

# Geometry Directed Browser for Personal Photographs

Aditya Deshpande\* Siddharth Choudhary P J Narayanan Krishna Kumar Singh

Kaustav Kundu Aditya Singh Apurva Kumar

Center for Visual Information Technology  
International Institute of Information Technology  
Hyderabad, India

## ABSTRACT

Browsers of personal digital photographs all essentially follow the slide show paradigm, sequencing through the photos in the order they are taken. A more engaging way to browse personal photographs, especially of a large space like a popular monument, should involve the geometric context of the space. In this paper, we present a *geometry directed photo browser* that enables users to browse their personal pictures with the underlying geometry of the space to guide the process. The browser uses a pre-computed package of geometric information about the monument for this task. The package is used to register a set of photographs taken by the user with the common geometric space of the monument. This involves localizing the images to the monument space by computing the camera matrix corresponding to it. We use a state-of-the-art method for fast localization. Registered photographs can be browsed using a visualization module that shows them in the proper geometric context with respect to a point-based 3D model of the monument. We present the design of the geometry-directed browser and demonstrate its utility for a few sequences of personal images of well-known monuments. We believe personal photo browsers can provide an enhanced sense of one's own experience with a monument using the underlying geometric context of the monument.

## Keywords

Structure from Motion, Localization, Photo-tourism, Photo Browser

## 1. INTRODUCTION

Digital cameras have become the standard method to capture one's visual experiences to relive later. Photographs taken by such cameras are today typically watched on a computer screen. Personal photo browsers allow the photogra-

\*Corresponding author

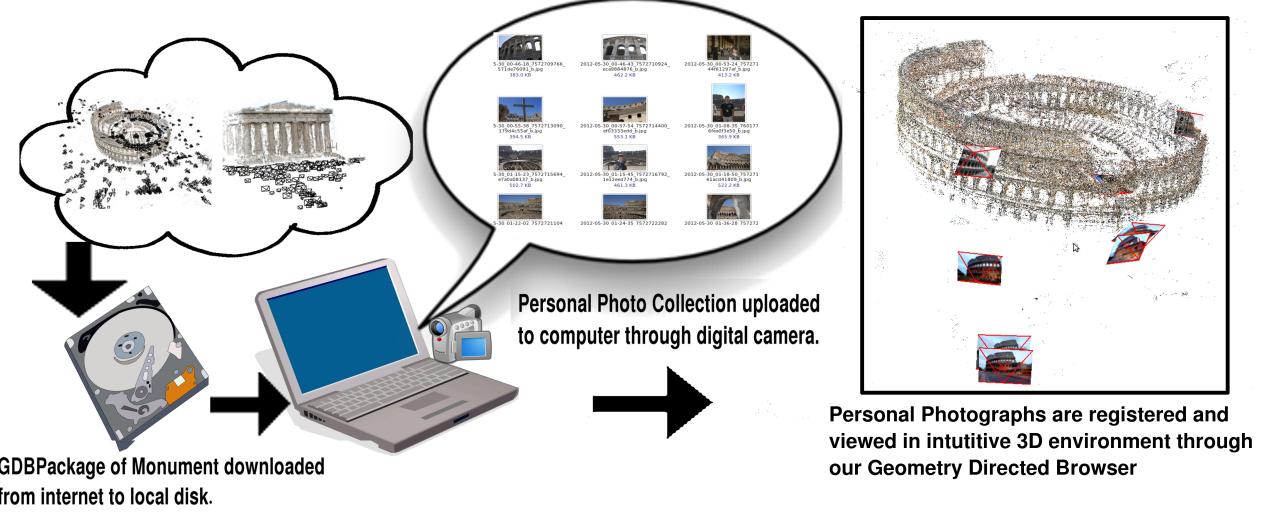
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICVGIP '12, December 16-19, 2012, Mumbai, India  
Copyright 2012 ACM 978-1-4503-1660-6/12/12 ...\$15.00.

phers to watch the pictures again. Photo browsers, however, follow the model of presenting the collection of photographs linearly in the order in which photos were taken, usually in a slide show mode. This model has practically not changed from the early days of digital photography, though there has been drastic changes in how humans interact with other digital data. The photo sharing sites like Flickr, Picasa, and Facebook, also provide similar browsing mechanisms, but enable one to share the photographs with a larger circle of people easily. Some tools have started to use the content in the photographs, for example, Zhang et al. perform automated annotation of family albums using a Bayesian framework to model the similarity of faces [15], Davis et al. in addition to traditional face recognition also use contextual data like time of capture, GPS tags, other people in the photo, indoor/outdoor scene etc. to annotate photographs [3]. This enhances the browsing experience of photographs of social engagements involving people. Such methods are still not available for photographs of monuments, cities, etc., that individuals continue to capture.

The photo tourism work [11] and the PhotoSynth system that grew from it [1] allowed 3D-aware browsing of a community photo collection (CPC) or images of a common space taken by several users over the years and shared on popular sharing sites. This method browses photographs of the collection by several individuals together. PhotoSynth in effect provides a 3D-directed slide-show of the collection of photographs, based on compute-intensive processing of the data using the structure-from-motion (SfM) procedure. Ordinary users are usually more interested in personally reliving the memories of their visits using photographs that they collect and not of the community collections of the same site.

In this paper, we present a method and a system that provides a rich, geometry-directed browsing experience to individuals of their own personal photographs of a place of interest. The browsing experience we provide is guided by the coarse 3D geometric structure of the underlying space. We use the point cloud obtained using SfM using a community collection of pictures of the same space to provide the geometric context. Such models are either already available or can be created easily for many (if not all) places of popular interest today. Creating this structure is an offline process and needs to be performed only once per monument. Our system has two modules. The first module processes the photographs the user uploads and quickly registers them to a pre-stored coarse 3D geometric structure using a package of information specific to the space under consideration. We adapt the state-of-the-art approaches to



**Figure 1:** Concept diagram of geometry-directed browsing. The user first downloads a GDBPackage for the sites of interest from a common gallery before going on the trip. On the road, sequences of photographs for each monument is uploaded from the camera to a laptop. The photos are registered to the space of the monument using the GDBPackage. The GDBrowser allows the user to browse through these registered photographs using an intuitive, geometry-directed mode.

perform this step. The second module provides an intuitive, geometry-directed method to browse these registered photographs, enhancing the user's personal experience of the visit within a non-intrusive geometric context. This is best thought of as another way to browse one's photographs of the space, beyond the standard linear method.

Figure 1 shows the concept behind our Geometry Directed Browser (GDB). An Internet gallery (or a shop) has the data packages – called GDBPackages – appropriate for several monuments of popular interest. Each package has the essential information needed by our system – the 3D point cloud, visibility matrix, etc. – in a suitable format. Users acquire packages needed for the monuments they intend to visit before the trip commences. During the trip, users capture photographs of his or her choice, using a digital camera (or a smart phone) of the monument. The photos are uploaded to a laptop (or other personal device) at the end of the day or periodically. A special program (or “app”) on this device will process and register the photographs to the 3D space of the monument using the GDBPackage. This step uses the state-of-the-art localization algorithm to compute the camera matrix corresponding to each user photograph. The registration is done on the user's compute device and not on the cloud, as good connectivity for transferring high-quality pictures is not expected on the road. The registered photographs can subsequently be browsed within the 3D geometric context of the monument using our browser.

Our objective is to provide geometry-based browsing of the personal collections, which typically contains a few tens of photographs of each monument. Thus, while photo tourism allowed one to be a tourist at a site using the pictures a large number of users have taken and shared, our system aims at providing a geometry-directed browsing experience for the photographs of an individual tourist. Our focus is on providing a good browsing experience to the users. We have built our system to be fast enough on a personal device like a laptop

top by minimizing the computation effort required at each step. Registering the photos to the monument takes a few seconds per photograph, which is roughly the same order of time required to transfer it to the camera. The computation can also overlap with the transfer. Browsing is interactive and fast as it mostly uses stored information and renders only a point cloud. We intend to extend the same facility to a more universal personal device like a smart phone in the future.

## 2. RELATED WORK

With the advent of digital cameras, it became possible to capture and store a photograph on a computer in digital format. Eventually, this also led to the development of photo browsers on the computers. The photo browsers provide a visualisation of a set of photographs clicked by the user on a display device, eliminating the need to print a hard copy of each and every photograph.

The photo browsers in traditional GUI-based computer systems have the following two basic views:

1. Explorer View: All the photographs in an album are visible in small frames called thumbnails.
2. Photo View: Once the user selects a particular photograph from the above explorer view, the user can view a detailed high resolution display of the selected photograph. In this view, the user can also directly toggle to previous/next photo in alphabetical or temporal order using some keyboard input and/or GUI object, thus simulating a slide show.

There are a large number of such photo browsers today. Eye of Gnome, Fspot, KSquirrel, Windows Photo Viewer, iPhoto and photo browsers of popular websites like Flickr, Picasa and Facebook, are all photo browsers that follow the above two views paradigm. Some sophisticated photo

browsers (viz. Picasa, iPhoto) use techniques of face recognition, scene classification, etc., to provide a meaningful photo browsing experience to the user [15, 3]. Photo browsers like Google Maps and Flickr also use the user-supplied geo-location or tagging information present in the photographs to overlay them on a map or a satellite image.

Despite recent advances in Structure-from-Motion [11, 12] and other 3D Computer Vision techniques, there has not been much work in developing photo browsers that help the user navigate his/her personal photo collection more intuitively and in a 3D space. Snavely et al. developed a system wherein they take a large unstructured community photo collection (CPC) as input to reconstruct the constituent 3D points and viewpoints (i.e. cameras) from them [11]. Using this information they developed the PhotoTourism system which grew into the PhotoSynth system from Microsoft. This is a photo explorer for the large collection of pictures obtained from the internet. This photo explorer provides the user with controls to navigate the reconstructed 3D space and the constituent cameras are visible as frustums. Once the user selects a frustum, the image from that camera is made visible. Users can use the navigation controls to see spatially nearby photos which are retrieved by maintaining the information about 3D points that overlap between any two photos. When a user wishes to go from one photo to another completely different photo (not necessarily in close spatial proximity), the viewpoint moves through a sequence of intermediate photo transitions. The dependence on these intermediate photos implies data storage to the order of the size of CPC (a few GBs). Thus, this approach has a high memory and computation overhead, Kumar et al. developed a computationally lighter approach where they perform only local reconstruction of a small set of overlapping images which is sufficient for the photo browsing task [13]. Their system depends on retrieving similar/overlapping images (with the current image) from the database to provide realistic pairwise photo-transitions. As per our analysis of data obtained from Flickr, personal photo collections of a large monument like Colosseum and Pantheon are in the range of 10 to 40 photographs only, moreover most of them will lack sufficient overlap to perform local reconstruction. To visualize these personal photo collections, this method will require additional storage of a subset of images from the CPC. Also, the resulting photo transitions will then consist of images from the CPC which might have seasonal variations, difference in time of day as compared to the user's photos and this will affect the visual appeal of the system to the user.

Snavely et al. [10] also use the positions of the cameras obtained from SfM to estimate interesting paths (along which the monument is typically photographed) specific to the monument in the input dataset. They then use this path in the photo transitions discussed above, to make them seem more realistic. VisualSfM is a visualization tool for SfM, but its focus is on visualizing the incremental process of SfM [14]. Once the SfM pipeline concludes, using this tool the user can view the 3D point cloud with cameras as frustums and also navigate through it. But the system does not provide a mechanism for transition between two photos, nor does it allow for registering new photos without running structure from motion pipeline.

There are two major categories of *localization* algorithms, one using image based search in a geo-tagged dataset and

the other using structure information present in the SfM dataset. In the image based search approach, Hays et al. use scene matching with a geo-tagged dataset to output a probability distribution over possible geo-locations of the query image [4], Panda et al. use instance retrieval in an annotated dataset to geo-locate user's photos of a digital heritage site on an underlying map [7]. There is also significant work done on localization of new images in a SfM dataset. Irschara et al. find point matches of query image with nearby (similar) images to localize the query image [5]. Li et al. prioritize points using visibility to find 3D-2D matches before localizing using RANSAC [6]. Choudhary et al. use the framework of visibility probability to guide the matching process and obtain high inlier matches for fast localization [2]. Sattler et al. use 2D-3D matches using lists of points that map to the same visual word for finding matches in a new image for localization [8]. In continuation with this work, Sattler et al. also devised a method that uses both 2D-3D and 3D-2D matches which allows them to give better runtime performance without affecting the accuracy [9]. We use the localization to SfM dataset category of localization algorithms since the 3D point cloud information in SfM dataset allows us to provide an intuitive 3D visualization. We specifically use the localization method of Choudhary et al. for our localization module [2]. This method does not depend on nearby images to register new photographs and hence the storage of entire CPC photos is not required. This localization method primarily requires only a cover set of reconstructed points and the visibility matrix (*point vs point common cameras*) to perform localization.

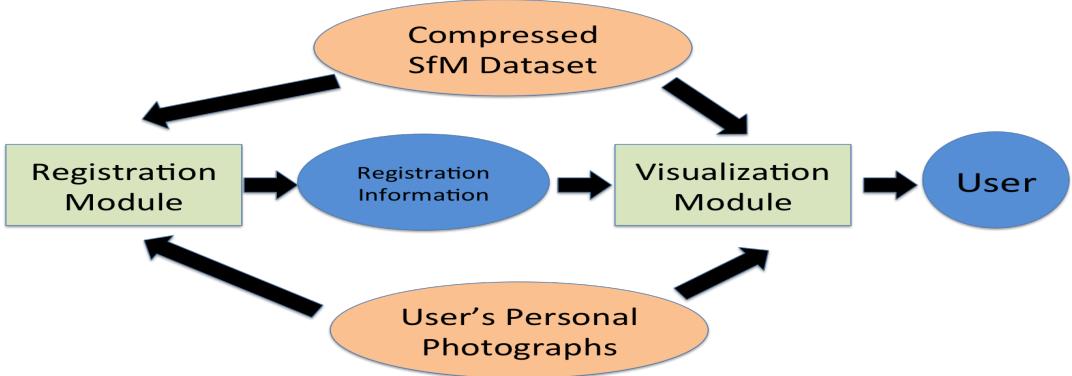
### 3. OUR METHOD

Our system is intended to work on low-end processors such as those present in a personal laptop. The system starts to work when a set of photographs are uploaded to it typically from a camera after a day at a monument. Registering the photos using pre-stored 3D information on the monument is the first step. The visualization module then allows the user to browse his or her images within the geometric context of the monument. Because of the reasonable storage and computation requirements, our method can also work on a small device like a tablet or a smart-phone, which also has a camera to capture photographs.

#### 3.1 Outline

Our Geometry Directed Browser(GDB) application has two parts: the registration module and the visualization module. The registration module processes the photographs after they are copied from the camera to the computer. The processing involves placing each photograph in the 3D coordinate space of the pre-computed coarse 3D reconstruction of the monument. Parameters such as the 3D position of cameras, the rotation matrix and the viewing direction are computed for each photograph in the user's collection, these are written to a file. This file is then used by the visualization module for providing interactive browsing of the photographs.

Both modules use a pre-computed GDBPackage containing essential geometric information about the underlying monument as input. This package contains a reduced point cloud model of the scene, their 2D coordinates and mean descriptors, visibility matrix and other essential information. This information is used by the registration module to place



**Figure 2: Block diagram of our personal photo browser system**

the user’s photos in 3D space and by the visualization module to create the 3D geometric context.

### 3.2 Registering Photographs

Our goal is to provide a geometry-directed browsing experience to the user. This needs coarse scene geometry; we chose a reduced SfM point set to supply the geometry as it helps us provide the feel of the geometry in a subtle and non-dominant manner. The user photographs are placed accurately (registered) in this SfM point space using a localization module. As a result of localization, we can now provide intuitive geometry directed browsing for the users personal photo collection.

The problem of finding the position and pose of the camera from a photograph of the scene is called the localization problem. Several localization approaches for SfM data have been reported recently [2, 5, 6]. Localization involves identifying a number of known 3D points in the given image and performing camera calibration using them. This is possible since 3D points and their 2D feature descriptions are available from the SfM dataset.

#### 3.2.1 GDBPackage: Compressed SfM Dataset

Localization requires sufficient information about the underlying space which is obtained from SfM reconstruction. Typical reconstructions of monuments run into thousands of images and millions of points, requiring gigabytes of data. It may not be feasible to handle and process such large datasets on the personal devices, which are our target platforms for this photo browser. We therefore propose that the heavy reconstruction and computation of other relevant information about geometric context of monument is done offline for each monument. We also use a lighter representation i.e. the compressed GDBPackage to minimize the pre-computed data that the user needs to download. The data in GDB-Package is similar to what is used by the recent paper [2]. The contents of the GDB package are the following.

1. A list of highly visible 3D points from the SfM dataset. This typically contains 25000 to 50000 points that are visible in most cameras. These are identified using a K-cover algorithm on the complete SfM dataset using the procedure and code provided by the authors of [2].
2. The track information for the points in terms of the bipartite graph of visibility [2, 6], giving for each point

the cameras that see it along with the 2D point and the mean feature descriptor. It also includes for each camera, the points seen in it.

3. The visibility matrix as a representation of the visibility probabilities [2], giving the number of cameras that see each pair of points. This information is needed specifically for localization using the recent method. A suitable representation is used for this very sparse matrix.

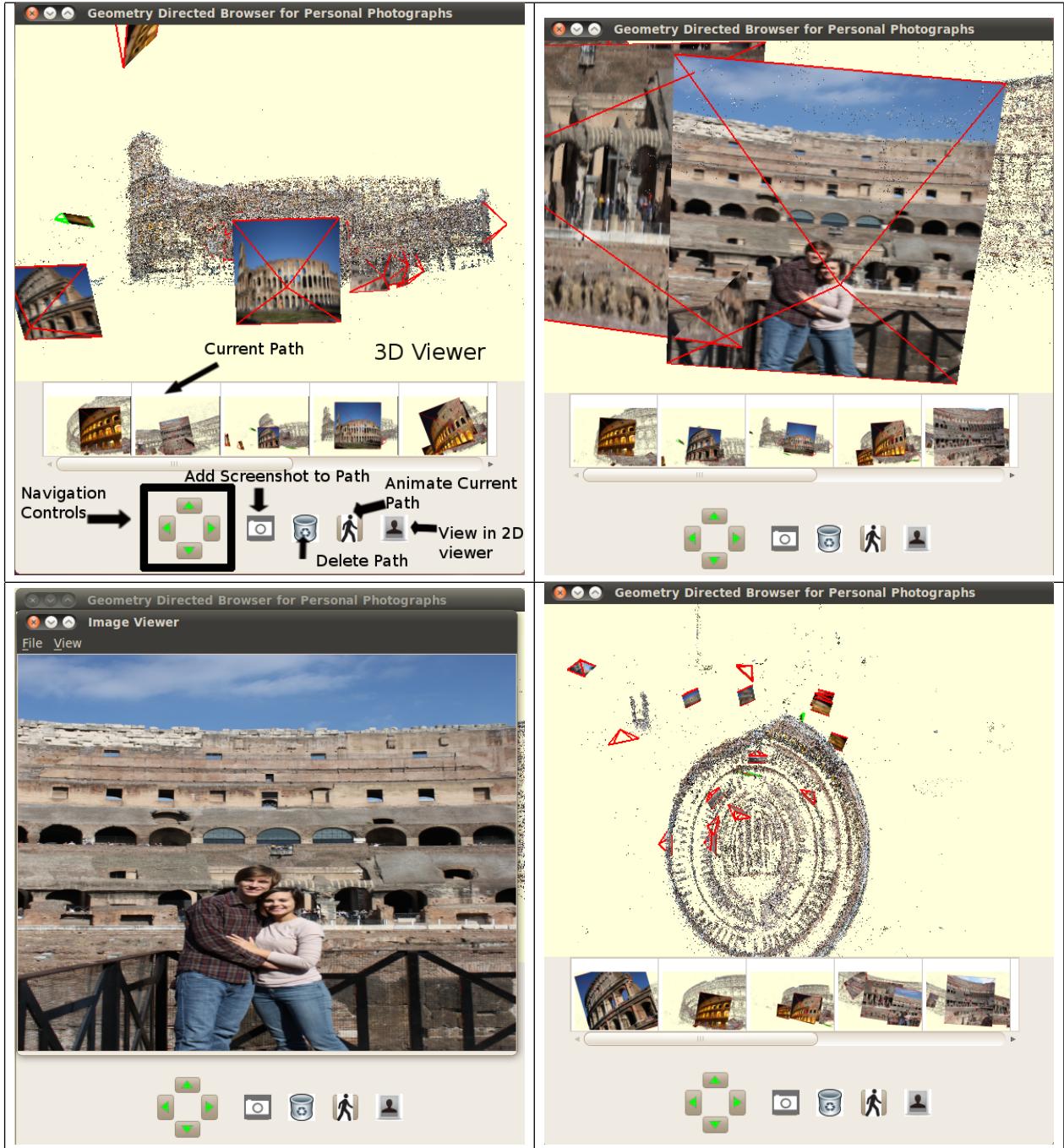
The compressed GDBPackage for a typical location with 50K points will require under 10M bytes for storage. Data to construct GDBPackages for popular tourist locations are either available already or can be computed from community photo collections. A gallery or a store for suitable compressed models for all popular tourist spot can be made available for public use (Figure 1). Tourists can download the data for each site they intend to visit before starting the trip. They upload their photos to the laptop periodically. The registration is then performed on the laptop, creating a representation that is ready for geometry-directed browsing.

#### 3.2.2 Localizing/Registering User Photographs

Localization provides the camera matrix for each photograph uploaded by the user. This process is also referred to as registration of user photographs. Thus, the location  $X_i$  and pose  $R_i$  is known for each photograph  $P_i$  in the monument space. Localization first matches a number of 3D points from the SfM dataset to 2D feature points in each image. If sufficient matches can be found, a standard camera calibration procedure can recover the camera parameters corresponding to the uploaded photograph.

Different localization methods have been proposed before. We use the state-of-the-art method that uses visibility probabilities [2]. This method is fast and hence most suitable to be used on a personal machine. The amount of prior data required by this method is limited to a fraction of reliable points and the visibility matrix. Other localization methods either use the whole dataset or are computationally more involved to be suitable for our goal. We use the code given by the authors of that paper and do not make any modifications to the basic method.

The localization is performed by (a) first identifying a seed point in the new photograph, (b) using visibility probability to guide the search through other points till 25 or so



**Figure 3:** Screenshots of the photo browser when viewing Colosseum photographs taken from personal photo collection of a Flickr user. From the left: (a) Explanation of different controls of the browser; (b) The view of a selected image in the geometric context; (c) The separate window which shows the full resolution image in traditional 2D viewer; (d) Top view of registered images for the colosseum model.

matches are found, and (c) using RANSAC to recover the camera matrix using the matches [2]. In our experience, a 25-photo sequence from a user can be localized at the rate of 1 second per image after the initial setup. The setup time mainly involves reading the geometry data, computing sifts of user photographs and setting up the data structures for the program.

### 3.2.3 Non-Localizable Photographs

Localization works best when many features of the monument space can be matched well in user photographs. Users, however, take photographs where matches may be few. This happens when individuals or groups of people are photographed in and around the space, something we all do. Users may also point the camera away to capture the crowds or the

scenery around the place of interest. This geometric context will not be present in the SfM package and hence such images are *non-localizable*. Since there is no way to find the matrices for them, we guess their positions from other clues so that a reasonable geometric context is available for them also. Modern digital cameras encode the time of image capture in the EXIF tags associated with each image. The time associated with a non-localized image in relation to its temporal neighbors can give a credible estimate of its position. We place the image at a location that is a weighted average of the locations of its immediate known predecessor and successor in time. Let the photograph  $P_i$  be a non-localized one with a capture time of  $t_i$ . Its predecessor  $P_{i-1}$  has a location  $X_{i-1}$  and time  $t_{i-1}$  and its successor  $P_{i+1}$  has a location  $X_{i+1}$  and time  $t_{i+1}$ . We guess the location  $X_i$  of  $P_i$  to be

$$X_i = \frac{X_{i-1}(t_{i+1} - t_i) + X_{i+1}(t_i - t_{i-1})}{t_{i+1} - t_{i-1}}. \quad (1)$$

This may not be the correct location from which the photograph was taken, but it suffices for display purpose as the location will be somewhere in between temporally nearby photographs. The discrepancy is also not very critical in this case as the non-localizable photograph captures less monument geometry. We can also use additional tags on the photograph such as the GPS coordinates, if present. The relative GPS displacement from the temporally neighboring photographs can give an even better estimate than above. The viewing direction can also be found by similar interpolation and thus we can also get an estimate of the pose for non-localizable images.

### 3.3 Browsing Photographs

#### 3.3.1 Navigation Controls

In our photo browser, we display the 3D point cloud of monument along with the registered photos in the *3D Viewer* part of visualization tool, as shown in Figure 3. The 3D point cloud is a result of SfM reconstruction of the monument and can be easily downloaded by the user, as already explained in the previous sections. Note that in this *3D Viewer* along with the 3D points, the cameras of the user's photos are also displayed. Once photos uploaded by the user have been localized, we use information obtained about their camera centres and viewing directions to display the cameras of user photos in the 3D point cloud space of the monument. On the 3D Viewer part, the user can use traditional mouse based 3D navigation controls similar to many computer games using (a) click and drag to rotate the scene, (b) use the scroll bar to zoom in and out. We also provide button based *navigation controls* to do the same task.

#### 3.3.2 Geometry Directed Browsing

Along with each camera, we display a small preview of the photo, analogous to a thumbnail in the traditional photo browsers. Once the user clicks and selects a photo (or thumbnail), we smoothly transition the viewpoint to a position near to the selected photo in the 3D world. We also interpolate the viewing direction along the path such that when we reach the destination point near the photo, the viewing direction of the viewer is nearly same as that of the selected photo in the 3D monument space. Smooth camera transitions give the user a sense of the geometry of the monument

and towards the end the user can see higher dimension and a more detailed version of selected photo. Such a photo, shown in the 3D world helps the user in understanding its relative position and viewing direction with respect to other nearby photos and the overall monument geometry.

We also provide a button to the user, so that she can switch to a standard 2D viewer, if needed, for the selected photo 3(c). This 2D view is included because we can fill up entire pixels on the screen just with the photo in this view and thus the user can look at finer details present therein.

Once a selected photo is displayed, the user is still allowed to navigate the 3D space and can transition to nearby photos by clicking on them. For example, in Figure 3(b) from the selected photo (right) the user can click on the nearby photo (left), the smooth transition makes the relative position and viewing direction between the two images clear. These photographs can be further apart than they are shown in the figure and in such a scenario the user can zoom out and perform the clicking on nearby photo.

#### 3.3.3 Generating Photo-Tours

In addition to the above, we enable the user to define a path through the 3D space in terms of a few key views. These views may involve the photographs taken by the user, but needn't. The selected path can be saved and played back, which creates smooth transitions between key viewpoints. This allows the user to create interesting photo-tours of the monument by choosing his/her own sequence of viewpoints in the 3D monument space. We refer to the viewpoints as screenshots. The user uses the *add screenshot to path* button to select a set of viewpoints one after the other. All the selected screenshots are displayed in the scroll bar at the bottom. After the user is happy with the selection of screenshots, the *animate current path or the walk button* starts the photo-tour. During the photo-tour, we smoothly interpolate the camera through the sequence of chosen screenshots. A video with a demonstration of all these features can be found in the supplementary material. The SfM reconstruction has many registered cameras and there has been previous work in finding natural paths along which a monument has been photographed [10]. If such information is made available in the GDBPackage of the monument, it is also possible to generate a fully automated photo-tour in addition to user generated ones for the personal photo collection.

## 4. RESULTS

### 4.1 Platform Details

We used an off-the-shelf desktop system to do our processing, also our method can be easily be ported to a laptop. The aim was to make geometry directed browsing possible on low end systems available in the public domain. The configuration details of the system we used are given in Table 1.

| Item        | Specification                    |
|-------------|----------------------------------|
| CPU         | Intel(R) Core(TM)2 Duo CPU E7500 |
| Clock Speed | 2.93GHz                          |
| RAM         | 2GB                              |

**Table 1:** Details of the platform used in our experiments.

## 4.2 Test Dataset

Users upload their personal photo collections on Flickr so as to allow easy sharing over the internet. The Flickr API allows us to download any public photos uploaded by the user. We performed a tag based search on the monument name to download sets of photos taken at the monument and later uploaded by various users. We used the meta data such as user id, date and time the photo was taken etc. accompanying these photos to identify personal photo collections of various users at these monuments. We only used data of the past two years (2012, 2011), since our SfM datasets are relatively old and there is less likelihood of these recent photos already being present in the SfM dataset. We ran the registration module to find the correct viewpoints (or cameras) for the photos in personal photo collection, and using this registration information we created an intuitive 3D visualization of the personal photo collection. The details for the test dataset are given in Table 2.

| # Photos<br>(Personal<br>Collection) | Monument      | # Photos<br>Registered | Time(secs)<br>per Photo<br>Registration |
|--------------------------------------|---------------|------------------------|---|
| 24                                   | Colosseum     | 21                     | 1.01                                    |
| 19                                   | Colosseum     | 14                     | 0.97                                    |
| 35                                   | Pantheon      | 22                     | 1.13                                    |
| 17                                   | Stone Chariot | 17                     | 1.07                                    |

**Table 2:** Details of the test dataset used in our experiments.

## 4.3 Screenshots of the Browser

Figure 4 shows some screenshots of the visualization for a personal photo collection of a user who visited the Pantheon. These images were downloaded from Flickr and have been taken by a single user on a single day. We can see from the results that images both in the interior and exterior of the monument can be viewed well in the 3D point cloud space of the monument. Similar results for the Colosseum and Stone Chariot (Hampi) personal photos dataset are show in Figure 5 and 6 respectively.

## 5. CONCLUSION AND FUTURE WORK

In this work, we have presented a visualization system that provides an intuitive photo browsing experience of personal photo collections with an approximate geometry of the underlying 3D space providing the context. We restrict our system requirements to that of a low end system easily available in the market, thus enabling the use of this visualization system in public domain. Mobile phones are equipped with sufficient compute power today and capture good quality photos. Our browser is currently heavy on storage and computation and is suitable for a laptop as designed now. Our future work will involve finding suitable representation, localization, and browsing schemes so as to enable geometry-directed browsing on personal smart-phones.

## 6. ACKNOWLEDGMENTS

We thank the Indian Digital Heritage (IDH) project of Department of Science and Technology, Govt. of India for the partial financial support towards this work.

## 7. REFERENCES

- [1] Photosynth - capture your world in 3d, <http://photosynth.net/>.
- [2] S. Choudhary and P. J. Narayanan. Visibility probability structure from sfm dataset and applications. In *Proceedings of the 12th European conference on Computer vision*, ECCV’12, 2012.
- [3] M. Davis, M. Smith, J. Canny, N. Good, S. King, and R. Janakiraman. Towards context-aware face recognition. In *Proceedings of the 13th annual ACM international conference on Multimedia*, MULTIMEDIA ’05, pages 483–486, 2005.
- [4] J. Hays and A. A. Efros. im2gps: estimating geographic information from a single image. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, CVPR, 2008.
- [5] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, CVPR, 2009.
- [6] Y. Li, N. Snavely, and D. P. Huttenlocher. Location recognition using prioritized feature matching. In *Proceedings of the 11th European conference on Computer vision: Part II*, ECCV’10, pages 791–804, 2010.
- [7] J. Panda, S. Sharma, and C. V. Jawahar. Heritage app: Annotating images on mobile phones. In *Indian Conference on Vision, Graphics and Image Processing*, 2012.
- [8] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *Proceedings of the IEEE International Conf. on Computer Vision*, ICCV, 2011.
- [9] T. Sattler, B. Leibe, and L. Kobbelt. Improving image-based localization by active correspondence search. In *Proceedings of the 12th European conference on Computer vision*, ECCV (1), pages 752–765, 2012.
- [10] N. Snavely, R. Garg, S. M. Seitz, and R. Szeliski. Finding Paths through the World’s Photos. In *ACM Transactions on Graphics (TOG)*, SIGGRAPH ’08, 2008.
- [11] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM Transactions on Graphics (TOG)*, SIGGRAPH ’06, pages 835–846, 2006.
- [12] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008.
- [13] K. Sriyan, S. Ahsan, S. Sinha, and C. V. Jawahar. Image-based walkthroughs from incremental and partial scene reconstructions. In *Proceedings of the British Machine Vision Conference*, BMVC, pages 116.1–116.11, 2010.
- [14] C. Wu. Visualsfm, <http://www.cs.washington.edu/homes/ccwu/vsfm/>.
- [15] L. Zhang, L. Chen, M. Li, and H. Zhang. Automated annotation of human faces in family albums. In *Proceedings of the eleventh ACM international conference on Multimedia*, MULTIMEDIA ’03, pages 355–358, 2003.



Figure 4: Screenshots of our viewer for a dataset of the Pantheon. (a) Correct correspondence between photo and point cloud for facade of the monument (b) The images in the interior are also relevant in the geometric context viz. you can relate the door in image (frontmost) to 3D point cloud (c) Zoomed out view of all images (d) Some non-localizable images, these you can see are close by views and lack sufficient monument geometry.

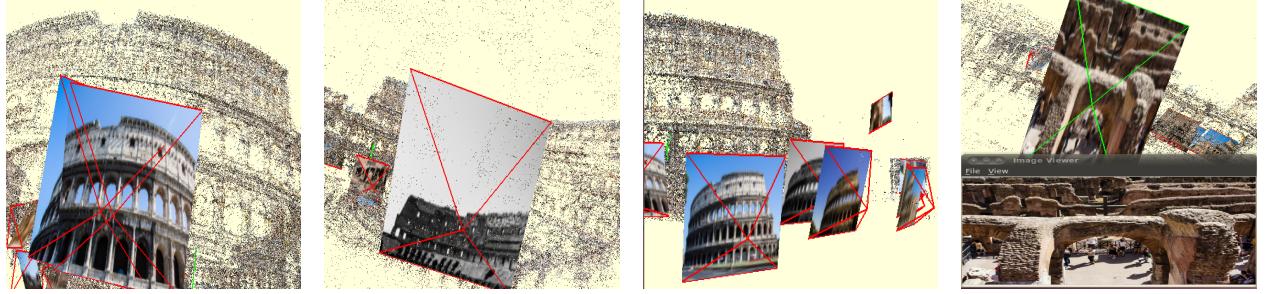


Figure 5: Screenshots of our viewer for a dataset of the Colosseum. (a), (b) Correct correspondence between photo and point cloud for the selected images. (c) A few images viewed in a zoomed out manner in their geometric context (d) A non-localizable image of the given dataset.

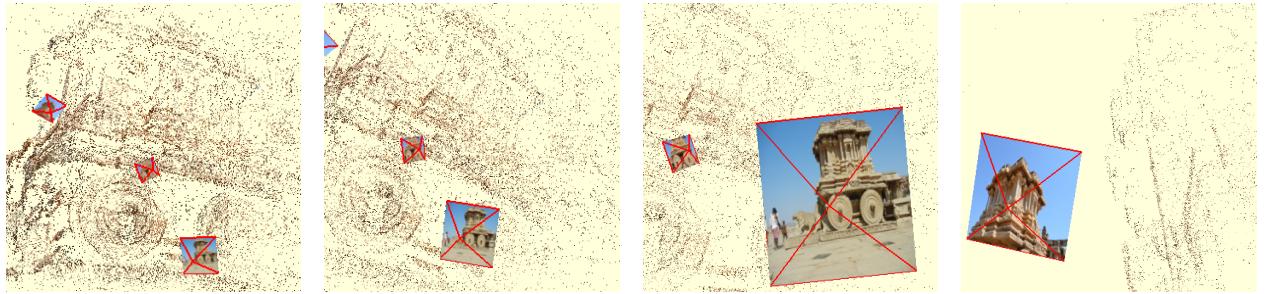


Figure 6: Screenshots of a few localized images out of the 17 image dataset of Stone Chariot, Hampi. We used a set of 217 images collected on our own for reconstructing the point cloud in above screenshots. We localize additional images to this reconstructed space and show the results of visualization through our viewer.