

An Approach to Maintaining Viewer Perspective in Interactive Virtual Tours

Thomas Carpenter, Gregory Doerfler, Thomas Way and Frank Klassner

Center of Excellence in Enterprise Technology

Department of Computing Sciences

Villanova University, Villanova PA 19085

thomas.way@villanova.edu

Abstract - *While virtual tours have been around for some time, the ability to get a true sense of the location has not been adequately developed. As part of a partnership between our department and the Vatican, we have created a virtual tour of the Sistine Chapel using multiple locations and interactive menus to allow users to feel as though they are actually there. Not only does the virtual tour give users a true sense of the place, but also it offers information about the chapel as well as the biblical history of paintings on the wall. The innovative aspect of this tour is the degree of attention given to maintaining viewer perspective at all times, so that movement appears both smooth and intuitive. In this paper, we discuss the overall development process of this virtual tour, and focus on the perspective maintaining techniques that were devised and evaluated.*

Keywords: Virtual Tour, Virtual Reality, Perspective, Stitcher, Krpano, XML.

1 Introduction

As part of an ongoing partnership between our home institution department and the Vatican, we created a virtual tour of the Sistine Chapel. This undertaking was a follow-up to a Vatican internship and study abroad experience in Rome during the fall semester of the junior year of a two student team (Carpenter and Doerfler). As a part of the internship, the team assisted Villanova research faculty with photographic shoots at various locations in and around the Vatican. The most notable location was the well-known and highly recognizable Sistine Chapel. At these photographic shoots, a special camera and rotating stand were used to take hundreds of images in the complete 360° panoramic range of directions from a central, fixed location, or in the case of the Sistine Chapel which has a

central half-wall that does not make a single location practical, two fixed locations.

Once the photographic images data was gathered, the images were digital “stitched” together into a seamless, 360° panoramic view. To perform this image concatenation, a software program called Stitcher Unlimited was used. Stitcher Unlimited (stitcher.realviz.com) was used to stitch together the images from the Sistine Chapel into two panoramas, each centered on a fixed point within one of the two sides of the Chapel. These panoramas were then edited to remove minor imperfections in the stitching and developed further using the interactive 3D panorama viewing system and scripting language, Krpano (krpano.com). The result of this extensive development effort was the production of a 3D interactive tour that incorporates context-sensitive menus, visual and contextual “hotspots” and an intuitive navigation system. This combination of features and functionality, together with the innovative perspective maintaining panning algorithm we devised and the uniqueness of the Sistine Chapel as a setting, potentially marks this project as a significant contribution to the field of virtual tours.

2 Virtual Tours

Virtual tours work by offering a location (an axis point) from which the user can “look around” – changing the perspective, rotating the camera, and adjusting the zoom. Clickable “hotspots” link two or more locations together and allow the user to move to the next location.

What makes a virtual tour a tour of some location and not simply a single-location panorama is this ability to transport from one specific location to another within the boarder context of the location being virtually visited. In this project, a novel algorithm was devised that determines the visually “best,” or most intuitive and visually pleasing, path when moving from specific

location to specific location with the tour’s virtual space. Additionally, an informative, hotspot-sensitive, interactive menu system was crafted to assist the user in navigating the space to meaningfully explore and learn about the many historical, architectural and artistic features within the Sistine Chapel.

2.1 Photography

In order to produce panoramic images of the Sistine Chapel, a 360° camera rig and rotational mounting device were used to manipulate the camera and accurately capture the complete range of vision. The operation of this camera rig and mount was mechanized to produce highly consistent results. Once the rig had captured some pre-determined number of images in a 360° range (for example, 15 images), the camera would automatically adjust by an increment, either tilting up or down, and take another 360° rotation of images. The result of this exhaustive photography was two sets of over 200 very high quality images from the two fixed location, with each of the sets containing significant overlap within the collections of images.

Responsibility for setting up and configuring the image capture equipment fell to the research faculty involved due to their experience with the equipment.

During the setup procedure, particular attention had to be paid to positing the camera properly on its mounting stand so that the back of the lens would sit precisely over the axis of rotation. If this careful calibration was not performed, images could be skewed slightly and may not line up later during the stitching procedure, producing a low-quality result. Extreme care was also taken in careful setting the focus and exposure levels of the camera so that images would be as clear as possible. The camera has manual focus capability only, with no ability to adjust focus during an image capture session, so a single focus setting had to be carefully chosen to produce the best overall results for the entire image-capture operation.

2.2 Stitching the Panoramas

The program “Stitcher Unlimited” was used to piece together the 200+ images that resulted from the image capture sessions inside the Sistine Chapel. While Stitcher does much of the work itself through “auto-stitching,” its graphical user interface and supplementary tools proved helpful when the results of stitching contained minor imperfections and had to be manually repair using the available tools.

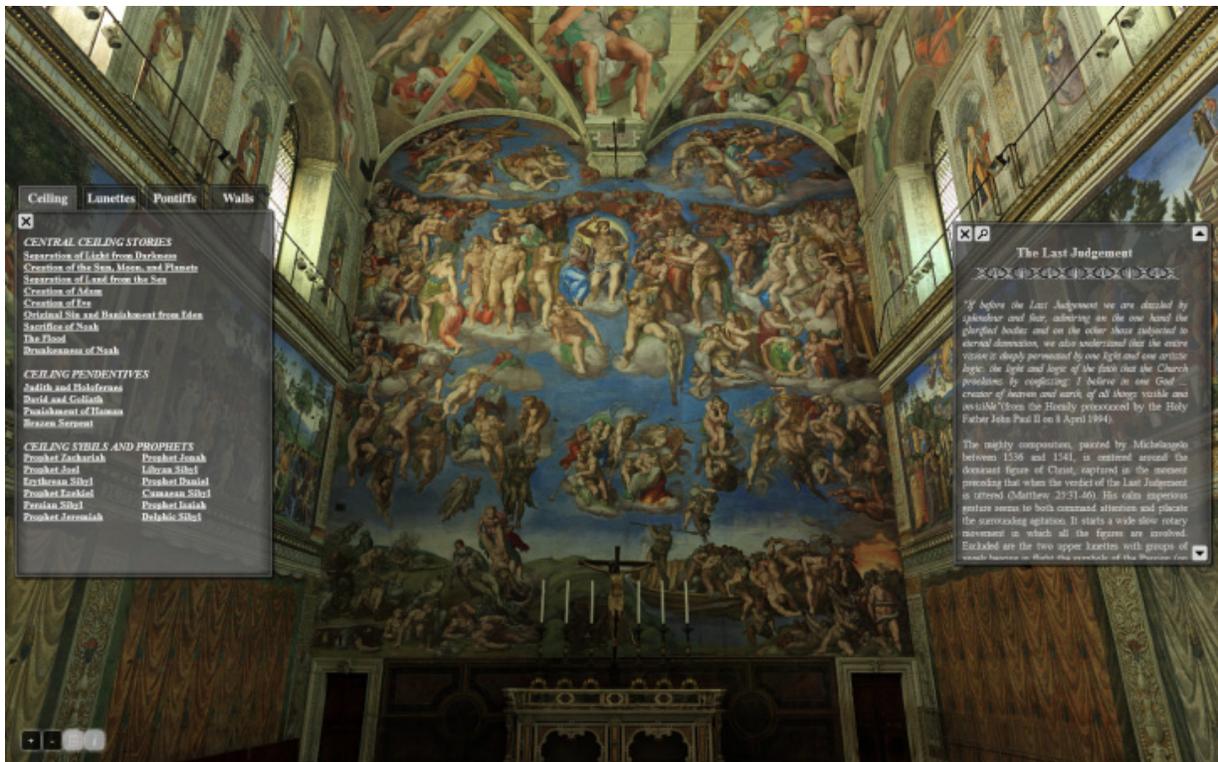


Figure 1. Screen capture of user interface, showing “The Last Judgment” in the Sistine Chapel.

The final output of this process was a massive, uncompressed digital image that is a flattened version of the entire 360° virtual tour. This uncompressed image was next processed using a plug-in tool of the Krpano program, which split the single tour image into over 500 images of many differently-sized rectangles. The Krpano program performed this splitting and selected rectangle sizes in order to best optimize the eventual online loading time of the finished, interactive, virtual tour product.

As a result of the vast quantity of images, and because the sets of images contained so much overlap, there was a significant amount of redundancy that needed to be managed. The solution to this redundancy problem involved selecting every second or third image in a sequence, which reduced the quantity of image input to Stitcher while providing ample information to form the complete 360° virtual tour image.

2.3 Interactivity

The next step in development of the virtual tour was incorporating interactivity, taking into account the needs and common behaviors of the viewer [6]. The initial task was to be able to view both of the single-location tours separately in the Krpano virtual tour viewing software. Internally, the tour is viewed by loading the web content for the tour into a web browser with also loads a corresponding file stored in the XML format that contains all content and commands to support the interactivity.

With the Krpano software, plug-ins were used such as the Options plug-in and Editor plug-in to obtain the coordinates for certain figures and features on the walls of the virtual tour and create interactive hotspots for these locations. An initial design component was the creation of a left-hand side, slide-out menu that listed wall figures and features. This initial menu was later abandoned in favor of a tabbed, fade-in and fade-out menu system that was more visually appealing.

Because the Krpano software provides no menu building plug-in, the menus were created by combining the capabilities of several plug-ins including the Text Fields, Images and Buttons plug-ins. The Text Fields plug-in provides limited HTML and CSS support for layout, so it was necessary to use multiple Text Fields to produce the illusion of a columnar layout.

Using Scroll Bars and Text Fields, a right-hand information menu was created that pops up when the

user clicks on a hotspot, providing relevant information on the figure, or figures, of interest. On this menu is a magnifying glass icon, which when clicked gives the user a “better view” of the hotspot. Since the user may or may not want hotspots to be enabled, they are disabled by default but can be turned on or off using a toggle button on the bottom left of the user interface screen.

Much of the look and feel of the interactivity is the result of the algorithm that was designed for this specific project, which seeks to maintain visual perspective and make movements feel natural to the viewer. The algorithm automatically determines parameters such as direction of motion, degree of zoom out and zoom in for producing a better viewing experience, and how best to go through the center door of the half-wall when transitioning from one fixed, central location to the other location within the Sistine Chapel virtual tour.

2.4 Algorithm for Maintaining Perspective

The most significant contribution to the field of virtual tour creation is the development of perspective maintaining interactivity that allows the user to get a true sense of the location. While many other virtual tours attempt to use multiple locations and hotspots, the common approach is to produce a visual blur while transitioning from one area to another [7]. This blur effect prevents the user from maintaining a true sense of the movement.

For this project, an algorithm was devised that attempts to produce movement such that the user loses perspective throughout the viewing experience, a desirable feature [4,5]. The algorithm was developed in a series of steps. Initially, a basic algorithm was created that produced simple movement from a current hotspot to the desired hotspot. This was adequate in some cases, although if the viewer was already zoomed in closely on a feature, the resulting movement was confusing and dizzying and viewers found themselves feeling lost in the virtual space. The result of this observation was a more generalize movement algorithm that calculates a desired movement path proportionally based on the distance to the current and target hotspots (Figure 2).

The perspective maintaining motion algorithm works as follows. First, the x and y coordinates of the current position and the position of the hotspot are acquired.

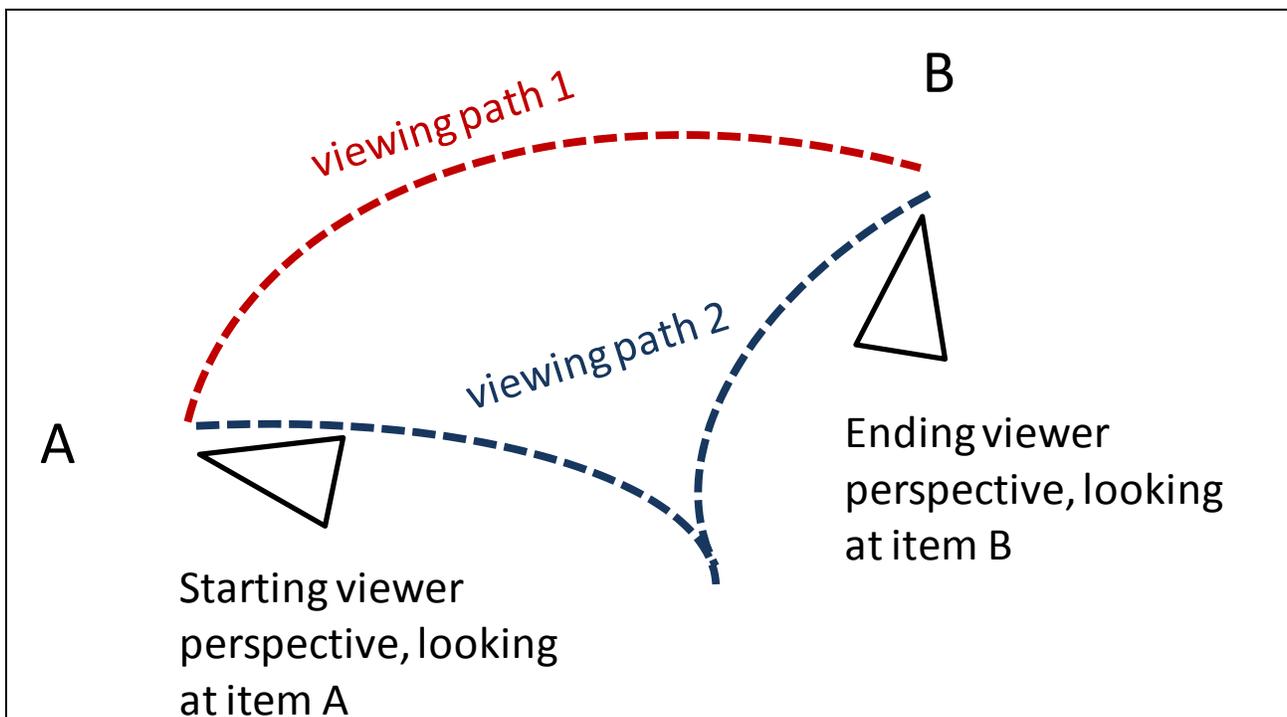


Figure 2. Comparison of naive, sweeping movement of the original viewing path 1 from viewing item A to viewing item B, which produces a less-desired viewer experience, with the improved back off, sweep, and zoom in movement of viewing path 2, which produces a consistent viewer experience that maintains perspective within context of the larger viewing area.

The Krpano viewer has a built-in function to get the current location, but the coordinate system can produce negative positions if the user's movement has been counter-clockwise. Therefore, if a position is negative, it is simply normalized by adding 360 to it, since with Krpano a location is guaranteed to be in the range of -360 to 360.

Next, the desired hotspot is compared with the bounds of each side of the Sistine Chapel tour to determine if it is in a different panoramic location.

The information for each hotspot is stored by simply creating an extra variable within the interactivity logic called **panoLoc**. If a hotspot is determined to be in the other location, motioned is created so that the viewer zooms toward the central door first before being panned to the new hotspot location.

The specifics of calculating this movement were determined to be that if the user is zoomed-in past a set

zoom value of 40 and the current position is not already near the door, then first the view is zoomed out to a set value of 15 and then movement to the door is produced. Otherwise, the view can be moved directly to the door and then zoomed-through as the corresponding XML interactivity file for the other location is loaded.

If the desired hotspot is within the current panoramic location, then an appropriate zoom value must be calculated for use in the movement. Since the possible movement to a hotspot can be clockwise or counter-clockwise, the appropriate distance is always calculated to be the shorter rotational distance. Therefore, the x distance to move in degrees, or Δx , will always be between 0 and 180.

Because of this restriction, x must be normalized based on whether it is either greater than 180 or less than -180. The calculation is performed as follows:

If the value of x is greater than 180, then calculate Δx to be $360 - \text{curX} + \text{gotoX}$.

If the value of x is less than -180 then calculate Δx to be $360 - \text{gotoX} + \text{curX}$.

If the value of x is any value in between -180 and 180, then calculate Δx to be $\text{curX} - \text{gotoX}$.

Δy is calculated simply by calculating $\text{curY} - \text{gotoY}$.

The x distance is calculated, and a ratio of distance versus the distance to the hotspot is determined. The zoom values range from 15-55, making a range of 40. Multiplying the distance ratio by 40 produces a zoom value in proportion to the calculated distance. Because the zoom values start at 15 and not 0, the zoom value must be normalized by adding 15 to it to ensure that is in a valid range.

A final observation regarding zoom and movement was that no zoom is needed if the distance from the current hotspot to the desired hotspot is some minimal distance. Through objective observation and experimentation with a variety of values, it was determined that any rotational distance between hotspots that is less than 60° does not require zooming out in order to maintain visual perspective. For motion further than 60° , the calculated zoom value should be used as it produces a visually consistent experience for the viewer.

This carefully designed movement with the interactive virtual tour provides virtual visitors with a real sense of being within and looking around inside the Sistine Chapel. Using the perspective maintaining algorithm, movement throughout the Chapel is natural and viewers maintain a consistent sense of space and location.

3 Challenges

While developing the virtual tour, several issues were encountered that necessitated adaptations to the approaches that were used. First, an observation that designating regions that were greater than a certain size as hotspots did not produce the desired 3D effect. The solution was to divide many of the hotspots into groups of “linked” hotspots. The added benefit of this approach is that it was possible to mask out protruding objects such as railings and light fixtures. In this way, moving the mouse over any of the “linked” hotspots automatically invokes the presentation of the

information associated with any of the associated “linked” hotspots so that information is presented to the user in a uniform and consistent manner.

Second, naming conventions were devised to produce consistency throughout the project. Because of the scale of the project, work was split between developers and when different naming conventions were used for variables or hotspots, significant effort was required to reconcile the functionality of individually produced components. By making naming uniform, unsurprisingly, the pieces of the project produced in parallel were integrated much more easily.

Third, it was realized that including all hotspot and data information into a primary XML interactivity file was cumbersome and could potentially reduce loading performance. The solution was to create a separate data XML file and two hotspot XML files, each of which were “included” into the main interface content as needed.

Finally, it was determined that incorporating JavaScript coding within the Krpano viewer scripting led to unreliability of the menu navigation system. The decision was made to program the interactivity purely within Krpano, which introduced additional challenge but resulted in a more dependable and stable result.

4 Conclusions

This project produced some significant contributions to the area of virtual tours, both in terms of the unique subject (the Sistine Chapel) and perhaps most importantly an algorithm for maintaining perspective when moving through a virtual tour space. While Krpano makes use of a developed scripting language, it is still evolving and therefore lacks some desired functionality and corresponding documentation.

The lack of working examples of other Krpano-based tours was one of the biggest challenges that made building the interactive menu system especially difficult. In our experience, this project has made the most extensive use of Krpano programming and functionality by a significant margin, and it is hoped that others producing such tours can benefit from this experience.

For anybody who has experienced a virtual tour, it is evident that interactivity of the tour enhances the experience for the user. It allows users to feel the real sense of the location as they, in this case, seamlessly move through the Sistine Chapel. Maintaining user perspective during automatic movement adds to the user

experience, combining ease of navigation with a sense of being present in the tour.

5 Future Work

While the current version of this interactive tour is good according to users who have tried it, there are improvements that could be made to movement functionality and loading time. Once deployed, the tour is likely to draw considerable traffic on the Internet, so more efficient techniques for loading the visual and data content and for moving throughout the virtual tour are needed.

Mobile application of programs seems to be rapidly moving to many different parts of software development and programming. Another extension of this virtual tour project would be to develop it for a mobile platform such as the iPhone or Droid. Until recently, Krpano used Adobe® Flash™ in its rendering of the virtual tours. Recently, Krpano added support for using JavaScript and CSS in XML files in order to display a virtual tour. While this does not apply to the use of plug-ins (such as menus) and hotspots, it could increase opportunities for improved interactivity in the future, including mobile platforms.

6 References

- [1] Andolina, S., A. Santangelo, and A. Gentile. "Adaptive Voice Interaction for 3D Representation of Cultural Heritage Site." *Intelligent and Software Intensive Systems*, 2010.
- [2] Leigh, Jason, Andrew Johnson, Christina Vasilakis, and Thomas DeFanti. "Multi-perspective Collaborative Design in Persistent Networked Virtual Environments." *Evl | Electronic Visualization Laboratory*, 2010.
- [3] User Documentation. Krpano.com, 2010. URL: <http://krpano.com/docu>, Accessed: Dec. 12, 2010.
- [4] Niklas Elmqvist, M. Eduard Tudoreanu, and Philippas Tsigas. Tour generation for exploration of 3D virtual environments. In *Proceedings of the 2007 ACM symposium on Virtual reality software and technology (VRST '07)*, Stephen N. Spencer (Ed.). ACM, New York, NY, USA, 207-210, 2007.
- [5] Niklas Elmqvist, Mihail Eduard Tudoreanu, and Philippas Tsigas. Evaluating motion constraints for 3D wayfinding in immersive and desktop virtual environments. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems (CHI '08)*. ACM, New York, NY, USA, 1769-1778, 2008.
- [6] Bastanlar, Y.. User Behaviour in Web-based Interactive Virtual Tours, In: *Proc. of 29th International Conference on Information Technology Interfaces*, Dubrovnik, Croatia, 2007.
- [7] Eun-Young (Elaine) Kang, Ilmi Yoon, "Smooth Scene Transition for Virtual Tour on the World Wide Web," *Sixth International Conference on Computational Intelligence and Multimedia Applications (ICCIMA'05)*, pp. 219-224, 2005.