writing when rendering the virtual objects. The haptic tool is subsequently rendered and the hardware accelerated z-buffer algorithm will deal with the occlusion problem simply and efficiently. This method relies on the silhouette of the geometric representation of the virtual objects matching sufficiently to the representation used in the generation of the movie. If the silhouettes do not sufficiently match then the tool will be occluded by the virtual objects incorrectly.

The haptic interaction can be utilised for other functions beyond simply touching the objects within the environment. Although touching them is particularly important when obtaining information regarding weight, surface contours, size

and three-dimensional information. Section 3.5 provides additional functions of the haptic device within the movie context.

3.4. Determining touchable objects

Given a particular path through the virtual environment only a subset of the objects visible in the scene will actually be candidates for haptic interaction. Consequently for each frame of the prescribed route a list is built that contains an index into an array of touchable objects. For each frame all the object's transformations are obtained and combined with the camera's transformation to determine the objects position relative to the camera. This information is utilised to permit a sphere check to determine if the object is touchable by the haptic tool at a given frame. This results in a list of touchable object indices for each frame of the movie. To reduce the memory requirement frame coherence is exploited allowing the list to be converted into a table of touchable object sets and a lookup table. This is achieved by reducing the original collection of touchable object lists into a unique set. A lookup table is constructed identical in size to the number of frames of the movie and each location of the table refers to one of the unique touchable object lists.

By determining the touchable objects as a preprocessing step the location of the touchable objects is known for each frame during haptic rendering in constant time.

3.5. Utilising haptic feedback for navigation

One challenge regarding the presented technique is to provide the user with a sense of immersion despite the fact that a section of pre-rendered movie is being played to them. Touching the virtual objects is a significant factor in providing this sense of immersion, however, the freedom to control their movement is also important for an immersive interactive application. The direction of travel within a pre-rendered movie can not be modified but the view direction and the speed of movement is permitted to be altered. In Section 3.6 a technique will be presented to enable the user to take control of the path they take. To avoid the use of additional input devices the view direction and playspeed will both be modified using the haptic feedback device. When the user presses and holds the white button on the device a virtual spring is initialised with one of its end points clamped at the current cursor position. As the user moves from this point the spring extends and once its length is greater than a certain threshold the scene will rotate. Two springs are utilised one for the x component and one for the y component, which perform scene rotations in the x and y axes respectively. A force is calculated based on the extension of the springs. The user is then able to link the magnitude of the force with the speed of rotation. The play speed of the movie can be modified in a similar fashion except that the blue button can be held. When either of the device's buttons are depressed the virtual

objects are no longer touchable. This is important to prevent confusion over forces generated from the interface functions and the interaction forces within the environment. The visual feedback utilised to exhibit the change of haptic function involves altering the haptic tool from a hand when touching virtual objects to 3D cursors for movie controls.

3.6. A haptic enabled user interface for a virtual cultural heritage tour

The proposed virtual cultural heritage tour enables a user to navigate a large area visualising artefacts within their correct geographical location. The nature of a guided tour system is exploited to enable high quality cube maps to be pre-rendered for each frame of the movie. This concept permits a user to look in any direction as they travel along their chosen route. Integrating the branching movie framework, [Che95], permits a user to navigate between the different prescribed paths through the environment. The branching movie framework describes the early computer graphics technique employed by the games industry, comprising a directed graph where edges represent movies and vertices represent connections between the movies. In the proposed technique many movies are generated and unlike the traditional branching movie framework connections are permitted at any point, owing to the nature of the cube map movie frames. Therefore users are able to select their choice of direction at key junction points in a seamless manner. For example, in a street scene key points would be located at junctions in the road network as well as at the entrance of buildings such as St Andrew's Hall.

Haptic feedback is incorporated to enable the routes through the scene to be selected, the velocity of the camera to be altered and the objects in the scene to be explored. By utilising the haptic feedback device for the entire user input the application is suitable for a standalone installation in a room with low ambient lighting. Exhibiting the work in this environment is particularly well suited to a kiosk in a museum and enhances the visual quality perceived on a standard monitor. Furthermore since the keyboard and mouse are no longer required, low lighting is acceptable. In addition the development of cheap consumer level 3DOF output haptic feedback devices ensures that a system consisting of a standard PC, haptic device and monitor is economically viable for both museums and education facilities.

During the running of the system the user is presented with further information concerning their current position within the cultural heritage site, as well as provided with context information with respect to the touchable objects. Both the temporal and spatial context information is in the form of a variety of media including short animations, images and text. The remaining two subsections provide details of the data structures required to maintain data concurrency and integrity, followed by the user interface design and

implementation specifics concerning the inclusion of haptic feedback.

3.6.1. Data structure

Figure 3 presents a class diagram illustrating the overview of the data structure developed. At the root of the data structure is the location class. A location contains a map of the area it covers plus a list of sites. Each site refers to a particular position on the location map and may have associated with it a name, movies and images that can be viewed from that position. A site may share this media with other sites at the same location. Therefore the location class contains a unique list of movies, objects and images and the list of sites indexes into it.

The MovieSpline class stores an index into the list of cube map movies. Each movie enables the user to seemingly travel from a particular location on the map to another location. A set of control points are stored and displayed on the map using a Catmull-Rom spline. This permits a marker to travel along the spline illustrating to the user their current location. The user can select these points to enter into the movie at their locations. A list of integers indexing into the content and 3d object lists is maintained. Associated with each index is a valid frame range to ensure that 3D objects and image based content are only updated within the valid frame interval. This provides context information to the user as they progress along the timeline. Furthermore, each object has an index associated with it to allow objects to display context information when inspected by the haptic tool.

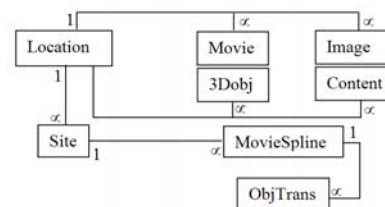


Figure 3: A class diagram for the data structures of the system.

3.6.2. Haptically enabled user interface

The user interface is divided into three main areas: the map area, content area and movie area. It is manipulated in two modes using the haptic feedback device. The most important mode is to manipulate the haptic device within a three dimensional volume for the movie exploration. However, the device is also used to select buttons and control the user interface in 2.5D. Initially the user manipulates a 2D cursor which is constrained to a plane giving the illusion of a flat area covering the entire screen. Areas of the interface are

divided with small walls which keeps the user in particular areas as they slide over the interface. Each map or content area may be panned horizontally and vertically by pressing the blue button down and dragging the device. A force is perceived as the cursor moves. The content in these areas may be scaled by holding the white button down and pushing the device into the screen. A key component of the interface is to enable the transition of the haptic device from the 2.5D interface mode to the 3D movie mode. This is achieved by pushing the device into the screen when the 2D cursor is displayed on top of the movie. The user must pull the device towards them to return to the 2.5D mode from the 3D movie mode.

Figure 4 shows the interface of the program with the hand cursor currently interacting with the objects displayed within the movie. The movie depicts a highly detailed model of the interior of St Andrew's Hall. Four objects were inserted into the model and rendered within the animation to illustrate how the techniques presented in this paper can be utilised. As the movie is playing at key points contextual information is displayed in a scrollable pane at the bottom of the interface. A sample of this information is displayed within the figure. The haptic enabled interface permits the software to be utilised using only a visual display and a haptic feedback device.



Figure 4: Image captured during the execution of the program.

4. Results

To facilitate the construction of test data and to ease the methods applicability, a MaxScript was written to enable all the required information to be extracted from 3ds Max automatically. The user defines any object they wish to be touchable using the "touch" keyword in the object's name and any camera from which to view the scene using the "movie" keyword in the camera's name. By executing the script six cameras for each of the indicated cameras are generated and

used to render the six movies for the cube map. All touchable objects are automatically translated to the origin and exported in .3ds format. Furthermore all the touchable object sets and camera and object transformation matrices are exported ready to be used by the C++ implementation.

The fundamental requirement of the proposed system is that it is capable of rendering both the graphics and the haptics at the desired rates of 60Hz and 1000Hz respectively. A high visual quality will be achieved using an offline renderer. To determine if this is possible seven test scenes were constructed using an arrangement of cubes surrounding a camera's path. By executing the script all the necessary models, transformations and movies were automatically obtained and sent to the OpenGL program. The script executed in under one minute for a scene containing four hundred objects. This result excludes the rendering time for the six movies. The simulation was performed and results for the graphics frame rate and haptic feedback rate were obtained, collecting one result every second. Table 1 presents the mean average results recorded. All the tests were conducted on a Dual 3.0GHz Xeon processor computer with 1GB of RAM and a Quadro FX 1440 graphics accelerator card. The haptic feedback is delivered via a Phantom Omni haptic feedback device produced by SensAble Technologies.

The first column of the table states the number of objects included in each test scene. The scenes were constructed by incorporating further objects radially from the camera path's location. Therefore once the scene has increased in size sufficiently the number of touchable objects becomes constant and this number is given in the second column of the table. It can be seen that as the number of touchable objects increases both the haptics and graphics rates remain within an acceptable threshold of the desired goal. Further testing was

# scene objects	# touchable objects	mean fps	mean haptic rate
20	20	604	997
50	50	502	998
100	100	459	998
200	187	295	998
400	279	257	997
800	393	220	998
1600	393	231	994

Table 1: Comparison of the graphics frame rate and haptic update rate against the number of objects included in the virtual environment.

conducted experimenting with the memory consumed by the movies. It is feasible to consider playing six 50mb movies on each of the faces of the cube at interactive frame rates. However, this is not a crucial factor since one of the systems benefits lies in the ability to construct smaller sections of movie files. This enables a user to change direction more frequently and give them more freedom to explore their surroundings.

5. Acknowledgements

The authors wish to thank the Urban Modelling Group at UEA (<http://www.urbanmodellinggroup.co.uk>), in particular David Drinkwater for providing photorealistic three dimensional models of St Andrew's Hall, Norwich UK, and its surrounding area. This work was funded by HEART (<http://www.heritagecity.org/>), acting as a single, co-ordinating organisation to strategically plan, regenerate, manage and promote all heritage resources in Norwich, and Spatial Metro Project (<http://www.spatialmetro.org/>), a transnational group of partners co-operating to find innovative ways to improve city centres for pedestrians.

5.1. Conclusions

A technique has been presented to enable users to interactively touch large high quality pre-rendered environments. By incorporating a haptic feedback device for navigation and exploration the user obtains a greater level of immersion despite the nature of the semi-prescribed path. Scenes can be constructed in 3ds Max with both dynamic cameras and scene objects and converted into a format suitable for an image based real-time rendering technique automatically by executing a single MaxScript file. This results in all the required information being easily produced for the virtual tour program. The techniques have been illustrated with a virtual tour of St. Andrew's Hall where the user is permitted to touch several objects located inside. Context information is provided to the user both as the movie proceeds and as they interact with objects in the scene.

The current method allows both the camera and the objects to undergo rigid body transformations. For future work techniques enabling the haptic tool to influence the movements of the virtual objects will be investigated.

References

- [AKB01] ADAMS R., KLOWDEN D., BLAKE H.: Virtual training for a manual assembly task. *Haptics-e 2(2)* (2001).
- [BBF*02] BERGAMASCO M., BROGNI A., FRISOLI A., SALVINI F., VIGNONI M.: Tactual exploration in cultural heritage. In *Proc. of the XIV Computer-aided Egyptology Round Table (IE2002)* (Pisa, Italy, July 2002).
- [Bre01] BREWSTER S.: The impact of haptic 'touching' technology on cultural applications. In *Proc. of EVA2001* (Glasgow, UK, 2001).
- [CBG00] CROSSAN A., BREWSTER S., GLENDYNE A.: A horse ovary palpation simulator for veterinary training. In *Proc. of PHANToM Users Research Symposium* (2000), pp. 79–86.
- [Che95] CHEN S.: Quicktime vr: an image-based approach to virtual environment navigation. In *Proc. of ACM Siggraph* (1995), pp. 29–38.
- [Che99] CHEN E.: A six degree of freedom haptic system for desktop virtual prototyping applications. In *Proc. of the First International Workshop on Virtual Reality and Prototyping* (1999), pp. 97–106.
- [GMS06] GAW D., MORRIS D., SALISBURY K.: Haptically annotated movies: Reaching out and touching the silver screen. In *Demo paper, 14th IEEE Haptics Symposium* (March 2006).
- [HBS97] HO C., BASDOGAN C., SRINIVASAN M.: Haptic rendering: Point - and ray - based interactions. In *Proc. of the Second PHANToM Users Group Workshop* (October 1997).
- [LD03] LAYCOCK R., DAY A.: Automatically generating large urban environments based on the footprint data of buildings. In *Proc. of ACM Symposium on Solid and Physical Modeling* (2003), pp. 346–351.
- [LWR*03] LOSCOS C., WIDENFELD H. R., ROUSSOU M., MEYER A., TECCHIA F., GALLO E., DRETTAKIS G., RECHE A., TSINGOS N., CHRYSANTHOU Y., ROBERT L., BERGAMASCO M., DETTORI A., SOUBRA S.: The create project: Mixed reality for design, education, and cultural heritage with a constructivist approach. In *ISMAR 03* (October 2003).
- [MJ06] MORRIS D., JOSHI N.: Hybrid rendering for interactive virtual scenes. In *Stanford University Technical Report CSTR 2006-06* (2006).
- [MPT99] MCNEELY W., PUTERBAUGH K., TROY J.: Six degree-of-freedom haptic rendering using voxel sampling. In *Proc. of ACM Siggraph 99* (1999), pp. 401–408.
- [OBP05] OWEN R., BUHALIS D., PLETINCKX D.: Visitors' evaluations of icts used in cultural heritage. In *Proc. VAST* (2005), pp. 129 – 136.
- [RKK97] RUSPINI D., KOLAROV K., KHATIB O.: The haptic display of complex graphical environments. In *Proc. of ACM Siggraph 97* (Los Angeles, CA, 1997), pp. 345–352.
- [RMJ*05] RIZZO A., MCLAUGHLIN M., JUNG Y., PENG W., YEH S., ZHU W.: Virtual therapeutic environments with haptics: An interdisciplinary approach for developing post-stroke rehabilitation. In *Proc. of CPSN05* (2005).
- [SMS*01] SNIBBE S., MACLEAN K., SHAW R., RODERICK J., VERPLANK W., SCHEEFF M.: Haptic techniques for media control. In *Proc. of the 14th Annual UIST 2001* (November 2001).
- [WSJH04] WILLIAMS R., SRIVASTAVA M., JR. R. C., HOWELL J.: Implementation and evaluation of a haptic playback system. *Haptics-e 3*, 3 (2004).
- [ZS95] ZILLES C., SALISBURY J.: A constraint based god-object method for haptic display. In *Proc. of the IEEE Conference on Intelligent Robots and Systems* (1995), pp. 146–151.