

City-Scale *Reality Modeling* from Community Photo Collection

Pierre Fite-Georgel*

Timothy Johnson

Jan-Michael Frahm

UNC - Chapel Hill

ABSTRACT

Many applications in Augmented Reality (AR) require the use of Reality Models - 3D information tailored for use in AR systems. Available virtual information obtained from CAD models and GIS data is not always trustworthy for registration in AR, as it often has not been verified and might not accurately represent reality. Nowadays, images of landmarks, available on the Internet, allow for automatic 3D modeling, and we propose to deploy these datasets to model the reality. This paper is the first to use city scale models reconstructed from Internet photo collection in an AR framework. Our method augments the information of the generated *Reality Models*, which are typically represented through texture images and 3D geometry (sparse or dense). We automatically annotate the 3D models by analyzing the rich information contained in user provided tags of the original images. Additionally, we introduce an automatic procedure to align new uncalibrated images to our reality model, thus providing a method for the transfer of augmentation from the 3D-model to an image.

1 INTRODUCTION

In Augmented Reality, 3D information can be used for augmentation and for registration. Unfortunately, the data available from 3D CAD models and 2D maps often deviates from the reality [7]. This lack of precision cannot always be compensated for with robust methods and could therefore lead to a wrong visual augmentation. Interactive systems [4, 13] exist to support an operator to verify a 3D model but they are still cumbersome. *Reality Models* have been proposed to overcome the challenges posed by unreliable data [7]. They can be used as reliable sources of 3D information for registration of an AR system. Unfortunately, techniques currently deployed to create these reality models require that an operator manually acquire the data - for example, Klinker et al. use a stereo-system to obtain a dense 3D model [7].

In this paper, we argue that by exploiting the recent progress of computer vision techniques, this manual image acquisition step can be avoided since this data is readily available through pictures on the Internet [15]. We deploy dense 3D models as delivered by [2] for augmentation with textual data from the user provided tags. To this end, we introduce a new registration pipeline to augment uncalibrated cameras using known dense models.

2 RELATED WORK

Our method uses the reconstruction pipeline of Frahm et al. [2] which advances photo collection modeling to reconstruct entire cities from large Internet photo-collections consisting of millions of images (for instance, a search for 'Rome' on the photo sharing website Flickr yields approximately 3 million images) in less than a day on a single PC. Their method deploys binarized GIST image appearance descriptors [17] for appearance clustering. The clusters facilitate the design of a reconstruction method whose complexity is approximately linear in the number of images. This provides up

to three orders of magnitude performance improvement over previous methods [15, 8, 1].

Image based localization is an active research topic in computer vision. Irschara et al. [5], used a bag-of-words approach combined with a vocabulary tree [10] to obtain the pose of a camera with respect to a known model. It requires a time and memory intensive offline process to create the search index and the inverted file. Li et al. [9] propose to improve bag of words approaches by prioritizing highly repetitive points of a set of registered images. In contrast we demonstrate that a compressed version of GIST as used in Frahm et al. [2] performs comparably well at a fraction of the computational cost, thus facilitating real-time operation.

Internet photo collections provide tags for a majority of the images which in some cases are even localized to specific image areas. The Photo-tourism system [15] describes a method to transfer these tags between images using multi-view geometry and occlusion detection to map the tags across views. Simon et al. [14] extend this to cluster sets of 3D points based on the cameras field of view, in order to automatically label 3D geometry using a complex graph optimization.

3 3D MODELING PIPELINE

In this section, we briefly describe the methods used to generate 3D models [2, 6]. Each image is described using a binarized GIST descriptor [11], which acts as a signature for the image. To group viewpoints the descriptors are clustered using a variant of the k-means algorithm. The obtained clusters are then verified for geometric consistency. Following verification based on the pairwise epipolar geometry, an "iconic" image is defined as the representative for each cluster [8]. This effectively reduces the redundancy of the data and the next steps can focus on the chosen iconics.

Following the iconic selection step, a set of overlapping iconics have to be identified to detect scene overlap. To avoid the prohibitively expensive exhaustive matching, the method deploys user provided geo-tags and appearance similarity by testing overlap between co-located iconics and iconics similar in appearance. This gives consistent connected sets of iconics that represent the different sites. Following this, a structure from motion process is performed using incremental bundle adjustment. Once the images are aligned, we compute multiple depth maps using a multi-view stereo plane sweep. The resulting depth maps are then fused to obtain dense models using a multilayer height-map [3].

These steps outlined above yield highly efficient system that leverages the inherent parallelism and redundancy in the data, in order to achieve a computational advantage. This parallelism enables the extensive use of commodity graphics hardware to improve computational efficiency. The system described in [2] is capable of processing almost 3 million images for a city like Berlin or Rome, in less than 24 hours using a single PC with two quad core CPUs and four Nvidia GTX 295 graphics cards.

4 TAGGING REALITY MODEL

Meta-information can be added to reality models that have been created using Internet photo collections. For example, models can be geo-localized [16] using the geotag attached the image. The textual tags that are included with the images can also be used to add information to the model. Tags, while a good source of additional

*e-mail: {george,tjohnson,jmf}@cs.unc.edu



Figure 1: **Dense reconstruction** of five landmarks in Berlin. From left to right: the Brandenburg gate, the Reichstag (parliament), Berliner Dom (cathedral), the Ishtar Gate and the ruins of Kaiser Wilhelm Church.

information, also contain a significant level of noise. We show in this section that a voting scheme can be used to obtain the information and suppress noisy tags. This new information augments the reality models and can be transferred to the user to improve his experience in an AR world.

When we download the images, in addition to the image files, we download meta-information such as acquisition date, upload date, type of license, focal length, geotags, and textual tags. On an individual image basis, these tags offer little information. For example, an image from the Brandenburg gate model has the tags: berlin, mitte, museumsinsel, museumisland, constructionsite, nightlights, brandenburggate, brandenburgertor, pariserplatz, unterdenlinden. Though these tags include some correct information for the gate, there is a fair amount of wrong information (e.g. museumisland). The goal is to thus filter out this “noisy” information.

First, a set of candidate tags is obtained by gathering all the tags present in all the images registered in a model. We then compute a proper description of the model by using a voting scheme with thresholding. Finally, trivial words are removed. For example, in the case of the Berlin dataset, we exclude Germany, Deutschland, Aeminia (Germany in Italian), Berlin, and Europe.

As a demonstration, here are the results of the 5 best tags from each model, along with the number of votes for each tag.

- **Brandenburg Gate:** brandenburggate (451), brandenburger-tor (437), gate (324), brandenburg (293), and tor (274).
- **Reichstag:** Reichstag (655), bundestag (126), parliament (91), architecture (69), and building (59).
- **Berliner Dom:** dom (146), berlinerdom (136), cathedral (106), church (72), and berliner (52).
- **Ishtar Gate:** pergamon (50), museum (48), pergamonmuseum (38), ishtargate (19), and babylon (14).
- **Kaiser Wilhelm Church** church (73), kaiserwilhelmgedächtniskirche¹ (30), gedächtniskirche (21), travel (18), and kaiser (16).

This attached data can then be used as a source of information for augmentation.

5 AR APPLICATIONS USING REALITY MODELS

In this section, we propose a method which uses the sparse and the dense scene model to efficiently augment the models. Our novel method conserves computational resources during the augmentation by only requiring the augmentation information and the binarized GIST in memory.

The dense reality models can be used to obtain a 3D pose, which can then be used for augmentation. However, we propose a pipeline

¹Gedächtniskirche: memorial church



Figure 3: **Feature Selection** in red are matches validated by the fundamental matrix F and the white circle are the one selected from the augmentation plane (white quadrangle).

that does not compute a full pose between the model and the camera to create an augmentation. Instead, with a view towards efficiency, we search for corresponding images in the set of pictures that were used to create the model (c.f. section 5.1). The fundamental matrix between the two images is then used to transfer an augmentation from the model to the image (c.f. section 5.2).

5.1 Registration to a Reality Model

When trying to register an image to a reality model, we first describe it using a binarized GIST descriptor. The descriptor is used to find the nearest neighbors in the reality model. The use of global image appearance descriptors serves to select a nearby viewpoint captured under similar illumination conditions, thus potentially allowing for illumination and location sensitive annotations. These candidate matches can be verified using SIFT followed by RANSAC. The model verified by the RANSAC is a fundamental matrix F . If the image is matched to one of the candidates (henceforth called a keyframe), the process is stopped as we can already relate the image to the model.

This method can be used not only to match an image to a single model, but it can be applied to a city-scale reality model. The image is matched to each model included in the city-scale reconstruction. Every model is described by its image binarized GIST. This is a memory efficient procedure, since each image is described by a 512 bit code. The Hamming distance can be used to prioritize the matching and improve the performance of the system.

In the following section, we show that only a fundamental matrix is necessary to transfer an augmentation from the model to the image.

5.2 Augmentation using a Reality Model

In order to transfer an augmentation from the model to an uncalibrated image, we use the available dense model. Using the dense geometry, we can define a planar surface around the area to augment, which allows us to define a homographic warp to transfer the

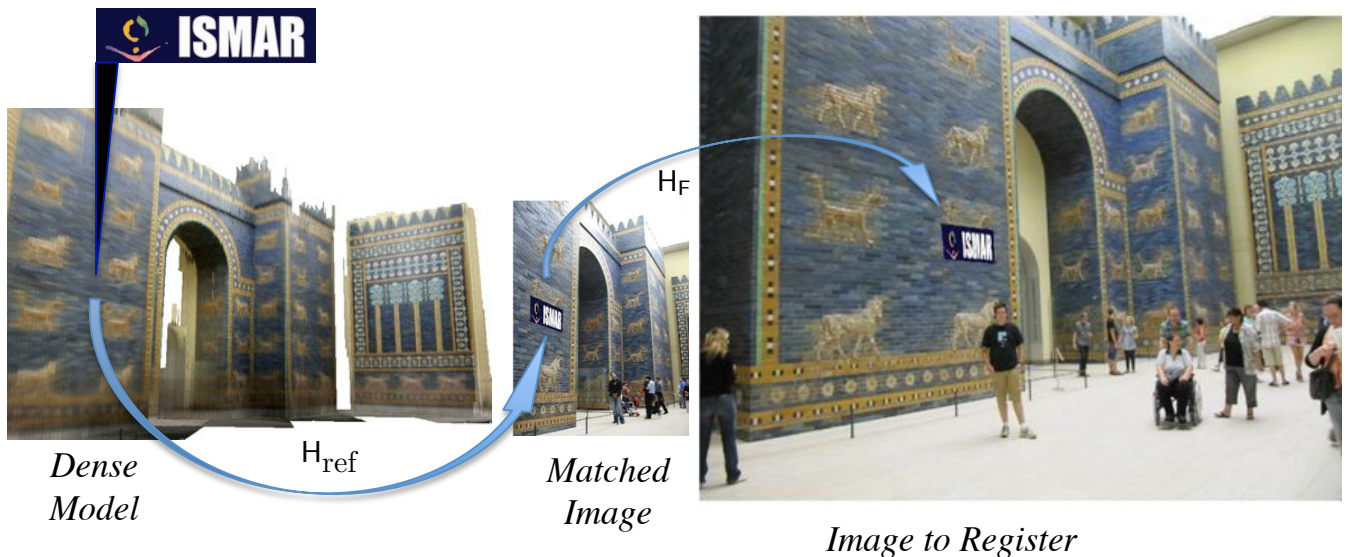


Figure 2: **Augmentation Scheme** A label to be augmented is attached to the model, its location is warped using H_{ref} to the keyframe to describe an area and a related homography H_F between the images.

augmentation. This is the only part of the approach that requires an access to the dense model and can easily be computed and stored beforehand. The planar surface that is defined by the augmentation location in the dense model is described with a normal vector \mathbf{n} and a distance d . Using the normal, we automatically compute the homography (H_{ref}) between the dense model and the keyframe by using the pose of the camera and the normal to the model around the location to be augmented in the model, as follows:

$$H_{ref} = K \left(R - \frac{\mathbf{nt}^T}{d} \right), \quad (1)$$

with K the matrix of internal parameters of the keyframe, R its rotation, and \mathbf{t} its translation.

This homography can be used to describe an area in the keyframe that acts as a support to compute the final warp. We now select the subset of feature matches (between the keyframe and the images) that are included in this area, as shown in figure 3. Using this subset, we can compute a homography H_F that describes the underlying 3D plane. The calculation is done using a direct linear transform.

By composing the two homographies ($H = H_{ref}H_F$), we obtain a mapping from the model to the image. This allows for an automatic augmentation without requiring any camera calibration for the new image, as shown in Figure 2.

6 EXPERIMENTAL RESULTS

In this section, we present registration results obtained using 5 different models of the city Berlin, see Figure 1. For each of the scenes represented by one of the models, we selected images which were not used for their creations in order to verify the proposed method for registration. We try to match each candidate image to every available model. For these experiments, we search 10 nearest neighbors and for the RANSAC we assume a lower bound of 50% on the inlier ratio, in order to guarantee a limited number of iterations. The results of these experiments are visible in Figure 4. All the images were registered to their corresponding models and no mis-registration happened. The run-time of the approach is limited by the extraction of GPU-SIFT, which is running at 10 Hz on an Nvidia Geforce GTX 295. The experiments were run on a single

CPU and GPU thread. The rest of the pipeline operates in real-time, as reported in [6].

7 FUTURE WORK

In this paper, we are currently using all images registered to the model which might include redundant information. It would be interesting to see the impact of generating a smaller set of images to uniformly cover the model, maybe by computing a uniform distribution in the GIST space. Additionally, we could imagine using non-scale invariant features to speed up the registration process as the images available in the model already cover the model at different scales.

8 CONCLUSION

In this paper, we presented a method to attach information to reality models, using information freely available from community photo-collection websites. This additional information can be transferred to new models. We also demonstrated the usability of a lightweight registration pipeline based on binary gist and presented an augmentation framework that works for un-calibrated cameras. Using this type of dense model, we can finally (AR) graffiti [12] historical sites without getting fined!

ACKNOWLEDGEMENTS

The authors wish to thank B. Clipp, E. Dunn, D. Gallup, Y. Jen, S. Lazebnik, M. Pollefeys, R. Raguram, and C. Wu for their help. This work was supported in part by Nvidia, NSF grant IIS-0916829, DOE grant DE-FG52-08NA28778.

REFERENCES

- [1] S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski. Building rome in a day. *International Conference on Computer Vision (ICCV)*, pages 1–8, Jul 2009.
- [2] J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys. Building rome on a cloudless day. *ECCV*, pages 1–14, Jun 2010.
- [3] D. Gallup, M. Pollefeys, and J.-M. Frahm. 3d reconstruction using an n-layer heightmap. In *Deutsche Arbeitsgemeinschaft für Mustererkennung (DAGM)*, 2010.



Figure 4: **Matching results.** Binary GIST descriptors are used to find neighboring images in the reality models, the resulting candidates are confirmed using SIFT and RANSAC.

- [4] P. Georgel, P. Schroeder, S. Benhimane, S. Hinterstoisser, M. Appel, and N. Navab. An industrial augmented reality solution for discrepancy check. *IEEE and ACM International Symposium on Mixed Augmented Reality (ISMAR)*, page 4, Sep 2007.
- [5] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. *CVPR*, pages 1–8, Mar 2009.
- [6] T. Johnson, P. Fite-Georgel, R. Raguram, and J.-M. Frahm. Fast organization of internet photo collections using cuda. In *CVGPU (ECCV Workshop)*, 2010.
- [7] G. Klinker, D. Stricker, and D. Reiners. The use of reality models in augmented reality applications. *LECTURE NOTES IN COMPUTER SCIENCE*, Jan 1998.
- [8] X. Li, C. Wu, C. Zach, S. Lazebnik, and J.-M. Frahm. Modeling and recognition of landmark image collections using iconic scene graphs. *ECCV*, pages 1–14, Jul 2008.
- [9] Y. Li, N. Snavely, and D. P. Huttenlocher. Location recognition using prioritized feature matching. *ECCV*, pages 1–14, Sep 2010.
- [10] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. *CVPR*, pages 1–8, Apr 2006.
- [11] M. Raginsky and S. Lazebnik. Locality sensitive binary codes from shift-invariant kernels. In *NIPS*, 2009.
- [12] J. Rekimoto, Y. Ayatsuka, and K. Hayashi. Augment-able reality: Situated communication through physical and digital spaces. *ISWC*, May 1998.
- [13] R. Schoenfelder and D. Schmalstieg. Augmented reality for industrial building acceptance. *IEEE Virtual Reality Conference (IEEE VR)*, pages 83–90, 2008.
- [14] I. Simon and S. Seitz. Scene segmentation using the wisdom of crowds. *ECCV*, 2:541–553, Aug 2008.
- [15] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. *ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 835–846, 2006.
- [16] C. Strecha, T. Pylvanainen, and P. Fua. Dynamic and scalable large scale image reconstruction. *CVPR*, pages 1–8, Mar 2010.
- [17] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large databases for recognition. In *CVPR*, 2008.