Assisted multi-view stereo reconstruction

Matteo Dellepiane, Emanuele Cavarretta, Paolo Cignoni, Roberto Scopigno
Visual Computing Lab
ISTI-CNR
Pisa, Italy
surname@*isti.cnr.it*

***Abstract*—Multi-view stereo reconstruction methods can provide impressive results in a number of applications. Nevertheless, when trying to apply the state-of-the-art methods in the case of a more structured 3D acquisition, the lack of feedback on the quality of the reconstruction during the photo shooting can be problematic.**
In this paper we present a framework for the assisted reconstruction from images of real objects. The framework is able to provide, in quasi-realtime, a sparse reconstruction of the scene, so that the user is able to spot the missing or problematic parts. Moreover, the framework is able to separate the object of interest from the background and suggests missing points of view to the user, without any previous knowledge of the shape of the scene and the acquisition path. This is obtained by analyzing the sparse reconstruction and the connection between the reconstructed points and the input images.
The framework has been tested on a variety of practical cases, and it has proved to be effective not only to obtain more complete reconstructions, but also to reduce the number of images needed and the processing time for dense reconstruction.

I. Introduction

Multi-view stereo reconstruction is a process concerning the automatic acquisition of objects and scenes models from multiple photographs. It aims at obtaining a 3D representation of a real object, in the form of a point cloud or a polygonal mesh, starting from a set of uncalibrated images. Potential applications of this technology include: construction of realistic object models for the movie, television, and video game industries; quantitative recovery of metric information for scientific and engineering data analysis; fast visualization via point-based rendering techniques; and object replication through fast prototyping technologies.
The diffusion of high-quality, low-cost consumer digital cameras and the advancement in the state-of-the-art of this research topic made multi-view stereo reconstruction a very promising technology widely available to the public. A significant example of this is the Microsoft PhotoSynth online service, but several other freeware or low-cost solutions are also available.
The structure of most multi-view stereo reconstruction solutions is based on an unsupervised strategy, where no previous information about the images is known. The reconstruction follows some steps which are usually fulfilled in a pipeline fashion (see also Figure 1-top). Although based on different steps, that could be driven by the user, the available systems are usually structured as a *black-box*, because the algorithms are optimized for the case where all the images are available from the beginning of the reconstruction. The whole process can take several hours, especially in the case of complex scenes with several tens of images. In some cases, the user can have an intermediate feedback by analyzing a sparse reconstruction of the scene, but if some additional images have to be added, the process must start again from scratch. While the results that can be obtained are impressive, it is harder (w.r.t. other acquisition technologies, like 3D scanning) to know in advance if an object will be entirely acquired or not. This is due to the intrinsic nature of the approach, which matches a type of feature which is only intrinsically related to visible features on the object. Moreover, the *black-box* approach prevents from having a real-time feedback on the completeness of the reconstruction: the user will not know if the acquisition is successful until the whole dataset has been processed. This could be problematic in a number of contexts, like archeological excavations, which are essentially a destructive process [1]. In this case, if we miss important view points in the photographic campaign, we will discover it when the digging process would have progressed further.
In this work we propose an approach for 3D object and scene reconstruction that provides a quasi-realtime, assisted sparse reconstruction. While the images are transferred from the acquisition device to the processing system (i.e. a laptop), the reconstructed scene is incrementally updated, and the system provides feedback to the user to allow him to produce a complete reconstruction of the scene, with a reasonable number of images. The main contributions of the proposed system are:

- An incremental, quasi-realtime approach for the calculation of the *sparse reconstruction* of the scene. The structure of the reconstruction scheme breaks the *blackbox* paradigm, enabling an interaction between the user and the system.
- An automatic method to identify the object(s) of interest for our reconstruction, in order to disambiguate it/them from the background, and to suggest the missing points of view that will help in the completion of the reconstruction.

- A system to reduce the number of images needed to cover and sample the surface of interest (defined according to our approach), thus reducing also the processing time to produce the dense reconstruction.

II. Related Work

In the last few years, multi-view stereo reconstruction has been a very active field of research. The proposed methods are essentially an extension of the idea of stereo reconstruction. The approaches for Simultaneous Localization And Mapping (SLAM) [2], [3], [4] usually provide an approximate reconstruction of the scene, achieving the result in nearly realtime. These methods rely on frame-to-frame video tracking, and they are not always reliable for a wide baseline case. For example, specific closing loop strategies could be needed [5]. In our case, given the generality of the scenes and the difference between the camera positions, the best approach is represented by the combination of Structure from Motion [6] and Multi View Stereo Reconstruction. This pipeline can slightly vary among the approaches, and an accurate overview is well beyond the scope of this paper. Nevertheless, the reconstruction procedure is usually divided in three main steps:

- *Features recognition and Matching*: the first stage consists in analyzing the input images, extracting a number of descriptive points, and matching them among all the possible couples of images. Several types of feature points can be considered, but the most robust and used are the Scale Invariant Feature Transform (SIFT) [7], that proved to be extremely flexible and to adapt well to 3D reconstruction. Moreover, a GPU implementation [8] enables to calculate the matching among big groups of images in a reasonable short time.
- *Camera Calibration and Bundle Adjustment*: starting from the matching table provided by the previous step, the camera parameters associated with each images, and the corresponding 3D points generated by the matching tracks can be estimated to create a sparse representation of the scene. This is usually obtained using a Bundle Adjustment strategy [9], that was studied and modified to handle very complex cases [10], [11], [12].
- *Multi-view dense reconstruction*: the result of the previous step can be (at least partially) used as a starting point to compute a denser reconstruction of the scene [13], [14], [15]. These methods can provide dense models, but the processing can take hours especially when dealing with hundreds of images.

The previous pipeline has become a standard for Multi-view Stereo Reconstruction, and it was implemented in several versions, both structured as webservices or closed systems, where the user must provide all the images which were acquired, and the interaction is limited to the possibility to tune the parameters of the various steps of the reconstruction.
Some attempts have been done to break this paradigm, by proposing methods to assist the user during the acquisition, essentially by estimating a set of missing view to be suggested to the user [16], [17], [18]. Our approach differentiates from the latter because there is no assumption on the object shape and the acquisition setup, and the object of interest and the suggested view(s) are generated only by analyzing the sparse reconstruction of the scene.

III. Motivation and approach

The goal of the present work is to actively assist the user while acquiring images for Multi-view Stereo Reconstruction. The aim is to reduce the number of needed images, and to be able to detect missing parts of the object of interest. In order to obtain this, it is necessary to break the *black-box* paradigm, to provide a feedback before the time-consuming part of the pipeline, i.e. the dense stereo reconstruction.
Figure 1 shows the different structure of our method w.r.t. the classic approach. On the top, the classic data processing pipeline is shown, with the user providing and initial dataset and receiving the final result. On the bottom, the structure of our incremental method where, once shot, any image is added to the reconstruction process. Hence, the point cloud is updated, and a feedback loop analyzes the scene and assists the user in completing the acquisition. The user can have a nearly realtime feedback about the quality and completeness of the reconstruction, in order to choose the new images without adding redundant data. Moreover, the system is able to suggest possible missing view(s), that could complete the portions of the object which show a lower density of sampled data. In order to obtain the new acquisition structure, it was necessary to structure the coarse reconstruction box (in green) in a new, incremental fashion, and design the new block (in red) regarding scene understanding and new views selection. The next subsections present the structure of these two blocks.

A. Incremental sparse reconstruction

The recent advancements in literature and implementation made the sparse reconstruction step a quite fast one, able to complete in minutes even with datasets made of several tens of images. Nevertheless, the current available tools perform feature matching and bundle adjustment in a non incremental way, needing all the images of the dataset.
Hence the reconstruction procedure was reorganized in order to be able to refine the 3D point cloud every time a new image is available. A different strategy is needed to handle the startup and the purely incremental phase.

Startup. One of the critical steps of the incremental approach is the startup of the reconstruction, where a suitable initial description of the scene is needed. If a wrongly estimated scene is calculated, the whole reconstruction process can fail. In the case of the *blackbox* approach, the strategy is the same as in Bundler [6], where (after the feature

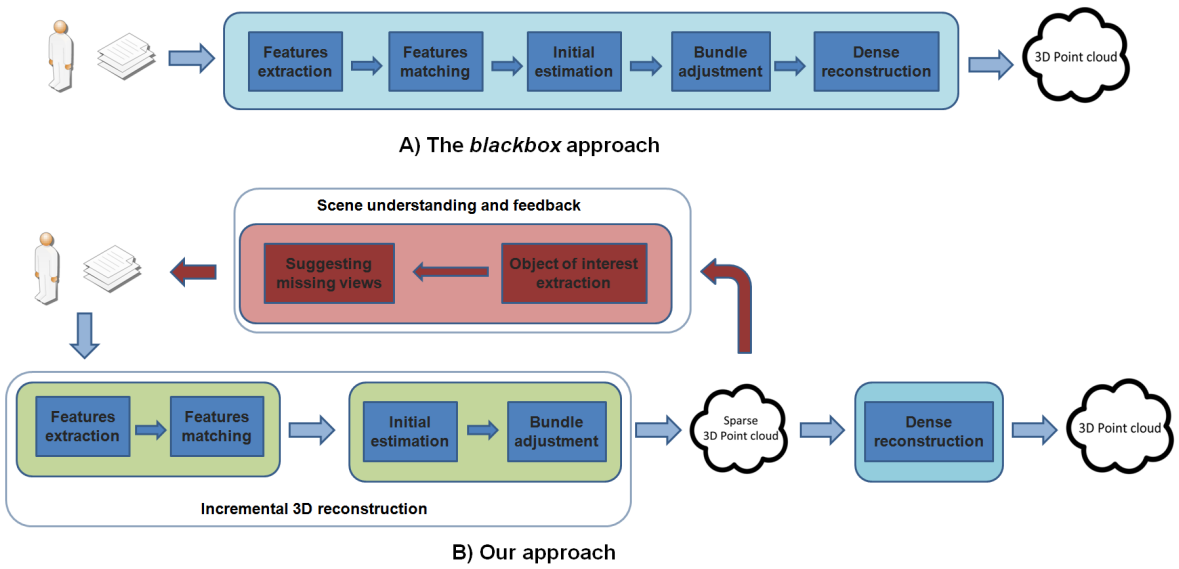

Figure 1. A) The structure of the black-box approach. B) The structure of the proposed method, with the feedback loop during acquisition

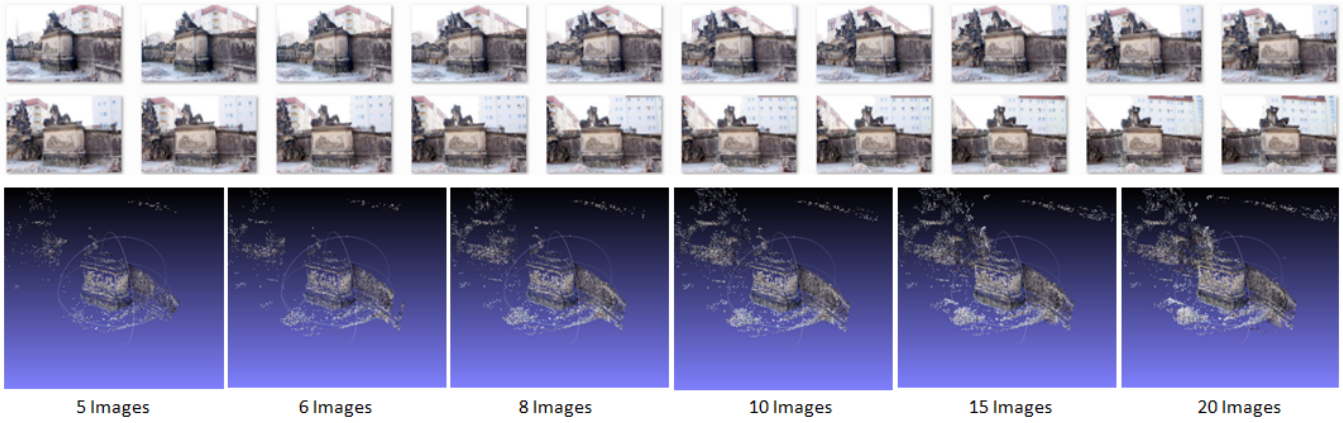

Figure 2. An example of the incremental reconstruction of a scene. Top: the 20 images used for the reconstruction. Bottom, the evolution with the insertion of new images.

extraction and matching) a starting group of four images is chosen as a starting set. In our system, it is not possible to choose the best images. Hence, the strategy for startup is as follows:

- The system is idle until the fourth image is received. Then, a first reconstruction is calculated.
- The reconstruction is checked in order to prevent degenerated configurations. This is achieved by analyzing the relative camera position, orientation, and focal length. If the position and orientation of the cameras are too similar, or the focal length is too different, the reconstruction is discarded.
- If the reconstruction is validated, the system starts the incremental mode (see below). Otherwise, the system waits for another image and applies the Bundler approach, by selecting the best four for an initial reconstruction, and then adding the others in an incremental way.

Incremental reconstruction. Once that the reconstruction has been initialized, the system waits for a new image. At its arrival, two steps are fulfilled:

- The features of the new image are extracted and matched with all the other images. The difference with respect to the classic approach is that the track table (the list of group of matches that concur in the reconstruction) has to be updated, by integrating the existing tracks, and adding new tracks that are created with the insertion of the new image. An initial estimation of the new camera parameters is calculated as well. If the initial estimation fails, the image is discarded (see below).
- The bundle adjustment is calculated on the updated set

of data. In the current system, we decided to implement the MulticoreBA [12], where all the data are taken into account and re-calculated. The choice of this solution was due to the fact that the BA is completed in a few seconds even when several images have been already acquired, and the re-calculation tends to strengthen the scene coherency. An incremental alternative to this was proposed by Lourakis [10]: this could speed up the reconstruction, while some assumptions on the acquisition strategy (overlap between temporally adjacent images) would be needed. Our choice was to put no assumption on the acquisition strategy, leaving the user free to choose its preferred sampling.

Figure 2 shows the evolution of the incremental reconstruction, where the point cloud size and density increases as soon as new images are added.

Recovering discarded images. Since the goal of the system is to get the best of the images taken by the user, a mechanism to recover the images previously discarded has been added. Every time that a new image is correctly added to the reconstruction, the system tries to add also the images that were discarded before, by trying to calculate an initial camera estimation, and eventually re-launching the BA. This mechanism enables to account for cases where the user does not provide an accurate coverage of the object, but some "bridging" images are provided in a subsequent moment. Hence, adding a single image could lead to a big improvement in the scene reconstruction.

B. Scene understanding and feedback

The evolution of the reconstruction already guides the user in the acquisition process, but we implemented a new method to analyze the reconstructed scene, so that a useful feedback to the user can be provided. Both of them are obtained by analyzing the reconstructed scene and the behavior of the user.

The first one is related to distinguish the object of interest (the one that the user is trying to acquire) from the background information, which is valuable for the reconstruction process, but usually makes harder to understand the rendered results. The second one analyzes the selected zone of interest to find the zones which are less densely sampled, and suggest possible points of view to speed up and complete the acquisition process.

1) Extracting the object of interest: The aim of this component is to discriminate the object of interest from the background. In order to obtain a generic model, independent from the type of object which is acquired, a quality measure is calculated for each point of the cloud. A first criterium could follow the amount of images that generate the point, but this is also related to the strength of the corresponding SIFT, which can be also part of the background. Another observation, which follows other works on object recognition and tracking [19], [20], is that the object of interest quite often lay in the center of the image. Following this assumption, we can assign to each reconstructed point a quality value which is related to the position in each image of the feature points which generated it. Hence, given a point P in the sparse reconstructed point cloud, generated by N images, we define its quality as:

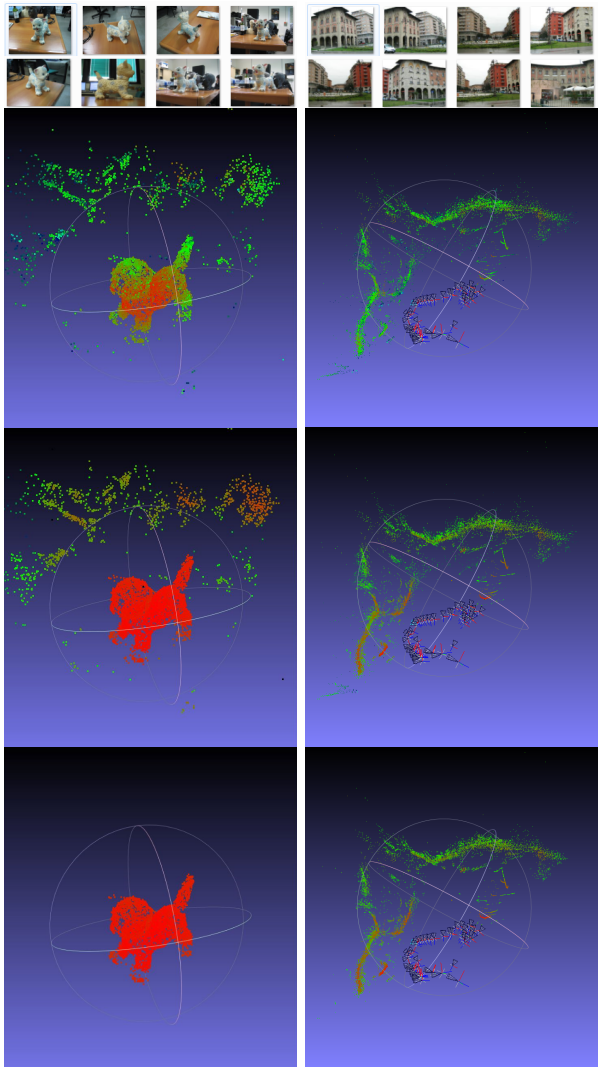

Figure 3. Two examples of the quality value for background removal (red: highest quality, blue lowest quality). Red indicates high quality, blue low quality. First row: a subset of the images. Second row: the quality value based on Equation 1. Third row: the smoothed quality value. Fourth row: the automatic background removal.

$$Q(P) = \frac{\sum_{i=1}^{N} \frac{max(|w_i/2 - x_i|, |h_i/2 - y_i|)}{max(w_i/2, h_i/2)}}{N} \quad (1)$$

where x_i and y_i are the coordinates of the feature in the i-th image which generated the point, w_i and h_i respectively

the width and height of the image. Two examples of the mapping of this quality function are shown in Figure 3, second row. The two sets are representative of two different acquisition types: a single object acquired with an hemispherical pattern and an architectural context acquired with a *panoramic* pattern. The use of the proposed quality value shows that in the first case, the central object already appears detached from the background, while in the second case the quality value is well distributed among the portions of the scene.

While it already shows the position and shape of the object of interest, the quality measure is not enough to automatically segment it from the background. Hence a pass of smoothing is applied to the quality values of each point. The smoothing is applied taking into account the density of the point cloud: each quality value is recalculated by averaging the neighbors which are at a distance which is below the average distance of the 60-neighbors. In this way, the parts of the point cloud which are of lower quality, but near to high quality and dense portions, gain value, while the low quality and sparse portions are hardly affected. The third row of Figure 3 shows the quality value after smoothing: the small cat is now fully separated from the background, while in the second row the quality value is more equally distributed in the scene.

After the smoothing step, a bigger difference is present between the dense object reconstruction and the sparse background. Taking advantage of the separation between the object and the background, a threshold cut can be automatically calculated by taking into account the histogram of the quality values. The histogram (organized in 40 bins) is analyzed by finding the maximum in the zone of high quality. Then, the histogram is analyzed in the decreasing direction of quality value, and the threshold to cut the background is set on the first local minimum. The last row of Figure 3 shows the automatic extraction of the object of interest: in the first case, the object is extracted. In the second case, the system is able to understand that almost the whole sampled set of points constitutes the scene of interest. Nevertheless, the user can manually tune the threshold, in order to better visualize the object of interest.

It's important to stress that the background information is only detected and hidden, but it remains an active part in the matching and reconstruction process.

2) Generating suggested views: Once that the object of interest has been detached from the rest of the scene, the system analyzes the sparse reconstruction in order to suggest missing points of view to complete the acquisition. The new views are created by taking into account the portions at lowest resolution, and generating a camera with similar parameters to the ones that were used for the reconstruction. The procedure is defined as follows:

- A reference point is found on the point cloud. The point represents the zone with less density in the point cloud, and it's selected by finding the point whose 60 nearest neighbors have the highest average distance.
- The camera intrinsic parameters are set identical to the other cameras of the reconstruction. The focal length is assigned as an average of the focal length of all the cameras.
- The direction of view is obtained by estimating the normal of the point using its neighbors [21]. The distance of the camera is set as an average of the camera distances of all the other views.
- Since no previous information about the scene orientation and only the direction of view is known, the other components of the orientation matrix of the camera are estimated by using the intersection between the direction of view and the ground plane which is defined by the reconstructor. Unfortunately, the ground plane of the scene is often different w.r.t. the real one. A small intervention of the user could help generating also the right orientation of camera. In any case, the proposed views are accurate enough to provide a valid suggestion to the user.

In order to generate views which are not too similar to the ones used for the reconstruction, the direction of view is slightly perturbed several times, and the chosen view is the one which is the most detached from the others. Figure 4 shows five suggested views generated on a dataset acquiring a statue. The Figure shows that the proposed views try to break the circular acquisition path, in order to cover some detailed part that needs peculiar points of view. Moreover, the proposed views are correctly concentrated on the part of the statue which exhibits more geometric detail.

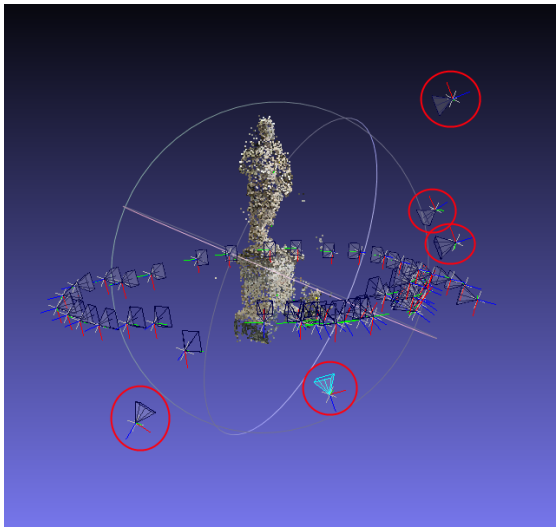

Figure 4. An example of five suggested views (indicated by red circles) in the context of the acquisition of a statue. The views concentrated on the more detailed part of the object, and try to suggest different directions of view.

C. Acquisition procedure and output

Given the components described above, the acquisition procedure using the proposed framework is organized as follows: the user starts acquiring the object, making a first initial coverage. The sparse reconstruction visually guides him to add missing images, until enough 3D data is available. The framework analyzes and separates the object of interest. The user is then able to ask for suggested views, which will focus on low density parts that are difficult to spot when analyzing the point cloud.
Once that the user is satisfied with the input dataset, it's possible to export it in a compatible format for a dense reconstruction [14]. Moreover, since the system has separated the object and the background, it is also possible to generate a mask for each image of the dataset, to shorten the reconstruction time. Since only a sparse description of the object is known, this mask is generated in a conservative way, by applying a series of dilation and erosion steps to close holes and preserve the silhouette of the object. Please also refer to next Section and Figure 6.

IV. Results

The framework was tested on a number of practical cases. The experimentation was conducted using a Canon Powershot S95, acquiring the images with a resolution of 5MPixels. The images were directly transferred (using an Eye-Fi SD card) to a laptop with 64bit OS, 8 Gb Ram and NVidia Quadro 2000M. All the processing was conducted on the same laptop, in order to test also the applicability of the on-the-field processing.
We present three of the test cases, which are paradigmatic of different acquisition strategies: Cat, a small object that can be acquired with an hemisphere of camera positions; Statue, which can be acquired with a circular path of cameras at similar heights; and City Wall, which represents the acquisition of an architectural element using a *panorama* style.
For each test case, we compared the results of our framework with the one obtained with one of the fastest state-of-the-art tools, VisualSfM [22]. After the use of our method and VisualSfM, the output data were processed with the same dense reconstruction method, PMVS2 [23]. For the analysis of data, we focused on three aspects: the number of images used in two approaches, the time needed for acquisition and processing (taking into account also the use of the automatic masks), and the final result. Table 5 shows some figures on this comparison.
In general, the number of acquired images is similar, the acquisition time is longer for our method, also because it contains the sparse reconstruction phase. Hence, the processing for the dense reconstruction is always longer when using VisualSfM+PMVS2, not only for the bigger amount of data calculated, but also because the matching and sparse reconstruction is part of its process. During the acquisition with our method, the time needed to add a single image was between 4 and 25 seconds, depending on a number of factors: number of images already in the input set, number of matches and tracks, size of the scene.
Analyzing each set, in the Cat case we used the same input dataset (acquired using our method) to compute the reconstruction. The results, in Figure 6, surprisingly show that VisualSfM+PMVS2 produces a bigger number of points, but it's not able to reconstruct the upper part of the Cat, although the images covered it (as our method shows). Moreover, the reconstruction obtained using masks does not depict the background, needing a much lower amount of work for data cleaning.
In the Statue dataset a similar number of images was

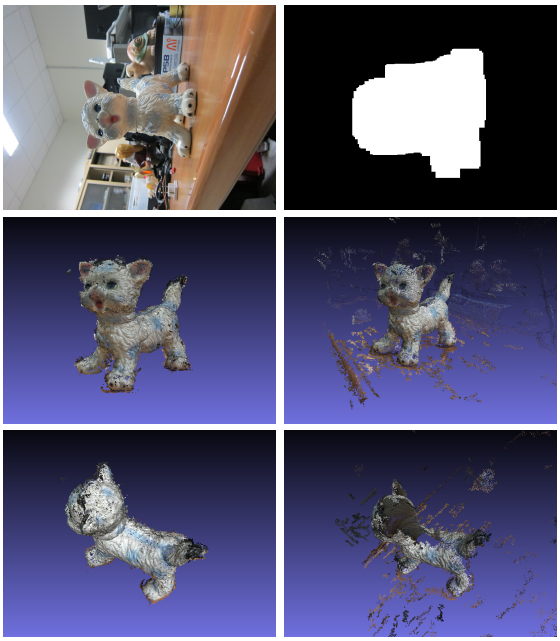

Figure 6. The Cat test set (32 images). First row, an example of an image and of the automatically calculated mask. Second and third row, snapshots of the reconstruction obtained with Our method and VisualSfm+PMVS2 [22]

acquired, but the processing time was definitely shorter in the case of our method. This was because our framework masked the background, obtaining the same number of points on the object of interest in a much shorter time, see Figure 7. In this case, the suggested views by our system could be only partially followed, since some of the points of view were not physically reachable from the ground level.

Finally, the City Walls case shows that dense reconstruction is similar for the two methods, although VisualSfM+PMVS2 is able to retrieve a bigger portion of the scene, and the data are less noisy. This could be due to an increased difficulty in handling planar objects and panoramic-like acquisitions. These are more prone to deformation and small matching errors, so that the incremental reconstruction

	$Our method + PMVS2$				$VisualSfM + PMVS2$			
	N.Images	Acq.Time	Proc.Time (masked)	N. points (masked)	N.Images	Acq.Time	Proc.Time	N. points
Cat	32	7 min.	176sec(90sec)	72114(51136)	32	-	480 sec	97800
$Statue$	31	12 min	480sec(290sec)	184546(133425)	37	7 min	980 sec	242263
$CityWall$	28	7 min	210sec(207sec)	126547(113174)	32	4 min	450 sec	179596

Figure 5. Table of data for the three test cases.

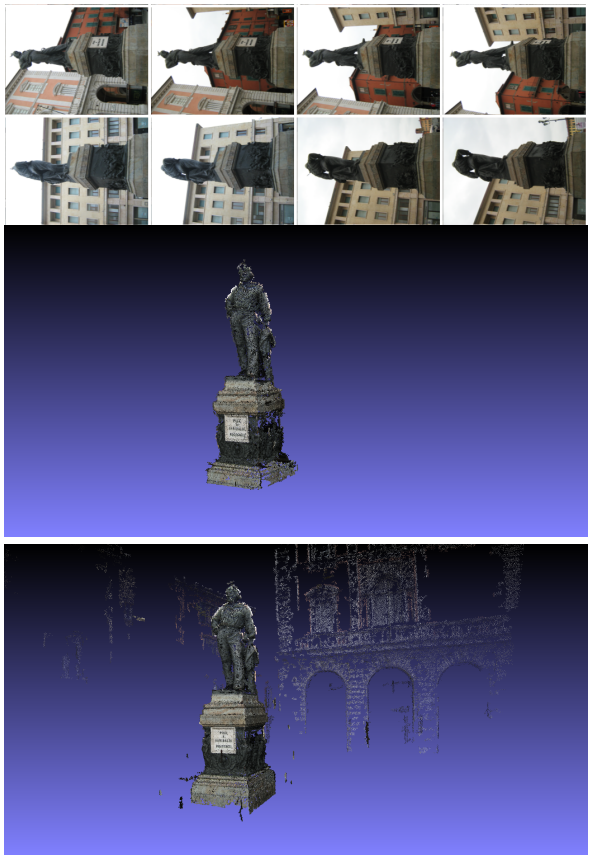

Figure 7. The Statue test set. First row, an example of some of the images used for the reconstruction. Second row, the reconstruction obtained with Our method (32 Images). Third row, the reconstruction obtained with VisualSfm+PMVS2 (37 images) [22]

Figure 8. The City Wall test set. First row, an example of some of the images used for the reconstruction. Second row, the reconstruction obtained with Our method (28 images). Third row, the reconstruction obtained with VisualSfm+PMVS2 (32 images) [22]

probably needs a more controlled situation, where the system asks for more images to strengthen the final 3D structure. Nevertheless, once again the background separation was able to preserve the whole object although the acquisition was not strongly focused on a portion of the scene.

In conclusion, the tests show that our system allows to process the input data in a incremental way and performs in a comparable way (if not better) w.r.t. one of the reference tools for Multi-view Stereo reconstruction. Further testing would be needed in the case of more complex acquisitions, dealing with hundreds of images. In order to do this, we need to improve the performances of the system to keep it quasi-realtime (see next Section).

V. Conclusion and future work

In this paper, we presented a framework for assisted reconstruction from images. The framework provides a sparse description of the scene in quasi-realtime, applying an incremental reconstruction that is updated every time a new image is available. The user can use the sparse reconstruction to check the completeness of the acquisition. Moreover, a novel analysis of the scene discriminates the object of interest and the background, and possibly suggests missing points of view, which will help in completing the sampling task. The framework proved to be effective regardless of the size and shape of the object of interest. The amount of processing make it impossible to work on current mobile devices, but the future improvements could include:

- An improvement of the incremental reconstruction

block, possibly implementing a fully incremental process, as described in Lourakis [10]. At the moment the system performs slowly when more than 100 images are taken into account.

- The parallelization of the reconstruction routine, without losing the need to have the system working on non-high-end devices.
- The extension of the system architecture to be used with mobile devices: in this case a client-server structure would be needed. The mobile could upload the images to a remote server, and retrieve the sparse reconstruction in a very short time.

REFERENCES

[1] M. Dellepiane, N. Dell Unto, M. Callieri, S. Lindgren, and R. Scopigno, "Archeological excavation monitoring using dense stereo matching techniques," *Journal of Cultural Heritage*, vol. available online, 2012, http://dx.doi.org/10.1016/j.culher.2012.01.011. [Online]. Available: http://vcg.isti.cnr.it/Publications/2012/DDCLS12

[2] J. Leonard and H. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," in *Intelligent Robots and Systems '91*, Nov, pp. 1442–1447 vol.3.

[3] A. Davison, I. Reid, N. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *Pattern Analysis and Machine Intelligence, IEEE Tr. on*, vol. 29, no. 6, pp. 1052–1067, June.

[4] G. Klein and D. Murray, "Parallel tracking and mapping on a camera phone," in *Proceedings of ISMAR '09*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 83–86. [Online]. Available: http://dx.doi.org/10.1109/ISMAR.2009.5336495

[5] K. Ho and P. Newman, "SLAM-Loop Closing with Visually Salient Features," in *IEEE International Conference on Robotics and Automation (ICRA)*, Apr. 2005.

[6] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3d," in *ACM SIGGRAPH 2006*. ACM, 2006, pp. 835–846. [Online]. Available: http://doi.acm.org/10.1145/1179352.1141964

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. C. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004. [Online]. Available: http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94

[8] S. N. Sinha, J. michael Frahm, M. Pollefeys, and Y. Genc, "Gpu-based video feature tracking and matching," in *Workshop on Edge Computing Using New Commodity Architectures*, 2006.

[9] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment - a modern synthesis," in *Proc of ICCV '99*. London, UK, UK: Springer-Verlag, 2000, pp. 298–372. [Online]. Available: http://dl.acm.org/citation.cfm?id=646271.685629

[10] M. I. A. Lourakis and A. A. Argyros, "Sba: a software package for generic sparse bundle adjustment," *ACM Transactions on Mathematical Software*, pp. 1–30, 2009.

[11] S. Agarwal, N. Snavely, S. M. Seitz, and R. Szeliski, "Bundle adjustment in the large," in *Proceedings of the ECCV'10*. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 29–42. [Online]. Available: http://dl.acm.org/citation.cfm?id=1888028.1888032

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

[13] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz, "Multi-view stereo for community photo collections," in *Proceedings of ICCV 2007*. Rio de Janeiro, Brazil: IEEE, 2007, pp. 265–270.

[14] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1362–1376, 2010.

[15] Y. Furukawa, B. Curless, S. M. Seitz, R. Szeliski, and G. Inc, "R.: Towards internet-scale multiview stereo," in *In: Proceedings of IEEE CVPR*, 2010.

[16] S. Wenhardt, B. Deutsch, E. Angelopoulou, and H. Niemann, "Active visual object reconstruction using d-, e-, and t-optimal next best views," in *CVPR*, 2007.

[17] E. Dunn and J.-M. Frahm, "Next best view planning for active model improvement," in *BMVC*, 2009.

[18] M. Trummer, C. Munkelt, and J. Denzler, "Online next-best-view planning for accuracy optimization using an extended e-criterion," in *Pattern Recognition (ICPR), 2010 20th International Conference on*, Aug., pp. 1642–1645.

[19] G. Kootstra, N. Bergstrom, and D. Kragic, "Using symmetry to select fixation points for segmentation," in *Pattern Recognition (ICPR), 2010*, Aug., pp. 3894–3897.

[20] N. D. F. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla, "Automatic 3d object segmentation in multiple views using volumetric graph-cuts," *Image Vision Comput.*, vol. 28, no. 1, pp. 14–25, Jan. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.imavis.2008.09.005

[21] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," *SIGGRAPH Comput. Graph.*, vol. 26, no. 2, pp. 71–78, Jul. 1992. [Online]. Available: http://doi.acm.org/10.1145/142920.134011

[22] VisualSfM, "A visual structure from motion system," Info on: http://homes.cs.washington.edu/ ccwu/vsfm/.

[23] Y. Furukawa and J. Ponce, "Accurate, Dense, and Robust Multi-View Stereopsis," *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pp. 1–8, 2007. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2007.383246