Exploring Cultural Heritage Sites through Space and Time

R. G. LAYCOCK, D. DRINKWATER, and A. M. DAY
University of East Anglia, UK

Traveling through a single virtual environment only tells part of the story; a particularly interesting aspect is to illustrate how an area has developed over time. This article presents a unified approach to illustrating four-dimensional data concerning a cultural heritage site. The proposed framework provides a semi-automatic approach to both reconstructing the environment and bringing all the time-dependent models into an intuitive visualization package. For each time period considered for reconstruction, the system requires a set of building footprint maps depicting the layout of the environment plus a few statistics. The statistics govern the construction of three-dimensional building models, allowing each building's architectural style, typical building height, and roof style to be altered. This information is automatically processed and converted into a form that can be visualized. By integrating high quality landmark buildings from laser scanning or interactive modelling packages into the automatically generated scene, the cultural heritage site is realized both in a spatial and temporal context. The visualization is achieved via a 4D navigable movie which is presented using two concrete implementations written using Flash and OpenGL. The OpenGL-based implementation allows a collection of 3DS Max scenes to be automatically visualized requiring only a set of camera paths identified by the user.

Categories and Subject Descriptors: I.3.6 [**Computer Graphics**]: Methodology and Techniques—*Interaction techniques*; I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*Geometric algorithms, languages, and systems*

General Terms: Algorithms

Additional Key Words and Phrases: Urban modeling, interfaces

ACM Reference Format:

Laycock, R. G., Drinkwater, D., and Day, A. M. 2008. Exploring cultural heritage sites through space and time. ACM J. Comput. Cult. Herit. 1, 2, Article 11 (October 2008), 15 pages. DOI = 10.1145/1434763.1434768 http://doi.acm.org/10.1145/1434763.1434768

1. INTRODUCTION

Visitors are attracted to many cultural heritage sites throughout the world, each site typically preserving one particular snapshot of a location. To promote a better understanding of the area and to fully appreciate the rich history of a location, it is important to be able to observe how a location has altered throughout the ages. By reconstructing three-dimensional models of an area from different time periods, a user is able to travel through and observe the environment from different points in space and time. These points in space may be unattainable since the physical location may no longer exist. If the physical location does exist, then the desired viewpoint may still be unavailable due to restricted access imposed for health and safety reasons or in the interest of site conservation. Therefore the virtual

Author's address: R. G. Laycock, School of Computing Sciences, University of East Anglia, Earlham Road, Norwich, UK, NR4 7TJ; email: rgl@cmp.uea.ac.uk.

environment promotes both novel views of the existing environment as well as providing access to sites residing in remote locations.

This article considers and provides novel solutions to two main issues. The first is concerned with the desire to rapidly construct large environments from different time periods and the second considers how the user can obtain information from different time periods during their traversal in an intuitive manner.

2. PREVIOUS WORK

A wealth of literature is available on constructing three-dimensional models of existing cultural heritage sites. It can be divided into two main categories: laser scanning [Stumpfel et al. 2003; Vosselman 1999; Weidner and Förstner 1995; Fruh and Zakhor 2001; Zhao and Shibasaki 2003] and photogrammetry [Kim et al. 2000; Dick et al. 2001; El-Hakim 2000; Liebowitz et al. 1999; Baillard and Zisserman 1999; Fitzgibbon and Zisserman 1998; Gool et al. 2002]. While these approaches have been used extensively to record, measure, and preserve cultural heritage sites, they are only capable of displaying the current state of the environment. (Although some work has been undertaken towards the reconstruction from historic photographs [El-Hakim et al. 2006]. This is suitable for reconstructing small environments or isolated buildings.) A limiting factor of the deployment of the image to model techniques for cultural heritage has been due to the lack of general purpose automated software, which has often been confined to research institutes. Through the EPOCH project, researchers at the University of Leuven have developed a Web-based interface capable of automatically processing a collection of images that a user uploads. The images are converted into a 3D model, and the result is delivered to the user [EPOCH-project 2007]. The resulting 3D model will only illustrate to the user what is visible in the images and has the potential of introducing holes in the model. Typically restoration of these automatically constructed environments into three-dimensional worlds involves the use of interactive modeling packages [Baracchini et al. 2004]. When considering modeling a large area for the depiction of carefully reconstructed landmarks or monuments within context, a reduction in the manual intervention is required.

Since 2001, procedural modeling techniques have been considered for the construction of virtual urban environments [Greuter et al. 2003; Parish and Mueller 2001; Suzuki and Chikatsu 2003]. Birch et al. [2001], developed an interactive modeling system capable of generating general built environments. Their modeling approach coupled a shell modeler with a window modeler to enable users to rapidly construct buildings of a variety of architectural styles, comprised of a number of storeys and facade features. The rapidly constructed buildings were subsequently fitted to historic maps by aligning them to a Bézier curve manually defined to follow the streets on a scanned historic map. The system enabled large environments to be constructed using a predefined set of interactively modeled buildings and was applied to produce a reconstruction of a medieval town. However, their approach employs a finite set of building models, the size of which is dictated by the interactive modeling efforts of the user. In the same year, Parish and Muller [2001] developed an entirely procedurally-based system for the automatic construction of large urban environments. The hypothetical location of cities was derived based upon high population density areas within a population density map. An L-system was generated to represent the road layout which connects the identified cities. The road layout provided the necessary two-dimensional information required in partitioning the remaining area into blocks. From these blocks further partitioning is performed to identify building plots which are subsequently populated with a building model constructed procedurally. Each building model was rudimentarily designed, leading to the extension work of Muller et al. [2006] to develop a structured grammar for constructing three-dimensional building models. The authors illustrated the structured grammar's utility in constructing artificial scenes based on architectural rules. One fundamental issue not considered is the automatic extraction of the true layout of the environment for recreation. In the paper of Birch et al. [2001] the buildings are assumed to follow the road layouts and, in Parish and Muller's 2001 paper, the layout of the environment is entirely synthetic.

For constructing the current environment, techniques from the remote sensing community can be consulted. For instance light detection and ranging (LIDAR) data and digitized building footprint and road layout information may be incorporated [Laycock and Day 2003; Vosselman and Dijkman 2001]. These data sources enable a three-dimensional scene to be constructed by extruding the building footprint information to a height value indicated by the LIDAR data. The fidelity of the model may be increased by the introduction of additional facade and roof geometry derived from multiple overlapping images [Baillard and Zisserman 1999; Fitzgibbon and Zisserman 1998] or laser scanning [Fruh and Zakhor 2001; Zhao and Shibasaki 2003]. At the same instant the geometry is derived, images are acquired and are used for texture mapping. However, for recreating historic environments, archive footage should be considered. Suzuki and Chikatsu [2003] presented a method to reconstruct Kawagoe in Japan based on archive maps. Their approach utilized the map to obtain the position of rectangular footprints by automatically extracting the four corners. This approach is limited in that it only recovered rectangular footprints and was only able to populate those footprints with one of six interactively modeled buildings.

The second issue considered in this article is the development of an intuitive user-interface capable of seamlessly traversing these high quality environments both through space and time. Frequently a user is presented with a computer animation to aid in explaining how an environment has altered throughout the ages. Recently Stumpfel et al. [2003] have used a combination of laser scanning and photometric stereo techniques to reconstruct the Parthenon. The resulting computer animation illustrates many of the sculptures which currently reside in the British museum in their correct context. For a user to obtain a more immersive experience, an increase in the level of interaction is required. Pletinckx et al. [2001], attempt to answer this issue by extending the QuickTimeVR interface. By considering the virtual reconstruction of St. Laurentius Church in Ename as a case study, the authors show how novel views may be obtained by horizontal displacement of the mouse cursor as in the traditional QuickTimeVR application. However, this operation is improved by enabling time travel using the vertical displacement of the mouse. Vergauwen et al. [2004] augment this concept with real images acquired from a helicopter, thereby providing the user with both the historic reconstructions and present day appearance of the location. This approach does permit an environment to be rotated and visualized from a single point at different time periods, but, it does not provide the user with the ability to traverse the environment. A system capable of permitting environment traversal based on images was developed between 1978–1980 at MIT. The solution involved a camera attached to a vehicle recording an image every 10 feet as the vehicle traveled along the road. The images were then played back to simulate the navigation of the real scene. This solution published in 1980 [Lippmann 1980] provides a suitable navigation method for prerendered movies but does not consider either panoramas during the scene traversal except at junctions or the notion of including multiple time periods.

Zuk et al. [2005] provide a small model to be visualized. The desired time period is displayed by selecting the appropriate time from the timeline. The approach is enhanced with the use of transparency to indicate the certainty of each part of the model. El-Hakim et al. [2006] also utilize three-dimensional models rather than images to facilitate navigation through both time and space using a timeline interface. However, this approach is not extendable to large high-quality scenes since many virtual models are required for each time period, and each geometric model potentially consists of millions of polygons and hundreds of texture maps. To achieve a high-fidelity result, these models must be rendered with a realistic lighting model in real time. Consequently, obtaining a smooth transition between models of different time periods becomes impractical on current consumer-level hardware.

In this article, a semi-automatic urban modeling approach capable of constructing large virtual urban environments from scanned two-dimensional maps is presented. To reconstruct a model for a time period without map data, it is possible to employ an archaeologist's hand-drawn sketch. The approach is able to reconstruct three-dimensional building models consisting of textured facades and roof geometry based on closed nonintersecting polygons. The four-dimensional QuickTimeVR concept which provided a view around a single viewpoint is built on enabling a user to traverse seamlessly through a large high-quality virtual environment visualizing the geographical location as it evolved over time. The approach is not constrained as previous techniques were by the amount of geometry, and it is not limited to providing panoramas at disjoint locations. This work has been applied to the reconstruction of Koblenz, Germany, and for visualizing St Andrew's Hall in Norwich, UK, from 1200AD to the present day.

3. URBAN MODEL CREATION

Visualizing isolated buildings or ruined monuments only conveys part of the story. During a virtual reality walkthrough or fly-by of a cultural heritage site, the user will have a more informative experience if the area surrounding the landmark building is provided; such an application provides the context for the landmark and better depicts its function. Motivated by this goal, the urban model creation is completed using three stages. The first stage considers modeling a large area from geographical datasets automatically. The second stage takes as input the automatically obtained model and enables a user to rapidly alter various features to improve the fidelity of the reconstruction. The third stage involves incorporating completely interactively generated models for the landmark buildings. Each of these stages will now be discussed in more detail.

3.1 Automatic Urban Modeling

The automatic urban modeling algorithms developed enable both the terrain and the buildings to be automatically generated. The terrain is generated from LIDAR data by converting a regular grid into a triangulated mesh. This forms a basis on which to insert the building models. Each building model consists of a footprint, a roof type, a building height, and a set of textured facades. The next four sections will discuss the automatic algorithms capable of reconstructing a historic city. The reconstruction of 19th-century Koblenz is used as a case study.

The urban modeling procedure begins with the identification of a set of building footprints. The footprints are nonintersecting closed polygons that are extracted from archived maps or sketches. Figure 1(A) illustrates a sample from a 19th-century map of Koblenz. It illustrates the existence of noise in the image which must be cleaned. To achieve this, the difference between multiple Gaussian filters applied across the image is taken, followed by a remapping of the grey levels, therefore enabling a binary threshold to segment the image into black boundary pixels and white background pixels. Figure 1(B) presents the results of applying this first preprocessing step to the grey-level image to obtain a binary image. To facilitate the identification of the building footprints, the user is required to indicate any building footprints, for extraction by placing a dot inside them. This is quickly achieved, and these dots are automatically identified by performing a connected component analysis on the image. Any connected region of black pixels that is sufficiently circular is chosen as a building indicator point, the connected pixels are converted to white, and their centroid is inserted into a list for further processing. The connected component analysis also enables the removal of regions of black pixels which are considered to be too small to be part of a building footprint. A threshold on the size of the connected black pixels is set to 50 pixels, which is suited to the map scale and enables the identification of small regions for removal. These small regions are likely to represent part of a text feature or map symbol.

(A) (B)

Fig. 1. (A) illustrates a photograph of part of an archived map representing the German city of Koblenz in the 19th Century. (B) presents the result after automatically preprocessing the image to obtain a binary image. (Reprinted with permission of Stadverwaltung Koblenz)

From each of the building indicator points, a seed fill algorithm, which will terminate on reaching a black pixel in the binary image, is initialized. Consequently, to ensure the seed fill algorithm does not escape through any small holes in the footprints, which may arise due to noise, a dilation is performed. A polygon for each flood fill region is defined by performing a boundary walk of the connected set of pixels. Figure 2(A) illustrates the regions in dark grey which have resulted from filling from the black building indicator points. The initial building footprint polygons are shown using light grey, partitioning the dark grey building footprint regions from the black building footprint pixels. The dilation stage of the algorithm has increased the thickness of the footprint pixels resulting in reducing the area bounded by the building footprints. Consequently, a refinement stage is undertaken employing an active contour model to expand each of the initial building footprints away from its centroid without allowing it to penetrate any other footprint. This is achieved by applying a distance transform to the binary image that indicates a high value for black pixels which are furthest from the building footprint regions depicted using light grey pixels. Therefore the active contour model is able to expand the initial building footprint polygon while the movement causes the vertices to move to a pixel value with a higher distance value, resulting in the polygon finishing on the center of the edges in the binary image as indicated in Figure 2(B).

3.2 Constructing Roof Models

The approach of the previous section allows for building footprints derived semi-automatically from archived maps and hand-drawn sketches. The footprints provide the layout for the urban environment, and the next stage involves converting the two-dimensional information into a coarse three-dimensional urban model. To reconstruct a modern day scene, LIDAR data may be employed to extrude the footprints

Fig. 2. (A) From each black building indicator point, a seed fill algorithm is performed to construct the dark grey building areas. The initial building footprint polygons are shown in light grey surrounding each building area. (B) By employing an active contour model, the footprints are adjusted from their original position, shown in light grey, to fit to the centers of the building footprint edges in the map. Their final position is illustrated using thin black lines.

into three-dimensional volumes. However, to reconstruct a historic scene, this LIDAR data does not exist. To this end, a simple interface has been designed that enables a user to define a height for an individual building or a region of the building. This approach permits building heights to either be set explicitly or to be randomly selected from a valid range of building heights. In addition, the valid range of building heights for the time period can be selected and more weight given to a particular height should it dominate. Regardless of the data sources employed, the footprints which result can be arbitrary in nature, therefore there is a need to develop new roof-modeling algorithms capable of constructing roof models of a variety of architectural styles for any given nonintersecting closed polygon.

The automatic urban modeling system implements the straight skeleton [Felkel and Obdrmálek 1998] to enable the construction of the hip roof style. These roof styles have been extended previously to include such styles as Gable, Gambrel, Mansard, Dutch-Hip, and Well, [Laycock and Day 2003]. To further increase the number of roof styles that can be created, an interface is set up enabling different combinations of the aforementioned roof styles. By altering a variety of parameters such as the inclination of the roof planes and the height of each roof style, many roof-modeling procedures can be generated. For instance, the particular Gambrel roof style employed for the buildings in Koblenz was achieved by combining two Mansard and one Gable roof. Each roof style can be quickly applied to individual buildings or groups of buildings. Given a selection of buildings, the percentage of each type of roof style used can be defined. This approach permits three-dimensional building models to be generated semi-automatically for all of the footprints extracted from a map. Typically this area will be used as the context in which to position further higher resolution geometric models.

Consequently, it is worth considering employing level-of-detail techniques to reduce the geometry on this part of the model to enable more of the triangle budget to be exploited by the important parts of the scene. However, this is not controlled in real time since an offline renderer is employed for all the rendering. Therefore, prior to the rendering, the scene is constructed such that more geometry is allowed in areas near the camera and less geometry is used for buildings further from the camera. The geometry simplification is achieved by processing the building footprints removing any vertices which are close to being collinear. A user-specified threshold is used for this measure, and a value of 20cm was used in the current reconstruction. For those buildings in the periphery, the buildings are approximated with rectilinear footprints. This can be done as only their facades and roofs will be partially visible, thus it is not necessary to represent the cross section of the footprint explicitly.

For these buildings, it is frequently the case that they started their life as a simple rectangular footprint, which over time became modified through the addition of extensions. Consequently, by solving the largest empty rectangle problem [Orlowski 1990] for each footprint, a dominant rectangle can be extracted. The procedure involves identifying the largest rectangle, which is interior to the given footprint. The largest empty rectangle problem is computed efficiently using an axis-aligned rectangle, therefore an orientated bounding box is computed for each building footprint and the footprint is rotated by the amount required to convert the orientated bounding box into an axis-aligned rectangle. Once the rectangle is identified, the footprint can be rotated back.

3.3 Applying Surface Texture Details

To improve the fidelity of the scene, images are applied to the faces of the geometry constructed via the automatic urban modeling procedure. A library of texture maps is defined and categorized by architectural style, aspect ratio, and the height of the texture map's content in meters. The last two parameters are present to ensure that a texture map is applied to a facade without causing significant distortion. The distortion is controlled by ensuring that no texture is applied to a facade that does not have the same aspect ratio within a 10 percent threshold, or the same height value within a 50cm tolerance. These constraints may be tightened up but doing so will incur a trade-off with the amount of geometry to render. Within each architectural style, the textures are grouped such that all the textures within a given group would appear correct if placed onto a single building. This enables consistent materials to be used on a building. Furthermore one texture per group is stored containing the wall material.

In a similar interface to that used for the roof-modeling stage, a user is provided with the functionality to apply textures of a particular architectural style to multiple buildings. For each building, the algorithm commences on the first group of texture maps for the given architectural style and applies the first image it finds which fits with a user-defined tolerance on the amount of image distortion to the facade. The remaining faces of the building are textured with images from this texture group. If no applicable texture map is identified in this group, the next group of textures is tried. If during the texture-mapping stage, no image can be found that can be applied to a particular facade, then that facade is subdivided and the process repeats. The subdivision terminates if the width of the facade becomes too small, and, in this case, a texture-containing the corresponding wall material is applied. To maximise the ability of the texture-mapping algorithm to identify texture maps which will exhibit minimal distortion artifacts, several textures are generated with different aspect ratios and intended heights. This will minimize the need to excessively subdivide many of the building facades, which could potentially lead to significant increases in geometry.

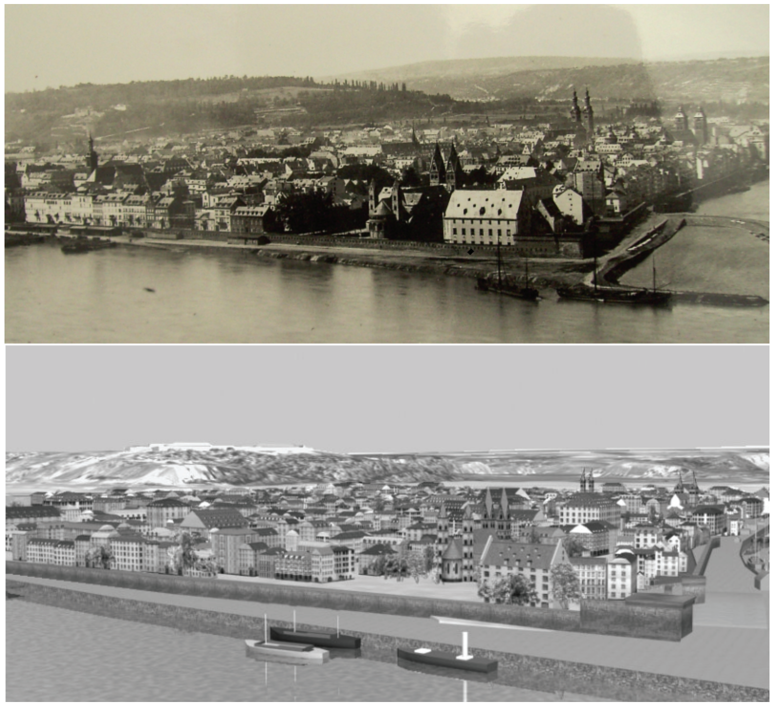

Fig. 3. The top of the figure illustrates an archive photograph of 19th-century Koblenz. (Reprinted with permission of Stadverwaltung Koblenz). The bottom image presents the result of the automatic urban modeling algorithm augmented with interactively modeled landmark buildings and trees.

Figure 3 illustrates the semi-automatically modeled and textured urban environment of 19th-century Koblenz. In this case study, textures were constructed from existing photographs.

3.4 Procedurally Generating Fine Geometry

A dominant factor affecting the visual quality of a large urban environment is the skyline of the city. This is one of the main motivations for generating appropriate roof-modeling algorithms. From observing archived photographs of the city of Koblenz, a key component to the skyline is clearly visible. A vast majority of the buildings contained multiple levels of dormer windows, which become smaller in size as they approach the roof apex. Consequently, procedural geometry techniques were developed to facilitate the integration of fine geometric details including dormer windows, eaves, gable ends, and chimneys.

Dormer windows are automatically constructed with their cross section extracted from a .tga file. The tga's alpha channel is used as a mask to distinguish between pixels relevant to the dormer window and the rest of the image. By applying the Canny edge detection algorithm to the alpha channel, line segments may be fitted to the resulting edge pixels. These line segments are combined with the boundary of the image to generate the cross section of the dormer window, which is subsequently texture mapped and extruded until it reaches the roof geometry automatically. A rule-baed system is in place to ensure that dormer windows are able to populate the roof models automatically, scaling in size as they approach the roof apex. The procedure for dormer window construction is illustrated in Figure 4.

Fig. 4. The top-left image presents the original .tga file using magenta to illustrate the transparent pixels in the texture. The image is processed to extract the boundary of the opaque region, shown using solid black lines in the bottom-left of the figure. The center image presents the result of converting this boundary into a polygon, extruding it, and applying the top-left image as a texture automatically. Multiple instances are procedurally inserted onto the roof of the buildings as presented in the right of the figure.

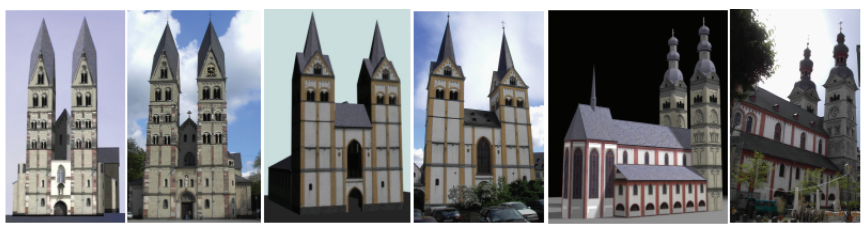

Fig. 5. Three pairs of images are illustrated with each pair representing a rendered and an original photograph of a 3DS Max-modeled landmark building.

3.5 Incorporating High-Quality 3D Models

For the landmark buildings in the scene, 3DS Max is employed to construct high fidelity reconstructions. By incorporating interactive modeling techniques, both high-quality exterior and interiors can be constructed representing hypotheses concerning their appearance. Figure 3 illustrates three models of significant buildings in Koblenz visualized within the automatically generated context. These models are illustrated in Figure 5 and were manually inserted into the virtual environment by aligning them with the terrain and the footprints extracted from the scanned images.

4. OVERVIEW OF THE 4D NAVIGABLE MOVIE

Handling the large quantities of geometry that are potentially obtained from laser scanning landmark buildings is a nontrivial task. This is especially true if it is important to maintain the fidelity of the reconstruction of the cultural artifact. If the high-quality mesh cannot be modified, then standard mesh decimation techniques for creating multiple levels of detail should not be applied since any modification will only introduce further errors between the model and the actual artifact. The number of polygons required for representing the artifact is only increased when considering the visualization of the artifact within the correct context. Consequently, rendering such a virtual environment with realistic lighting in real time is a challenging endeavor. Further difficulties arise given that multiple versions of the virtual environment exist at different time periods. Therefore an image-based solution is proposed to handle all the geometry and textures based on the 4D Quicktime VR system [Vergauwen et al. 2004]. The 4D QuicktimeVR solution extended Quicktime's ObjectVR concept [Chen 1995] of taking a series of

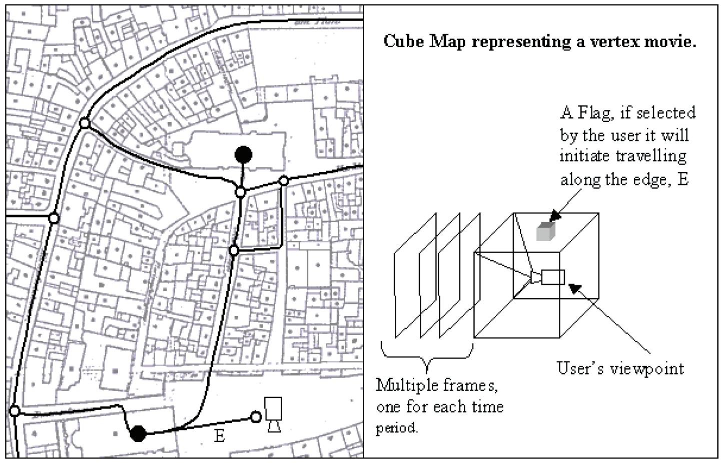

Fig. 6. The left image illustrates the map of Koblenz augmented with a graph representing the 4D navigable movie. Each solid black circle indicates a cultural heritage site, and the black outlined circles represent junctions in the network. The illustration shows a user currently located at the bottom-right vertex of the graph. The right image presents the cube map being used to approximate the plenoptic function at this point. Each face of the cube contains many images representing the different time periods that have been reconstructed.

views from around an object which could be searched by moving the mouse horizontally to the left or the right. This system has been taken up by many Web sites providing a user-intuitive method of viewing around an object. Vergauwen et al. [2004] extended this technique into four dimensions, including time, by allowing a user to also move the mouse vertically and blend time periods. This approach functions for a single viewpoint and this article extends this concept to a four-dimensional navigable movie, which permits the user to travel through space as well as time.

The remainder of this section will introduce the abstract concept and conclude by providing two concrete examples implementing the technique in Macromedia Flash and OpenGL. Each implementation offers benefits over the other in particular situations depending on the application, and these will be discussed.

4.1 4D Navigable Movie Creation

The 4D navigable movie consists of a directed graph with each vertex of the graph representing either an important location to visit on the cultural heritage site or a junction. For each vertex of the graph, a camera will be positioned at that location in the model and a set of images will be obtained to approximate the plenoptic function [Adelson and Bergen 1991]. The plenoptic function is a five-dimensional function representing the light intensity received at every point in three-dimensional space from every direction in the scene. These directions may be parameterised using spherical coordinates. Given this information, a user will be able to view in all directions from that point in the model. A similar procedure will be conducted for all the edges of the graph in both directions, consequently permitting a user to obtain a view from every vertex and every point along all edges of the graph. Multiple graphs with the same position of vertices and edges are used to acquire a set of images from each of the different time-dependent models. Figure 6 illustrates the directed graph and takes a closer look at one of the vertices where the plenoptic function has been approximated using a cube map.

Visualization commences by entering the system at a vertex. The user is subsequently permitted to translate the mouse horizontally and effectively turn his head to look in different directions. By moving the mouse vertically a blend is performed to change the user's view across different time periods. As the user rotates horizontally, a signpost appears indicating the location of the start of an edge; let this edge be denoted by E. The signpost is represented with an icon, which is selected using the left mouse button. On selection of this signpost, the user will rotate until their view matches the view of the first image in the sequence of images representing the path along edge E. Once the user has been orientated correctly, the remaining images for the edge will be displayed. At any point, the user can pause the playback or allow it to play at a fixed rate. Furthermore, as the users feel as though they are travelling along the edge in the model, they will be permitted to move their mouse horizontally and vertically to orientate their look direction and alter the time period, respectively. This extension provides the user with the ability to seamlessly traverse a cultural heritage site visualized at different time periods, bringing together all the models into a unified framework.

4.2 Implementing the 4D Navigable Movie in Flash

In this section, the abstract concept of the 4D navigable movie is implemented using Macromedia Flash. Macromedia Flash is useful for this purpose since it supports the playback of animations and has the ability to easily handle the clicking of the signposts, which indicate the presence of a pathway. Furthermore, Macromedia Flash is a powerful tool enabling this work to be extended to include both 2D and 3D graphics and animation. The 4D navigable movie is created as follows.

For each vertex, a camera is positioned in the model and is subsequently rotated through 360 degrees about the y axis to achieve a vertex movie comprising 360 frames. The resulting movie is imported into Flash and embedded into the timeline. By setting the key frame interval equal to 1, actionScript can be written to control the animation enabling it to play forwards and backwards as the user translates the mouse horizontally across the screen. The procedure is repeated for each of the time-dependent models at the given vertex location. By layering multiple movies onto the timeline, the user is able to translate the mouse vertically and control an alpha blend. The blend permits each movie to be accessed in chronological order.

For each edge, a camera is initialized at its start and travels along the edge capturing frames of the movies. The resulting edge movies are inserted into the timeline and are connected into the 4D navigable framework using a signpost. In flash, the signpost consists of a button which is inserted into the vertex movie, and when it is clicked, it will redirect, the flash to play the edge movie. However, before the redirection occurs, the 360-degree movie is played to rotate the user from its current frame to the angle equal to the starting frame of the edge movie by the shortest path. This provides a seamless transition from the vertex movie to the start of the edge movie. On reaching the end of the edge movie, the end vertex movie is controlled such that it starts from the frame equal to the angle of the end of the current edge movie. The procedure is repeated for each time-dependent model in a similar way to that incorporated for the vertex movie, enabling an alpha blend.

By conducting this process for all edges and vertices and for each time-dependent model, the user is able to traverse the environment from vertex-to-vertex while obtaining temporal reconstructions at any point along the route. It has the advantage that the system can be easily incorporated into a Web interface or surrounded by a custom-built interface in Flash. Figure 7 illustrates the view obtained from a given point during the running of the 4D navigation movie. As the scene continues to grow in size, it becomes impractical to combine all the movies into a single flash file, primarily due to the time required to start the application. However, this can be resolved by partitioning the flash file into multiple files each representing a particular set of vertex or edge movies. Each set contains those movies acquired from the same geographical location but from a different time period. A single file can be used to dynamically load the files at runtime enabling a pseudo-infinite 4D navigable movie that, if augmented with additional text to explain when a file is loaded, can be run directly from a DVD. The

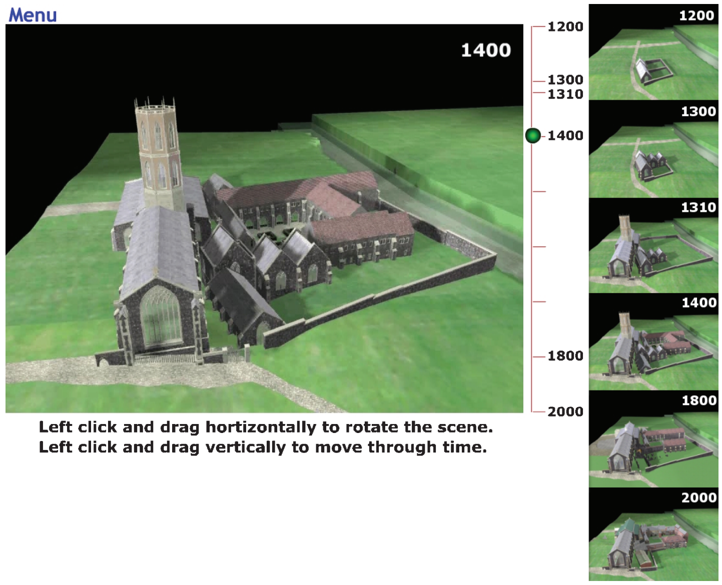

Fig. 7. The Flash interface is augmented with a timeline which may be clicked to directly move to a particular time period. The right-hand side illustrates the viewpoint of the model at six different time periods.

benefit of using Flash is in the speed with which further media can be incorporated into the system. This could be annotations describing various aspects of the environment or text referring to particular cultural artifacts. Extending the system to handle further event driven aspects is readily achieved. The disadvantage of the approach is that the user is not able to rotate their look direction as they traverse an edge of the graph. Furthermore, the approach only approximates the plenoptic function for 360 azimuthal angles, given one zenith angle. The next section deals with this issue.

4.3 Implementing the 4D Navigable Movie in OpenGL

In order to represent all the viewing directions considered by the plenoptic function at a given viewpoint, six images are acquired for each vertex in the 4D navigable movie. These six images, each acquired using a camera with a field of view of 90 degrees and orthogonal to each other, are applied to the six faces of a cube. By fixing the user's viewpoint in the center of this cube, the user is able to perceive a full panorama as documented by Chen [1995]. To facilitate the traversal from one cube map to another, the approach is repeated to capture multiple cube maps for each frame along an edge movie. The number of cube maps collected along the edge is calculated based on the distance covered by the edge, the desired constant frame rate of the movie, 25 *fps*, and the desired speed of traversal along an edge that the movie should simulate, $2ms^{-1}$. Therefore a user is permitted to fully orientate their look direction both while located at vertices and along edges. It is feasible to decode multiple movies as the user travels along an edge or rotates at a vertex, and therefore a user can at any point translate their mouse vertically to blend between the images on the faces of the cube using multitexturing.

The 4D navigable movie implementation in OpenGL has been written to ensure that it can be set up directly from the time-dependent models automatically. This is achieved by first labeling all the camera paths in each 3DS Max file with the keyword "movie". The camera paths represent the edges that the cameras will traverse during offline rendering. In 3DS Max, these edges are represented using splines to ensure a smooth scene traversal. A script implemented in MaxScript processes this information to generate a graph and, for each edge, six cameras are constructed and used to obtain the corresponding six movies. Furthermore, the ends of the splines are used to generate a set of vertex cube maps. The graph along with the vertex and edge cube maps are exported and processed automatically by the OpenGL implementation to initialize all the cube maps and connect them into the correct configuration. The signpost, represented as 3D boxes labeled with their corresponding destination, are automatically positioned into the vertex cube maps. The selection buffer is employed to receive the hit information, which informs the program which 3d box has been selected, and hence the corresponding destination the user wishes to explore next. The application developer is required to enter the 3DS Max file names containing each of the time periods in chronological order to ensure the alpha blending is performed in the correct order. Using the Maxscript file, an application developer can enter the file names for a set of 3DS Max scenes, which are subsequently converted automatically into a 4D navigable movie system.

5. RESULTS

To test the urban modeling approach, the reconstruction of 19th-century Koblenz was undertaken. An 1850's map was photographed and registered into a single mosaic. After this alignment, the mosaic was processed to reduce the noise artifacts, and building footprint indicator points were inserted. It took approximately half a day to perform this preprocessing which included the time required to label 1000 building footprints. The automatic algorithm subsequently proceeds to process the mosaic in order to extract all the building footprints, convert them into three-dimensional models and apply texture maps. This stage of the algorithm is completed automatically in less than 40 minutes on the 5000 by 5000 pixel resolution image representing the 1000 buildings of Koblenz. The PC used for this result was a 2GHz Xeon processor with 512MB of RAM.

Once the automatic urban model is created, a user can rapidly alter building heights, roof styles, texturing, and procedural geometry by selecting individual or regions of buildings. For this model, only the buildings on the river frontage were altered in less than 30 minutes. While the automatic modeler is processing, 3DS Max was employed for modeling the important landmark buildings including the various churches, bridges, and cathedrals. Each church required four hours to model, with two hours required to convert the original photographs captured on location into rectified texture maps.

To test the 4D navigable movie technique, an area surrounding St Andrew's Hall in Norwich was developed with models ranging from 1200AD to the present day. Each model was verified working closely with historians. A graph of vertices and edges was constructed using a spline in 3DS Max, and MaxScript was employed to automatically construct the renders for both the Flash and the OpenGL implementations. This alleviated the need to define the six camera configuration or to manually manage the rendering process. Given all the rendered movies, it currently takes approximately two days to compile the Flash interface, although this could be automated using actionScript. However, the MaxScript used to produce the movies for the OpenGL implementation also defines the necessary information required to enable branching between cube maps in OpenGL. Consequently, the OpenGL setup can be achieved automatically, scripted via the result produced by the MaxScript.

6. CONCLUSIONS

In this article, two approaches have been combined into a unified framework to permit users to traverse a virtual environment both through space and time. The first approach discussed the use of an urban modeling system comprised of three levels of detail. It was shown how a scanned archive map or hand-drawn sketch could, with limited preprocessing, be converted into a three-dimensional urban model. The fidelity of the resulting model was improved in consultation with historians by altering roof styles, building heights, facade textures, and fine geometric details on a building or collection of buildings. This is achieved using a simple semi-automatic interface, and it allows the alterations to be rapidly conducted. Buildings fundamental to the reconstruction, that require a high accuracy and reside in the foreground of the animation, are modeled interactively or obtained by remote sensing techniques. These are inserted into the correct geographical context and modified to illustrate them in previous centuries using the knowledge of historians coupled with archive data.

All the automatically generated information is combined into an intuitive interface called the 4D navigable movie. The abstract concept presented allows a user to navigate an environment following predefined routes. At any point in their traversal, the user is able to translate the mouse vertically to move in chronological order through the various time periods reconstructed. The abstract concept is realized in two concrete implementations using Flash and OpenGL for displaying three-dimensional models of St Andrew's Hall, Norwich, UK from 1200AD to the present day. The construction of the OpenGL interface is entirely scripted enabling the automatic construction of a 4D navigable movie from a set of given 3DS Max files.

7. FUTURE WORK

The image-based solution reduces the rendering speed requirement permitting large high-quality environments visualized under realistic lighting models to be traversed. Traversing an environment on fixed paths is ideal for many applications including a virtual guided tour of a cultural heritage site. For future work, we intend to augment the existing 4D navigation movie with a virtual tour guide to provide both spatially and temporally relevant information to the user. Increasing the senses utilized by the user further enhances the user's immersive experience. The audio and visual feedback provided by a virtual tour guide and 4D navigable movie will increase the senses exploited by the system and will consequently lead to an enhanced user experience.

An alternative function of the 4D navigable movie is to illustrate many potential reconstruction scenarios for a given site, as opposed to using the fourth dimension for time. In this way, a user would be able to iterate over and navigate many potential scenarios, perhaps incorporating an expert's opinion.

ACKNOWLEDGMENTS

The authors wish to thank HEART (http://www.heritagecity.org/) for acting as a single, coordinating organization to strategically plan, regenerate, manage, and promote all heritage resources in the Norwich. We also thank Spatial Metro Project (http://www.spatialmetro.org/), a transnational group of partners cooperating to find innovative ways to improve pedestrian usage in city centers, for supporting this work.

REFERENCES

Adelson, E. H. and Bergen, J. R. 1991. The plenoptic function and the elements of early vision. In *Computational Models of Visual Processing*, MIT Press, 3–20.

Baillard, C. and Zisserman, A. 1999. Automatic reconstruction of piecewise planar models from multiple views. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*. 559–565.

Baracchini, C., Brogi, A., Callieri, M., Capitani, I., Cignoni, P., Fasano, A., Montani, C., Nenci, C., Novello, R., Pingi, P., Ponchio, F., and Scopigno, R. 2004. Digital reconstruction of the arrigo vii funerary complex. In *Proceedings of the Conference on Virtual Reality, Archeology, and Cultural Heritage (VAST)*. 145–154.

Birch, P., Browne, S., Jennings, V., Day, A., and Arnold, D. 2001. Rapid procedural-modeling of architectural structures. In *Proceedings of the Conference on Virtual Reality, Archeology, and Cultural Heritage (VAST)*. 187–196.

Chen, S. 1995. Quicktime vr: an image-based approach to virtual environment navigation. In *Proceedings of ACM SIGGRAPH*. 29–38.

Dick, A., Torr, P., Ruffle, S., and Cipolla, R. 2001. Combining single view recognition and multiple view stereo for architectural scenes. In *8th IEEE International Conference on Computer Vision (ICCV'01)*. Vancouver, 268–274.

El-Hakim, S. 2000. A practical approach to creating precise and detailed 3d models from single and multiple views. In *International Archives of Photogrammetry and Remote Sensing, XXXIII 33*, 202–209.

EL-HAKIM, S., MACDONALD, G., LAPOINTE, J., GONZO, L., AND JEMTRUD, M. 2006. On the digital reconstruction and interactive presentation of heritage sites through time. *In Proceedings of the Conference on Virtual Reality, Archaeology, and Cultural Heritage (VAST).* 243–250.

EPOCH-PROJECT. 2007. www.epoch.eu.

FELKEL, P. AND OBDRMÁLEK, S. 1998. Straight skeleton implementation. In *Proceedings of the 14th Spring Conference on Computer Graphics*. 210–218.

FITZGIBBON, A. AND ZISSERMAN, A. 1998. Automatic 3d model acquisition and generation of new images from video sequences. In *Proceedings of the European Signal Processing Conference*. 1261–1269.

FRUH, C. AND ZAKHOR, A. 2001. Fast 3D model generation in urban environments. In *Proceedings of the International Conference on Multisensor Fusion and Integration for Intelligent Systems*. Baden-Baden, Germany, 165–170.

GOOL, L. V., TUYTELAARS, T., FERRARI, V., STRECHA, C., WYNGAERD, J., AND VERGAUWEN, M. 2002. 3D modeling and registration under wide baseline conditions. In *Proceedings of Photogrammetric Computer Vision*. 3–14.

GREUTER, S., PARKER, J., STEWART, N., AND LEACH, G. 2003. Real-time procedural generation of 'pseudo infinite' cities. In *Proceedings of the 1st International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia (GRAPHITE'03)*. ACM Press, New York, NY, 87–95.

KIM, Z., HUERTAS, A., AND NEVATIA, R. 2000. Automatic description of complex buildings with multiple images. In *Proceedings of the 5th IEEE Workshop on Applications of Computer Vision*. 155–162.

LAYCOCK, R. AND DAY, A. 2003. Automatically generating large urban environments based on the footprint data of buildings. In *Proceedings of the ACM Symposium on Solid and Physical Modeling*. 346–351.

LIEBOWITZ, D., CRIMINISI, A., AND ZISSERMAN, A. 1999. Creating architectural models from images. *Comput. Graph. Forum 18*, 3, 39–50.

LIPPMANN, A. 1980. Movie-maps: An application of the optical videodisc to computer graphics. In *Proceedings of SIGGRAPH Computer Graphics*. 32–42.

MUELLER, P., WONKA, P., HAEGLER, S., ULMER, A., AND GOOL, L. V. 2006. Procedural modeling of buildings. *ACM Trans. Graph. 25,* 3, 614–623.

ORLOWSKI, M. 1990. A new algorithm for the largest empty rectangle. *Algorithmica 5*, 65–73.

PARISH, Y. AND MUELLER, P. 2001. Procedural modeling of cities. In *Proceedings of ACM SIGGRAPH*. 301–308.

PLETINCKX, D., SILBERMAN, N., AND CALLEBAUT, D. 2001. Presenting a monumnet in restoration: The saint laurentius church in ename and its role in the francia media heritage initiative. In *Proceedings of the Conference on Virtual Reality, Archaeology, and Cultural Heritage (VAST)*. 197–204.

STUMPFEL, J., TCHOU, C., YUN, N., MARTINEZ, P., HAWKINS, T., JONES, A., EMERSON, B., AND DEBEVEC, P. 2003. Digital reunification of the parthenon and its sculptures. In *Proceedings of the Conference on Virtual Reality, Archaeology, and Cultural Heritage (VAST)*. 41–50.

SUZUKI, S. AND CHIKATSU, H. 2003. Recreating the past city model of historical town kawagoe from antique map. *International Archives of Photogrammetry and Remote Sensing XXXIV-5/W10*.

VERGAUWEN, M., PLETINCKX, D., WILLEMS, G., VERBIEST, F., GOOL, L. V., AND HELSEN, T. 2004. As time flies by: Mixed image and model-based rendering of an historical landscape from helicopter images. In *Proceedings of the Conference on Virtual Reality, Archaeology, and Cultural Heritage (VAST)*. 241–250.

VOSSELMAN, G. 1999. Building reconstruction using planar faces in very high density height data. In *ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery*. 87–92.

VOSSELMAN, G. AND DIJKMAN, S. 2001. 3D building model reconstruction from point clouds and ground plans. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Services 34*, 3, 37–44.

WEIDNER, U. AND FÖRSTNER, W. 1995. Towards automatic building extraction from high resolution digital elevation models. *ISPRS J. 50*, 4, 38–49.

ZHAO, H. AND SHIBASAKI, R. 2003. Reconstructing a textured CAD model of an urban environment using vehicle-borne laser range range scanners and line cameras. *Machine Vision Appl. 14*, 35–41.

ZUK, T., CARPENDALE, S., AND GLANZMAN, W. D. 2005. visualizing temporal uncertainty in 3D virtual reconstructions. In *Proceedings of the Conference on Virtual Reality, Archaeology, and Cultural Heritage (VAST)*. 99–106.

Received January 2008; revised June 2008, July 2008; accepted August 2008