Fast Robust Reconstruction of Large Scale Environments

Jan-Michael Frahm, Marc Pollefeys, Svetlana Lazebnik, Brian Clipp, David Gallup, Rahul Raguram, Changchang Wu
Department of Computer Science
University of North Carolina *at* Chapel Hill
Chapel Hill, North Carolina, 27599-3175
Email: jmf@cs.unc.edu

***Abstract*—The approach presented in this paper tackles the active research problem of the fast automatic modeling of large-scale environments from videos with millions of frames and collection of tens of thousands of photographs downloaded from the Internet. The approach leverages recent research in robust estimation, image based recognition and stereo depth estimation. The high computational speed is achieved through parallelization and execution on commodity graphics hardware. The approach achieves real-time reconstruction from video and reconstructs within less than a day from tens of thousands of downloaded images on a single commodity computer. We demonstrate modeling results on a variety of real-world video sequences and photo collections.**

I. Introduction

Fully automatic modeling of large-scale environments has been a long-standing research goal in photogrammetry and computer vision. Detailed 3D models automatically acquired from the real world have many uses including civil and military planning, mapping, virtual tourism, games, and movies. In this paper we present a system approaching fully automatic modeling of large-scale environments, either from video or from photo collections.

Recently, mapping systems like Microsoft Bing Maps have started to use 3D models of cities. These systems achieve impressive results for modeling large areas with regular updates, but they still have many limitations. Namely, they require a human in the loop for delivering models of reasonable quality; the models have a very low complexity and do not provide enough detail for ground-level viewing; availability of the models is restricted to only a small number of cities across the globe. By contrast, our scalable system can automatically produce 3D models from ground-level images with a higher level of detail at high speeds and low cost.

All key components of our modeling pipeline have been optimized for efficiency by using the temporal relationships of video frames or for photo-collections combining image recognition and geometric constraints to establish spatial relationships efficiently. The registered images/frames are then registered into models with several thousands to millions of frames and for videos dense scene models are obtained. Figure 1 shows examples of our state of the art models.

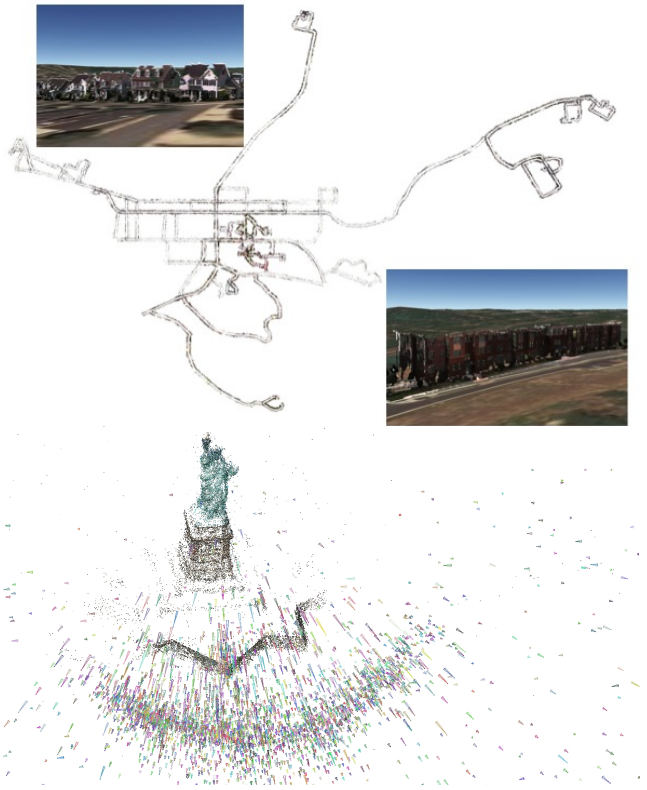

Fig. 1. The top shows an overview model of Chapel Hill reconstructed from 1.3 million video frames on a single PC in 11.5 hrs. On the bottom a model of the statue of liberty is shown. The reconstruction algorithm registered 9025 cameras out of 47238 images downloaded from the Internet.

II. Overview

Our 30 Hz real-time 3D reconstruction from video can operate from video alone or use the video streams of multiple cameras mounted aided by additional sensors, such as differential GPS and an inertial sensor (INS), to reduce accumulated error (drift) and provide geo-location. The system exploits the temporal order of the video frames, which implies a spatial relationship between adjacent frames, to efficiently perform camera motion tracking and dense geometry computation. Internet photo collections are not ordered and typically highly

contaminated. We determined for three datasets the degree of contamination by labeling the dataset for the "Statue of Liberty" (47238 images with approximately 40% outliers), the dataset for "San Marco" (45322 images with approximately 60 % outliers) and for "Notre Dame" (11928 images with approximately 49%). Our system uses appearance clustering combined with multi-view geometry to obtain pairwise image relationships. Besides modality specific adaptations the processing pipelines for both types of input (video and photo collections) share most of the same algorithmic components. The design of the systems aims at efficient reconstruction through parallelization of the algorithms, enabling their execution on the graphics processor (GPU). The major computational blocks of our system are:

- **Local correspondence estimation** establishes correspondences for salient feature points to neighboring views. In the case of video, feature correspondences are determined through KLT-tracking(Section IV-A1). For photo collections, our system first finds neighboring views for each image through clustering of global image descriptors (Section IV-C1) and then uses SIFT features [1] for detailed verification (Section IV-A).
- **Camera pose/motion estimation** from local correspondences is robustly performed through ARRSAC, an efficient RANSAC technique (Section IV-A2). It determines the inlier correspondences and the camera positions with respect to the previous images.
- **Global correspondence estimation** is performed to enable global drift correction. This step searches beyond the temporal neighbors in video and beyond the neighbors in the cluster for photo collections for images with overlap to the current image (Section IV-B).
- **Bundle adjustment** uses the global correspondences to reduce the accumulated drift of the camera registrations from the local camera pose estimates.
- **Dense geometry estimation** is performed from video streams in real-time to extract the depth of all pixels from the camera (Section V). Our system uses a two-stage estimation strategy. First we use efficient GPU-based stereo to determine the depth map for every frame. Then we use the temporal redundancy of the stereo estimations to filter out erroneous depth estimates and fuse the correct depth estimates.
- **Model extraction** is performed to extract a triangular mesh representing the scene from the depth maps.

The next section will survey the relevant literature for the above system components. Section IV-A will introduce our algorithm for real-time 3D reconstruction from video. Section IV-C will detail the adaptations that are necessary to improve efficiency in reconstruction from unstructured photo collections. Finally, Sections V and **??** will describe algorithms for creating dense stereo reconstructions and polygonal models from video streams.

III. Related Work

In the last few years, there has been a considerable progress in the area of large-scale reconstruction from video for urban environments and aerial data, as well as from Internet photo collections. Systems for urban reconstruction from video were proposed in [2], [3], [4]. These systems partially relied on a human in the loop or expensive capture equipment and, except our previous work [4], did not achieve fast reconstruction. To achieve fast processing and efficient visualization, Cornelis et al. [5] proposed to model facades as ruled surfaces parallel to the gravity vector.

One of the first works to demonstrate 3D reconstruction of landmarks from Internet photo collections is the *Photo Tourism* system [6]. This system achieves high-quality reconstruction results with the help of computationally exhaustive pairwise image matching combined with global bundle adjustment after inserting each new view. This process is particularly inefficient for heavily contaminated collections. Aiming at lower computational cost the system in [6], constructs *skeletal sets* of images from the collection whose reconstruction provides a good approximation to a reconstruction involving all the images [7]. That work still needs to compute the expansive pairwise matching. Using the independence of the pairwise evaluation Agarwal et al. [8] address this computational challenge by using a computing cluster with up to 500 cores to reduce the computation time significantly. The efficiency of our system is made possible by a hierarchical reconstruction approach starting from a set of *canonical* or *iconic* views [9], [10] representing salient viewpoints and parts of the scene. By contrast to previous systems, we view summarization as an image organization step that *precedes* 3D reconstruction, and we find iconic images using relatively simple 2D appearance-based techniques.

After discussing the above modeling systems we now discuss prior work on the main system components described in the previous section in more detail.

The first step in our systems is to establish local correspondences between the video frames or the different images of the photo collection respectively. Due to the large dynamic range of outdoor video we use our extended KLT tracker [11], which tracks the camera gain [12]. In the case of Internet photo collections, we do not have a natural linear ordering of images and have heavily contaminated collections. We extract the subsets of images that observe common 3D scene structure through 2D appearance descriptors inspired by [13], [14] Then we employ SIFT matching [1] to establish the local correspondences within each group of related images.

After establishing the local correspondences, we next determine the camera positions and orientations. As many other systems we leverage the work in multiple view geometry and typically alternates between robustly estimating camera poses and 3D point locations directly [15], [16]. Often bundle adjustment [17] is used in the process to refine the estimate. In [18], a technique for out-of-core bundle-adjustment is proposed, which takes advantage of this redundancy of photo

collections by locally optimizing the "hot spots" and then connecting the local solutions into a global one.

Having registered all the cameras, we next compute a dense 3D representation of the scene. Given the dense stream of viewpoints from video, we perform dense stereo. We refer the reader to [19], [20] for surveys of binocular and multiple-view stereo algorithms. Our system uses an extended version of Yang and Pollefeys [21] approach. Many other approaches target urban environments using the fact that they predominantly consist of planar surfaces [22], [23], or even more strict orthogonality constraints [24]. To ensure computational feasibility, large-scale systems generate partial reconstructions, which are afterwards merged into a common model. Conflicts and errors in the partial reconstructions are identified and resolved during the merging process [25]. Alternative, approaches like Koch et al. [26] presented a volumetric problem formulation.

Finally, we should note that our work does not address the *temporal* aspect of urban modeling, namely, the fact that cities evolve over time. Introducing this aspect is a challenging long-term research direction. One of the preliminary works to this end is the 4D Atlanta project of Schindler et al. [27].

IV. Camera Pose Estimation

In this section we discuss the methods used for camera registration in our system. First, Section IV-A will discuss camera registration methods for video sequences, which take advantage of the known temporal relationships of the video frames. Second, in Section IV-C we discuss the extension of camera registration to unordered Internet photo collections.

A. Camera Pose from Video

Reconstructing the structure of a scene from images begins with finding corresponding features between pairs of images. In a video sequence we can take advantage of the temporal ordering and small camera motion between frames to speed up correspondence finding. This allows our system to treat the local correspondence estimation as a tracking problem (Section IV-A1). Then the local correspondences are used to estimate the camera motion through our recently proposed efficient adaptive real-time random sampling consensus method (ARRSAC) [28] (Section IV-A2). Alternatively, if GPS and inertial measurements are available, the camera motion is estimated through a Kalman filter, efficiently fusing visual correspondences and the six degree-of-freedom (DOF) camera poses as detailed in [4].

1) Local Correspondences: Our system uses the Kanade-Lucas-Tomasi feature tracking [11] as differential tracking method that first finds strong corner features in a video frame, which are then tracked. To accommodate the large change in intensity occuring between frames in outdoor scenes we use our gain adaptive KLT tracker [12]. Specifically its GPU based implementation [29].

2) Robust Pose Estimation: The estimated local correspondences typically contain a significant portion of erroneous correspondences. To determine the correct camera position we apply our efficient adaptive Real-Time Random Sample Consensus (ARRSAC) algorithm [28] to estimate the relative camera motion through the essential matrix. While being highly robust ARRSAC overcomes the significant computational expense of RANSAC, with a runtime exponential in outlier ratio and model complexity. ARRSAC is capable of providing accurate real-time estimation over a wide range of inlier ratios. To achieve significant computational savings and to meet a fixed time budget estimation ARRSAC moves away from the traditional hypothesize-and-verify framework of RANSAC to a parallel evaluation scheme. For the case of epipolar geometry estimation, for instance, ARRSAC operates with estimation speeds ranging between 55-350 Hz.

B. Global Correspondences

Since our system registers cameras sequentially, in the absence of GPS the obtained registrations are always subject to drift. Each small inaccuracy in motion estimation will propagate forward and the absolute positions and motions will be inaccurate. It is therefore necessary to do a global optimization step afterwards using constraints that are capable of removing drift.

Registering the camera with respect to the previously estimated path provides an estimate of the accumulated drift error. Our method determines the path intersection using the images only by evaluating the similarity of SIFT-features [1] in the current frame to all features in all previous views. We use our SIFT-GPU implementation, which can extract SIFT features at $12Hz$ from 1024×768 images on an NVidia GTX280. Our method employs a vocabulary tree [30] combined with scene summarization as introduced in [31] to avoid exhaustive search. The obtained list of potentially overlapping views is tested for valid two-view relationship to the search image. This test uses a GPU based putative feature matching and the ARRSAC framework achieving more than 10 Hz verification rates as demonstrated in [31]. This allows us to establish a registration with the previously seen scene part in an online fashion.

Since the initial sparse model is subject to drift due to the incremental nature of the estimation process, the reprojection error of the global correspondences is typically higher than the error of the local correspondences. In order to obtain results with the highest possible accuracy, an additional refinement procedure, generally referred to as bundle adjustment, is necessary [17].

For video sequences, the bundle adjustment can be performed incrementally, adjusting only the most recently computed poses with respect to the existing already-adjusted camera path. This allows bundle adjustment to run in real time, although with slightly higher error than a full offline bundle adjustment. This bundle adjustment technique is also used extensively by our system for processing photo collections as described below.

C. Camera Pose from Image Collections

The main difference from the case of video where the temporal order of video frames implies a spatial relationship

between the corresponding cameras, photo collections downloaded from the Internet do not have any intrinsic ordering. Moreover, these collections tend to be highly contaminated with outliers. We use an efficient solution by taking advantage of the redundancy inherent in Internet photo collections, stemming the tendency of people to take pictures from very similar viewpoints and with very similar compositions.

1) Efficiently Finding Corresponding Images in Photo Collections: To efficiently identify related images in photo collections, our system uses the *gist* feature [32], which encodes the spatial layout of the image and perceptual properties of the image. The gist feature was found to be effective for grouping images by perceptual similarity and retrieving structurally similar scenes [33]. To achieve high computational performance, we developed a highly parallel gist feature extraction on the GPU obtaining a 368-dimensional vector as a representation of each image in the dataset. The implementation on the GPU improves the computation time by a factor of 100 compared to a CPU implementation (timings are given in Table I).

Given that photos from nearby viewpoints with similar camera orientation have similar gist descriptors we use k-means clustering to effectively cluster viewpoints. At this point we aim for an over-segmentation since that will best reduce our computational complexity in subsequent steps. This is key to the overall efficiency of our system since this early grouping allows us to limit all further geometric verifications and to avoid an exhaustive search over the whole dataset as in [6].

The clusters loosely grouping the dataset, although it is sensitive to image variation such as clutter (people in front of the camera), lighting conditions, and camera zoom. We found that large clusters are typically almost outlier-free (examples shown in [34]), while the smaller clusters have significantly more noise and contamination. Next we enforce multi-view constraints to ensure that the images of each cluster observe the same static scene.

2) Iconic Image Registration: To facilitate efficient registration, we want to find a representative image (iconic) for each cluster of views. We can then remove unrelated images by enforcing a valid two-view geometry with respect to the iconic for each image in the cluster. Given the large feature motion in photo collections, similarly to the global correspondence computation in Section IV-B, we use SIFT features [1], putative correspondence estimation on the GPU and the ARRSAC from Section IV-A2 for two-view geometry estimation to determine the local correspondences between the images. To achieve robustness to degenerate data, we combine ARRSAC with our robust model selection method, dubbed QDEGSAC [35]. For more detail on this process we refer to [34]. After this geometric verification, each cluster only contains images showing the same scene, and each cluster is visually represented by its iconic image.

3) Global Registration Using the Iconic Scene Graph: After using the clusters to establish local image relationships and to remove unrelated images, the next step is to establish a global relationship of the images. For efficiency reasons, we use the small number of iconic images to bootstrap the global registration for all images. The exhaustive pairwise two-view relationships for all iconics are computed again using ARRSAC (Section IV-A2). This defines a graph of pairwise global relationships between the different iconics, which is called the *iconic scene graph*. The graph represents the global registration of the iconics based on the 2D constraints given by the epipolar geometry or the homography between the views. Next, our system performs an incremental structure from motion process using the iconic scene graph to obtain a registration for the iconic images (see [34] for details). The obtained 3D models for the independent components of the iconic scene graph are refined by bundle adjustment [17]. Typically the iconic scene graph contains multiple independent or weakly connected components. These components may correspond to parts of the scene that have no visual connection to the other parts, interior parts of the model, or other scenes consistently mislabeled (for example, Ellis Island is often labeled as "Statue of Liberty", as shown in Figure 2). Hence we repeat the 3D modeling until all images are registered or none of the remaining images has any connections to any of the 3D sub-models.

Dataset	Gist & clustering	Geometric	Re-clustering	registered views	total time
Liberty	0:25 h	3:08 h	3:21 h	13888	6:53 h
SM	0:23 h	3:42 h	2:47 h	12253	6:52 h
ND	0:0 3 h	1:19 h	1:02 h	3058	2:25 h

TABLE I
COMPUTATION TIMES FOR THE PHOTO COLLECTION RECONSTRUCTION FOR THE STATUE OF LIBERTY DATASET, THE SAN MARCO DATASET (SM) AND THE NOTRE DAME DATASET (ND).

To obtain more complete 3D sub-models than is possible through registering the iconics, our system searches for non-iconic images that support a matching for the iconics in two different clusters. Once a sufficient number of connections is identified between two clusters, our method uses all constraints provided by the matches to merge the two sub-models to which the clusters belong. The merging estimates the similarity transformation between the sub-models until no more additional merges are possible.

After the registration of the iconic images, we have a valid global registration for the images of the iconic scene graph. In the last step we extend the registration to all images in the clusters corresponding to the registered iconics using ARRSAC (Section IV-A2) and the matches to the iconic combined with periodic bundle adjustment. Results of our 3D reconstruction algorithm are shown in Figure 2.

V. DENSE GEOMETRY

The above described methods for video and for photo collections provide a registration for the camera poses, or external and internal camera calibrations for every image

Fig. 2. Top: Model from the interior of "Ellis Island" erroneously labeled as "Statue of Liberty". Middle: 3D reconstruction of the San Marco Dataset with 10338 cameras. Bottom: Reconstruction from the Notre Dame dataset with 1300 registered cameras.

in the scene. The knowledge of camera parameters can be used to generate an image-based browsing experience for Internet photo collections, as shown by the "PhotoTourism" paper [6]. Camera registration can also be used as input to dense stereo methods. Dense stereo for Internet photo collections is currently a very difficult research problem due to the irregular distribution of camera viewpoints, wide variation in lighting conditions, and the lack of photometric camera calibration. One initial method is that of [36], and it requires significant computational effort and can only be applied on a small scale. We have not yet attempted efficient large-scale estimation of dense geometry from Internet photo collections, but we have developed a real-time large-scale stereo system for video streams.

We adopt a stereo/fusion approach for dense geometry reconstruction. For every frame of video, a depth map is generated using a real-time GPU-based multi-view plane-sweep stereo algorithm [12]. Since it is a multi-view stereo, it robustly handles occlusions, a major problem in two-view stereo. The stereo depth maps are then fused using a visibility-based fusion method, which also runs on the GPU [37]. Beyond removing

Fig. 3. Our depthmap fusion method combines multiple stereo depthmaps (middle image shows an example depthmap) to produce a single fused depthmap (shown on the right) with greater accuracy and less redundancy.

outliers from the depth maps, it also combines depth estimates to enhance their accuracy. Given the redundancy in video (each surface is imaged multiple times), fused depth maps only need to be produced for a subset of the video frames. This reduces processing time as well as the 3D model size.

Our plane-sweep stereo algorithm is an extension to dense geometry of the algorithm of [38] and computes a depth map by testing a family of plane hypotheses, and for each pixel recording the distance to the plane with the best photo-consistency score. One view is designated as reference, and a depth map is computed for that view. For each plane, all matching views are back-projected onto the plane, and then projected into the reference view through homography mapping [39]. It can be performed very efficiently on the GPU. Once the matching views are warped, the sum of absolute differences (SAD) is computed to score the photo-consistency. To handle occlusions, the matching views are divided into a left and right subset, and the best matching score of the subsets is kept [40]. To normalize intensity differences in the image due to different exposures the image intensities are multiplied by the relative exposure (gain) computed during KLT tracking (Section IV-A1). The stereo algorithm can produce a 512×384 depth map from 11 images (10 matching, 1 reference) and 48 plane hypotheses at a rate of 42 Hz on an Nvidia GTX 280.

After depth maps have been computed, a fused map is computed for a reference view using the surrounding stereo depth maps [37]. All points from the depth maps are projected into the reference view, and for each pixel the point with the highest stereo confidence is chosen. Pixel wise stereo confidence is computed based on the shape of the cost function for the pixel (details are given in [37]). This confidence measure prefers depth estimates with low uncertainty, where the matching score is much lower than scores for all other planes.

For each pixel, after the most confident point is chosen, it is scored according to visibility constraints. Supporting points are those that fall within 5% of the chosen point's depth value. Valid points, are replaced with a new point computed as the average of the chosen point and support points. See Figure 3.

Currently our dense geometry method is used only for video. The temporal sequence of the video makes it easy to select nearby views for stereo matching and depthmap fusion. For photo collections, a view selection strategy would be needed that identifies views with similar content and compatible appearance (day, night, summer, winter, etc.). This is left as future work.

The final step of our system extracts a triangular mesh model and obtains the textures from the while removing double surfaces in the final models. For details we refer to [4].

VI. Conclusions

In this paper we have presented methods for real-time reconstruction from video and for fast reconstruction from Internet photo collections on a single PC. We demonstrated the computational performance of the methods on a variety of large-scale datasets. Efficiency was achieved through parallelization of many computations involved, enabling an execution on the graphics card as a highly parallel processor. Additionally, for modeling from Internet photo collections we combine constraints from recognition with geometric constraints, leading to an orders of magnitude more efficient image registration than any existing system.

Acknowledgment

We would like to acknowledge the DARPA UrbanScape project, NSF Grant IIS-0916829, other funding from the US government, and our collaborators David Nister, Horst Bishof, Arnold Irschara, Christopher Zach, Amir Akbarzadeh, Philippos Mordohai, Paul Merrell, Chris Engels, Henrik Stewenius, Brad Talton, Liang Wang, Qingxiong Yang, Ruigang Yang, Greg Welch, Herman Towles, Xiaowei Li.

References

[1] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[2] A. Gruen and X. Wang, "Cc-modeler: A topology generator for 3-d city models," *ISPRS Journal of Photogrammetry & Remote Sensing*, vol. 53, no. 5, pp. 286–295, 1998.

[3] Z. Zhu, A. Hanson, and E. Riseman, "Generalized parallel-perspective stereo mosaics from airborne video," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 226–237, 2004.

[4] M. Pollefeys, D. Nister, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewenius, R. Yang, G. Welch, and H. Towles, "Detailed real-time urban 3d reconstruction from video," *International Journal of Computer Vision*, vol. special issue on Modeling Large-Scale 3D Scenes, 2008.

[5] N. Cornelis, K. Cornelis, and L. Van Gool, "Fast compact city modeling for navigation pre-visualization," in *Int. Conf. on Computer Vision and Pattern Recognition*, 2006.

[6] N. Snavely, S. M. Seitz, and R. Szeliski, "Modeling the world from Internet photo collections," *International Journal of Computer Vision*, vol. 80, no. 2, pp. 189–210, November 2008.

[7] ——, "Skeletal sets for efficient structure from motion," in *CVPR*, 2008.

[8] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski, "Building Rome in a day," in *ICCV*, 2009.

[9] S. Palmer, E. Rosch, and P. Chase, "Canonical perspective and the perception of objects," *Attention and Performance*, vol. IX, pp. 135–151, 1981.

[10] T. L. Berg and A. C. Berg, "Finding iconic images," in *The 2nd Internet Vision Workshop at IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[11] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 1981, pp. 674–679. [Online]. Available: citeseer.ist.psu.edu/lucas81iterative.html

[12] S. Kim, D. Gallup, J.-M. Frahm, A. Akbarzadeh, Q. Yang, R. Yang, D. Nistér, and M. Pollefeys, "Gain adaptive real-time stereo streaming," in *Int. Conf. on Vision Systems*, 2007.

[13] T. L. Berg and D. Forsyth, "Automatic ranking of iconic images," University of California, Berkeley, Tech. Rep., 2007.

[14] F. Schroff, A. Criminisi, and A. Zisserman, "Harvesting image databases from the web," in *ICCV*, 2007.

[15] P. Beardsley, A. Zisserman, and D. Murray, "Sequential updating of projective and affine structure from motion," *Int. J. Computer Vision*, vol. 23, no. 3, pp. 235–259, Jun-Jul 1997.

[16] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, vol. 23, no. 1, 2006.

[17] American Society of Photogrammetry, *Manual of Photogrammetry (5th edition)*. Asprs Pubns, 2004.

[18] K. Ni, D. Steedly, and F. Dellaert, "Out-of-core bundle adjustment for large-scale 3d reconstruction," in *ICCV*, 2007.

[19] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. of Computer Vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[20] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Int. Conf. on Computer Vision and Pattern Recognition*, 2006, pp. 519–528.

[21] R. Yang and M. Pollefeys, "Multi-resolution real-time stereo on commodity graphics hardware," in *Int. Conf. on Computer Vision and Pattern Recognition*, 2003, pp. 211–217.

[22] T. Werner and A. Zisserman, "New techniques for automated architectural reconstruction from photographs," in *European Conf. on Computer Vision*, 2002, pp. 541–555.

[23] S. N. Sinha, D. Steedly, and R. Szeliski, "Piecewise planar stereo for image-based rendering," in *International Conference on Computer Vision (ICCV)*, 2009.

[24] Y. Furukawa, B. Curless, S. M. Seitz, , and R. Szeliski, "Manhattan-world stereo," in *Computer Vision and Pattern Recognition (CVPR)*, 2009.

[25] P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J.-M. Frahm, R. Yang, D. Nister, and M. Pollefeys, "Real-Time Visibility-Based Fusion of Depth Maps," in *Proceedings of International Conf. on Computer Vision*, 2007.

[26] R. Koch, M. Pollefeys, and L. Van Gool, "Robust calibration and 3d geometric modeling from large collections of uncalibrated images," in *DAGM*, 1999, pp. 413–420.

[27] G. Schindler, P. Krishnamurthy, and F. Dellaert, "Line-based structure from motion for urban environments," in *3DPVT*, 2006.

[28] R. Raguram, J.-M. Frahm, and M. Pollefeys, "A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus," in *ECCV*, 2008.

[29] C. Zach, D. Gallup, and J. Frahm, "Fast gain-adaptive KLT tracking on the GPU," in *CVPR Workshop on Visual Computer Vision on GPU's (CVGPU)*, 2008.

[30] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *CVPR*, 2006.

[31] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof, "From structure-from-motion point clouds to fast location recognition," in *In Proceedings of IEEE CVPR*, 2009, pp. 2599–2606.

[32] A. Oliva and A. Torralba, "Modeling the shape of the scene: a holistic representation of the spatial envelope," *IJCV*, vol. 42, no. 3, pp. 145–175, 2001.

[33] M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg, and C. Schmid, "Evaluation of gist descriptors for web-scale image search," in *International Conference on Image and Video Retrieval*, 2009.

[34] X. Li, C. Wu, C. Zach, S. Lazebnik, and J.-M. Frahm, "Modeling and recognition of landmark image collections using iconic scene graphs," in *ECCV*, 2008.

[35] J.-M. Frahm and M. Pollefeys, "RANSAC for (quasi-) degenerate data (QDEGSAC)," in *CVPR*, vol. 1, 2006, pp. 453–460.

[36] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz, "Multi-view stereo for community photo collections," in *ICCV*, 2007.

[37] P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J.-M. Frahm, R. Yang, D. Nister, and M. Pollefeys, "Fast visibility-based fusion of depth maps," 2007.

[38] R. Collins, "A space-sweep approach to true multi-image matching," in *Int. Conf. on Computer Vision and Pattern Recognition*, 1996, pp. 358–363.

[39] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[40] S. Kang, R. Szeliski, and J. Chai, "Handling occlusions in dense multi-view stereo," in *Int. Conf. on Computer Vision and Pattern Recognition*, 2001, pp. 103–110.