Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 10, No. 2, 2022, 245-256.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

A new Approach to Estimate Motion and Structure of a Moving Rigid Object in a 3D Space with a Single Hand-Held Camera

Reza Serajeh[1], Amir Mousavinia[2*] and Farzad Safaei[3]

1. Faculty of Electrical Engineering, K.N.Toosi University of Technology, Tehran, Iran.
2. Faculty of Computer Engineering, K.N.Toosi University of Technology, Tehran, Iran.
3. Faculty of Informatics, University of Wollongong, Wollongong, Australia.

Article Info

Article History:
Received 05 September 2021
Revised 02 February 2022
Accepted 14 March 2022

DOI:10.22044/jadm.2022.11167.2267

Keywords:
Structure from Motion, Dynamic Scene 3D Reconstruction, Moving Rigid Object 3D Motion Estimation.

**Corresponding Author:*
moosavie@kntu.ac.ir(A. Mousavinia).

Abstract

The classical structure from motion algorithms are widely used in order to estimate the 3D structure of a stationary scene with a moving camera. However, when there are moving objects in the scene, if the equation of the moving object is unknown, the approach fails. This work demonstrates that when the frame rate is high enough and the object movement is continuous in time, meaning that the acceleration is limited, a simple linear model can be effectively used in order to estimate the motion. This theory is first mathematically proven in a closed-form expression, and then optimized by a non-linear function applicable for our problem. The algorithm is evaluated both on the synthesized and real data from the Hopkins dataset.

1. Introduction

There have been many studies around camera motion estimation (known also as odometry) and 3D structure estimation due to their many interesting applications in different areas such as robot navigation and map generation [1]. This part first explains an introduction to it, and then represents a problem description for this paper.

1.1 Related works

The field of 3D motion and structure estimation has been studied in different known classical approaches named Structure From Motion (SFM), and Simultaneously Localization And Mapping (SLAM). SLAM is useful for online cameras that use prior sequenced knowledge, while SFM is an offline algorithm used for a bunch of stored images. However, both of these approaches estimate the camera motion and 3D structure of a scene from 2D image sequences taken from a moving camera. This goal is achieved by considering the geometry between different views of the static scene and application of triangulation [2] on the image corresponding points found by the methods such as SIFT [3], SURF [4], ORB [5] or LIFT [6]. When these different views are under control, the 3D structure can be even optimized to reduce localization error and enhance depth estimation accuracy by improving the camera arrangement [7, 8]. Additionally, in order to estimate and optimize the solution in the presence of noise, filter-based approaches such as Kalman filter [9] and particle filter [10] or bundle adjustment (BA) [11] have been introduced.

However, the classical methods represent the geometry and mathematics for only the static scene, while, in practice, the environment is typically dynamic including the moving objects that do not follow the geometry.

In order to tackle this issue, the algorithms such as RANSAC [12], PROSAC [13], and MLESAC [14] are represented to just find and remove the outliers caused by the moving objects or noises and then rely on only the feature points located on the other static part. More practically, J. Engel *et al*. have proposed a large-scale algorithm LSD-SLAM as a robust solution useful for this situation [15]. A semantic visual SLAM named DS-SLAM has been proposed by C. Yu *et al*. [16] towards

dynamic environments. DM-SLAM combines an instance segmentation network with the optical flow to tackle the issue [17], and a deep learning approach has been used by M. S. Bahraini *et al*. [18].

However, this does not solve the issue of the moving objects themselves, and only concentrates on the static part by ignoring the moving parts. In other words, the mentioned algorithms work well on the dynamic scenes by just eliminating the moving objects such that they do not estimate their 3D structure and motion.

By usage of a laser scanner, C.-C. Wang and C. Thorpe [19], and C.-C. Wang *et al*. [20] have represented Simultaneous Localization, Mapping, and Moving Object Tracking (SLAMMOT) in order to tackle the issue. However, adding of laser scanner itself is a limitation that does not allow the method to be applicable for any captured video without a laser scanner.

In the domain of dynamic scene 3D reconstruction, instead of 3D structure estimation for the moving objects, different approaches only segment the moving objects from the static scene [21]. Multi-Body Structure from Motion (MBSfM), as an extension of SFM, stands for the methods in this field that detect and segment different rigid moving objects existent in the scene, and reconstruct their 3D structures in their own coordinate, separately [22, 23]. In other words, MBSfM segments different rigid moving objects into their different objects' motion clusters, where each object is separately represented by a specific motion and 3D structure of itself similar to the conventional SFM [24, 25]. Non-Rigid Structure from Motion (NRSfM) is also another extension of SFM for non-rigid moving objects [26-30].

However, by detection and segmentation of the moving rigid objects, each object can be only separately processed by the conventional SFM to estimate the object's 3D structure (up to an unknown scale) and its consequent motion for the camera. This means that each moving object, separately, has its structure and camera motion independent from other parts of the scene.

1.2 Problem description

The 3D estimation of motion and structure of a moving rigid object in the coordinate of the main scene is a substantial and challenging problem. The main difficulty is that the triangulation [31] used for estimation of the static 3D structure of the scene is not valid for the estimation of moving object 3D structure when besides the camera motion, the object itself is also freely moving in a 3D space without known moving parameters. In other words, there is a short Δt time between each consecutive frame when the object can move and therefore add noises to the corresponding points and their disparities. This noise can be incrementally integrated over the next coming frames, and therefore, adds a high uncertainty to the corresponding points, and consequently, makes a big 3D reconstruction error. For example, in Figure 1, the typical SFM fails for reconstructing the moving car 3D structure.

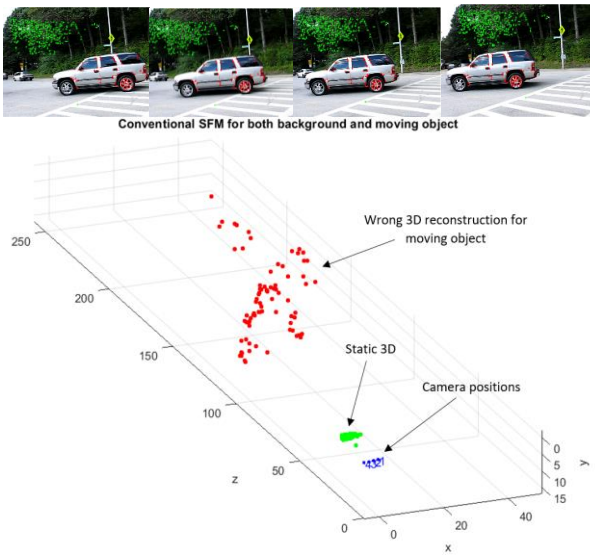


Figure 1. Conventional SFM fails to reconstruct 3D structure of moving object.

Therefore, when both camera and object are freely moving in a 3D space, there is a problem with the SFM approaches to estimate the 3D motion and structure of the moving object in the coordinate of other parts of the scene. In this condition, since the corresponding points are caused by the consequent movement of both the camera and the object, there is no unique motion and position of the moving 3D object's point to satisfy the geometry for the corresponding points without any constraint.

In other words, by looking at Figure 2, any 3D point existent on a projection ray on one frame can lie on another frame's corresponding projection ray with different specific motions. This means that there are infinitive 3D points and motions that can together satisfy the geometry. In order to solve this underdetermined problem, Avidan and Shashua [32, 33] have introduced a trajectory triangulation, assuming that the 3D point is traveling on a 3D unknown line in different frames. By this constraint, they require at least 5 frames to make a linearly solvable system. In their other research work [34], they have assumed that the object is traveling over a conic

section, where this constraint requires 9 frames to solve the equations.

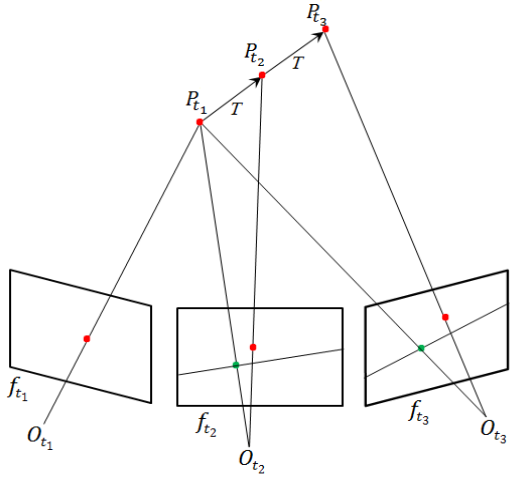


Figure 2. By fixing translation T between 3 consecutive frames for a 3D moving point P , it can uniquely lie on the corresponding optical rays, where red points on the frames are the corresponding points. In the case of $T = 0$, the corresponding points of the first frame in other frames will be located on the epipolar lines (green points).

Following the research work, triangulation has been generalized by Kaminski and Teicher [35] to transform the non-linear trajectory problem into a linear problem by polynomial representation. Park *et al*. [36] have estimated robustly the 3D points by least squares. Particle filter has also been used by Kundu *et al*. [37] to estimate and optimize iteratively the position and velocity of the moving objects using a Bearing Only Tracking (BOT). In a more recent work, H. S. Park *et al*. [38] have inspected the possibility of accurate 3D reconstruction by using reconstructability criteria represented in [36]. For this, they required to generate a low amount of basis vectors where the object trajectory should be approximated by a linear combination of them. This allows them to constrain the issue and solve an underdetermined system. However, this trajectory approximation is dependent on how the trajectory fits the basis vectors.

In our work, considering a physical constraint of having a constant speed for a short time, we can solve the issue without constraining the whole trajectory and then simplify it into a closed-form equation.

Since the 3D position of a physical rigid object such as a vehicle is continuous in time (the speed is continuous or the acceleration is limited), therefore, we can approximate the object speed by a constant value for a short time, for example, 3 consecutive frames. In other words, we assume that the speed in each time step is $v_{t_n} = v_{t_{n-1}} + \Delta v$ and when $\Delta t \to 0$ then $\Delta v \to 0$. Therefore, for a small number of consecutive frames K, the approximation $v_{t_n} \approx v_{t_{n-K}}$ is valid. However, the acceptable change of speed is correlated with the camera frame per second (fps) such that with higher camera fps, we can keep the approximation still valid for faster object speed changes.

This allows us to fix the speed, and consequently, the translation between frames for the moving object for a short period of 3 frames, as can be seen in Figure 2. We show that by this constraint, the rigid object's 3D motion and structure are recoverable by the only usage of minimum 3 consecutive frames. As it is shown in Figure 2, when the translation T is constant between frames, the 3D point P can be uniquely reconstructed such that it satisfies the geometry.

Our method represents a closed-form solution for this issue. This solution initializes a non-linear optimization to also handle noise. The theory is first mathematically presented and proven, and then validated on simulated data and frames where the results are also compared with a recent deep model. Additionally, the method is tested on real image sequences in order to visualize the result in practice. To represent our method, the rest of this paper is organized as what follows. In Section 2, the method is mathematically represented. Section 3 concisely explains the algorithm. Section 4 represents the experiments on both the simulated data and the real data. Finally, Section 5 describes the conclusion.

2. 3D motion and structure estimation of moving object

In this section, the theory for estimation of both the 3D motion and the structure of a moving rigid object is represented in two parts, workflow and problem formulation.

2.1. Workflow

Since the real moving objects such as vehicles have a continuous motion, we assume that the speed is constant (the acceleration is zero) for a short period of time. In this case, the motion of a rigid object can be estimated by a 3D translation vector, valid for a couple of consecutive frames. As shown in Figure 3, in our approach, given the camera poses and the moving object's corresponding points, we estimate the 3D structure and 3D motion of the moving object in the coordinate of the static scene. The static scene structure and camera motions can be estimated by the conventional SFM. An example of this process can be seen in Figure 4.

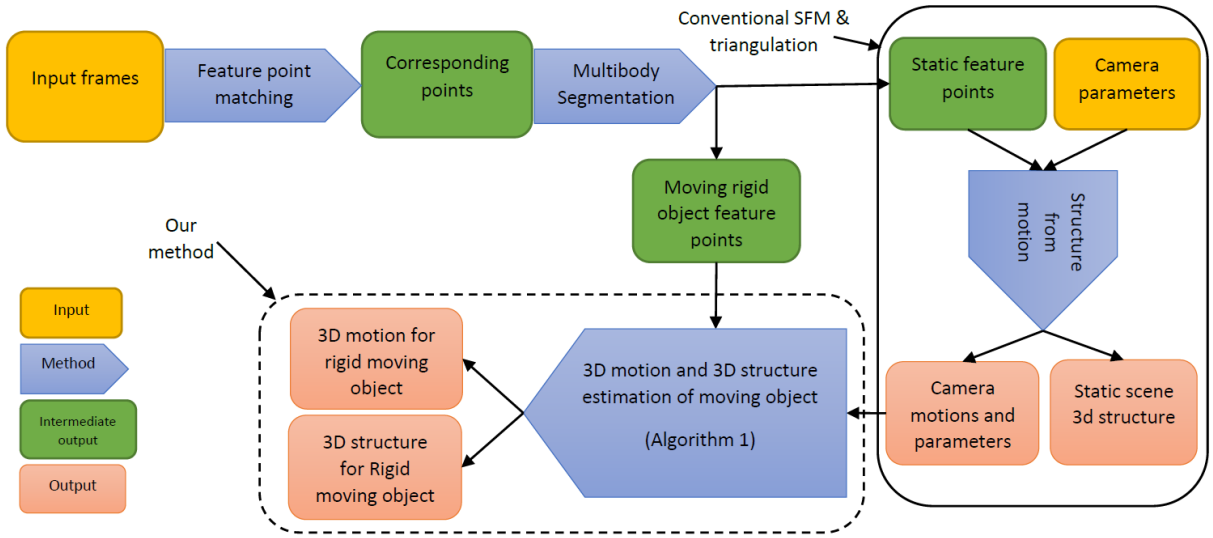


Figure 3. This block diagram shows at first the steps required for conventional approaches to reconstruct camera 3D motion and 3D structure of the static scene, and secondly shows where our approach is placed to estimate the 3D motion and 3D structure of a moving rigid object.

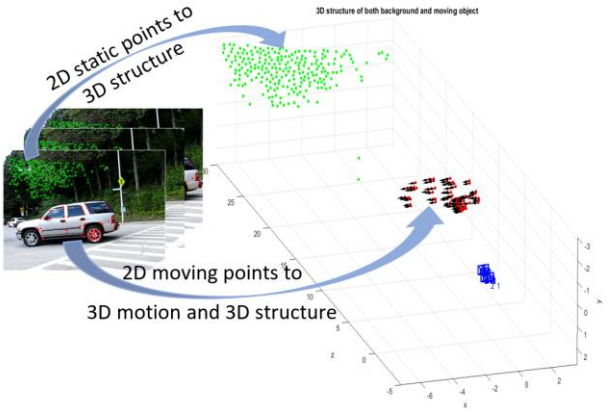


Figure 4. While the conventional SFM can reconstruct the 3D camera motions (in blue) and 3D structure of the static scene (in green), our approach can estimate the 3D motion (in black) and 3D structure (in red) of moving objects in the coordinate of the static scene.

2.2. Problem formulation

We first consider $P_t^n = \begin{bmatrix} X_t^n & Y_t^n & Z_t^n & 1 \end{bmatrix}^T$ as the nth 3D point at the frame time t on a moving rigid object, where $n = 1:N$, and N is the number of key points on the object. Then by approximating the object motion for each pair of consecutive frames as a fixed translation vector $T_t = \begin{bmatrix} T_t^x & T_t^y & T_t^z & 0 \end{bmatrix}^T$ for a couple of frames, the position of P_t^n over these frame time steps is:

$$P_{t+k\dot{o}}^n = P_t^n + kT_t \tag{1}$$

where $\dot{o}$ is a short time-step between two consecutive frames, and $k = 0:K$ refers to the frame index. In our equations, integer $K > 1$ is enough small to make it possible to fix the translation T over a couple of consecutive frames and also to provide at least 3 required frames for our equations.

At time $t + k\dot{o}$, the projected point through its image is $\Pr\left(P_{t+k\dot{o}}^n\right) = \left[u_{t+k\dot{o}}^n, v_{t+k\dot{o}}^n\right]^T$, where by considering the projection matrix as $\pi_{t+k\dot{o}}$ (a 3×4 matrix), the equations are as follow:

$$p_{t+k\dot{o}}^n = \begin{bmatrix} x_{t+k\dot{o}}^n \\ y_{t+k\dot{o}}^n \\ z_{t+k\dot{o}}^n \end{bmatrix} = \pi_{t+k\dot{o}} P_{t+k\dot{o}}^n = \pi_{t+k\dot{o}}(P_t^n + kT_t) \tag{2}$$

$$u_{t+k\dot{o}}^n = \frac{x_{t+k\dot{o}}^n}{z_{t+k\dot{o}}^n} \tag{3}$$

$$v_{t+k\dot{o}}^n = \frac{y_{t+k\dot{o}}^n}{z_{t+k\dot{o}}^n} \tag{4}$$

Generally, by expanding the equations above for all u, v, π, k, and P, we have:

$$u = \frac{\pi_{11}(X+kT_x) + \pi_{12}(Y+kT_y) + \pi_{13}(Z+kT_z) + \pi_{14}}{\pi_{31}(X+kT_x) + \pi_{32}(Y+kT_y) + \pi_{33}(Z+kT_z) + \pi_{34}} \tag{5}$$

$$v = \frac{\pi_{21}(X+kT_x) + \pi_{22}(Y+kT_y) + \pi_{23}(Z+kT_z) + \pi_{24}}{\pi_{31}(X+kT_x) + \pi_{32}(Y+kT_y) + \pi_{33}(Z+kT_z) + \pi_{34}} \tag{6}$$

where we can re-write the equations above through the linearly represented equations as follow:

$$\begin{aligned} &(\pi_{11}k - u\pi_{31}k)T_x + (\pi_{12}k - u\pi_{32}k)T_y \\ &+(\pi_{13}k - u\pi_{33}k)T_z + (\pi_{11} - u\pi_{31})X \\ &+(\pi_{12} - u\pi_{32})Y + (\pi_{13} - u\pi_{33})Z = u\pi_{34} - \pi_{14} \end{aligned} \tag{7}$$

$$\begin{aligned}&(\pi_{21}k - v\pi_{31}k)T_x + (\pi_{22}k - v\pi_{32}k)T_y \\ &+(\pi_{23}k - v\pi_{33}k)T_z + (\pi_{21} - v\pi_{31})X \\ &+(\pi_{22} - v\pi_{32})Y + (\pi_{23} - v\pi_{33})Z = v\pi_{34} - \pi_{24}\end{aligned} \quad (8)$$

Therefore, each projected point in each frame provides two equations. This means that on each time step t, by having K frames and N points, there are $2\times N\times(K+1)$ equations available. These equations can be written for any $\pi_{t+k\dot{o}}$, $u^n_{t+k\dot{o}}$, $v^n_{t+k\dot{o}}$, T_t, and P^n_t. The first three $\pi_{t+k\dot{o}}$, $u^n_{t+k\dot{o}}$, and $v^n_{t+k\dot{o}}$ are known parameters, while the other T_t and P^n_t are unknown, and T_t is constant for all the P^n_t located on the moving rigid object. Consequently, the total number of unknown parameters equals $3\times(N+1)$. This linear system can be written as follows:

$$A_t U_t = b_t \quad (9)$$

where A_t and b_t are known coefficients and biases made by equations 7 and 8, and U_t is an unknown vector containing all the unknown parameters as below:

$$U_t = \begin{bmatrix} T^x_t \\ T^y_t \\ T^z_t \\ X^1_t \\ Y^1_t \\ Z^1_t \\ \vdots \\ X^N_t \\ Y^N_t \\ Z^N_t \end{bmatrix} \quad (10)$$

The closed-form for this system that minimizes square error $\|AU-b\|^2$ is:

$$\hat{U}_t = \left(A_t^T A_t\right)^{-1} A_t^T b_t \quad (11)$$

When the system does not have any noise, $\hat{U}_t$ is equal to U_t. However, in practice, by having more noises included in the system, it is just a rough approximation of U_t. In order to optimize it for a more robust and accurate solution, a non-linear optimization is used to minimize the projection error where the optimization is initialized by $\hat{U}_t$:

$$\tilde{U}_t = \underset{U}{\arg min}\left(\sum_{k,n}\left\|\Pr\left(P^n_{t+k\dot{o}}\right) - \left[u^n_{t+k\dot{o}},\ v^n_{t+k\dot{o}}\right]^T\right\|^2\right) \quad (12)$$

This non-linear optimization helps to make the system more robust in the presence of noise.

3. Algorithm

We assume that after applying the conventional SFM on the static part of the scene for the calibrated camera, all the projection matrices $\pi_{t+k\dot{o}}$ are already estimated for all frames. Additionally, the moving object is already segmented by motion segmentation, and thus the corresponding points $u^n_{t+k\dot{o}}$, $v^n_{t+k\dot{o}}$ on the moving object are also available. Then our algorithm will do the process as below for all the moving objects to reconstruct the 3D structure of the moving rigid objects and their 3D motions:

Algorithm 1. Steps to reconstruct 3D motion and 3D structure of moving rigid objects.

1) Inputs:
   a. Projection matrices π for all frames
   b. Moving rigid objects' corresponding points u and v for all the frames
2) For each rigid object:
3) For all $K>1$ consecutive frames at time t and for all N points on the object, generate:
   $$A_t U_t = b_t$$
4) Compute $\hat{U}_t = \left(A_t^T A_t\right)^{-1} A_t^T b_t$ and optimize the term below initialized by $\hat{U}_t$:
   $$\tilde{U}_t = \underset{U}{\arg min}\left(\sum_{k,n}\left\|\Pr\left(P^n_{t+k\dot{o}}\right) - \left[u^n_{t+k\dot{o}},\ v^n_{t+k\dot{o}}\right]^T\right\|^2\right)$$
5) If the time is not ended, then $t \leftarrow t+1$ and go back to step 3.
6) Output:
   a. 3D structure of moving rigid objects in different times t
   b. 3D motion of moving rigid objects in different times t

4. Experiments

Recently, for a single image depth estimation taken by a single hand-held camera without the usage of any other equipment such as LIDAR, one of the very highlighted methods is the usage of deep learning to estimate the depth by only considering the image RGBs [39-43].

Although these approaches are generally handy and work in the presence of noise for a dense structure estimation, in our specific application where both the camera and the object are moving, the presented approach in our paper highlights different advantages, mentioned below:

- Deep learned approaches for single images just estimate the depth per each frame independently, while, we consider the relation of consecutive frames such that we can calculate the 3D motion of moving objects as well as the depth.

- These methods are dependent on the training data domain but we generalize our solution by mathematically modeling the geometry.
- They estimate the depth by an unknown complex model, while we compute the depth through a simple explainable model.
- They require a high-performance hardware for training and even maybe for testing, while we provide a simple closed-form term as a solution that can be run on typical machines.

In this section, our method is tested on both the simulated data and real video data in order to validate the theory, and show the effectiveness of our approach. For real data, the videos are taken from the Hopkins database [44]. Furthermore, we made a known virtual scene including a moving object using 3D MAX to compare our approach with a recent deep learned model AdaBins [43], and show its efficiency.

4.1. Approach validation on simulated data

In order to validate the equations, the synthetic data is generated based on a pinhole camera model. In this model, first, N number of randomly located 3D points P_t^n are considered. Additionally, $K>1$ consecutive frames in different camera poses (random translation and rotation) are considered, where the focal length is f . Then all N points are translated by a fixed translation T from each frame to the next one, and then projected through the corresponding frame. This allows us to generate all $\left[u_{t+k\dot{o}}^n,\ v_{t+k\dot{o}}^n\right]^T$ and $\pi_{t+k\dot{o}}$ as the known parameters, and then compute $\hat{U}_t$ and $\tilde{U}_t$ as the linear and non-linear estimations of T and all P_t^n, respectively.

In this experiment, all parameters are randomly initialized in a typical range reported in the table below:

Table 1. Parameters and their range used for randomly generated simulation data.

Parameter	Range
Focal length f	(10, 100) mm
Camera translation in 3D space	(0.05, 0.3) m in each direction
Camera rotation (roll, yaw, pitch)	(0, 20) degree for each direction
Number of consecutive frames (K)	(3, 10)
Number of points on the moving object (N)	(10, 100)
Object distance from the camera	(1, 50) m
Object translation (T_x, T_y, T_z)	(-1, 1) m for each element
Object size	< 2 m in each direction

In order to compute the error of 3D structure reconstruction, one of the well-known criteria is the re-projection error when we do not have the 3D structure ground truth. However, since in this experiment we have the ground truth available, we directly compute the average l2-norm distance of 3D structure and motion estimation with their ground truth as the error criteria. For this goal, 4 error criteria are introduced as follow:

Translation estimation errors:

- Linear model error: $\hat{e}_T = \left\|\hat{T}_t - T_t\right\|_2$
- Non-linear model error: $\tilde{e}_T = \left\|\tilde{T}_t - T_t\right\|_2$

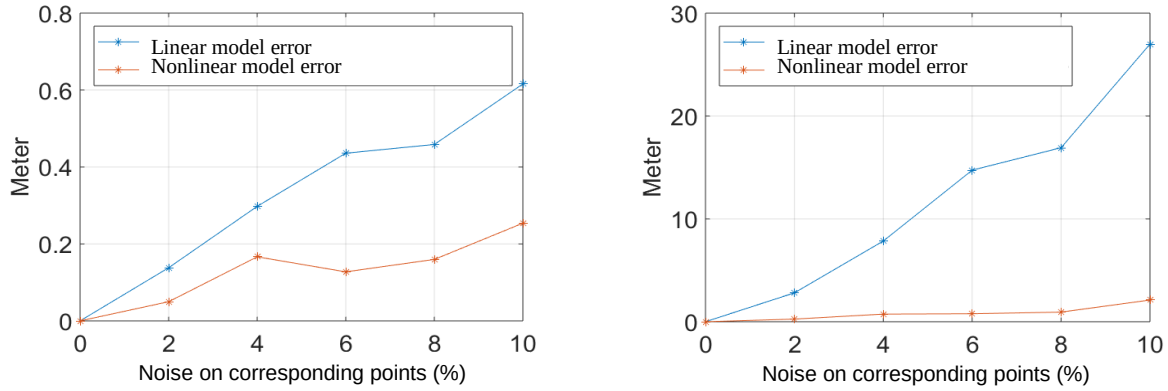
3D structure estimation errors:

- Linear model error: $\hat{e}_P = \dfrac{\left\|\begin{bmatrix}\hat{P}_t^1\\ \vdots \\ \hat{P}_t^N\end{bmatrix} - \begin{bmatrix}P_t^1\\ \vdots \\ P_t^N\end{bmatrix}\right\|_2}{N}$
- Nonlinear model error: $\tilde{e}_P = \dfrac{\left\|\begin{bmatrix}\tilde{P}_t^1\\ \vdots \\ \tilde{P}_t^N\end{bmatrix} - \begin{bmatrix}P_t^1\\ \vdots \\ P_t^N\end{bmatrix}\right\|_2}{N}$

In order to optimize our non-linear function, we use the Quasi-Newton method.

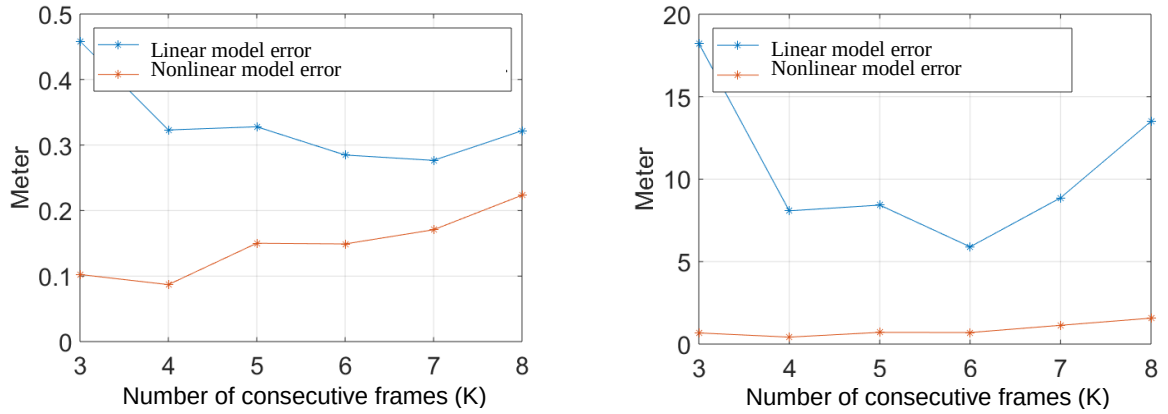
Since the major source of the noise existent in the model is the corresponding points noise, to model the effect of that, a uniform noise is added to the corresponding points, and the effect of that is evaluated in the following. The noise is randomly added up to a certain percent of disparity mean over all images. By increasing the noise percent from 0 to 10, we show $\hat{e}_T$, $\tilde{e}_T$, $\hat{e}_P$, and $\tilde{e}_P$ concerning the percent of noise in Figure 5. The experiments are repeated for 100 randomly initialized run, and the averages are shown in this figure.

As it is shown in this figure, when there is no noise added to the system, the equations are valid such that both the linear and non-linear solutions can exactly find both the 3D structure and the 3D motion of moving objects where the errors are zero. However, by adding noise to the corresponding points, the errors of estimation get higher, where in all cases, the non-linear optimization estimates the object's 3D structure and 3D translation more accurately.



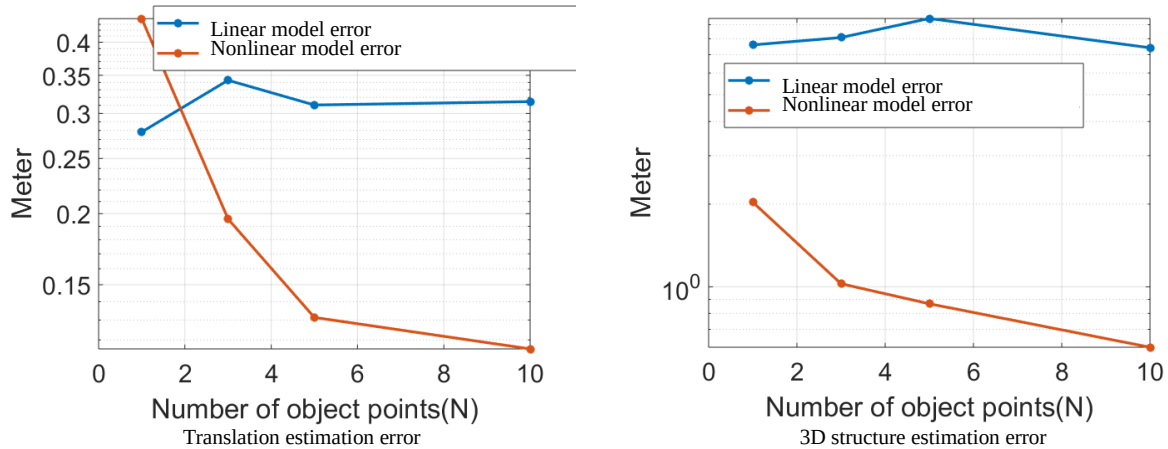
Translation estimation error 3D structure estimation error

Figure 5. Average error for estimation of 3D motion and 3D structure of moving rigid objects in the presence of noise. The error is shown for both the linear and non-linear solutions.



Translation estimation error 3D structure estimation error

Figure 6. Average error for estimation of 3D motion and 3D structure of moving rigid objects concerning different numbers of consecutive frames K .

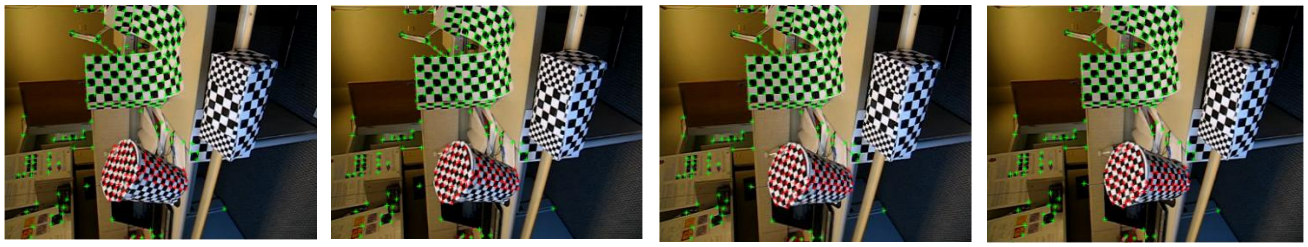


Translation estimation error 3D structure estimation error

Figure 7. Average error for estimation of 3D motion and 3D structure of moving rigid objects concerning different numbers of points N on the object.

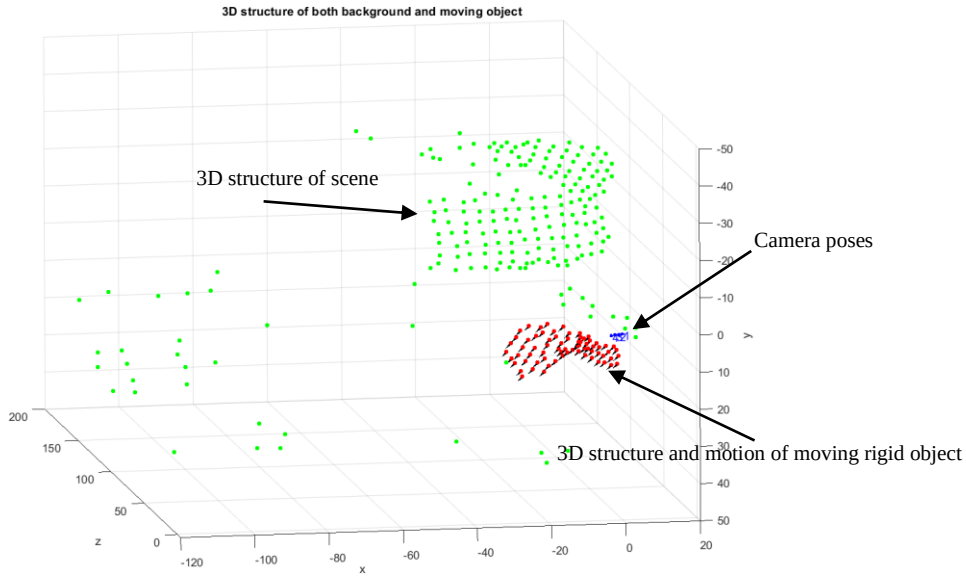
In addition to the effect of noise on the system, the effect of two other factors, the number of consecutive frames K where the object translation is estimated as constant for and the number of points on the moving object N are also validated on the accuracy of the system in the presence of noise. In order to show the effect of these factors, we fix the noise of the corresponding points on 5%, and then the errors of the model are shown concerning these two factors in Figures 6 and 7.

These figures show that the number of K does not have necessarily an increasing or decreasing effect on the accuracy of estimation; instead, to decrease the error of the model in the presence of the corresponding points noise, having more points (larger N) on the object is considerably effective.

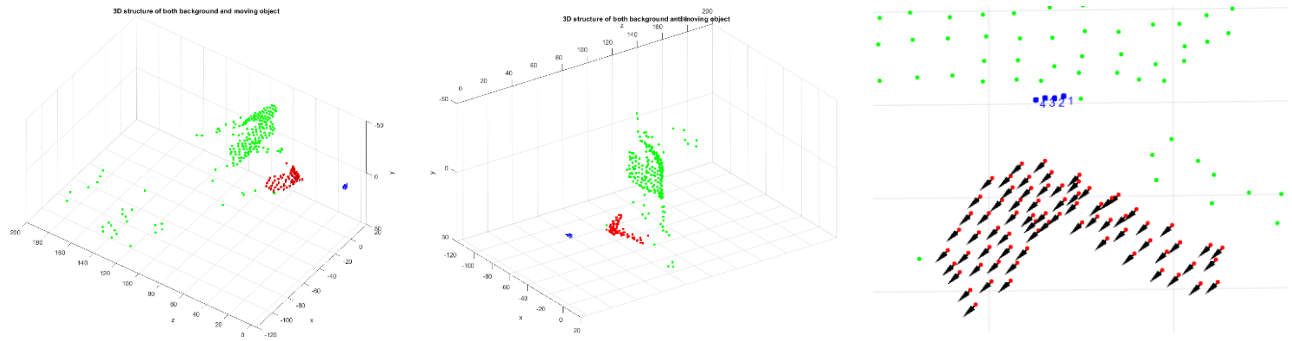


Frame 1 Frame 2 Frame 3 Frame 4

Figure 8. Interior video frames taken from Hopkins database; Green points: corresponding points for the static scene; Red points: corresponding points on the moving rigid object.



3D structure and 3D motion



Left view Right view Frontal magnified view

Figure 9. Reconstructed structure for both static scene (in green) and moving rigid object (in red), in addition to the 3D motion of moving object (black arrows) are shown for the frames represented in Figure 8. For a better visualization, different views are shown.

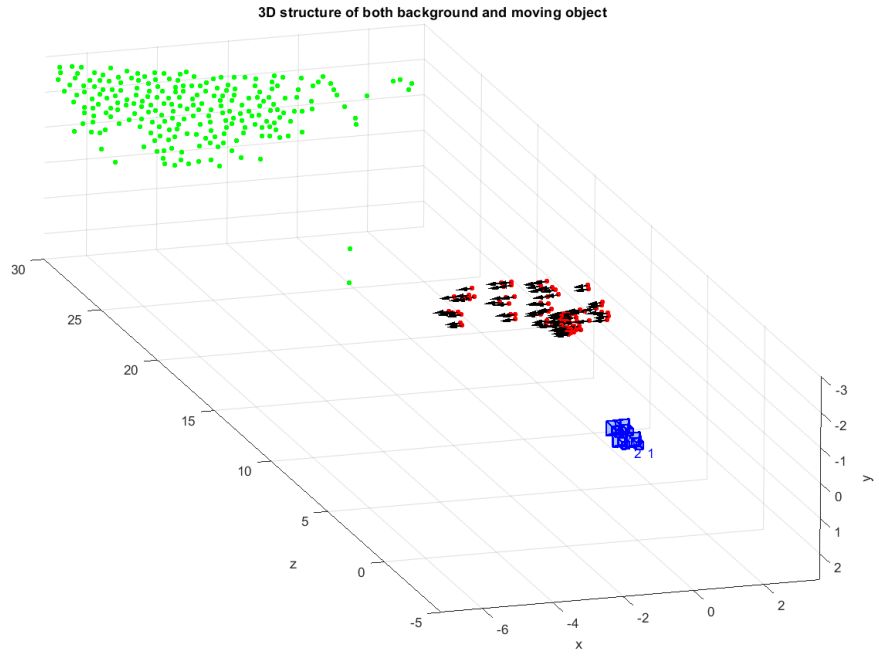
4.2. Result on real data

In this section, the method is tested on the real video data taken from the Hopkins database [44]. In this database, the corresponding points on the static and moving object are already segmented and given where we can test our algorithm to reconstruct the 3D structure and motion of the moving object in the static scene coordinate.

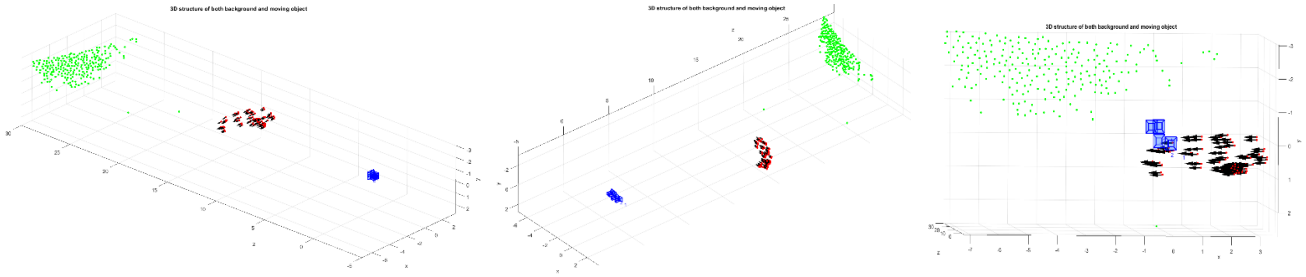
In the video samples tested here (Figures 8-11), there are two segments of points: the points on the static scene (the green points) and the points on the moving rigid object (the red points). In these scenes, the object is moving while the camera is also freely moving. By having the intrinsic camera parameters, the motion of the camera is estimated by applying the conventional SFM on the static part of the scene, which results in the projection matrices for all the frames. It additionally gives us the 3D structure of the static scene up to an unknown scale.

Frame 1 Frame 2 Frame 3 Frame 4

Figure 10. Exterior video frames taken from Hopkins database; Green points: corresponding points for the static scene; Red points: corresponding points on the moving rigid object.



3D structure and 3D motion



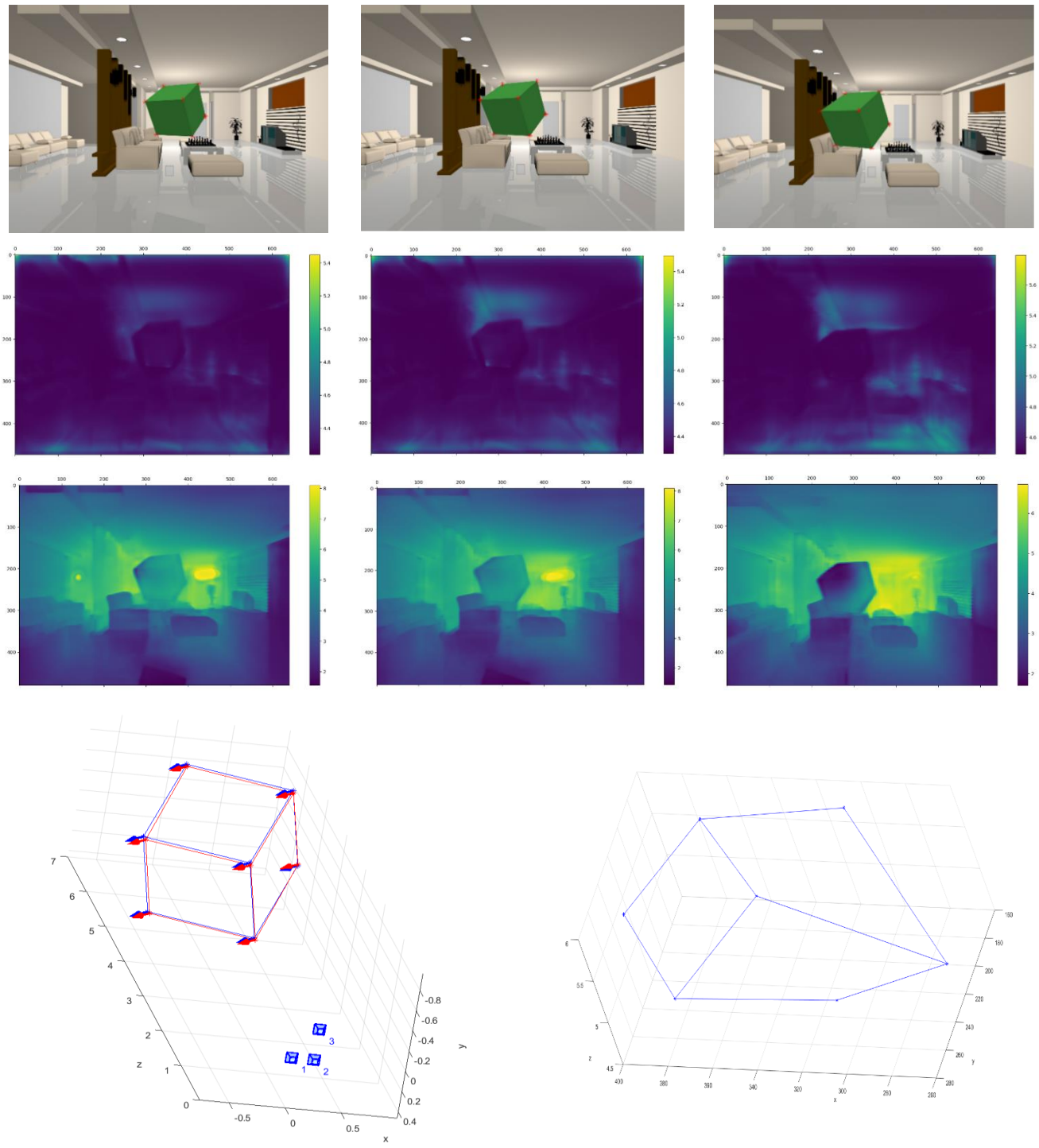
Left view Right view Frontal magnified view

Figure 11. Reconstructed structure for both static scene (in green) and moving rigid object (in red), in addition to the 3D motion of moving object (black arrows) are shown for the frames represented in Figure 10. For a better visualization, different views are shown.

Then by having the corresponding points on the moving object and the projection matrices, our method is used to estimate both the 3D structure and the 3D motion of the moving object in the coordinate of the static scene.

As it is shown in Figure 8 for an interior scene, the camera is moving to the left for 4 frames, while the basket itself is rotating down simultaneously. The red points shown in Figure 9 show the 3D structure of the moving object, where the motion is also estimated as black arrows. In another test, Figures 10 and 11 represent an example of a real vehicle moving in an exterior static scene.

In these figures, similarly, the 3D structure and 3D motion of the moving object are estimated for 4 frames to show the effectiveness of the presented method on the real data.



Our method result (blue: ground truth; Red: our result)

Single image depth estimation by a deep network

Figure 12. Comparison of our approach with deep learned model AdaBins on simulated frames. First row: the frames generated by 3D MAX (sorted from left to right); second row: AdaBins result when the model is trained with KITTI database; third row: AdaBins result when the model is trained with NYU database; fourth row left: 3D motion and 3D reconstruction of moving object calculated by our approach where the ground truth is in blue and our result is in red; fourth row right: result of 3D reconstruction from AdaBins trained with NYU where the scale is unknown, and the shape is deformed.

4.3. Result comparison on simulated frames

In this part, a virtual scene is simulated by 3D MAX, where the geometry is known to us, as it is shown in Figure 12. We made a moving one-meter cubic object inside. Then both our approach and a new complex deep learned model AdaBins [43] that is used for single image depth estimation are compared in this figure to show the effectiveness of our approach in a specific case study, where both the camera and the object are freely moving.

As it is shown, our approach reconstructed the 3D structure of the moving rigid object more precisely than the deep model. This is because it

models the geometry between the consecutive frames. In addition, using our approach, the 3D motion of the moving object between the frames is also calculated, while the deep model does not give this information because it works per each image separately.
This figure also shows that the result of the deep model can variate when the training data changes (from KITTI to NYU) but our mathematical modeling generalizes the approach since it is not dependent on training data.

5. Conclusion

The usage of conventional SFM for a moving rigid object in the static scene is not possible without constraint since both the camera and the object itself are moving freely and simultaneously. In this work, since the motion of a moving rigid object was continuous, the speed of that for a short period of time was approximated as a fixed vector, and therefore, zero-acceleration. Using this constraint, we could find a closed-form solution to both reconstruct the 3D structure of the object and also estimate the 3D motion of that using only 3 consecutive frames. Additionally, in order to make the solution robust for noise, a non-linear optimization was used. The theory was first explained mathematically, and then validated on the simulated data. Finally, it was tested on the real image sequences as well as the simulated frames in order to show how effectively our method solved the issue.

References

[1] M. R. U. Saputra, A. Markham, and N. Trigoni, "Visual SLAM and structure from motion in dynamic environments: A survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 2, p. 37, 2018.

[2] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[3] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91-110, 2004.

[4] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*, 2006: Springer, pp. 404-417.

[5] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *ICCV*, 2011, vol. 11, no. 1: Citeseer, p. 2.

[6] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "Lift: Learned invariant feature transform," in *European Conference on Computer Vision*, 2016: Springer, pp. 467-483.

[7] H. Kamali Ardakani, S. A. Mousavinia, and F. Safaei, "Camera Arrangement using Geometric Optimization to Minimize Localization Error in Stereo-vision Systems," *Journal of AI and Data Mining*, vol. 9, no. 3, pp. 295-307, 2021.

[8] M. Karami, A. Moosavie Nia, and M. Ehsanian, "Camera Arrangement in Visual 3D Systems using Iso-disparity Model to Enhance Depth Estimation Accuracy," *Journal of AI and Data Mining*, vol. 8, no. 1, pp. 1-12, 2020.

[9] J. Civera, A. J. Davison, and J. M. M. Montiel, *Structure from motion using the extended Kalman filter*. Springer Science & Business Media, 2011.

[10] M. Pupilli and A. Calway, "Real-Time Camera Tracking Using a Particle Filter," in *BMVC*, 2005: Citeseer.

[11] H. Strasdat, J. M. Montiel, and A. J. Davison, "Visual SLAM: why filter?," *Image and Vision Computing*, vol. 30, no. 2, pp. 65-77, 2012.

[12] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381-395, 1981.

[13] O. Chum and J. Matas, "Matching with PROSAC-progressive sample consensus," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005, vol. 1: IEEE, pp. 220-226.

[14] P. H. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Computer vision and image understanding*, vol. 78, no. 1, pp. 138-156, 2000.

[15] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European conference on computer vision*, 2014: Springer, pp. 834-849.

[16] C. Yu *et al.*, "DS-SLAM: A semantic visual SLAM towards dynamic environments," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018: IEEE, pp. 1168-1174.

[17] J. Cheng, Z. Wang, H. Zhou, L. Li, and J. Yao, "DM-SLAM: A Feature-Based SLAM System for Rigid Dynamic Scenes," *ISPRS International Journal of Geo-Information*, vol. 9, no. 4, p. 202, 2020.

[18] M. S. Bahraini, A. B. Rad, and M. Bozorg, "Slam in dynamic environments: A deep learning approach for moving object tracking using ml-ransac algorithm," *Sensors*, vol. 19, no. 17, p. 3699, 2019.

[19] C.-C. Wang and C. Thorpe, "Simultaneous localization and mapping with detection and tracking of moving objects," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, 2002, vol. 3: IEEE, pp. 2918-2924.

[20] C.-C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous localization,

mapping and moving object tracking," *The International Journal of Robotics Research*, vol. 26, no. 9, pp. 889-916, 2007.

[21] M. Derome, A. Plyer, M. Sanfourche, and G. L. Besnerais, "Moving object detection in real-time using stereo from a mobile platform," *Unmanned Systems*, vol. 3, no. 04, pp. 253-266, 2015.

[22] J. P. Costeira and T. Kanade, "A multibody factorization method for independently moving objects," *International Journal of Computer Vision*, vol. 29, no. 3, pp. 159-179, 1998.

[23] Y. Murakami, T. Endo, Y. Ito, and N. Babaguchi, "Depth-Estimation-Free condition for projective factorization and its application to 3d reconstruction," in *Asian Conference on Computer Vision*, 2012: Springer, pp. 150-162.

[24] R. Sabzevari and D. Scaramuzza, "Monocular simultaneous multi-body motion segmentation and reconstruction from perspective views," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014: IEEE, pp. 23-30.

[25] R. Sabzevari and D. Scaramuzza, "Multi-body motion estimation from monocular vehicle-mounted cameras," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 638-651, 2016.

[26] C. Bregler, A. Hertzmann, and H. Biermann, "Recovering non-rigid 3D shape from image streams," in *cvpr*, 2000, vol. 2, no. 2: Citeseer, p. 2690.

[27] Y. Dai, H. Li, and M. He, "A simple prior-free method for non-rigid structure-from-motion factorization," *International Journal of Computer Vision*, vol. 107, no. 2, pp. 101-122, 2014.

[28] S. Kumar, Y. Dai, and H. Li, "Multi-body non-rigid structure-from-motion," in *2016 Fourth International Conference on 3D Vision (3DV)*, 2016: IEEE, pp. 148-156.

[29] M. Paladini, A. Del Bue, M. Stosic, M. Dodig, J. Xavier, and L. Agapito, "Factorization for non-rigid and articulated structure using metric projections," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009: IEEE, pp. 2898-2905.

[30] J. Xiao, J.-x. Chai, and T. Kanade, "A closed-form solution to non-rigid shape and motion recovery," in *European conference on computer vision*, 2004: Springer, pp. 573-587.

[31] R. I. Hartley and P. Sturm, "Triangulation," *Computer vision and image understanding*, vol. 68, no. 2, pp. 146-157, 1997.

[32] S. Avidan and A. Shashua, "Trajectory triangulation of lines: Reconstruction of a 3d point moving along a line from a monocular image sequence," in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, 1999, vol. 2: IEEE, pp. 62-66.

[33] S. Avidan and A. Shashua, "Trajectory triangulation: 3D reconstruction of moving points from a monocular image sequence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 4, pp. 348-357, 2000.

[34] A. Shashua, S. Avidan, and M. Werman, "Trajectory triangulation over conic section," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, vol. 1: IEEE, pp. 330-336.

[35] J. Y. Kaminski and M. Teicher, "General trajectory triangulation," in *European Conference on Computer Vision*, 2002: Springer, pp. 823-836.

[36] H. S. Park, T. Shiratori, I. Matthews, and Y. Sheikh, "3D reconstruction of a moving point from a series of 2D projections," in *European conference on computer vision*, 2010: Springer, pp. 158-171.

[37] A. Kundu, K. M. Krishna, and C. Jawahar, "Realtime multibody visual SLAM with a smoothly moving monocular camera," in *2011 International Conference on Computer Vision*, 2011: IEEE, pp. 2080-2087.

[38] H. S. Park, T. Shiratori, I. Matthews, and Y. Sheikh, "3D trajectory reconstruction under perspective projection," *International Journal of Computer Vision*, vol. 115, no. 2, pp. 115-135, 2015.

[39] S. Qiao, Y. Zhu, H. Adam, A. Yuille, and L.-C. Chen, "Vip-deeplab: Learning visual perception with depth-aware video panoptic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3997-4008.

[40] S. Lee, J. Lee, B. Kim, E. Yi, and J. Kim, "Patch-Wise Attention Network for Monocular Depth Estimation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, vol. 35, no. 3, pp. 1873-1881.

[41] S. Aich, J. M. U. Vianney, M. A. Islam, M. Kaur, and B. Liu, "Bidirectional attention network for monocular depth estimation," *arXiv preprint arXiv:2009.00743*, 2020.

[42] F. Aleotti, G. Zaccaroni, L. Bartolomei, M. Poggi, F. Tosi, and S. Mattoccia, "Real-time single image depth perception in the wild with handheld devices," *Sensors*, vol. 21, no. 1, p. 15, 2021.

[43] S. F. Bhat, I. Alhashim, and P. Wonka, "Adabins: Depth estimation using adaptive bins," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4009-4018.

[44] R. T. a. R. Vidal, "A Benchmark for the Comparison of 3-D Motion Segmentation Algorithms," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, doi: 10.1109/CVPR.2007.382974.

رویکردی جدید برای تخمین حرکت و ساختار یک جسم متحرک صلب در فضای سه بعدی با استفاده از یک دوربین دستی متحرک

رضا سراجه[۱]، امیر موسوی نیا[۲،*] و فرزاد صفایی[۳]

[۱] دانشکده برق، دانشگاه صنعتی خواجه نصیرالدین طوسی، تهران، ایران.

[۲] دانشکده کامپیوتر، دانشگاه صنعتی خواجه نصیرالدین طوسی، تهران، ایران.

[۳] دانشکده انفورماتیک، دانشگاه ولونگونگ، ولونگونگ، استرالیا.

ارسال ۲۰۲۱/۰۹/۰۵؛ بازنگری ۲۰۲۲/۰۲/۰۲؛ پذیرش ۲۰۲۲/۰۳/۱۴

چکیده:

رویکردهای مرسوم SFM به طور گسترده به منظور بازسازی سه بعدی صحنه ثابت با استفاده از تخمین حرکت دوربین، استفاده شده‌اند. با این حال، زمانی که در صحنه اجسام متحرک با مدل حرکتی نامشخص وجود دارند، این روش ها با مشکل روبرو می‌شوند. این مقاله نشان می‌دهد، زمانی که تعداد فریم های ویدئو به اندازه کافی زیاد باشد و حرکت شئ در صحنه پیوسته در زمان و یا به عبارتی با شتاب محدود باشد، می‌توان به صورت خطی، حرکت و ساختار شئ را تخمین زد. این تئوری در ابتدا به صورت یک معادله ریاضی بیان شده است و در نهایت با استفاده از یک بهینه سازی غیر خطی بهبود داده شده است تا راه حل مسئله مذکور را ارائه نماید. این الگوریتم بر روی داده های شبیه سازی شده و همچنین فریم های واقعی گرفته شده از پایگاه داده Hopkins، ارزیابی شده است.

کلمات کلیدی: تخمین سه بعدی صحنه با استفاده از مدل حرکت دوربین، بازسازی سه بعدی صحنه متحرک، تخمین سه بعدی مدل حرکت یک شئ صلب.