Geo-registered 3D Models from Crowdsourced Image Collections

Jan-Michael Frahm[1], Jared Heinly[1], Enliang Zheng[1], Enrique Dunn[1], Pierre Georgel[3], and Marc Pollefeys[1,2]

[1] University of North Carolina *at* Chapel Hill
[2]Eidgenössische Technische Hochschule Zürich
[3]Dekko Inc.

Abstract

In this paper we present our system for scalable, robust, and fast city-scale reconstruction from Internet photo collections obtaining geo-registered dense 3D models. The major achievements of our system are the efficient use of coarse appearance descriptors combined with strong geometric constraints to reduce the computational complexity of the image overlap search. This unique combination of recognition and geometric constraints allows our method to reduce from quadratic complexity in the number of images to almost linear complexity in the Internet photo collections size. Accordingly, our 3D modeling framework is inherently better scalable than other state of the art methods and in fact is currently the only method to support modeling from millions of images. In addition, we propose a novel mechanism to overcome the inherent scale ambiguity of the reconstructed models by exploiting geo-tags of the Internet photo collection images and readily available StreetView panoramas for fully automatic geo-registration of the 3D model. Moreover, our system also exploits image appearance clustering to tackle the challenge of computing dense 3D models from an image collection that has significant variation in illumination between images along with a wide variety of sensors and their associated different radiometric camera parameters. Our algorithm exploits the redundancy of the data to suppress estimation noise through a novel depth map fusion. The fusion simultaneously exploits surface and free space constraints during the fusion of a large number of depth maps. Cost volume compression during the fusion achieves lower memory requirements for high-resolution models. We demonstrate our system on a variety of scenes from an Internet photo collection of Berlin containing almost three million images from which we compute dense models in less than the span of a day on a single PC.

1 Introduction

City models from aerial images have recently been commercially introduced. The biggest limitation is that when observed from the ground the texture and geometry have very limited resolution. The next generation of 3D models will need to employ ground based imagery to overcome this limitation. To obtain those models, we have proposed the first real-time 3D reconstruction system, which collects ground reconnaissance video [12] in the range of multiple million frames for small cities and computes 3D models. Alternatively, researchers proposed to use crowd sourced Internet photo collections to avoid the image collection effort required for city-scale 3D modeling [1, 3, 9, 16]. These Internet Photo Collections (IPC) typically include several million images for a city and cover the sites of interest in the city. The major challenge added compared to video is there no information about the spatial ordering of the views is provided, i.e. there is no information about which views overlap with each other. The massive amount of unordered image data requires one to solve the overlap search efficiently to obtain a highly scalable 3D reconstruction system that meets the demands of 3D modeling from millions of ground reconnaissance images.

In this paper we discuss our system for city-scale 3D reconstruction from crowd sourced photo collections, which can compute city-scale 3D models from three million images on a single PC in less than the span of a day. Our method is designed to reduce the computational complexity of the reconstruction problem to close to linear for the most expensive parts of the reconstruction pipeline. We achieve this by jointly using geometric constraints and recognition based constraints allowing for a highly efficient method. In contrast to most other crowd based 3D modeling algorithms our system efficiently solves the high resolution dense reconstruction problem by exploiting the redundancy in the estimated multi-view geometries enforcing surface and free-space constraints. Additionally, the proposed method is able to use readily available geo-registered street view imagery (for example, those provided by Google StreetView) to automatically scale and geo-reference the computed 3D city models. In summary our method is currently outperforming the state of the art methods in crowd sourced 3D reconstruction by three orders of magnitude in performance while in contrast to those methods also solving the dense reconstruction problem.

2 Related Work

There are currently two main classes of city scale reconstruction algorithms, the first uses bag of words methods to identify scene overlap, which is then used to bootstrap large-scale structure from motion registration delivering camera registration and sparse 3D point clouds of the scene [1, 16]. Even with a cloud computer of 62 CPUs these methods only scale to the registration of a few hundred thousand images within 24 hours, not meeting the demand of true Internet scale photo collections. The second class of methods uses appearance based image grouping followed by epipolar geome-

try verification to identify overlapping images. A set of identified characteristic views (iconics) is afterwards exploited to bootstrap the efficient image registration through structure from motion [3, 9]. These methods can scale to the processing of a few million images on a single PC within 24 hours as shown in our work [3].

Bag of words methods typically provide a higher degree of resulting 3D model completeness in the reconstruction than the appearance clustering approaches. In order to achieve higher completeness they compromise the computational complexity of the method. The first approach in this category was the seminal PhotoTourism [16] method, which deployed exhaustive search for the overlap search. Agarwal et al. [1] employed feature based recognition and query expansion to improve the computational complexity of the overlap detection, scaling to the processing of a few hundred images on a cloud computer (62 CPUs). In contrast, our method scales to millions of images on a single PC while maintaining model quality and producing dense 3D models.

Appearance based clustering has the strong advantage of scalability, which has been shown in our work [3] to outperform the most efficient bag of word approach [1] by at least three orders of magnitude. This is a result of the close to linear computational complexity of the method introduced in [3]. Additionally, we deliver the largest so far obtained 3D models from IPCs.

After the appearance grouping for overlap search, both types of methods use traditional structure from motion techniques to solve for camera registration [7, 13]. They exploit error mitigation through increasingly larger bundle adjustment, which optimizes the 3D structure simultaneously with the pose of the cameras [1, 3, 16, 17]. This step is one of the remaining performance bottlenecks in the large-scale reconstruction systems, despite the recent progress in the computational performance of bundle adjustment through parallelization [19] or hierarchical processing as proposed by Ni et al. [10]. Alternatively, Crandall et al. [2] proposed to use a discrete continuous method to limit the number of bundle adjustments. However, it requires a computationally expensive discrete optimization partially executed on a 200 core cluster to initialize the continuous optimizer. Using this method limits the drift drastically and allows for large-scale reconstruction. Snavely et al. [17] propose to construct a minimal set of views for reconstruction (skeletal set), which provides the same reconstruction accuracy. The major drawback for the computational complexity with the skeletal set is that it still requires a full pair-wise matching of the image connection graph. In contrast, our approach utilizes high-level appearance information to reduce the inherent redundancy in the data by obtaining an iconic scene representation of the photo collection [3, 9].

Goesele et al. [6] use a patch growing to obtain a dense 3D model from IPC data. They start from the sparse structure from motion point cloud. Then, the participating views for each patch are carefully selected based on resolution, color, and viewing direction. Furukawa et al. [4] proposed another view selection approach for the computation of the dense 3D point cloud. Both Goesele et al. [6] and Furukawa et al. [4] are computationally highly demanding compared to our method. We extend our previous approach from Gallup et al. [5], which achieves scalable reconstruction for a limited

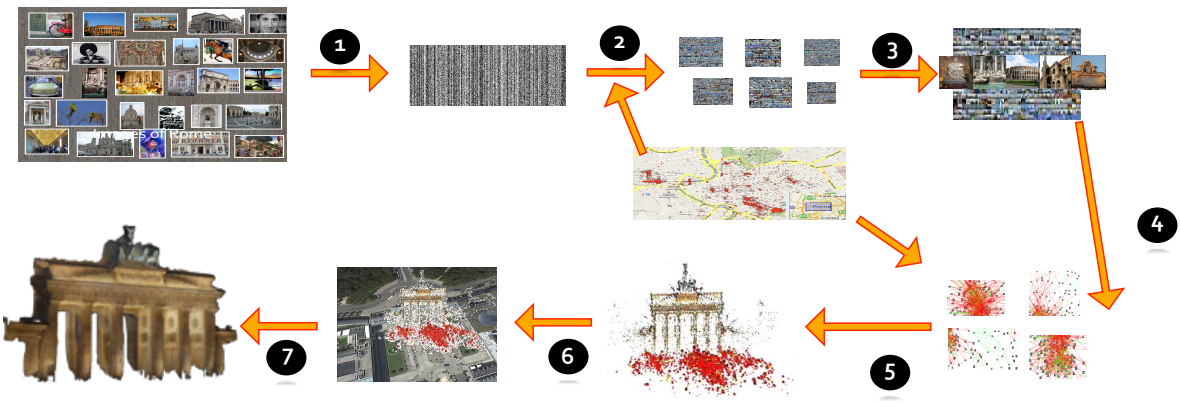

Figure 1: Overview of the processing steps.

number of views by decoupling the reconstructions in two of the three dimensions of the 3D space. Our extension overcomes the limitation of Gallup et al. [5] to only be able to use a limited number of views for the dense computation.

3 Scalable Crowd Sourced Modeling

Our 3D modeling technique consequently limits the computational complexity of each step of the system. This leads to an overall computational complexity that is close to linear in the initial registration process, which involves millions of images for a typical city-scale reconstruction. An overview of the steps of our method is shown in Figure 1. We evaluate the method on an IPC from Berlin that contains approximately 2.8 million images downloaded from Flickr. All reported execution times are on our PC with dual quadcore Xeon 3.33 Ghz processors, four NVidia 295GTX commodity graphics cards, and 48 GB RAM.

Appearance based grouping (Steps ❶-❷) Our approach uses appearance clustering for determining overlapping views. To facilitate the clustering we exploit the *gist*-descriptor [11] to obtain a compact 368 dimensional vector describing each image. This descriptor mostly depends on the edge structure of the image, the texture roughness of the image, and the global color distribution in the image. Hence, it will not change with small viewpoint changes but it will significantly change for scene changes and wide baseline viewpoints. Using this insight, our method clusters the gist descriptors of the images, which equates to a viewpoint clustering. For the Berlin dataset we obtain about 4GB of descriptors compared to the IPC of approximately 650 GB.

Given that clustering is an inherently parallel problem it would be optimal to execute it on commodity graphics hardware (GPUs), which are highly parallel. Even the 4GB of gist descriptors exceed the memory of typical graphics hardware. To ob-

Figure 2: Left: Appearance cluster from the Berlin dataset. Right: Geometrically verified cluster from Berlin dataset.

tain a more compressed representation we apply a random basis projection followed by a binarization [14]. This guarantees that with an increasing number of bits the Hamming distance approximates the Euclidian distance of the original descriptors. The best tradeoff between approximation accuracy and memory usage for our datasets was found at 512 bits. To cluster using the Hamming distance we employ a k-medoids algorithm on GPU, which is a k-means for non-metric spaces.

For a better cluster initialization we approximate the spatial distribution of the geotagged images with our cluster initialization. As shown in our work [3] this increases the number of registered cameras by 20% compared to a random initialization. Figure 2 illustrates an example viewpoint cluster. The advantage of our appearance clustering approach is that it has in practice linear complexity in the number of images.

Geometric cluster verification (Step ❸) Due to the coarseness of the 512 bit binary descriptors the above appearance clustering leads to noisy clusters. To remove the unrelated views we enforce a valid epipolar geometry between the images of a cluster. A naive verification strategy would evaluate all view pairs within a cluster, which is of quadratic complexity in the number of images. To avoid the combinatorial explosion, our method first identifies a set of n mutually consistent views ($n = 3$ for all results in the paper). Then, it selects the iconic view as the view that has the most correspondences to the other views. All remaining images are only verified against the iconic. For high verification performance we exploit USAC, a fast RANSAC scheme [15] leading to verification rates of approximately 450Hz. Figure 2 shows the verified clusters. The iconic of each cluster is then used in the following computation to represent the cluster for the initial registration.

Camera registration (Steps ❹-❺) After identifying all iconic views our method establishes the registration between the different iconics. Li et al. [9] introduced the iconic scene graph to represent the relationships between the iconic views. The graph has an edge between two iconics if there is a valid epipolar geometry between the two images. We take the concept of the iconic scene graph, which was designed for landmarks where all iconics show the landmark and extend it to city scale modeling by establishing local iconic scene graphs for each separate site in the dataset. To compute the local iconic scene graph we test for mutual overlap between the iconics by enforcing the epipolar geometry with USAC [15]. To avoid the quadratic complexity of an exhaustive test we use the binary descriptors from step ❶ by testing each iconic

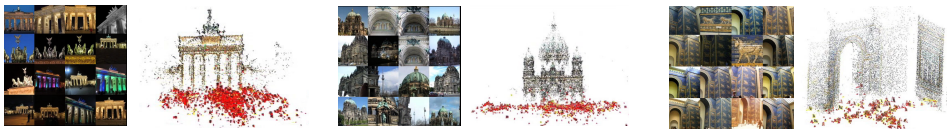

Figure 3: Example iconics and registered cameras for three different sites in Berlin: Brandenburg Gate, Berlin Dome, Ishtar Gate.

for potential overlap with the $k = 10$ nearest neighbors in the binary descriptor space. To foster further connections we exploit the spatial location of the cluster by verifying potential connections to other iconics in the spatial vicinity. The location of the iconic is obtained through kernel voting using the geo-located images within the cluster.

Then, we perform structure from motion combined with bundle adjustment for each local iconic scene graph. Next, all images of all clusters will be registered into the reconstruction. Example registrations and iconics are shown in Figure 3.

Geo-registration (Step ❻) Given the inherent scale ambiguity for structure from motion we propose a scale estimation and geo-registration. We empirically observed that typically there are very few precisely geo-localized photos in the IPC, which often prohibits accurate geo-registration except for some large-scale models as shown by Kaminsky et al. [8]. Hence, we exploit the embedded tags (automatic and user clicked) only to obtain a rough position through kernel voting, where each geo-localized image casts a Gaussian distributed vote with $\sigma = 8.3$m and a three σ cutoff distance. The rough estimate for the model location is then obtained as the center of gravity of the largest connected component. In the rare case of no registered geo-located images in the IPC we propose to use the text tags of the IPC images to obtain a rough geo-location through a search on Google maps.

To support more accurate geo-location our algorithm employs ubiquitously available Google StreetView panoramas in the model's vicinity, which have high accuracy geo-location and orientation[1]. Given that our obtained camera registration is in a Euclidean space we transform the latitude/longitude coordinates of the StreetView panoramas into the Universal Transverse Mercator (UTM) grid, which provides a Euclidean space too and allows the use of Euclidean distances during the alignment optimization.

Next, we perform a feature based registration of the StreetView panoramas into our 3D model using essentially the registration process of step ❺ with a viewing ray based panorama registration. After the registration of the panoramas into the 3D model we can utilize their coordinates in the 3D model and their known UTM coordinates to estimate the transformation between the 3D model coordinate system and the UTM coordinate system with USAC. Figure 4 illustrates a geo-registered reconstruction.

[1]The panoramas are automatically downloaded through the Google StreetView API (http://code.google.com/apis/maps/index.html)

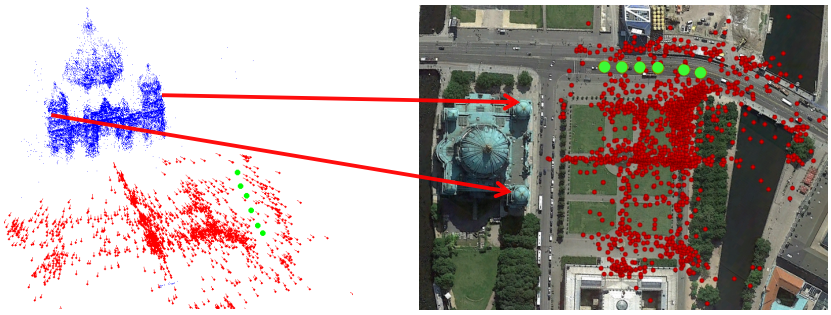

Figure 4: Geo-registered cameras for the Berlin dome. StreetView panoramas in green and IPC cameras are in red.

Figure 5: Example cluster images used for stereo computation and their depth maps

Dense Scene Modeling (Step ❼) After obtaining the geo-registered cameras our method computes a dense 3D model for each site. The major challenge for IPCs is the varying appearance of the scene in the different views, which poses significant challenges for the image based correlation. We exploit our gist-based clustering to overcome this challenge as it groups the images not only by viewpoint but also by color. Hence, the resulting clusters are similar in color and allow for a GPU based normalized cross correlation plane sweeping stereo algorithm to be executed. Example cluster images and their depth maps are shown in Figure 5. As a result of the varying appearance, the resulting depth maps contain significant noise. To combat the noise for the final 3D model we use the redundancy of the depth maps by fusing them into a mutually consistent 3D model.

Our fusion method [5, 20] uses a volumetric representation for the scene to enable the fusion of a large variety of viewpoints. Volumetric methods have the inherent cubic memory complexity, which prohibits high-resolution models. In contrast to traditional volumetric models, our fusion [20] only exercises quadratic memory complexity. Hence, it is able to fuse the depth maps of each site into a high-quality 3D model representation.

Our heightmap fusion [5, 20] takes a set of depth maps along with the external and internal camera parameters as input. From the camera parameters we automatically extract the ground plane using a technique similar to Szeliski's method [18], which fits a plane through the x-axes of the landscape mode cameras and the y-axes of the portrait mode cameras. The ground plane then serves as a basis for the heightmap representation with the height direction along the normal of the estimated ground plane. We chose a quantization grid for the ground plane to effectively define the ground sampling distance

Figure 6: Fused 3D models

of our obtained model. Given that our model is correctly scaled we can set the ground sampling distance directly in units of meter. We typically chose a value between 5 cm and 30 cm.

For each column (volume above the area of a ground sampling point) we then perform a voting process for its occupancy in the height direction using the same quantization as the ground sampling distance. In the voting, each pixel in each depth map votes for an occupied cell (voxel in the column) at the depth of the observed surface. For cells between the camera and the occupied cell it votes for empty and for all cells behind the observed surface its vote exponentially declines from occupied to a indecisive vote. After the votes of a column have been accumulated our technique determines the occupied segments within the column using a regularization to minimize the number of occupied segments (for more details please see Gallup et al. [5]). Please note that our method not only uses the surface constraints in the voting but also enforces free space constraints, which we found to be highly efficient to suppress noise in the fused model. We execute this process on commodity graphics hardware to exploit its given parallelism. This fusion process [5] is limited by the memory, i. e. it can only fuse a limited number of depth maps per column. Naively this can be overcome by either streaming all the depth maps from the hard disk for each column, which is prohibitively expensive in time to due disk bandwidth, or alternatively the cost volume has to be kept in memory to avoid streaming the depth maps. The later is prohibitive for high resolution models due to the cubic memory complexity.

To enable the fusion of all available depth maps our method [20] exploits Haar-wavelet compression for the cost volume, which accurately preserves the transitions between occupied and empty segments along the column. This enables the algorithm to present the cost function of each column to be represented with a constant number of coefficients k. We empirically found k=30 to perform well and used this for all our experiments. Accordingly, the memory to store the cost function only grows quadratically with the ground sampling distance allowing our method to perform high-resolution 3D modeling on commodity graphics hardware using all available depth maps as described in Zheng et al. [20]. Figure 6 shows some of the resulting models for the Berlin dataset. The overall processing time of the IPC was 23 hours and 35 minutes.

4 Conclusion

In this paper we introduced our algorithm for the computation of geo-registered dense 3D models from Internet Photo Collections, which outperforms any existing method for reconstruction of sparse models by at least three orders of magnitude. The performance and scalability of our system results from a consequent reduction of computational complexity using geometric and recognition constraints. Our technique also solves the automatic geo-registration by exploiting the readily available StreetView imagery along with its precise location information. Our novel depth map fusion is able to fuse the information from all available depth maps of a site and hence allows us to obtain highly detailed dense noise free 3D scene models.

References

[1] Sameer Agarwal, Noah Snavely, Ian Simon, Steven Seitz, and Richard Szeliski. Building rome in a day. *IEEE International Conference on Computer Vision (ICCV)*, pages 1–8, Jul 2009.

[2] D Crandall, A Owens, N Snavely, and D Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. *CVPR*, 2011.

[3] Jan-Michael Frahm, Pierre Fite-Georgel, David Gallup, Timothy Johnson, Rahul Raguram, Changchang Wu, Yi-Hung Jen, Enrique Dunn, Brian Clipp, Svetlana Lazebnik, and Marc Pollefeys. Building rome on a cloudless day. *European Conference on Computer Vision (ECCV)*, pages 1–14, Jun 2010.

[4] Yasutaka Furukawa, Brian Curless, Steven M. Seitz, and Richard Szeliski. Towards internet-scale multi-view stereo. In *CVPR*, 2010.

[5] D. Gallup, M. Pollefeys, and J.M. Frahm. 3d reconstruction using an n-layer heightmap. *Pattern Recognition*, pages 1–10, 2010.

[6] Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M. Seitz. Multi-view stereo for community photo collections. In *ICCV*, 2007.

[7] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[8] R.S. Kaminsky, N. Snavely, S.M. Seitz, and R. Szeliski. Alignment of 3d point clouds to overhead images. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pages 63–70. IEEE, 2009.

[9] Xiaowei Li, Changchang Wu, Christopher Zach, Svetlana Lazebnik, and Jan-Michael Frahm. Modeling and recognition of landmark image collections using iconic scene graphs. *European Conference on Computer Vision (ECCV)*, pages 1–14, Jul 2008.

[10] Kai Ni, Drew Steedly, and Frank Dellaert. Out-of-core bundle adjustment for large-scale 3d reconstruction. *IEEE International Conference on Computer Vision (ICCV)*, 2007. Notes Large Structure from motion.

[11] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.

[12] M. Pollefeys, D. Nistér, J.M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.J. Kim, P. Merrell, et al. Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision*, 78(2):143–167, 2008.

[13] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, 2004.

[14] M. Raginsky and S. Lazebnik. Locality sensitive binary codes from shift-invariant kernels. In *NIPS*, 2009.

[15] Rahul Raguram, Ondrej Chum, Marc Pollefeys, Jiri Matas, and Jan-Michael Frahm. USAC: A Universal Framework for Random Sample Consensus'. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[16] Noah Snavely, Steven Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. *ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 835–846, 2006.

[17] Noah Snavely, Steven Seitz, and Richard Szeliski. Skeletal graphs for efficient structure from motion. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–11, May 2008.

[18] Richard Szeliski. Image alignment and stitching: A tutorial. *Foundations and Trends in Computer Graphics and Computer Vision*, 2:1–104, 2006.

[19] C. Wu, S. Agarwal, B. Curless, and S.M. Seitz. Multicore bundle adjustment. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3057–3064. IEEE, 2011.

[20] E. Zheng, E. Dunn, R. Raguram, and J.M. Frahm. Efficient and scalable depthmap fusion. *British Machine Vision Conference*, 2012.