Journal of AL-Qadisiyah for computer science and mathematics Vol.10 No.1 Year 2018
ISSN (Print): 2074 – 0204 ISSN (Online): 2521 – 3504

2D to 3D conversion using a pair of images
"KLT algorithm as a case of development and study"

Sarmad Nihad Mohammed

University of Kirkuk

Department of Computer Science, College of Computer Science

Sarmad_mohammed@uokirkuk.edu.iq

Recived : 13\11\2017 **Revised : 30\11\2017** **Accepted : 18\12\2017**

Available online : 26 /1/2018

DOI: 10.29304/jqcm.2018.10.1.352

Abstract., This paper investigate how motion between two images is affecting the reconstruction process of the KLT algorithm which we used to convert 2D images to 3D model. The reconstruction process is carried out using a single calibrated camera and an algorithm based on only two views of a scene, the SFM technique based on detecting the correspondence points between the two images, and the Epipolar inliers. Using the KLT algorithm with structure from motion method shows the incompatibility of it with the widely-spaced images. Also, the ability of reducing the rate of reprojection error by removing the images that have the biggest rate of error. The experimental results are consisting from three stages. The first stage is done by using a scene with soft surfaces, the performance of the algorithm shows some deficiencies with the soft surfaces which are have few details. The second stage is done by using different scene with objects which have more details and rough surfaces, the algorithm results become more accurate than the first scene. The third stage is done by using the first scene of the first stage but after adding more details for surface of the ball to motivate the algorithm to detect more points, the results become more accurate than the results of the first stage. The experiments are showing the performance of the algorithm with different scenes and demonstrate the way of improving the algorithm where it found more points from images, so it builds more accurate 3D model.

Keywords. SFM, Conversion Algorithms, 2D into 3D, Computer Vision, KLT algorithm.

1.Introduction. The ability of the vision of living creatures in receiving the real world as a three-dimensional scene motivates pioneers of the computer vision community to determine methods to simulate this ability. The solutions to this problem are divided into two groups, the first by acquiring a three-dimensional model directly from the real world by using special cameras such as a stereoscopic dual-camera with the ability to generate a three-dimensional model directly from a real-world scene. The second is by using two-dimensional data as inputs for algorithms designed particularly for the conversion of two-dimensional models into three-dimensional models. The role of these algorithms is to reconstruct a three-dimensional model based on the structure of the two-dimensional data which is missing the third dimension (the depth information) of the real world. The missing depth information is the result of the inadequacy of the traditional camera to obtain the third dimension from a captured scene, hence the role of algorithms to overcome this problem.

2. Why Do We Need to Convert the Two-Dimensional into Three-Dimensional. In general, there is more than one reason to convert two-dimensional images into three-dimensional models. The enormous amount of two-dimensional data in the past and the present in addition to the traditional devices for capturing scenes from the real world are the most important reasons. At this point, a trend where the role of conversion algorithms from 2D to 3D for generating three-dimensional models is becoming more popular. The accuracy of these algorithms, which differ from each other, depends on elements such as time consumption and the precision of the output model. [1]

The way that the mind behavior to generate the 3D model from the real world are considered as a base to compute the 3D geometry from 2D geometry or the structure from motion. The nonlinear approach is a technique which is employ this behavior to recover the structure and motion by minimize the value of the nonlinear cost function [2].

3. Challenges Facing Conversion Techniques. The challenges facing the techniques of conversion from the two-dimensional model to the three-dimensional model are divided into two groups. The first group covers every algorithm and several problems which must be solved by applying these algorithms. The second group of challenges involves specific types of algorithms considered to be high quality conversion techniques.

The first group of challenges includes three tasks which are solvable with every conversion algorithm. These tasks include [3][4]:

- **Apportionment of depth.**
- **Check of convenient disparity**
- **Padding of the exposed regions**

The second group (as shown below) of these challenges could be named as typical problems, which require high quality conversion algorithms in order to execute them. Those problems such as:

- **Semi-transparent objects (glass)**
- **Repercussion**
- **Foggy translucent objects**
- **Thin objects such as fur or hair**
- **Noise effects such as film grain**
- **The quick and unorganized motion in a scene**
- **Small pieces such as snow, rain and explosions**

4. 2-D and 3-D. The process of transformation from 3D space to a 2D plane can be illustrated with a pinhole model (Figure 1), which consists of a plane R, called the image plane and a point C, the optical center, which does not belong to the image plane.

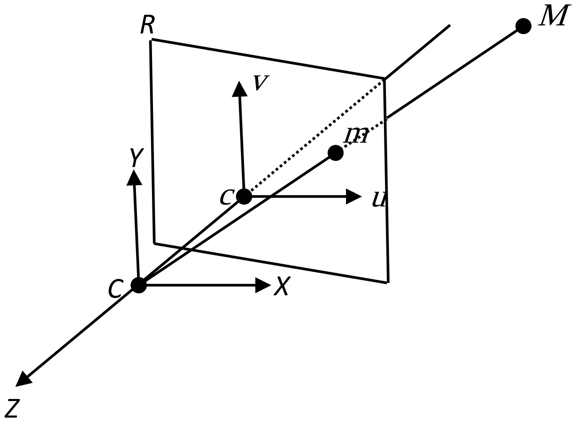


Figure 1 Pinhole$Model^1$

M object has a projection on the image plane R at the m point, and that projection represented by intersection of the optical ray (C, M) and the image plane R. The principal point c represents the center of the perpendicular of the optical axis on the image plane. The camera coordinate system (CCS) could be carried out with the center C and two axes (X and Y) which are parallel to the image plane (u, v) and the third axis Z corresponds the optical axis. The distance between the center C and the image plane represent the focal length f. [5]

5. The Relationship between the Camera and the Real World. In general, all images that we have represent the reflection of any object in our world, so those images represent the results of the relationship between cameras and the real world, and each point in the image has a corresponding point in the real world. Clearly, the position of any object in an image depends on its position in the real world. In fact, after the camera captures any scene, we obtain a 2D image coordinate $P(u,v)$ from 3D points (scene coordinates) $P(X,Y,Z)$. [6]

6. Camera Calibration. Camera calibration is the process of estimating the internal camera parameter (intrinsic parameter) that relates the direction of rays through the optical center to coordinates on the image plane. The importance of the internal camera parameter lies in the need for building 3D models of the world using a camera with a known intrinsic parameter [6].

7. STATE-OF-THE-ART. The procedure of obtaining structure from a set of images began in the 1980s [7-10]. Normally, structure from motion is initially approached by placing a set of obvious characteristics that are found in two image structures. This is commonly denoted as the correspondence problem solution. Then, the proportional motion of these characteristic correspondences is given the structure of the environment [11]. The conventional estimation of structure from motion mostly uses two images obtained from a single camera to slant the field of view of 45 to 60° [12-14]

8. Related Works. Using the structure and motion together under the name of structure from motion to reconstruct the three-dimensional model from multiple images is considered to be a significant topic in computer vision research. The pioneers in the field of computer vision have proposed many techniques to fill the lacunae in the structure from motion approach.

Zhengyou Zhang [18] used structure and motion from two perspective views based on the essential parameters, a fundamental matrix and Euclidean motion.

The problem with this technique is that the results mostly are not good enough due to the sensitivity of the second step to the incipient guess and the difficulty of obtaining an accurate incipient estimate from the first step. In order overcome this problem, Zhengyou Zhang proposed an approach by imposing the fundamental matrix (zero-determinant constraint). Unlike [18], Frank et al [19] introduced another technique by using the structure from motion without correspondence. This method exceeded the traditional techniques that require the presence of a known correspondence point [12] or calibrated images from a known camera viewpoint [13] or known shape [14]. Furthermore, this method deals with non-sequential images which are taken from vastly different viewpoints.

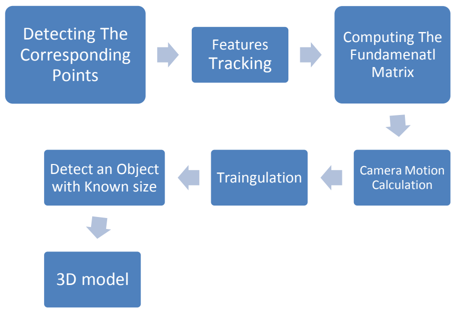
Masahiro [15] introduced a method of using the structure from motion in map reconstruction. This method was a system of three-dimensional simultaneous localization and mapping (SLAM), which is based on the SFM scheme. The steps of this method are as follows:

- Basic Framework
- Feature Tracking
- Initial Estimation

The first step considers the three-dimensional SLAM as a set of images obtained from a monocular camera. The three-dimensional map is represented as three-dimensional points from the feature points tracked through the set of images. The second step occurs based on KANADE-LUCAS-TOMASI [16]. The third step occurs by using the factorization method [17].

The precision and robustness of this method is based on the selection of the baseline distance, so the proper baseline selection depends on standards for object shape reconstruction and the camera pose estimation.

9. The Proposed Method. According to the title of the paper, the technique of reconstructing a three-dimensional model from a pair of two-dimensional images depends on structure and motion. In order to obtain this information, there are a number of steps to follow. First, we need a static scene with an object of known size (in our scene, the object is a ball of size 10 cm) this size is considered as a scale factor to reshape the 3D model, and a calibrated camera to obtain two views. After obtaining the real data in two images, the work of the KLT algorithm begins at this step. The workings of this algorithm are presented in the following sections as the next diagram representing.


Detecting The Corresponding Points
Features Tracking
Computing The Fundamenatl Matrix
Camera Motion Calculation
Traingulation
Detect an Object with Known size
3D model

Processing Diagram

9.1 Detection of The Correspondence Points. In order to continue to the others step, it is necessary to find the correspondence points. Therefore, the best features need to be detected in order to track from image to image. This process is carried out by using the minimum eigenvalue algorithm as proposed by C. TOMASI & J. SHI [23], and as the below equation shows:

$$R = \min(\lambda_1, \lambda_2)$$

where (λ_1, λ_2) represents the eigenvalues and the window (corner) is accepted if those eigenvalues are greater than the predefined threshold value (λ) as shown below:

$$\min(\lambda_1, \lambda_2) > \lambda$$

According to the C. Tomasi & J. Shi method, the strongest corners will be found in the image, which is a grayscale image.

9.2 Features Tracking. This step begins after finding the strongest corners (best features) from the first image. The role of this process is to track those features in the second image. This process is carried out by using the KLT algorithm (KANADE-LUCAS-TOMASI) [24]. The goal of this algorithm is to find the specific location of a specific point in the second image according to the first image. This is achieved with the following equation:

$$\bar{V}_{opt} = G^{-1}\bar{b}$$

9.3 Computing the Fundamental Matrix. The computation of the fundamental matrix from the correspondence points which are detected is the first step, the next equation is used to compute the fundamental matrix F:

$$X'^{T}FX = 0$$

Where X' and X represents corresponding points of a pair images.

9.4 Camera Motion Calculation. In this section, we will estimate the position and orientation of a calibrated camera. Normally, there are two views, hence there are two poses. Both poses are relative to each other as denoted by the fundamental matrix F. The camera poses are computed up to scale and the position denoted a unit vector.

9.5 Triangulation. The three-dimensional positions of the matched points can be determined by triangulating.

9.6 Detect an Object with Known Size. This process is carried out by using the MSAC algorithm (M-estimator sample consensus). The fitting of a sphere to an inlier point cloud using an object with known size is here a ball of size 10 cm.

10. Experimental Results. The experiments were carried out on an ordinary PC equipped with the following specifications:

- System Type: 64-bit operating system, x64-based processor.
- Edition: Windows 10 Home.
- Processor: Intel (R) Core (TM) i3-2310M CPU @ 2.10 GHz.
- RAM: 4.00 GB.

The input images were obtained from a digital camera (NX3000) equipped with:

- 20.3 MP APS-C CMOS Sensor.
- 16-50 mm Power Zoom Lens.
- 1/4000 sec Shutter Speed.

All experiments were carried out using the MATLAB R2015b software package. The methodology of the paper was based on the technique of ‘structure from motion ‘using KLT algorithm, but by using a single calibrated camera with the camera calibration application in MATLAB and by obtaining two views of the scene with a little motion for the second view. The algorithm that will create the three-dimensional model of the scene, from a pair of two-dimensional images following several steps, as the next section shows.

- The first step is carried out by loading a pair of images of the scene obtained by using the above camera.
- Next, the camera parameters are obtained by loading the camera calibration. To understand the mean reprojection error, which represent the difference in distance between the actual scene and the estimated one, we show below the equation of mean projection error:

$$\sum d\,(x_i, \hat{x}_i)^2 + d\,(x_i', \hat{x}_i')^2 \;\ldots\ldots\ldots\; 1$$

The unit of the reprojection error in pixel, so less than one it will be acceptable rate as shown in figure 2.

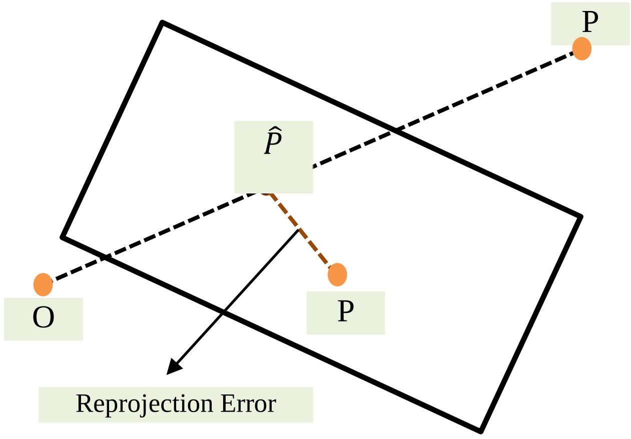


Figure 2: The reprojection error

- Camera calibration.
- In order to avoid any lens distortion effects on the accuracy of the final reconstruction, MATLAB offers a simple function for this purpose which straightens any lines that may deform due to the radial distortion of the lens.

- At this step, the algorithm detects the corresponding points between the two images. This process can be carried out in a number of ways; however, here, the motion occurs not too far from the first position, so the KLT algorithm (KANADE–LUCAS–TOMASI) is suitable to create the point correspondences.
- Computing the fundamental matrix is carried out at this point and according to the results, the inlier points are obtained, and those points match the Epipolar constraints.
- The computation of the camera position, which consists of the translation and rotation, is carried out by using the CameraPose function in MATLAB.
- The three-dimensional locations of the matched points found in the fourth step are reconstructed using the triangulation function.
- The Plot Camera and the PcShow functions are used to display the three-dimensional point cloud.
- In order to detect the actual scale factor, the algorithm uses an object with known size, so the scene contains a ball with a known radius (of 10 cm). The PcFitSphere function fits a sphere to the point cloud to detect the ball.
- The final step is the metric reconstruction, which mean the coordinates of the three-dimensional points will be in centimeter due to the actual radius of the ball which was 10 cm.

The following images show the results of the above steps with multiple different scenes, and each image has a title to clarify its identity. The time consumed by the algorithm to reach the results was different in each test, where the 1st test consumed 102 seconds, the 2nd test consumed 280 seconds, and the 3rd one consumed 131seconds. The results are shown below:

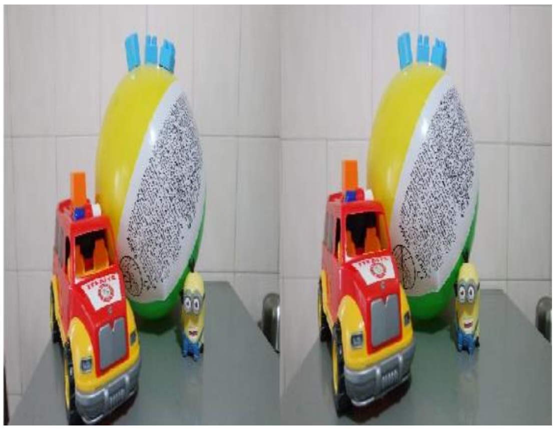


Figure 3: The original images

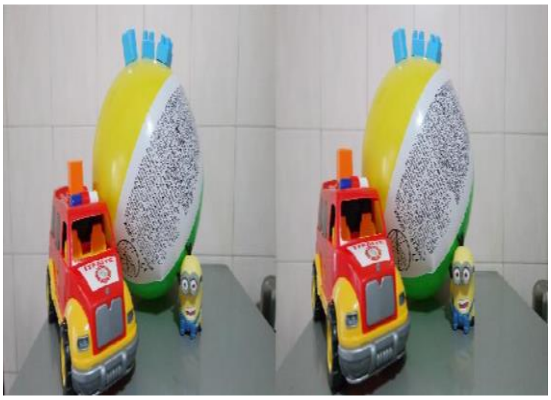


Figure 4: The Undistorting images

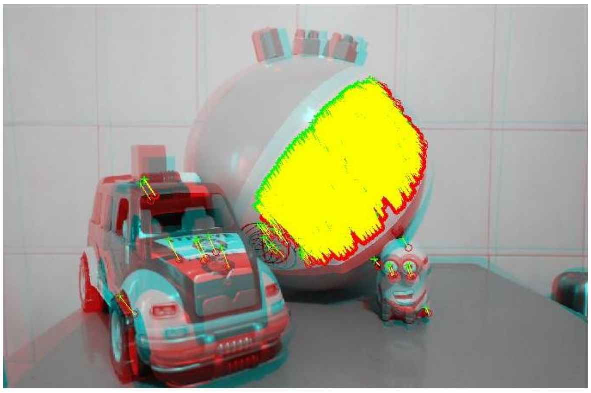


Figure 5: Strongest corners from the first image

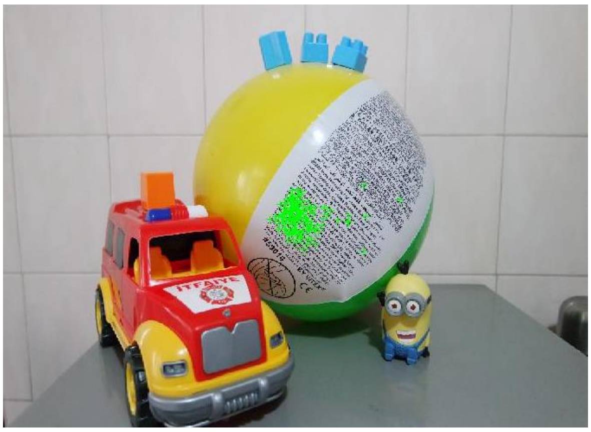


Figure 6: The Tracked features

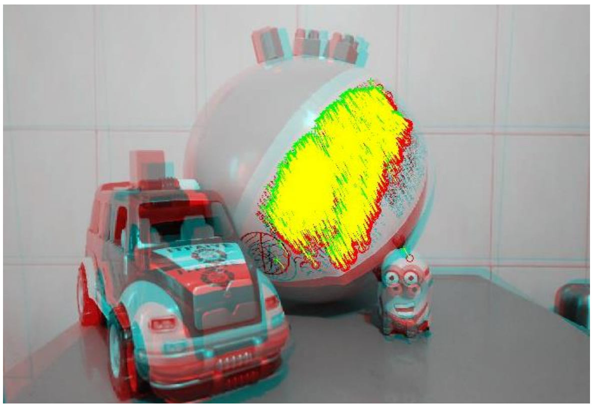


Figure 7: The Epipolar inlier

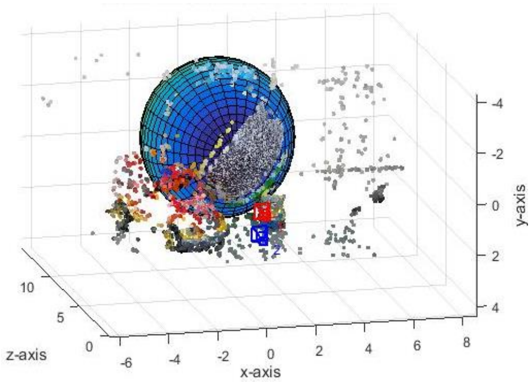


Figure 8: Estimated size and location of the ball

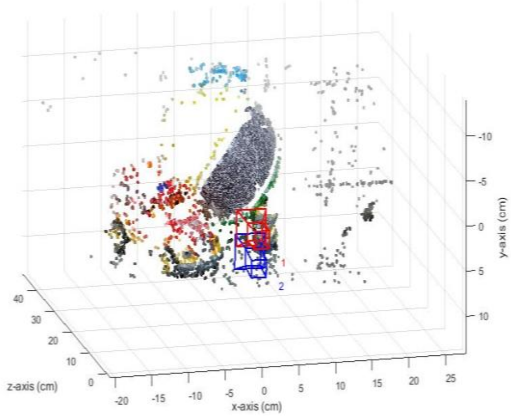


Figure 9A: Metric reconstruction of the scene

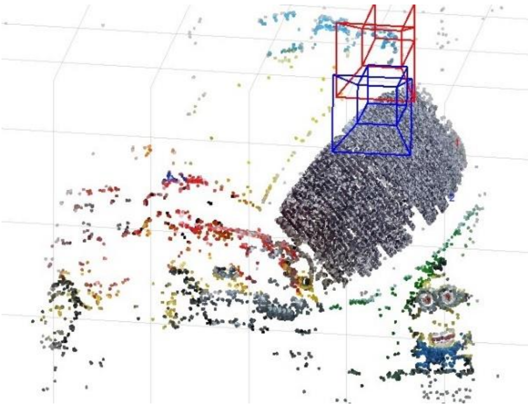


Figure 9B: Metric reconstruction of the scene with another position

In order to test the algorithm with another scene, which consisted of different ball with rugged surface and objects with more details, we repeat the execution of the code and the results were as shown:

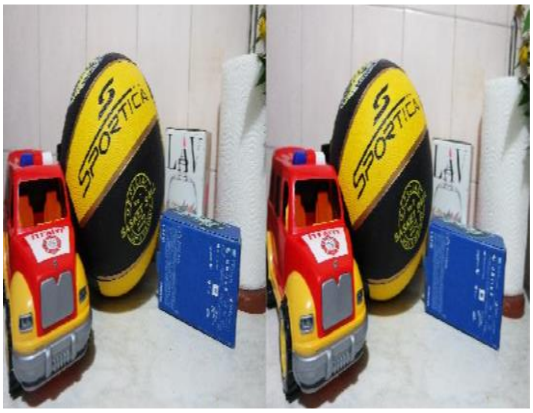


Figure 10: The original images (second test)

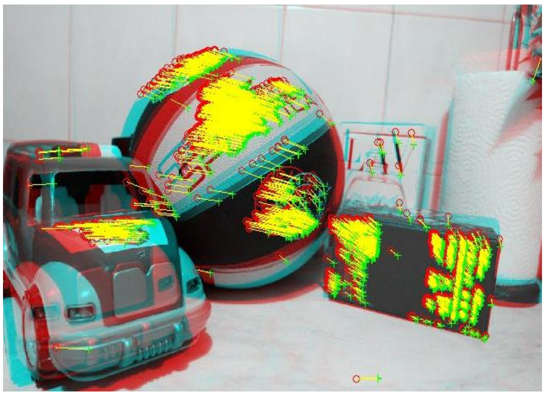


Figure 13: Tracked features (second test)

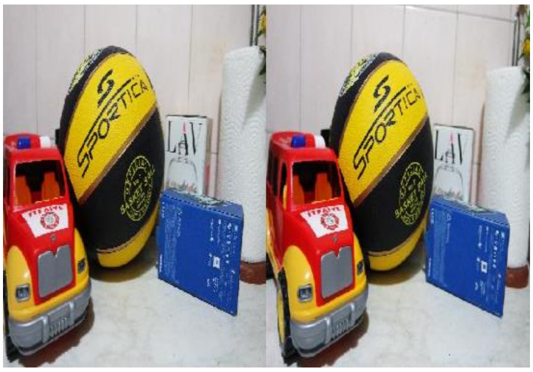


Figure 11: Undistorted images (second test)

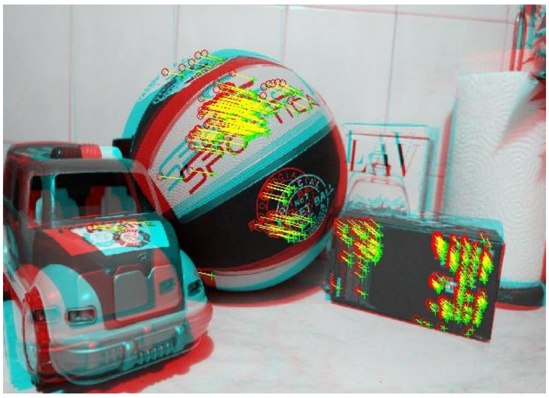


Figure 14: Epipolar inlier (second test)

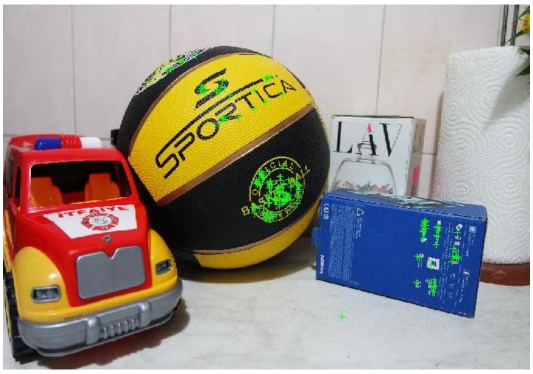


Figure 12: Strongest corners from the first image (second test)

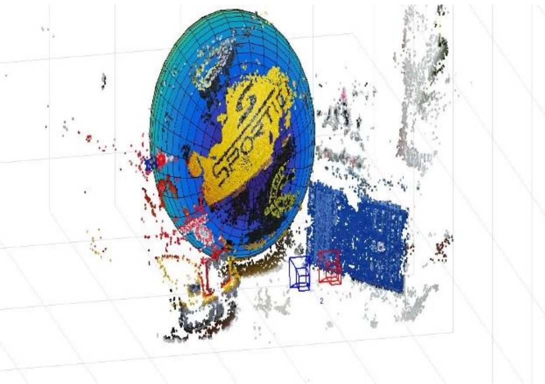


Figure 15: Estimated size and location of the ball (second test)

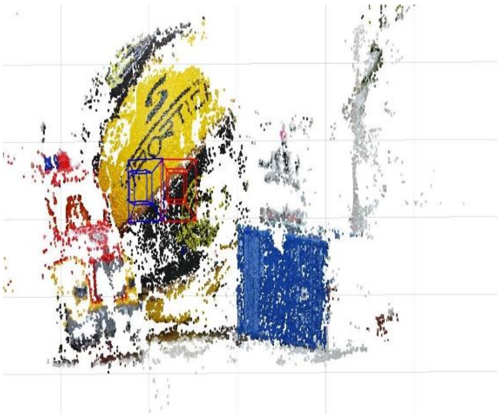


Figure 16A: Metric reconstruction of the scene (second test)

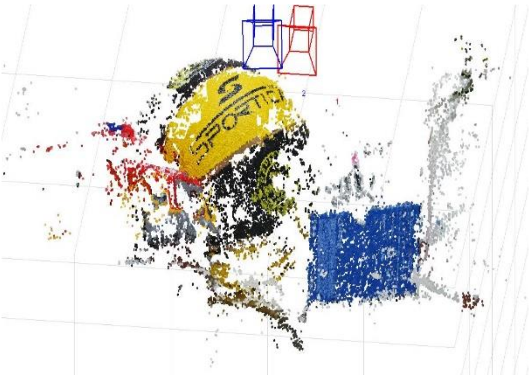


Figure 16B: Metric reconstruction of the scene with another position (second test)

The first scene (Figure 3) contained a ball with a soft surface which had some parts with only one color. We added some details to this ball in order to induce the algorithm to detect more matching points, and the results were as shown below:

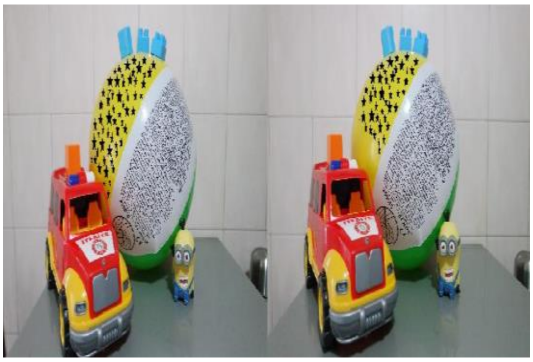


Figure 17: The original images (3rd test)

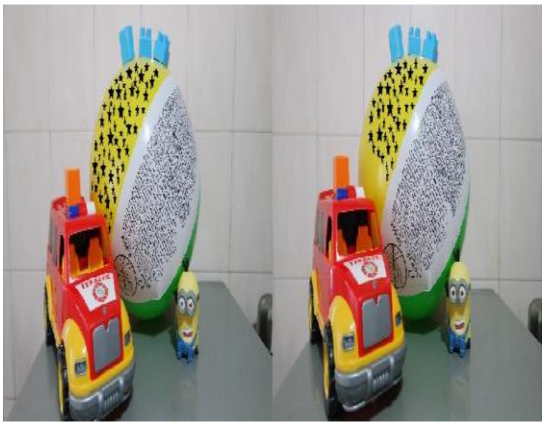


Figure 18: The undistorted images (3rd test)

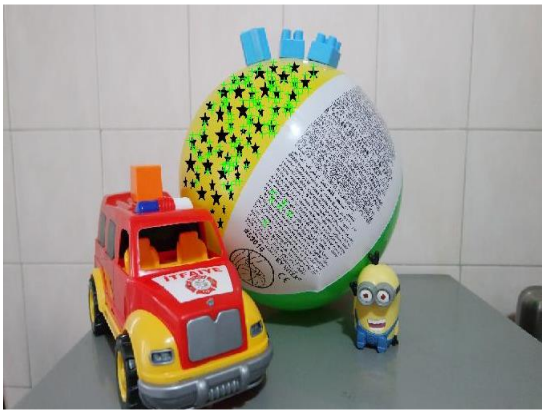


Figure 19: Strongest corners from the first image (3rd test)

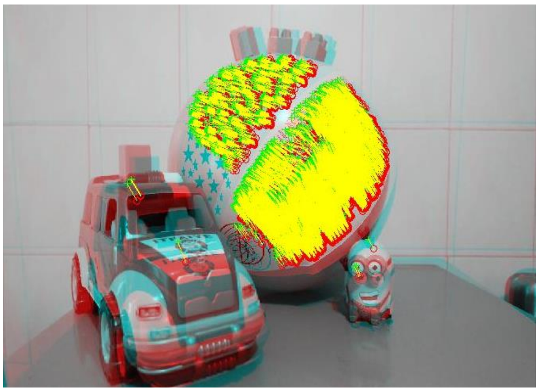


Figure 20: The Tracked features (3rd test)

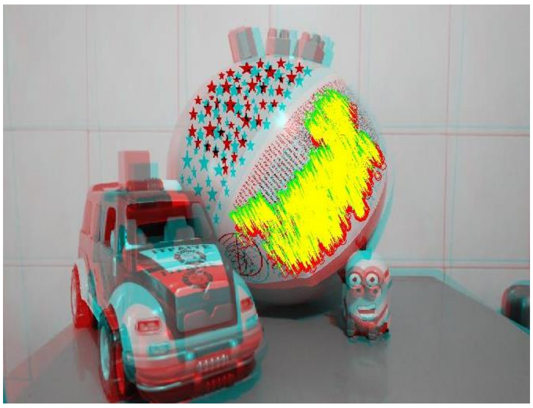


Figure 21: The Epipolar inlier (3rd test)

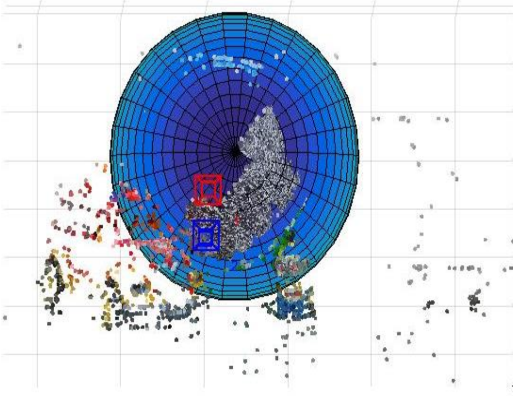


Figure 22: Estimated size and location of the ball (3rd test)

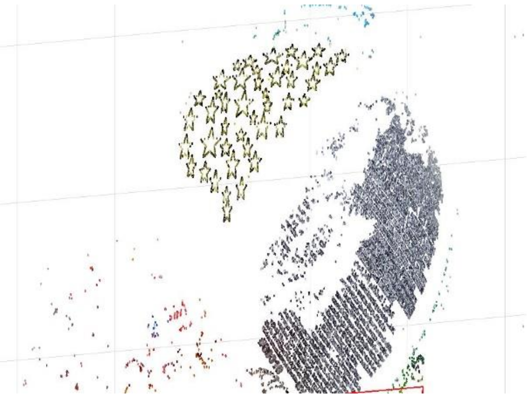


Figure 23A: Metric reconstruction of the scene (3rd test)

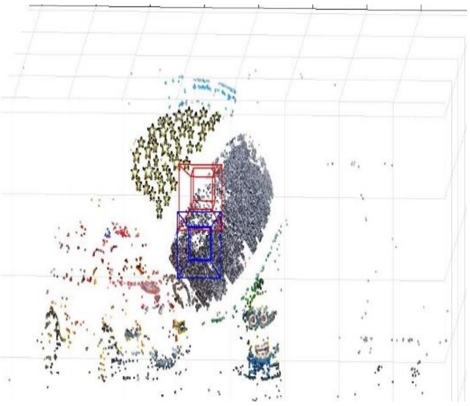
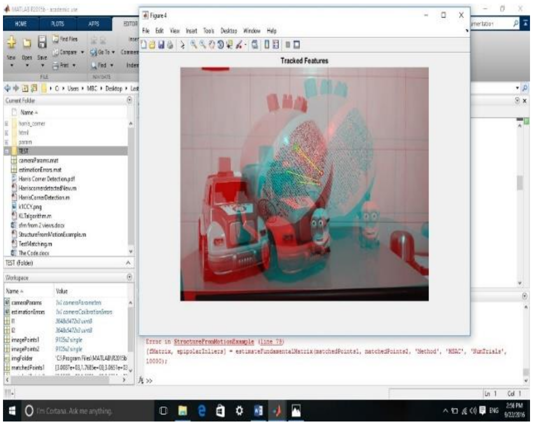


Figure 23B: The metric reconstruction of the scene with another position (3rd test)

As we said when the distance between the images is not too far, so the KLT algorithm will work properly, but when the distance become more than 5 cm (distance between the camera and the scene was 50 cm) the algorithm fails to match the points between the images as shown in the figure 24.



```
Error in StructureFromMotionExample (line 73)
[fMatrix, epipolarInliers] = estimateFundamentalMatrix(matchedPoints1, matchedPoints2, 'Method', 'MSAC', 'NumTrials',
10000);
```

Figure 24: KLT algorithm error

11.Real Data and Numerical Results

- **First Test**: In Figure 3, we show two real images of a constructed composite scene. This scene represents a difficult set of data due to its soft surface. We have covered the images with matched points using the KLT algorithm technique. When the sixth step of the algorithm is applied to the matched points of the real data, the motion estimate is a single matrix (1×3) for translation and a double matrix (3×3) for rotation, as shown below:

$$t = \begin{bmatrix} -0.27 \\ 3.16 \\ 0.3 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.99 & 0.005 & -0.034 \\ -0.002 & 0.99 & 0.096 \\ 0.035 & -0.095 & 0.99 \end{bmatrix}$$

As Zhengyou Zhang [32] used the same technique that we followed in our method and according to the available numerical data from his method, the translation and rotation data was as shown below:

$$t = [-9.6, 1.85, -1.75]^T$$

$$R = [-2.1, 4.2, 1.04]^T$$

The remaining data obtained from the experimental results are as follows:

		M.P. error 0.77
All colors		19961856x3 uint8
Ball Prop.	Parameters	[0.57, -0.91,10.6,3.13]
	Center	[0.57, -0.91,10.6]
	Radius	3.13
Camera parameters	Radial Distortion	[-0.099,0.12]
	Tangential Distortion	[0, 0]
	Estimate Skew	0
	Intrinsic Matrix	[3.9,0,0;0,3.9,0;2.7,1.85,1]
	Focal length	[3.9,3.9]
	Principal Point	[2.7,1.85]
Fund. Matrix		[1.3, -3.07,0.01; -7.28, -2.6,0.001; -0.01, -2.1,0.9
Scale Factor		3.18

The Numerical Result Data (First Test)

- **Second Test**: The second test carried out by using another scene as shown in the figure 10, and the numerical results data as shown below:

$$t = \begin{bmatrix} -3.3 \\ 0.14 \\ -0,47 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.99 & -0.02 & -0.08 \\ 0,02 & 0.99 & 0.015 \\ 0.087 & -0.018 & 0.99 \end{bmatrix}$$

		M.P. error 0.77
All colors		19961856x3 uint8
Ball Prop.	Parameters	[-0.99, -0.68,9.7,2,92]
	Center	[-0.99, -0.68,9.7]
	Radius	2.92
Camera parameters	Radial Distortion	[-0.099,0.12]
	Tangential Distortion	[0, 0]
	Estimate Skew	0
	Intrinsic Matrix	[3.9,0,0;0,3.9,0;2.7,1.85,1]
	Focal length	[3.9,3.9]
	Principal Point	[2.7,1.85]
Fund. Matrix		[2.02, 6.49, -0.001; -4.02, -3.88,0.01; 2.33, -0.01, 0,99]
Scale Factor		3.41

The Numerical Result Data (2nd Test)

- **Third Test:** The third test carried out by using the first scene but with adding some more details as shown in the figure 17, and the numerical results data as shown below:

$$t = \begin{bmatrix} -0.25 \\ 3.34 \\ -0,01 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.99 & 0.002 & -0.031 \\ 0,001 & 0.99 & 0.11 \\ 0.031 & -0.11 & 0.99 \end{bmatrix}$$

		M.P. error 0.77
All colors		19961856x3 uint8
Ball Prop.	Parameters	[0.69, -0.98,9.64,2,98]
	Center	[0.69, -0.98,9.64]
	Radius	2.98
Camera parameters	Radial Distortion	[-0.099,0.12]
	Tangential Distortion	[0, 0]
	Estimate Skew	0
	Intrinsic Matrix	[3.9,0,0;0,3.9,0;2.7,1.85 ,1]
	Focal length	[3.9,3.9]
	Principal Point	[2.7,1.85]
Fund. Matrix		[1.7, 3.06 ,0.01; -5.49, -4.03,0.003; -0.01, -0.00, 0,99]
Scale Factor		3.35

The Numerical Result Data (3rd Test)

12. Discussion: The image resolution used in the algorithm was 5472×3648. Initially, the algorithm begins in the first test with loading a pair of images (Figure 3), followed by the camera calibration stored in the camera parameters object loaded, which included the camera intrinsic matrix, the radial distortion and the estimated skew. According to the value of the skew, which here is zero, there is no distortion in the lines of the lens. The next process aims to remove any bends in the lines of the lens, and as the skew is zero, there is no need for this step (4). Later, the feature points will have been detected in this step from the first image (Figure 5) and, as mentioned above, are carried out by using the KLT algorithm.

The point tracker is created to find the correspondence points between the images (Figure 6). In order to specify the Epipolar constraints, the fundamental matrix is estimated, and by computing the fundamental matrix, the inlier points will be established and matched to the Epipolar constraints (Figure 7). Before the final step in the algorithm, the camera position (R, t), which represents the external parameters, are computed. Later, by using the sphere function to fit the point cloud in order to find the size and location of the ball in the scene (Figure 8).

Finally, the coordinates of the three-dimensional points in centimeters are determined according to the actual size of the ball (Figure 9 A and B). The final result of reconstruction of the three-dimensional model was not good due to the holes in the model; therefore, it was necessary to fill the uncovered areas

Zach et al [20] in their methods using four different datasets, and by adding more points where are reduced the of error, except the third dataset where the error is increased, and this issue is left without explaining in their paper. Those results shown in the next table.

Dataset	#Images	#3D points	Init. Image error	#Added points	Final image error
1	175	43553	2.17	1497	2.14
2	186	47756	6.18	5605	4.89
3	99	31876	1.77	5747	6.75
4	191	60997	3.3	1556	2.4

The results of Zach et al method[12]

In our method we used different types of scenes in order to demonstrate the behavior of the algorithm. The numbers of three-dimensional points, which are the algorithm obtained from the first scene (figure 3), are 19333 points. After adding more details to the first scene, and by using the same algorithm (figure 17), the numbers of three-dimensional points are increased from 19333 points to 22195 points. In the second test we are using different scene (figure 10), which is have more colors and details, the result of using such a scene was obtaining more 3D points. Where the numbers of 3D points are increased from 22195 points to 59413 points. Next three tables are clarifying all those results which were carried out from the three tests.

The original scene (1st Test)		
3D points	Image points	Matched Points
19333	30306	19333
The original scene after modifying (3rd Test)		
3D points	Image points	Matched Points
22195	39402	22195
Different scene with more details (2nd Test)		
3D points	Image points	Matched Points
59413	247519	59413

Numbers of points according to different scenes

Zach et al [20] in their method were added more points in order to reduce the rate of error, where the approach of the proposed method in this paper is motivate the algorithm to obtain more matched points by using scenes rich in details.

The limitations of the previous algorithm were found in the fourth step of the feature detection, where the KLT algorithm will not work probably if the space between the obtained images is too great (Figure 10). Next, the tracker features had some difficulties detecting the soft surfaces in the scene (Figure 6), so we added some details to this surface in order to motivate the algorithm to detect more matching points (Figure 17). Then, the same steps which mentioned above were executed. As the final result of the third test shown in the figures 23 A, B, the algorithm detects more points and reconstruct new model with more points.

The second test was carried out by using a different scene (Figure 10), after executing the algorithm, the results were more accurate than the first test due to the details of the scene which was had more colors than the first scene (figure 3).

13. Conclusion: In this paper, we concentrated on the KLT algorithm based on two views. The experimental results in the previous section have some limitations that need improvement and complementary solutions.

The results of the experiments show the insufficiency of KLT algorithm when the distance between images becomes more than 5 cm. Also, we figure out the possibility of reducing the rate of reprojection error by removing the images that have the biggest rate of error.

The experimental results are consisting from three stages. The first stage is done by using a scene with soft surfaces, the performance of the algorithm shows some deficiencies with the soft surfaces which are have few details. The second stage is done by using different scene with objects which have more details and rough surfaces, the algorithm results become more accurate than the first scene. The third stage is done by using the first scene of the first stage but after adding more details for surface of the ball in order to motivate the algorithm to detect more points, the results become more accurate than the results of the first stage. The experiments are showing the performance of the algorithm with different scenes and demonstrate the way of improving the algorithm.

In spite of the limitations mentioned above, the algorithm creates three-dimensional models that depend on only two views with the model being meaningful according to the original scene. Moreover, the work of the algorithm is quite good due to the rating of the mean projection error, which was 0.94 and decreased into 0.77.

14. Future Work: Researchers in this field may use this paper in investigations of two-dimensional to three-dimensional conversion algorithms. They can deal with the limitations mentioned herein by finding alternative algorithms instead of using the KLT algorithm to cope with widely-spaced images, or improve the 3D-model for greater accuracy. Also, they can estimate the depth information using other algorithms, and comparing the results with the current one in order to clarify the strengths and weaknesses of each algorithm.

References

[1] Ji Zhang et al, (2011), “**Pose-Free Structure from Motion Using Depth from Motion Constraints**”, IEEE TRANS. ON IMAGE PROCESSING, VOL. 20, NO. 10, PP 2937-2953.

[2] Tony Jebara et al, (1999), “**3D Structure from 2D Motion**”, IEEE SIGNAL PROCESSING MAGAZINE, VOL. 16, ISSUE: 3, PP 66 – 84**.**

[3] Scott Squires, (2011), “**2D to 3D Conversion**”, ADAPT Web Site, effectscorner.blogspot.com.

[4] Jon Karafin, (2011), “**State-Of-The-Art 2D to 3D Conversion and Stereo VFX**”, International 3D Society University, Presentation 3DU-Japan event in Tokyo.

[5] Olivier Faugeras and Quang-Tuan Luong, (2001), “***The Geometry of Multiple Images***”.

[6] Simon J.D. Prince, (2012), “***Computer Vision Models, Learning and Inference*** “.

[7] LONGUET-HIGGINS, (1981), “***Computer Algorithm for Reconstructing a Scene from Two Projections***,” Nature 293, PP:133–135.

[8] Prazdny, (1983), “***On the Information in Optical Flows***,” Computer Vision, Graphics, and Image Processing VOL 22, No 2, PP:239 – 259.

[9] Tsai et al, (1981), “***Estimating Three-Dimensional Motion Parameters of a Rigid Planar Patch***,” Acoustics, Speech and Signal Processing, IEEE Trans on 29, PP:1147-1152.

[10] Soatto et al, (1998), “Reducing “***structure from motion***” a general framework for dynamic vision. Part 1: modelling.,” International Journal of Computer Vision 20, PP:993-942.

[11] Joseph and J. Sean, (2013), "***Structure from Motion in Computationally Constrained Systems***", Micro- and Nanotechnology Sensors, Systems, and Applications V, edited by Thomas George. Saiph Islam, K. Dutta, Proc. of SPIE Vol. 8725, 87251G.

[12] M. Pollefeys, (1999), "***Self-Calibration and Metric Reconstruction in Spite of Varying and Unknown Intrinsic Camera Parameters***", Int. J. of Computer Vision, VOL.32, No.1, PP:7-25.

[13] K. Kutulakos, (1999), "***Theory of Shape by Space Carving***", In Proc. Seventh Int. Conf. on Computer Vision, PP:307-314.

[14] D. Lowe, (1991), "***Fitting Parameterized Three-Dimensional Models to Images***", IEEE Trans. on Pattern Analysis and Machine Intelligence, VOL.13, No.5, PP:441-450.

[15] Zucchelli, (2002), "***Optical flow based structure from motion***", Numerical Analysis and Computer Science. Stockholm: (Royal Institute of Technology).

[16] Astrom et al, (2000), "***Solutions and Ambiguities of the Structure and Motion Problem for 1D Retinal Vision***," J. Math. Imaging Vis. VOL 12, No 2, PP:121-135.

[17] Li et al, (2006), "***Structure from Planar Motion***", IEEE Trans on Image Processing 15, PP:3466-3477.

[18] Zhengyou Zhang, (1997), "***Motion and Structure from Two Perspective Views: From Essential Parameters to Euclidean Motion Via Fundamental Matrix***", Journal of the Optical Society of America, VOL.14, No.11, PP:2938-2950.

[19] Frank Dellaert et al, (2000), "***Structure from Motion without Correspondence***", Computer Science Department & Robotics Institute Carnegie Mellon University, Pittsburgh PA 15213, IEEE 1063-6919/00.

[20] Christopher Zach et al, (2012), "***Discovering and Exploiting 3D Symmetries in Structure from Motion***", IEEE Conference, Computer Vision and Pattern Recognition, DOI: 10.1109/CVPR6247841, PP: 1514-1521.

[21] Klingner et al, (2013), "***Street View Motion-from-Structure-from-Motion***", IEEE Int. Conference on Computer Vision, DOI 10.1109/ICCV, PP: 953-960.

[22] Yongjun Zhang et al, (2015), "***Optimized 3D Street Scene Reconstruction from Driving Recorder Images***", Remote Sens. 7, PP: 9091-9121, ISSN 2072-4292 DOI: 10.3390/RS70709091.

[23] C. TOMASI & J. SHI, (1994), "***Good Features to Track***," Proc. of CVPR'94, PP: 593-600.

[24] Derek Hoiem, 2012, "***Feature Tracking and Optical Flow***", Computer Vision, CS 543/ECE 549, Illinois Univ.

تحويل الصور الثنائية الابعاد الى نماذج ثلاثية الابعاد بإستخدام زوج من الصور "خوارزمية كي ال تي كحالة تطوير ودراسة"

سرمد نهاد محمد
جامعة كركوك
كلية العلوم ، قسم علوم الحاسوب

المستخلص :

تدرس هذه الورقة كيفية تأثير الحركة بين صورتين لمشهد واحد على عملية تشكيل النموذج الثلاثي الابعاد بإستخدام خوارزمية كي إل تي (*KLT*) والتي استخدمت لتحويل صور ثنائية الأبعاد إلى نموذج ثلاثي الأبعاد. تتم عملية إعادة تشكيل النموذج الثلاثي الابعاد باستخدام كاميرا واحدة وخوارزمية تقوم على استخدام اثنين فقط من الصور لمشهد واحد، واعتماد تقنية (*SFM*) على أساس الكشف عن النقاط المشتركة بين الصورتين، و إبيبولار إنليرس. استخدام خوارزمية (*KLT*) مع طريقة (*SFM*) يدل على عدم قدرة الخوارزمية على النجاح في العمل عندما تكون المساحة بين الصورة الاولى والثانية شاسعة. كما وتم تقليل معدل الخطأ عند اعادة تسقيط المشهد من الواقع الى الكاميرا عن طريق إزالة الصور التي لديها أكبر معدل الخطأ. تم العمل على تشكيل نموذج ثلاثي الابعاد بأستخدام ثلاثة مشاهد مختلفة على ثلاثة مراحل. المرحلة الأولى تمت باستخدام مشهد ذو أسطح ناعمة، ويظهر أداء الخوارزمية بعض أوجه القصور مع الأسطح الناعمة التي ليس لها سوى القليل من التفاصيل. اما المرحلة الثانية تمت باستخدام مشهد مختلف مع الكائنات التي لديها المزيد من التفاصيل والأسطح الغير ملساء، وهنا نتائج الخوارزمية اصبحت أكثر دقة من المشهد الأول. المرحلة الثالثة تمت باستخدام المشهد الأول في المرحلة الأولى ولكن بعد إضافة المزيد من التفاصيل على سطح الكرة لتحفيز الخوارزمية لكشف المزيد من النقاط، وهنا اصبحت النتائج أكثر دقة من نتائج المرحلة الأولى. وتظهر التجارب أداء الخوارزمية مع مشاهد مختلفة وتظهر طريقة تحسين الخوارزمية حيث وجدت المزيد من النقاط من الصور، لذلك تم تشكيل النموذج الثلاثي الابعاد بصورة أكثر دقة.