3D Construction From a Sequence of Images

A. ALLOUACHE(1), A. BAZOULA(2), M.S. DJOUADI.
Laboratoire Robotique et Productique
UER Automatique
École Militaire Polytechnique.
alouache_12@live.fr (1), abdelouahab.bazoula@gmail.com (2)

Abstract **-** Building 3D models for the real world from images has attired a lot the intention of researchers in the last years, because this does not require expensive devices, just cameras for the acquisition of images, and this can be viable when it is not possible to use active sensors such as laser scanner. In this paper, we present the full pipeline of building a 3D model from multiple images using a mobile camera. Our algorithm requires a sequence of images taken by an intrinsically calibrated camera, then we estimate the camera positions from the images and a 3D point cloud of the captured scene. From the point cloud, we construct a mesh model then we finish our work by mapping a texture for the mesh model to generate a textured 3D model.

Index Terms **-** *SURF, SFM, Texture Mapping.*

I. INTRODUCTION

Building a realistic 3D model for the world has attired a lot the attention of researchers since decades. Even it seems easy to create a simple 3D model using a software like Maya or 3D Max, however the generation of a photorealistic complex object needs more efforts. Automatically reconstructing such 3D models requires first acquisition devices that can extract a 3D information of the scene. Then further modeling, computation and algorithms are usually required. For a long time, Laser scans (or similar techniques) have been extensively used, for example to produce 3D models in computer graphics. Usually associated with geometry post-processing (such as recovering smooth polygonal surfaces from point clouds), they provided nice results that are still used nowadays as references [1] or ground truth data. On the other hand, cameras provide more information such as color and textures. Used in stereo pairs or in multi-view settings with correct geometry modeling, they provide useful information. This can be used to reconstruct shapes in environments where active systems would not work, such as the reconstruction of large scale objects or environments [2].

The work presented in this paper is organized as follows: in section II we give a short review about the most important approaches and techniques that aim to solve the problem of building a 3D model for the real world, and the remainder of this paper focuses on the structure from motion (SFM) technique, which is implemented here. We assume that we have a sequence of images, taken by a partially calibrated camera see section III. Section IV presents all the necessary concepts, tools for implementing the 3D structure from motion algorithm, like features detection, matching which is the basic part of any SFM algorithm, the pseudo code of the implemented SFM algorithm will be presented in this section. Section V assumes that we dispose a colored 3D point cloud from which we see how to build a mesh model and generate a textured 3D model. Experimental results are given in section VI and conclusions are given in section VII.

II. TECHNIQUES

A. Shape from X

Shape from X is a generic name for techniques that aim to extract shape from intensity images. The approaches that uses a single image tries to extract a certain characteristics in order to determine the depth information of the captured scene. This characteristic information may be brightness (shape from shading) [3], texture deformation (shape from texture)[4], and variations in the focus parameters (shape from Focus/Defocus) [5].

B. Stereo vision

The principle of using stereo images is inspired from the human vision, which captures depth information using different sensory cues [6]. The stereoscopic vision uses the information obtained from the projection of the object, which is observed from two different views; the displacement of the two images can be used to triangulate the position of the object. Stereo vision assumes that we know the position of the stereo cameras in space, the distance that is separating the two cameras' centers. The stereo system must solve two problems [7], the first is stereo matching by meant which parts of the

stereo images are similar to each other and the second one is the 3D reconstruction by triangulation.

C. *Shape from motion*

Shape from motion (already called structure from motion SFM or structure and motion estimation SAM), SFM refers to the process of estimating simultaneously the 3D geometry of the scene (structure) and the positions of the cameras (motion). It exploits the matching points between the images. Fig 1. shows an example of a 3D model constructed by moving the camera into 8 different positions in the scene.

The first work on SFM including two and multi views structure appeared since the 1980s by Longuet-Higgins [8], a relative orientation evaluation technique was introduced.

The development of the technical SFM for the multi views had been presented after, this multi views include the methods of factorization [9] and the methods of global optimization [10], [11], [12].

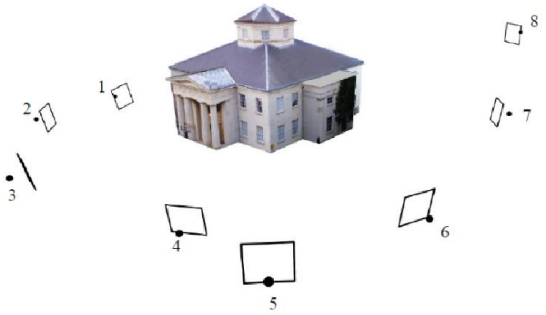

Figure 1. Structure and motion estimation

Later, bundle adjustment [13] in photogrammetry has made its way toward computer vision, for an optimal estimation of 3D geometry and the cameras parameters [14]. SFM has been the most popular algorithm for the 3D modelling from images and it is based on finding matching points between the images. The most SFM algorithms are cited in the literature [15], [14].

III. PARTIAL CAMERA CALIBRATION

A. *Pinhole camera model*

The simplest camera model and the widely used one in computer vision is the pinhole model of Fig 2. proposed by Hall [16]. The internal geometry and the position and orientation of the camera in the scene are modeled. It defines the basic projective imaging geometry with which the 3D objects are projected onto the 2D image plane.

The camera can be modelled by a set of intrinsic and extrinsic parameters. The intrinsic parameters are those that define the optical properties of the camera such as the focal length, the aspect ratio of the pixels, and the location of the image centre where the optical axis intersects the image plane. The extrinsic parameters define the position and orientation (pose) of the camera with respect to some external world coordinate system.

In this model, a scene view is formed by projecting 3D points into the image plane using a perspective transformation.

$$sx = K[R \quad T]X \tag{1}$$

Where: R (3x3) rotation matrix, T (3x1) translation vector and K (3x3) is a camera matrix, or a matrix of intrinsic parameters.

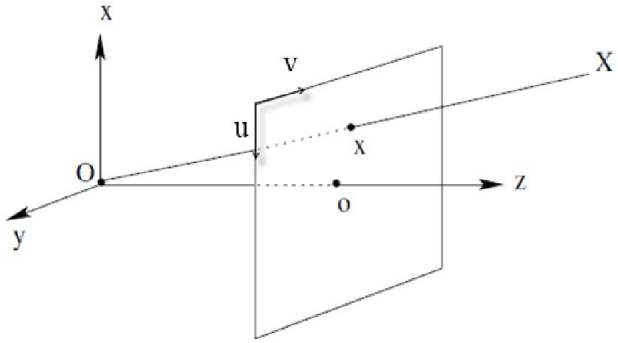

Figure 2. The Pinhole camera model

Equation (1) may be rewritten as:

$$\begin{bmatrix} su \\ sv \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2}$$

B. *Camera calibration*

The calibration means the determination of the projection matrix of equation (3) thus computing the different intrinsic and extrinsic camera parameters. Many standards methods exist for doing calibration [7] such as Tsai, Zhang, Faugeras and Toscani…the principle of these methods is using a calibration pattern, that disposes some features with known geometry and they should be easily extracted from their corresponding calibration images. Camera calibration for a single and stereo camera can be done by a Matlab toolbox [17].

Since we are working to estimate the camera motion from the image sequence so at the beginning we need just a partial calibration by meant the intrinsic camera matrix K presented in equations (1) and (2) then from each image we estimate the camera motion , thus the corresponding rotation matrix and translation vector.

IV. 3D STRUCTURE AND CAMERA MOTION ESTIMATION

In this section, we present how to estimate the 3D structure of the scene and the camera motion from a sequence of images. At the end of this section we present the implemented SFM algorithm, the basic part of our algorithm consist of features detection and matching , here we interest to points as type of features instead of lines because the process is complex so the

points are easier to manipulate and they don't take a lot of memory size.

A. Features detection and matching

Feature detection and matching techniques are important in many applications in computer vision and are core areas in 3D reconstruction. Some early development introduced finding feature points that are corner-like [18]. Interestingness could also be evaluated as high change of intensity using various sliding window functions and this is the backbone of the work done by people like Förstner [19] and then Harris and Stephens [20]. While eigenvalue based detectors like this, such as the Harris detector has rotation invariance, they are non-invariant to image scale.

More modern techniques such as Lowe's Scale Invariant Feature Transform (SIFT) use sampling of scale space to detect points that are invariant to changes in both scale and rotation [21]. This inspired the development of the Speeded Up Robust Feature (SURF) detector and has some gains in speed and robustness [21]. Features matching is done by comparing the Euclidean distance between the vectors descriptors and those that have the minimum distance their corresponding points are considered matching points.

B. Fundamental and essential matrices estimation

The fundamental matrix (denoted by *F*) and the essential matrix (denoted by *E*) are two very useful mathematical objects for 3D reconstruction. They are mostly similar, except that the essential matrix is assuming usage of calibrated cameras.

a) Fundamental matrix

Given two images with a set of correspondence features, then the image points satisfy the relation

$$x'^{T} F x = 0$$

Where *F* is the fundamental matrix of dimension 3x3 and rank 2. Finding 7 or more good matching points will allow for the estimation of the Fundamental matrix which would describe the necessary of the so called epipolar geometry [13].

Fig 3. shows the concept of epipolar geometry where a plane could be bounded at the 3D location of the point and two other locations of where this point appears on two pictures. All possible configurations of this plane will include epipolar line segments that intersect at an epipole on each image. This means knowing where the points are on an epipolar line segment means its partner is on the corresponding epipolar line segment of the second pictures.

b) Essential matrix

The essential matrix, a 3x3 sized matrix, imposes a constraint between a point in one image and a point in the other image with

$$x'^{0} E x = 0$$

Another important fact we use is that the essential matrix is all we need in order to recover both cameras for our images, although only up to scale; but we will get to that later. So, if we obtain the essential matrix, we know where each camera is positioned in space, and where it is looking.

C. Essential matrix decomposition

Four possible decompositions of the essential matrix [11]:

$$P_1 = [I \quad 0] \quad P_2 = [R \quad T]$$

$$P_1 = [I \quad 0] \quad P_2 = [R^0 \quad T]$$

$$P_1 = [I \quad 0] \quad P_2 = [R \quad ¡ \quad T]$$

$$P_1 = [I \quad 0] \quad P_2 = [R^0 \quad ¡ \quad T]$$

In order to select the correct pair of cameras projection matrices we should triangulate a set of matching points (see section *D*) and we take $P_1; P_2$ the pair that is generating 3D points in front of the two cameras see Fig 4. where the best solution is the one in the upper left.

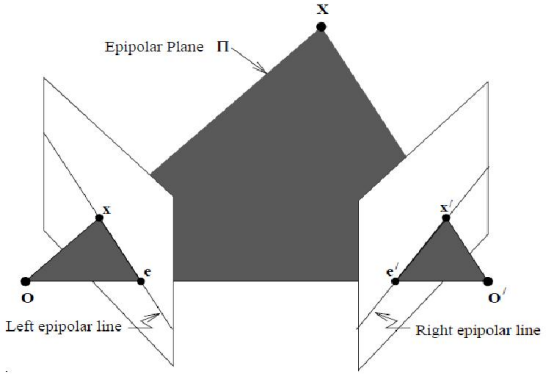

Figure 3. The epipolar geometry

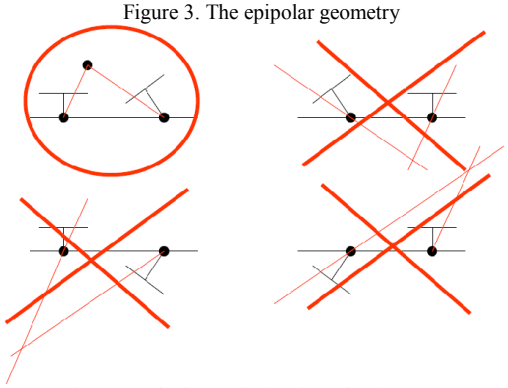

Figure 4. The best solution for selecting P_1, P_2

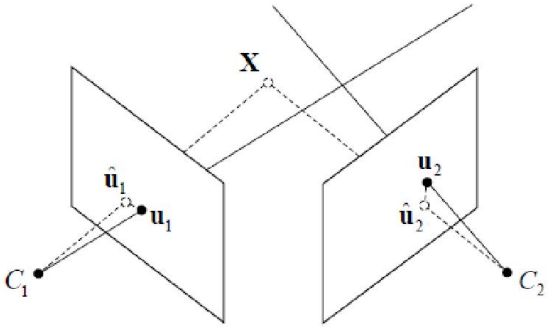

Figure 5. Triangulation illustration

D. 3D point triangulation

Up to this step, we have recovered the two camera projections matrices from the essential matrix and we dispose the set of matched points from the two views, we can estimate the 3D structure of the matched points by triangulation [22]. As an illustration let us look to Fig 5 a 3D point X can be computed from its measured pixel positions $(u_1; u_2; :::)$ in two or more views $(C_1; C_2; :::)$. Ideally, X should lie at the intersection of the back projected rays (solid lines). However, because of measurement noise, these rays will not generally intersect. Hence X should be chosen so as to minimize the sum of squared errors between measured and predicted pixel positions.

$$X = \arg\min_X \sum_{i=1}^{\Gamma} \| u_i - \hat{u}_i(P_i; X) \|^2$$

Where u_i and $\hat{u}_i(P_i; X)$ are the measured and predicted image positions in view *i*.

E. Bundle adjustment

One of the most important part of an SFM method is refining and optimizing the reconstructed scene, also known as the process of Bundle Adjustment (BA) [23]. This is an optimizing step where all the data we gathered is fitted to a monolithic model. Both the position of the 3D points and the positions of cameras are optimized, so reprojection errors are minimized (that is, approximated 3D points are projected on the image close to the position of originating 2D points). Fig 6. shows a simple setup with 3 images and 4 3D points that are visible in every image. The cameras and 3D points have been recovered from the 2D features extracted in the images, The process of bundle-adjustment aims at minimizing the reprojection error of the reconstructed 3D points in the images using the computed cameras.

F. The overall SFM algorithm

The overall problem of constructing 3D points out from multiple images can be broken down into two main steps: Initial 3D model from two images, and Refinement through adding more images. The initial 3D model is important as its accuracy will determine the quality of the model as more points are added. Additional points are added relative to the initial model as it serves as the base where everything is built upon. Every addition is based on the previous result and the process is iterative. The overall problem broken up into steps is outlined SFM Algorithm

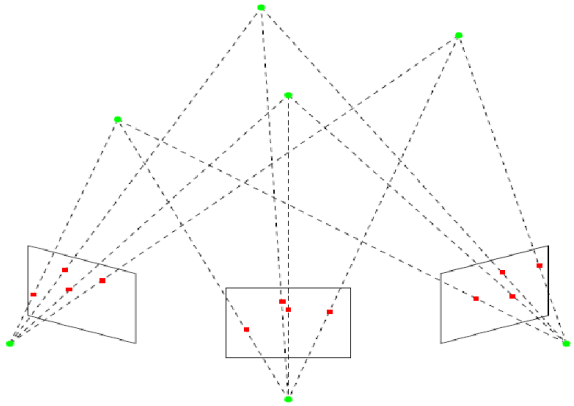

Figure 6. Bundle adjustment process

```
// SFM Algorithm
Input: K (3x3),
      Sequence of n images, n not large
For the 1st image and the 2nd image
       1. Detect features in each image
       2. Match features
       3. Estimate F12
       4. Deduce E12
       5. Decompose E12 to get P1, P2.
       6. Reconstruct the 3D points
       7. Get the colors of the 3D points.
   end
For i=3 to n do
       For images i and i+1
              Repeat steps 1-7
           end
      8. Refine the overall process by
        bundle adjustment
end
Output: n (3x4) camera matrices,
        (.ply) file containing the
        Colored 3D point cloud.
```

V. Mesh Building and Texture Mapping

Once we have a colored 3D point cloud, we can proceed to mesh building or usually called model fitting which gives a polygon mesh from the point cloud that gives another type of scene representation. We used MeshLab software [24], which is widely used for 3D point cloud processing like selecting, smoothing and coloring meshes, surface reconstruction and texture mapping. Texture mapping means we fill each patch of the mesh by a texture patch from the different images.

VI. Experimental Results and Simulation

Here we present our experimental results obtained using different datasets so visual results and some statistical results are given. We worked with OpenCV 2.4.6 that disposes the most functions that we need in our SFM algorithm.

A. *Visual results*

It is important to mention the hardware specifications that we worked with, so we used a computer that disposes 4 GB of RAM, CPU of Intel i3 generation, Intel graphic card.

The obtained results of the constructed 3D models are shown in Fig 7. We have taken 3 examples of sequence of images.

B. *Results evaluation*

We have tried examples of different scale scenes where scene1 is a small object, scene 2 is a wall and the last one is a building. An important point is that our 3D reconstruction is up to scale means we don't specify the real measurements directly from the point cloud.

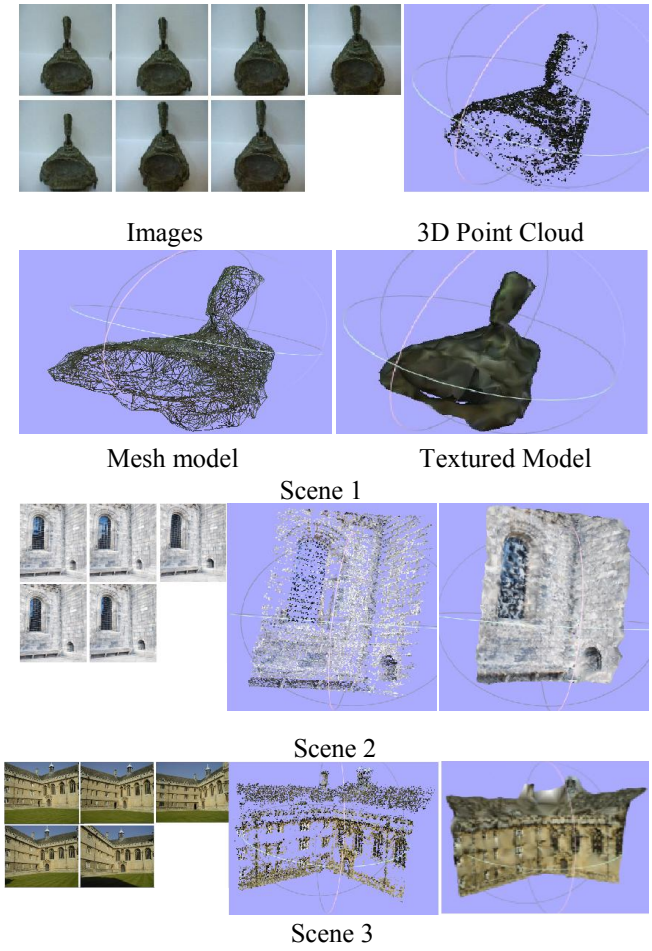

Figure 7. The constructed 3D models

While working with 3D data, we cannot say that the result is correct by looking simply at reprojection error measures or raw point information. On the other hand, if we look at the point cloud itself we can immediately verify whether it makes sense or there was an error.

We have taken a short sequence of images, but using a long sequence requires sometimes using for example GPU processor, for accelerating the process. The size of images in each sequence plays also an important role in the process

VII. CONCLUSION

We have presented the complete process of building 3D models for different real scenes, using sequence of images taken by an intrinsically calibrated camera. Based on estimating the camera motion and the 3D structure of the scene, we reconstruct a 3D point cloud; then we exported it as a binary file to the software MeshLab, for building a textured 3D model that can be used in different 3D file formats. We have implemented an incremental SFM algorithm means we treat the images in a sequential manner. The result of structure from motion can be exploited in different applications like: gaming and computer graphics, augmented reality and image based 3D modelling as in our case, robotic for autonomous navigation and guidance. A lot of modern military applications requires gathering as much as possible information from different environments, the SFM will be very helpful application.

Further perspectives on this work can be trying videos as input instead of images, then video processing tools are necessary for sampling frames then filtering good frames for the 3D reconstruction.

The camera is calibrated in advance; however automatically calibrating the camera from images is very interesting, this especially when using images taken by different cameras instead of a single camera.

Moving objects in the scene would cause the program of reconstruction to fail; possibly most of research in this field will be geared toward non-rigid structure from motion.

References

[1]. R. Furukawa, H. Kawasaki, R. Sagawa, and Y. Yagi. Shape from grid pattern based on coplanarity constraints for one-shot scanning. IPSJ Transaction on Computer Vision and Applications, pages:139–157, 2009.

[2]. The Stanford 3D scanning repository http://graphics.stanford.edu/data.

[3]. Zhang R, Tsai, P.S, Cryer J. E, and Shah M. Shape from shading: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8), pages:690–706,1999.

[4]. White R, Crane K, and Forsyth D.A. Capturing and animating occluded cloth. ACM Transactions on Graphics, 26(3), 2007.

[5]. Nayar S. K, Watanabe M, and Noguchi M. Real-time focus range sensor. IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(12), pages:1186–1198, 1996.

[6]. Reichelt S, Haussler R, Futterer G, Leister N: Depth cues in human visual perception and their realization in 3D displays. In three-dimensional visualization and display, 2010.

[7]. Emanuele trucco, Alessandro Verri. Introductory techniques for 3D computer vision. Edition 1998.
[8]. Longuet-Higgins, H.C. A computer algorithm for reconstructing a scene from two projections. Nature, 293(5828),pages:133-135, 1981.
[9]. Tomasi C, 1992. Shape and motion from image streams under orthography: a factorization method. *Int. J. Comput. Vis*, 9(2), pages:137-154, 1992.
[10]. Spetsakis M, Aloimonos J. A multi-frame approach to visual motion perception. Int. J. Comput. Vis, 6(3), pages:245-255,1991.
[11]. Szeliski R, Kang S.B. Recovering 3D shape and motion from image streams using nonlinear least squares. J. Vis. Commun. Image Represent. 5(1), pages: 10- 28,1994.
[12]. Oliensis J. A multi-frame structure-from-motion algorithm under perspective projection. Int. J. Comput. Vis, 34(2-3), pages:163-192, 1999.
[13]. Triggs B, Mclauchlan P, Hartley R, Fitzgibbon A. Bundle adjustment: a modern synthesis. LNCS pages:298-375, 2000.
[14]. Hartley R, Zisserman A. Multiple View Geometry in Computer Vision. Cambridge University Press,Cambridge, UK, 2004.
[15]. R Szeliski. Computer Vision: Algorithms and Applications. Springer, 2011.
[16]. E. L Hall, J. B. K Tio, C. A McPherson, and F. A. Sadjadi, "Measuring curved surfaces for robot vision", Computer Journal, pages: 42-54, December 1982.
[17]. Jean-Yves Bouguet, Camera Calibration Toolbox for Matlab. www.vision.caltech.edu.
[18]. Moravec H. The Stanford cart and the CMU rover. Proceedings of the IEEE, 71(7), pages: 872-884, 1983.
[19]. Förstner W. A feature-based correspondence algorithm for image matching International Archives Photogrammetry & Remote Sensing, 26(3), pages:150–166,1986.
[20]. Harris C, Stephens M. J. A combined corner and edge detector. In Alvey vision conference, pages:147–152, 1988.
[21]. Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pages:346-359, 2008
[22]. R. I. Hartley and P. Sturm. Triangulation. In American Image Under- standing Workshop, pages 957-966, 1994.
[23]. Triggs B, Mclauchlan P, Hartley R et Fitzgibbo, «Bundle adjustment: a modern synthesis» LNCS, vol. 1883, pages. 298-375, 2000.
[24]. http://meshlab.sourceforge.net/