International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:10, 2014

A Fast and Robust Protocol for Reconstruction and Re-Enactment of Historical Sites

S. I. Abu Alasal, M. M. Esbeih, E. R. Fayyad, R. S. Gharaibeh, M. Z. Ali, A. A. Freewan, M. M. Jamhawi

***Abstract*—**This research proposes a novel reconstruction protocol for restoring missing surfaces and low-quality edges and shapes in photos of artifacts at historical sites. The protocol starts with the extraction of a cloud of points. This extraction process is based on four subordinate algorithms, which differ in the robustness and amount of resultant. Moreover, they use different -but complementary- accuracy to some related features and to the way they build a quality mesh. The performance of our proposed protocol is compared with other state-of-the-art algorithms and toolkits. The statistical analysis shows that our algorithm significantly outperforms its rivals in the resultant quality of its object files used to reconstruct the desired model.

***Keywords*—**Meshes, Point Clouds, Surface Reconstruction Protocols, 3D Reconstruction.

I. Introduction

3D reconstruction from images is a reverse process for obtaining 3D scenes from 2D images. Every 3D reconstruction method has its appropriate instruments that document shapes and poses of the targeted objects. The data retrieved from the instruments are processed in order to build a 3D model using specialized software [1]. Most of photo reconstructing methods are based on the projection of corresponding 3D points for specific images. These points will be the "sight" of the reconstructed model. It is challenging to determine which point is the starting point in the comparison with the corresponded point in another image. Thus, when working with two images the position of a 3D point could be determined as the result of the intersection of two projected rays. This method is called "Triangulation" [2]. The main idea of this method is the nexus between diverse views that carry the information that corresponds to sets of points that have some structure, and that structure is related to the different poses of the camera as well as its calibration.

In recent years, there has been a demand for 3D content creation. Unfortunately, many existing 3D model reconstruction software lack high fidelity capabilities. This weakness might be related to the lack in utilizing digital imaging facilities such as cameras. Therefore, if additional reconstruction methods are applied, we could provide more realistic models. Structure from motion (SFM) is Surface reconstruction software obtains 3D points from images by computing the three dimensional structure relying on motion data or information existing in the images captured from different angles [1]. This method generates the model from a set of photos.

The system takes a set of images as input and generates a 3D point cloud of the scene; the portion that was used of each photo, and calibration of the cameras. Then a good-enough dense point cloud is generated through triangulation. We use VisualSFM (VSFM) [1] that integrates and improves the following algorithms: an incremental SfM system, SIFT [3] on the GPU, and the Multicore Bundle Adjustment. For the sparse reconstruction, VisualSFM provides a simple interface to run. The theory of structure from motion allows projection matrices and 3D points to be computed using only corresponding points in each view. Triangulation and Projection could be applied by using other reconstruction software to have high quality models. The software that we are discussing in this paper are: SURE [4], CloudCompare [5] and MeshLab [6] which is based on the Ball Pivoting Algorithm [7]. These software contain many algorithms that differ in the influence, effectiveness, and robustness for holes and sharp edges. These differences will be discussed in the next sections of this paper.

II. Proposed Work

A. Visual Structure from Motion (VSFM)

VisualSFM [1] is an open source GUI that integrates and combines diverse SFM software. It is used for 3D reconstruction using structure from motion. It is capable of processing dozens of photos to create a dense cloud of points. VisualSFM runs efficiently due to its exploitation of multicore parallelism in feature detection, feature matching, and bundle adjustment. The most important features of VisualSFM are summarized as follows:

1. Compute Missing Image Matches: VisualSFM attempts to match all the photos with dependence on how the photos were taken. Any areas that cannot be cross matched will cause fragmentation to the model. This step detects image features and determines feature correspondences. It generates a connected graph like images with common

This work was supported in part by the ENPI CBCMED (International Augmented Med project) under Grant I-A/1.2/113

S. I. Abu Alasal, E. R. Fayaad, M. M. Esbeih are with the Computer Information Systems Department, Jordan University of Science & Technology, Irbid, Jordan 22110 (e-mail: sanaasal11@gmail.com, emanruba@gmail.com, madleen.mousa@gmail.com).

M. Z. Ali and R. S. Al-Gharaibeh are with the Computer Information Systems Department, Jordan University of Science & Technology, Irbid, Jordan 22110 (e-mail: mzali@just.edu.jo, , rami@just.edu.jo).

A. A. Freewan is with the Visual Communication and Design Department, Jordan University of Science & Technology, Irbid, Jordan 22110 (e-mail: aafreewan@just.edu.jo).

M. M. Jamhawi is with the Urban Planning Department, Jordan University of Science & Technology, Irbid, Jordan 22110 (e-mail: mjamhawi@just.edu.jo).

features; key points are processed in parallel to obtain their orientations and descriptors. This process is carried out using Scale Invariant Feature Transform (SIFT) algorithm; the GPU implementation of SIFT [3].

2. Compute the sparse 3D: Reconstruction runs bundler and generates a sparse point cloud. The relevant match among images is calculated to determine the 3D positions of each point in a relative coordinate system. This process is a precursor to the dense reconstruction, but is normally very quick. Dense matching and point cloud generation on the set of images a process known as PMVS/CMVS. PMVS2 is multi-view stereo software that takes a set of images and camera parameters as input and reconstructs the 3D structure of the scene as a dense points set. SFM can generate good coverage dense point cloud that can be triangulated using standard software such as MeshLab. Reconstructing multiple images using structure from motion can be divided into two distinct stages: feature matching and incremental structure from motion. The algorithm to find corresponding points among images is shown in Fig. 1. [1].
3. Bundle adjustment: Bundle Adjustment is used to estimate a set of related parameters that more accurately predict the locations of the observed points in the used images. Adjustment minimizes the re-projection error using nonlinear least-squares algorithms between the image locations of observed and predicted image points [9].

```
Repeat
For each feature point in the first image
{
   Compute the epipolar line in the second
   image
     Compute the epipolar line in the
      third image 4
     If the epipolar line intersects only
        one feature point{
        Compute the epipolar line for
        matching feature point in the
        other image.
        Intersection with epipolar line
        from first image gives the third
        point.
        Remove triplet from the List.
Until no feature point can be matched
uniquely.
```

Fig. 1 Algorithm to find corresponding points between images [8]

B. Surface Reconstruction (SURE)

SURE [4] is software for creating 3D point cloud based on the Semi-Global Matching (SGM) method, in which matching results of low resolution pyramids are used to limit ranges in disparity search for high resolution pyramids. We show that memory requirements as well as processing times can be significantly decreased when the quality of disparities estimations is reduced. Merging redundant disparity estimations of multiple stereo models increases the precision and robustness of the generated point cloud. Based on basic principles of epipolar geometry, it uses a time efficient algorithm for outlier detection and object point triangulation. It minimizes the re-projection error as a representation of an approach of a multi-view stereo (MVS) method [10]. The processing order in SURE based on its basic modules is illustrated in Fig. 2.

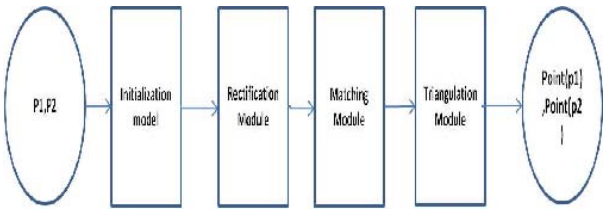

Fig. 2 Flow chart of main modules in SURE

Main modules in SURE can be summarized as follows:

1. **Rectification Module:** The first process in this module matches virtual images with the same centers as original, but with updates in rotations and internal parameters. The next step defines two 3×3 matrices, minimizes the rotation of the image planes, maps each point to the rectified image planes, having the same y-coordinates.

After deriving the hymnographies, the module calculates the grey values for pixels at positions in the rectified frames. Integer coordinates are mapped to the original images and the respective grey values are interpolated [2].

2. **Mathematical:** Let P_b and P_m be a pair of images to be rectified and P_b^r, P_m^r are the resultant epipolar (images P_b^r and images P_m^r) virtual images providing the same optical centers as original P_b and P_m but possess updated rotations and internal parameters. The next step is to define two 3×3 matrices M_b, M_r relating the homogeneous image coordinates ab, am in the original images and a_b^r, a_m^r in rectified images according to

$$a_b^r = M_b x_b \tag{1}$$

$$a_m^r = M_m x_m \tag{2}$$

The inverse mapping can be simply calculated as

$$a_b = M_b^{-1} a_b^r \tag{3}$$

$$a_m = M_m^{-1} a_m^r \tag{4}$$

Despite differences in approaches of the methods, original P_b and P_m are warped such that lines are horizontal and an arbitrary object point Z_i is mapped to the rectified image planes of P_b^r and P_m^r possessing the same y-coordinates, therefore

$$a_b^r(a_b^r, y^r, 1) = a_m^r(a_m^r, y^r, 1) \tag{5}$$

Also the optical rays and distances between object points and perspective can be calculated as:

$$C_m^r - Z_i = C_m - Z_i \quad (6)$$

$$C_{mb}^r - Z_i = C_b - Z_i \quad (7)$$

3. **Matching Module:** this module is used for image matching. It is a stereo method based on SGM but with extension to the classical approach. The input for this module is rectified images from the preceding rectification module. The output of this module is the raster data set representing the difference of each base image pixel with respect to the matched image. The distance D_b^r between camera centre C_b^r and object point X_r on the optical ray can be calculated as

$$D_b^r = \frac{B\sqrt{(x_b^r)^2 + (y_b^r)^2 + 1}}{d} \quad (8)$$

4. **Structure Computation (Triangulation) Module:** The inputs of the triangulation module are orientations of rectified/original base, match images, and the correspondent difference images. The output is a 3D point cloud or a depth image.

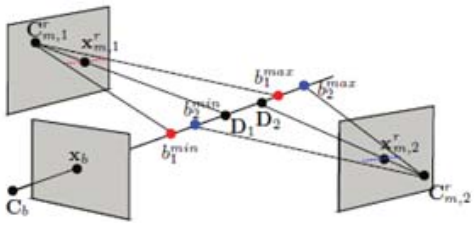

Fig. 3 Confidence intervals of disparity estimations

C. MeshLab

MeshLab is 3D mesh processing software that is well known in the fields of 3D development [6]. MeshLab is oriented to the management and processing of large and unstructured meshes of point triangulation. The software uses what is known as Octree data structuring. The automatic mesh cleaning filters include: removal of duplicated, unreferenced vertices, none manifold edges, and vertices and null faces. Remeshing tools [6] support high quality simplification that is based on quadric error measurement with various kinds of subdivision surfaces and two surface reconstruction algorithms. For noise removal, commonly present in acquired surfaces, MeshLab applies various kinds of smoothing filters and tools for curvature analysis and visualization. It registers multiple range maps based on the Iterative Closest Point algorithm (same as CloudCompare).

The Ball Pivoting Algorithm (BPA) [7] is a powerful heuristic for reconstructing a triangular mesh surface from a set of scattered 3D points. The method is strongly linked to the Alpha (α)-shapes theory [11] where sampling conditions for the Ball-Pivoting algorithm could be deduced from it.

1. **Data Structure of (BPA):** This method requires two important data structures. Firstly, a point sorting and searching structure to quickly access range neighborhoods of a given position. Secondly, surface meshes connectivity structure. This structure is a manifold with holes-type structure. The surface is guaranteed to be self-intersection free.
2. **Search Structure of (BPA):** To access neighboring points we use an octree as a search structure that divides the space into cells containing the points. The acceleration of the search structure is done through locational codes.

D.CloudCompare

CloudCompare is 3D point cloud processing software [5]. It handles triangular meshes and calibrated images. It provides a set of tools for editing; rendering 3D point clouds and triangular meshes. It offers advanced processing algorithms, such as methods for performing Projections (axis-based, cylinder or a cone unrolling) and Registration (Iterative Closest Point). Some of its features include:

1. Distance computation (cloud-cloud or cloud-mesh nearest neighbor distance "Hausdorff distance") [12]: measures how far two subsets of a metric space are from each other. It turns the set of non-empty compact subsets of a metric space into a metric space in its own right.
2. Statistics computation (spatial Chi-squared test).
3. Segmentation (connected components labeling, front propagation based).
4. Geometric features estimation (density, curvature and roughness). CloudCompare can handle unlimited scalar fields per point cloud on which various dedicated algorithms can be applied (smoothing, gradient evaluation, statistics, etc.). A dynamic color rendering system helps the user to visualize per-point scalar fields in an efficient way. Therefore, CloudCompare can also be used to visualize N-D data

III. Proposed Algorithm

We are presenting our proposed algorithm using a case of a historical artifact in a museum. Firstly, when the VSFM gets sufficient number of images for the historical artifact, it will search for all Featured Points (Fp) that exist in each image to compute the epipolar line in the other images. In the case of two or more images, the algorithm will clean the array that contains the images then uses the Fp and the value of the epipolar (C) in matching the intersections between the chosen Fp. If there is no Fp to catch then the RANSAC [8] algorithm will be used. The next step is to search for triplets to remove and that to eliminate the duplications. Those resulting Fp will be the output of the VSFM. Secondly, the .out file that results from the VSFM will be input to the SURE toolkit performing the model rectification. SURE uses the Fp and another data analyzed by the VSFM to generate the virtual images. SURE will then do model matching which uses the function of the updated rotation.

The point cloud (Pc) is generated by the orientation and

based on the distance between the rays (X) and the Fp. The registration method is used based on a special mode of rendering supposed to be the cylindering which is one of the OPENGL rendering modes. We should compute the distance between the Pc values. This data will be saved in a .las file that will be generated by the SURE toolkit and will then become the input of the CloudCompare software.

Both of CloudCompare and the MeshLab can build an Octree structure for the triangulation method of meshing. The two software packages differ in the algorithms that each uses for such purpose. Our next step is to use CloudCompare to check whether the point cloud in the .las file contains any background points in the model or they are merely points that are part of the model itself. It then builds the octree using a matrix that contains the Pc as neighbors (Pi) by searching in them using the RANSAC algorithm. If no neighbor around Pc is found, it will tack the point that resulted in the .las file.

Nonetheless, the resulting mesh is still not good enough in terms of quality and smoothness. Moreover, it has noisy points around the model which causes problems when separating the background points from the featured points of the cloud. Finally, the desired model is generated in the .obj file format to be subsequently fed as input for MeshLab software. MeshLab reconstructs the model again using the Ball-Pivoting Algorithm (BPA) which loops to get an active edge for the matrix. If the Pc is a seed for the triangle then it will start building the octree using the activated edge. Building the octree also is based on searching for neighbors for the activated seed. Following mesh development, a reconstruction function will run, that is based on finding the seed triangle again, then expanding a triangulation for the new seed and the extracted Pc. Then, it identifies the candidates to be used to find a vertex (v) to creating facet with the edge (r). The last step is to parallelize the points with the cells which can be processed simultaneously and duplicated in the same color. The pseudo-code that summarizes how the proposed protocol works is presented in Fig. 4.

IV. Experimental Setup and Analysis of Results

A. Experimental Setup

The experiments were conducted on a personal computer with an Intel Core i7-2640m CPU @ 2.80GHz processor, 4GB RAM, and 64-bit window 7. In VSFM we used multiple images of sample artifacts and computed 3D reconstruction as start. In SURE we used -fold 4 as a main parameter when reconstructing the models. In CloudCompare, the .las files which created by SURE were opened in point cloud to start the Distance computation by "Hausdorff distance". In MeshLab we used Surface Reconstruction by by Ball Pivoting method that started with 0 ball pivoting radius up to 0.574823 world unit. These values are limited by the hardware specification. We attempted to develop the model with several trials which are explained as follows:

1. VSFM and MeshLab: we took nine photos for the sculpture "Tomb Stone" at Dar-As-Saraya Museum in Irbid city in Jordan. The VSFM software processed the images and produces .out file which fed input for the (BPA) in the MeshLab. Fig. 5 shows the resulting model. The poor model highlights the importance of applying the (MVS) methods in reconstruction with the combination of CloudCompare meshing techniques and comparison of the point clouds.
2. VSFM with SURE and CloudCompare: after processing the .out file in the SURE software and determining the minimum number of points that must be observed in the stereo model, we used the .las file as an input for the CloudCompare. The resulting model is shown in Fig. 6.

Proposed Algorithm protocol to reconstruct models using photos:
Where (F_p) is the featured point, (D) the distance, (X) the ray, (P_c) the point cloud, (P_i) the neighbor, (r) the active edge, (v) the vertex.

```
getImage(image)
  foreach F_p in image
   C=right_epipolar(image)
   if C>=1 then
     clean_list()
     match_intersections(C,F_p)
     Remove_triplet()
     generate_virtual_images(C,F_p)
     update_rotation(F_p)
     P_c=orientation(D,X, F_p)
     register(P_c, "cylinder")
     compute_distances(P_c)
   end if
  end
     for P_c in P_i
           if P_i [row][column] is not background then
             if neighbors is empty then
            newP_i=P_i[row][column]+P_i[nextrow][nextcolumn]
             end if
            while true
         r=get_active_edge(newP_i)
               if P_c=findSeedTriangle() then
                  buildOctree(P_c,r)
                  getNeighbors(P_c,r,D)
                  Reconstruct(P_c,r)
                  T=fineSeedTriangle(P_c,r)
                  T_t=ExpandTriangulation(T,P_c,r)
                  v=findCandidate(e,P_c,r)
                  Paralleliz_Points()
               else
               end if
                  mark_as_boundary()
             end
           end if
   end
  return
```

Fig. 4 Algorithm to reconstruct models

Fig. 5 MeshLab output using ball-pivoting and poisson

The default value of the (-fold) command is 3 and was not manipulated. However, for a better resolution and enhanced appearance, this value must be increased. Even an increase by only 1 (-fold parameter = 4) is sufficient to leverage the quality of the produced model.

Fig. 6 CloudCompare output model using Poisson algorithm

3. VSFM, SURE, CloudCompare and MeshLab: this trial combines all the previous steps in one and applies the proposed protocol as indicated in the algorithm in Fig. 4. In this trial we noticed that SURE does outstanding job in the reconstruction process using the control files, and in running the individual modules like rectification, matching (SGM) and triangulation.

CloudCompare is to visualize the *.las* file by producing a set of points for each resulting image and is then superimposed on the 3-*D* plane. CloudCompare also had the ability to convert the data to another format to be processed by MeshLab. MeshLab applied the (BPA) for surface reconstruction, thus bringing higher smoothness and noiseless points. Finally, we applied cleaning and healing tools. The version of the model that was developed using our protocol is shown in Fig. 7. Evidently, this is a clearer and smoother version with minimum number of holes.

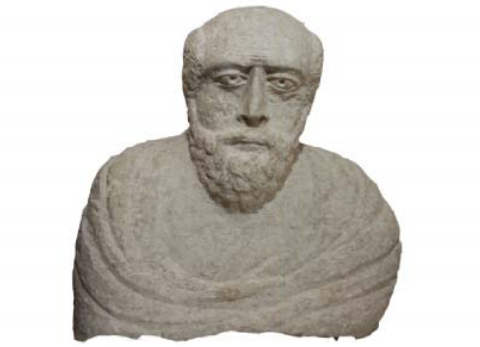

Fig. 7 Proposed protocol output model

B. Results and Analysis

The first experiment produced a model of poor quality, suffering surface incompletion and with far distances among the extracted points. There were 1953 vertices that resulted from this process. In the second experiment the effect of SURE reconstruction methods became tangible, especially with increasing the (-fold) parameter value to 4. CloudCompare visualized the resulted .las file and increased the resolution, yet edges and were still present. Increasing the points size two times resulted in vertices values of 581976, 622593 (average of nearly 602248 vertices). In the third experiment (Fig. 7) we merged the two previous experiments into one to gain the result of 4280651 vertices in average after applying the (BPA) five times using the same number of photos and the same parameters values.

Table I shows a comparison between the experimental setups and the obtained results. Combining the four toolkits in the order specified and with the suggested parameter values produces an average value of 4280651 vertices without noisy points or sharp edges and with smoother surfaces. MeshLab can use both of the Poisson and the (BPA) to reconstruct the surface while CloudCompare use only Poisson. Thus, in the proposed protocol both of them are used after reconstructing the model by the indicated toolkits.

TABLE I
COMPARISON BETWEEN RECONSTRUCTION PROTOCOLS

Concept	Protocol		
	Meshlab	***CloudCompare***	***Proposed Protocol***
Vertices	1953	602284[a]	4280651[a]
Reconstruction algorithm	Poisson, Ball-Pivoting	Poisson	Both Poisson, BPA and others

[a]: is the average vetrices resulted

Fig. 8 shows a plot of the differentiation of number of vertices in each experimented protocol. It is evident that the proposed protocol has the highest number of vertices compared to MeshLab and CloudCompare. These huge gaps between the commonly-used protocols are attributable to the use of the (MVS) methods that exist in the SURE. These methods can collect more featured point in the point cloud that would be visualized by the comparing methods and distance computation using the CloudCompare.

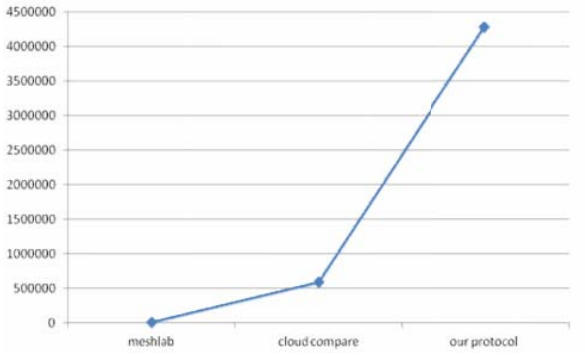

Fig. 8 Number of vertices in each protocol

After applying Ball-Pivoting algorithm for 5 times using MeshLab on the model that was created by CloudCompare, the number of vertices dramatically increased as shown in Fig. 9. This is a consequence of building the Octree five times which increased the correlation of the triangulated mesh.

V. Conclusions

In this paper, we introduce a new protocol that represents a novel method for reconstructing models of historical artifacts. Our method is based on generating point clouds from photos taken by a digital camera. The points in a cloud are triangulated using meshing algorithms and techniques with octrees structuring. The proposed protocol provides a good dense sample of points collected as a smooth model.

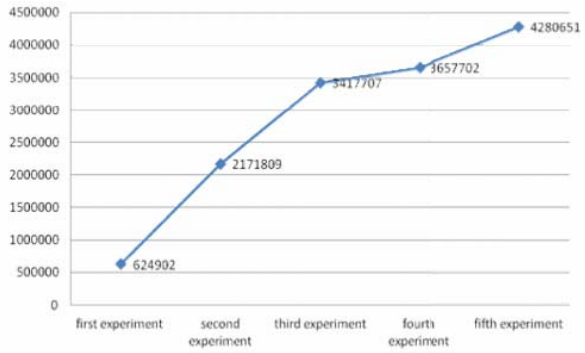

Fig. 9 Number of vertices for each experiment in our protocol

The quality of the density points depends on using the Ball-Pivoting Algorithm that reconstructs the messing surfaces. The quality of produced model is noiseless with lower number of holes. These improvements are the result of the combined framework providing all necessary tools to cut the noisy edges and bad points. The experiments and analysis of the results demonstrate CloudCompare's effectiveness in producing the necessary .obj file as input for MeshLab. This shows that CloudCompare is adequate for increasing the number of simulated points for the used images. Moreover, CloudCompare is sufficient when collecting points for the used images, while MeshLab is more appropriate for model reconstruction.

Acknowledgment

We give our thanks to the Department of Antiquities in Jordan for facilitating our access to Dar-As-saraya Museum in Irbid. Also, we express our gratitude to the authors of the open-source software related to this research. Without their product, we could not have comprehended the algorithms that each step of our reconstruction process is based on; including Virtual SFM, SURE, CloudCompare, MeshLab.

References

[1] J.C. Torres, G. Arroyo , C. Romo , J. De Haro, "3D Digitization using Structure from Motion" . CEIG - Spanish Computer Graphics Conference, 2012.
[2] R. Hartly, P. Sturm "Triangulation, Computer Vision and Image Understanding" Volume 68, Issue 2, November 1997, pp. 146–157.
[3] Jiang, X. Li, and G. Zhang, "SIFT Hardware Implementation for Real-Time Image Feature Extraction," Circuits and Systems for Video Technology, 2014.
[4] Mathias Rothermel, Konrad Wenzel, Dieter Fritsch, Norbert Haala, "SURE Photogrammetric surface reconstruction from imagery," unpublished.
[5] F. Memoli, G. Sapiro, "Comparing point clouds", SGP '04 Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing.
[6] Cignoni, M. Callieri, M. Dellepiane, F. Ganovelli, G. Ranzuglia "MeshLab: an open-source mesh processing tool" Eurographics Italian Chapter Conference (2008).
[7] F. Bernardin, J. Mittleman, H. Rushmeier "The Ball-Pivoting Algorithm for Surface Reconstruction," IEEE Transaction on Visualization and Computer Graphics, Vol. 5, No.4, October-December1999.
[8] O. Chum, T. Werner, and J. Matas, "Epipolar geometry estimation via RANSAC benefits from the oriented epipolar constraint," in Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference, 2004, pp.112-115 Vol. 1.
[9] S. Holmes, G. Sibley, G. Klein, and D. W. Murray, "A relative frame representation for fixed-time bundle adjustment in SFM," in Robotics and Automation, 2009, pp. 2264-2269.
[10] T. Tung, S. Nobuhara, and T. Matsuyama, "Complete multi-view reconstruction of dynamic scenes from probabilistic fusion of narrow and wide baseline stereo," in Computer Vision, 2009 IEEE 12th International Conference (2009), pp. 1709-1716.
[11] S. Park, S. lee, J. Kim "A surface reconstruction algorithm using weighted alpha shapes" FSKD'05 Proceedings of the Second international conference on Fuzzy Systems and Knowledge Discovery - Volume Part I, pp. 1141-1150.
[12] M. Tang, M. Lee, Y. Kim, "Interactive Hausdorff Distance Computation for General Polygonal Models" ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2009 Volume 28 Issue 3, August 2009 Article No. 74.

Sanaa I. Abu Alasal was born in (Irbid Jordan, 19/11/1992). Has a Bachelor Degree in computer information systems from Jordan University of Science and Technology Irbid, Jordan, 2014.

She worked as a Mobile Augmented Reality Developer for International Augmented Med Project at Jordan University of Science and Technology 2013-2014. Currently, she works as an office administrator and researcher at CTS Jordan Company.

Ms. Abu Alasal has been awarded the best poster presentation award at the sixth annual undergraduate research conference on applied computing, Zayed University April 30- May 1, 2014, Dubai, UAE.