

A scalable platform to share and navigate 360° cylindrical panoramas

Abhinav Asthana
BITS-Pilani
abhinav@bits360.com

Padmanabhan Thangaraj
BITS-Pilani
padmanabhan@bits360.com

Abstract

We present the creation and development of BITS360, a web-based framework that allows users to create & navigate a 360° horizontal panoramic environment. Our work is orthogonal to previous methods for creating multi-viewpoint panoramas, gigapixel images and split-scan panoramas. We introduce waypoints and overlays as essential elements in virtual reality and present a method of creating & linking waypoints and overlays to enhance panorama navigation and utility. We present the evolution of a framework that enables users to upload, view and annotate panoramas. This work as a whole documents, the steps in improvement of the BITS360 UI based on comprehensive user base feedback.

1. Introduction

We describe here a virtual tour system called BITS360 which builds upon previous such attempts. A large number of panoramas have been added from all over the world using the BITS360 system. In this paper, we present the current system along with some new techniques that can be incorporated to make virtual tours more useful using a collaborative approach.

The major contribution of our work is the development of a practical approach to panorama integration using current techniques and technologies on a web-based platform. Whereas a large amount of work is

available in literature to generate virtual reality, their interactivity and the ability of users to contribute to the creation of the virtual environment has almost completely been ignored from academic literature. BITS360 is designed to let the novice user define his own VR in the panoramic imaging domain and share it over the net. Our focus has been constantly to design the most effective interface towards the effort to realize a VR environment created by its users.

1.1 Related Work

A “panorama” typically refers to single-viewpoint panoramas, which can be created by rotating a camera around its optical center [1]. Multi-view point panoramas [2] on the other hand promise to deliver an equivalent experience by using a relatively sparse set of photographs captured with a handheld still camera that is moved along the scene.

Google’s Street View [3] provides a comprehensive 360° horizontal and 290° vertical panoramic street view. However, this is created centrally using a camera bike/car with no opportunity for user involvement and annotation. Street View clones like Immersive Media[4], MapJack [5], NORC[6], Seety[7], Cyclomedia [8], Everyscape[9], Driveme[10] & LocationView[11] also require images captured from public roads at street level, using a car to ferry the camera system. Location View, however, is optimized to provide a higher frame rate and seamless transition between frames, enabling continuous, lifelike motion and

surroundings that are animated rather than static. Location view maps (unlike those of any other solution) are updated at least once every 2 years at significant recurring cost. Most commercial implementations are unable to provide mapping for indoor environments. In such a scenario, it is imperative for UI frameworks to evolve into autonomous systems that provide users the tools to upload and cross-link panoramas generated using off-the-shelf cameras.

An alternate approach has been to hierarchically organize user generated images by Geo-Relevance[12]. McCurdy and Griswold's RealityFlythrough[13], Snaveley et al.'s PhotoTourism[14] and Microsoft's PhotoSynth[15] organize the images of a single event in zoomable 3D space using full camera parameters and common reference points within images. These techniques assume the availability of a large user base to generate content for their algorithms. An all-together different approach is taken by Pixelcase that provides aerial panoramas [19].

The primary motivation for our work has been to provide user's complete flexibility in creating individual panoramas of a scene using off the shelf cameras. A full view cylindrical panoramic mosaic is created from image sequences using conventional image processing techniques[14]. Currently Hugin [16], an open source effort provides photographers and enthusiasts a very easy way of creating panoramas. Panoramic photography is easy but its mass acceptance has failed due to the lack of a platform to share the images. The rest of the paper is organized as follows: Section **Error! Reference source not found.** describes the architecture of the BITS360 platform that was used to prototype our approach. Section **Error! Reference source not found.** describes how user's can navigate the 360° panoramas. Section 4 describes how the platform links multiple panoramas together using waypoints. Section 5 details overlays that can annotate a panorama. Section 0 documents other features such as the Snapshot tool, and Google Maps Integration. Section 7 shows

further improvements being made on the BITS360 experimental platform.

2. Development technologies

The BITS360 platform is based on the standard model of client-server interaction. There are 3 components of the platform: Web Client, Integration Server, and Web Services like Google Maps and Geo-Location services.

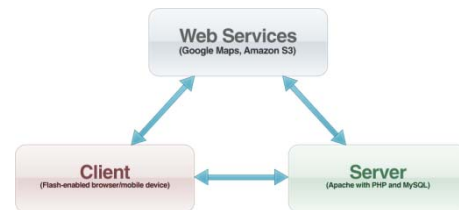


Figure 1. BITS360 platform

On the client side, BITS360 requires that the Adobe Flash Player plug-in be installed in the browser. Adobe Flash is a technology from Adobe Systems, which enables rich media applications on the Internet. We chose Flash because of its high browser penetration- around 99%[11]. This lets almost everyone to access the BITS360 virtual tour without having to install additional software. In contrast, other rich media technologies like Java and QuickTime have a lower installed base. One other significant advantage of using Flash is that a large number of web services support Flash and provide code libraries to pull content into Flash, which we can use to enrich the environment. BITS360 uses the Google Maps library to display geographical maps. Users tend to prefer the Flash version of web sites to partially HTML version. User characteristics and web site purpose can also influence the perceived usability of Web sites [22].

On the server side, BITS360 uses PHP as the scripting language and MySQL as the database for storing panoramic meta-data. As both are open source technologies the setup costs are minimal [23]. The PHP/MySQL combination has proved to be highly scalable [24]. The combination has

been tested on high traffic web portals including the likes of Yahoo, Facebook and Digg.com [25] and hence enables us to implement our architecture on a large scale.

3. BITS360 navigation system

3.1 Storing and displaying panoramas

BITS360 is capable of displaying cylindrical panoramas. We have used cylindrical panoramas because of their ease of construction [6]. To build a cylindrical panorama, a sequence of images is taken by a camera mounted on a leveled tripod. These can be then stitched together using commercial packages like ArcSoft. The system will also support other wide-angle formats such as 180-degree and 270-degree imagery.

BITS360 stores the panoramas as JPEG files. The JPEG compression algorithm works out best for storing high-resolution photographs. The file sizes are low without loss of significant quality. On an average a BITS360 panorama weighs around 380 Kilobytes for a resolution of 4663 x 489 pixels. Due to such low file sizes the BITS360 system can be scaled up for a large number of panoramas efficiently. In a future version of BITS360 we aim to put these panoramas on a cloud based storage solution like Amazon S3. To upload the panorama on the BITS360 system the user has to fill up the form shown in Figure 2. Using the Google Maps widget the user can also add the latitude and the longitude where the panorama was taken.

Figure 2. Automated upload of user panorama

Information related to the panorama is stored in the database and is pulled up whenever the client requests the panorama to be displayed.

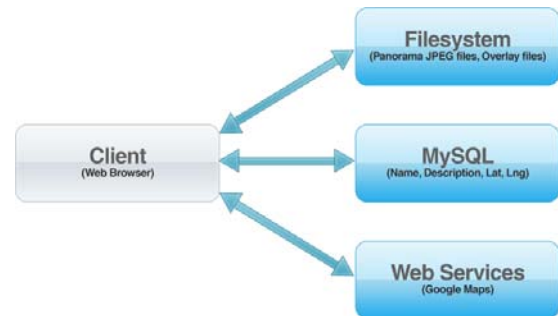


Figure 3. Displaying the panorama on the client

3.2 Interaction with the panoramas

BITS360 panoramas can be moved around using the mouse. The mouse pointer changes dynamically to show the direction in which the panorama is moving. Also instead of just coming to an abrupt halt- akin to hitting a brick wall- when the mouse button is released the panorama eases out and then stops. This results in a smoother experience for the user. This can also be used for flicking the panorama instead of the standard scrolling method. The browser is well integrated with the virtual tour system. The panorama expands to the full width of the browser to provide as much viewing area as possible. Furthermore, panoramas can be bookmarked like ordinary websites. The forward and back browser buttons work like normal HTML websites. The absence of this functionality is a common usability problem in websites rendered entirely in the Flash environment [21].

4. Linking panoramas together

BITS360 uses a unique waypoint system to allow users to jump from one panorama to another. Though horizontal panorama viewers, with large databases do exist like Arounder [18], Full screen QVTR [19] and Vray [20], however, none of them capture the relevance of each panorama by its geographic reality in the virtual space, which

we try to achieve by using the waypoint system. Waypoints act as hyperlinks in a virtual environment. The BITS360 waypoint system allows panorama contributors to link together panoramas effectively and with very low maintenance effort.

4.1 Displaying Waypoints

4.1.1 First approach

The first version of BITS360 had circular buttons as waypoints. As there can be more than one panorama in a particular direction, we had to bunch together several links in a list. On clicking the circular button a list is shown containing this list. The problem with this approach is that the user has to click twice to go to another panorama. This can get quite cumbersome after spending some time in the virtual environment.



Figure 4. Default waypoint state



Figure 5. Panorama list on clicking the waypoint

Another approach could be to display a list of nearby panoramas on the GUI. However, this does not give an idea about how the panoramas are spatially located with reference to the panorama being viewed.

4.1.2 New system

Waypoints for individual panoramas are stored in the database as co-ordinates with

reference to the panoramic image. However, it'll be cumbersome for the user when many waypoints overlap over each other. We have solved this problem by replacing all the waypoints which overlap over each other by a single button.



Figure 6. Waypoint system



Figure 7. Clustering waypoints

The top right corner of the button shows the number of panoramas being linked to through this waypoint. On clicking the panorama the button expands to show the list of panoramas being linked to. The user can scroll through the list while keeping the mouse button pressed. If the user doesn't find a panorama of interest he can move the mouse pointer away and the list would collapse into the button again.



Figure 8. Multiple waypoints as a list

This process takes only one mouse click and also solves the problem of waypoints clustering together in a close bunch.

4.2 Creating Waypoints

4.2.1 Traditional approach

Links to other panoramas in virtual tour systems are usually created manually. These links are stored as co-ordinates with respect to the image and stored in a database. When the panorama is to be displayed, the viewer creates hotspots on the panoramic image. Creating waypoints manually is a tedious process. An earlier version of our work had a system in which the user could select a panorama from a map and drag a waypoint over the image to its correct place. This had to be done for every panorama to which the user would want to link to. If there are 15 panoramas, then this will have to be repeated 15 times. And when some other user adds a new panorama, all nearby panoramas would have to be updated. In an environment with a large number of users, the virtual tour experience can quickly stagnate as new waypoints won't appear quickly enough.

4.2.2 New approach

The new version of BITS360 uses a simple yet effective technique to solve the above problem. After a user uploads the panorama, he is taken to the screen shown in Figure 8.



Figure 9. Aligning multiple panorama

All the panoramas falling inside the brightened portion of the circle will show up as waypoints in the panorama being edited. The user can expand the circle using a control widget on the perimeter of the circle. Using another control widget, the user can rotate the panorama to align it with a

waypoint or a physical landmark. Effectively, we are aligning the panoramic image with a geographical point. This allows us to calculate the angle between this panorama and other panoramas lying within the circle as shown in Figure 10.

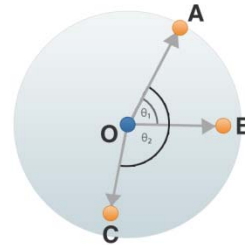


Figure 10. Calculating waypoint angle

This angle can then be used to map the corresponding waypoint's x co-ordinate on the JPEG image. For example the x co-ordinate for panorama B on the JPEG image for panorama A is given by

$$x = (\text{angle between OA \& OB}) * \text{width of } \frac{A}{2\pi}$$

The advantage of this method is that once the panorama is aligned with a latitude and longitude on the map, new panoramas in the specified radius will show up without the need for user intervention.

The waypoint co-ordinates are calculated at the time when the panorama is aligned. Also, when the client requests a panorama, the system also checks whether new panoramas have been added within the radius specified by the user. If a new panorama is added, its co-ordinate is calculated and stored in the database for quick retrieval later.

A limitation of this method is that the available information is insufficient to calculate the y co-ordinate of the waypoint on the image. However, displaying them in the middle of the panorama has proven satisfactory in a majority of cases. If required, the waypoint can be dragged in the panorama editing mode to change the y co-ordinate as well as x co-ordinate for finer adjustments.

5. Overlays

The usefulness of the panorama can be greatly improved if more information can be provided about the location or physical structures over the panorama itself. The overlay system aims to achieve this in a way which doesn't mar the user experience while navigating the virtual environment.

Overlays can range between anything from text to images and even video clips. The current version of BITS360 supports text overlays.

Overlays can be created using the panorama editor and then dragged and placed at an appropriate location. The panorama owner would be able to control what information is placed on his panorama. In case any inappropriate content is posted, it can be removed by the owner. Both the control for viewing overlays and waypoints can be turned off with a right click drop down menu as well as on a small tab on the top right corner.



Figure 11. Overlays

6. Other Features

6.1 Snapshot Tool

The BITS360 UI also provides the user a simple way to click a snapshot, by which he or she gets an image of a small portion of the panorama. The interface has been designed similar to the viewport of a camera to make this an intuitive interaction. This

feature can be expanded to email the pictures directly to friends.

6.2 Google Maps Integration

With Google Maps integration, the panoramas are now geo-tagged and hence have the scalability to map the entire globe. It increases the relevance of the panoramas, and gives the user the possibility to create a VR experience of any place mapped by satellite imagery today. Google Maps also provides the network of panoramas other information like routes, transportation services, nearby interests etc. The Map can be accessed with a button on the bottom right corner on the panorama as shown in Fig. 12, and has all the necessary tools for navigation such as panning, dragging and zooming.

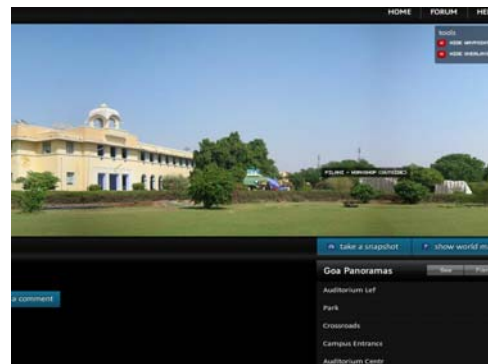


Fig 12. Snapshot Tool and Google Maps Button on the bottom right corner of the panorama

7. Conclusion

Future versions of BITS360 aim to provide more tools to users to enable them to customize the virtual environment. Users would be able to create their personal copy of the virtual environment as well as share it with others. The panoramas can be directly embedded as current technologies for video such as YouTube do.

Overlays can be in the form of audio and video allowing people to share their

experiences of a particular location. A wide variety of educational as well as tourist related applications are possible using overlays. Another source of overlays can be news portals or video sharing portals. Content can be pulled up directly from these portals and mapped on to the panoramic image using the waypoint system described in this paper. BITS360 will be the standard for 360-degree panoramic photography and hence would be an obvious choice at an organizational level in the areas of heritage monuments, architecture and real estate.

In accordance with the simplistic nature of this VR experience, BITS360 gives the user an opportunity to photo-document ones travels, journeys, locations and experiences in a unique way. It also pushes the boundaries for creative mixed media interactions with the possibilities of videos and sound overlays from the locations for artists and enthusiasts alike.

7. References

- [1] R. Szeliski and H. Shum. Creating full view panoramic image mosaics and environment maps.
- [2] A. Agarwala, M. Agrawala, M. Cohen, D. Salesin, and R. Szeliski, Photographing long scenes with multi-viewpoint panoramas in *ACM Trans. Graph.*, vol. 25, pp. 853-861, 2006.
- [3] Google Maps Street View. <http://maps.google.com/help/maps/streetview/>.
- [4] ImmersiveMedia. <http://www.immersivemedia.com>.
- [5] MapJack. <http://www.mapjack.com/>.
- [6] S.C. eXtreme Soft Group. <http://www.norc.ro/>.
- [7] Seety Ltd. <http://www.seety.co.uk/>.
- [8] CycloMedia Technology B.V. <http://www.cyclomedia.com/>.
- [9] EveryScape 2009. <http://www.everyscape.com/>.
- [10] DriveMe. <http://www.driveme.in/>.
- [11] LocationView Corp. <http://www.locaview.com/>.
- [12] B. Epshtein, E. Ofek, Y. Wexler, and P. Zhang. Hierarchical photo organization using geo-relevance in *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems* Seattle, Washington: ACM, 2007.
- [13] N. J. McCurdy and W. G. Griswold. A systems architecture for ubiquitous video in *Proceedings of the 3rd international conference on Mobile systems, applications, and services* Seattle, Washington: ACM, 2005.
- [14] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D in *ACM SIGGRAPH 2006 Papers* Boston, Massachusetts: ACM, 2006.
- [15] Photosynth. <http://labs.live.com/photosynth/>.
- [16] Hugin - Panorama photo stitcher, <http://hugin.sourceforge.net/>
- [17] Pixelcase, Aerial Photography of New York City, <http://www.pixelcase.com.au/vr/2009/newyork/>.
- [18] Arounder. <http://www.arounder.com/>.
- [19] Full Screen QVTR. <http://www.fullscreenqvr.com/>.
- [20] VRWAY <http://www.vrway.com/italian/default.html>.
- [21] Pilun Piyasirivej. Towards usability evolution of Flash websites.
- [22] Pilun Piyasirivej. Usability of Flash and HTML web sites, A comparative study using a Contingent Heuristic Approach.
- [23] UV Ramana and TV Prabhakar. Some experiments with the performance of LAMP architecture.
- [24] The PHP Scalability Myth. http://onjava.com/pub/a/onjava/2003/10/15/php_scalability.html.
- [25] Digg PHP's Scalability and Performance. http://www.oreillynet.com/onlamp/blog/2006/04/digg_phps_scalability_and_perf.html.