



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

IHTISHAM ALI
VISUAL SIMULTANEOUS LOCALIZATION AND MAPPING IN AN
ACTIVE DYNAMIC ENVIRONMENT

Master of Science Thesis

Examiners: Prof. Atanas Gotchev
and Prof. Jouni Mattila
Examiner and topic approved on 16
December 2016

ABSTRACT

IHTISHAM ALI: Visual SLAM in an Active Dynamic Environment

Tampere University of technology

Master of Science Thesis, 52 pages

August 2017

Master's Degree Programme in Automation Engineering

Major: Factory Automation and Industrial Informatics

Minor: Learning and Intelligent Systems

Examiners: Professor Atanas Gotchev and Professor Jouni Mattila

Keywords: SLAM, VSLAM, dynamic, GMM, stereo, segmentation, motion detection

In recent years, the work on simultaneous localization and mapping has matured significantly. Robust techniques have been developed to explore and map a static environment in real-time. However, the problem of localizing and mapping a dynamic environment is still to be solved. The dynamic part of the environment not only makes the localization difficult but it introduces a diverse set of challenges to the existing problems such as detecting, tracking and segmenting the moving objects, and 3D reconstruction of the moving objects and/or static environment.

This thesis focuses on studying the problem of simultaneously localizing and mapping a actively dynamic environment. A comprehensive review and analysis of the state-of-the-art methods is provided for both static and dynamic cases. A stereo camera is used to explore the dynamic environment and obtain semi-dense point clouds for the image sequence. The proposed approach is a variant of the standard ICP where the outliers of the registration process are not discarded. All 3D points are assigned a confidence measure based on their association in their respective neighborhood. The confidence measure decides if a 3D point is classified static or dynamic in the global map. Hence, the approach does not require any prior information about the environment or the moving objects. In the latter part of this study, the moving objects are segmented in 3D space and 2D images for any potential future analysis. The framework is tested with highly dynamic scenes from both indoor and outdoor environments. The results demonstrate the effectiveness of the proposed approach.

PREFACE

The research for this thesis work was motivated by an initial demonstration work at 3D Media research group, at TUT.

I would like to thank Atanas Gotchev and Jouni Mattila for their guidance and feedback on my thesis work. I am greatly thankful to Olli Suominen, who not only supervised my work but introduced me to the topic, to begin with. It is unarguably due to his patience and guidance that I could learn and complete this work. Moreover, a lot of credit goes to my friends, especially, Shadman Siddiqui, who helped along this lengthy process, always offering support and advice.

And finally, I can't thank enough my family for their never-ending love and support.

Tampere, 02.08.2017

Ihtisham Ali

CONTENTS

1.	INTRODUCTION	1
2.	BACKGROUND	3
2.1	SLAM and Visual SLAM	3
2.2	State-of-the-art for VSLAM in Dynamic Environment	5
2.3	Stereo Camera Geometry and Calibration	7
2.3.1	Camera Model.....	7
2.3.2	Camera Calibration	9
2.3.3	Epipolar Geometry & Image Rectification	9
2.3.4	Salient Feature detection and triangulation.....	11
2.3.5	Dense Disparity Estimation	13
2.3.6	Comparison of Feature matching and Dense Stereo Estimation....	15
3.	PROPOSED METHOD	17
3.1	Framework	17
3.2	Global Cloud Model.....	18
3.3	Point Cloud Alignment and Data Association	18
3.4	Merging and Confidence Gain	22
3.5	Confidence Reduction	24
3.6	Removal of Unstable points	25
3.7	Extraction and Clustering of Dynamic Points.....	25
3.8	Segmentation of Dynamic objects (2D images).....	28
3.9	Mask Obtained from clustered points	28
3.10	Motion Mask obtained using GMM.....	29
3.11	Segmentation using combined masks	32
4.	EXPERIMENTAL SETUP AND RESULTS	34
4.1	Case 1: Object introduction to scene and removal.....	35
4.2	Case 2: Passer-by in a moving scene	38
4.3	Case 3: Outdoor environment mapping	40
4.4	Case 4: Narrow Corridor with Semi-Transparent surfaces	42
5.	CONCLUSIONS.....	46
	REFERENCES.....	48

LIST OF FIGURES

Figure 1.	<i>Pinhole camera model. A Point P in 3D space is mapped to a 2D point p on image plane I by the ray connecting P with the center of projection C.</i>	7
Figure 2.	<i>Points following the constraints of Epipolar geometry (reconstructed from [37]).</i>	10
Figure 3.	<i>Stereo pair Rectification (a) Original images acquired using the Zed camera; the straight white line passes through same image coordinates (b) Rectified pair; the same visual points in left and right pair lie on the white line which depicts the Epipolar line.</i>	11
Figure 4.	<i>Feature point detection, tracking and triangulation (a, b) Stereo pair acquired using the Zed camera (c) Corner Feature points detected in the left frame of stereo pair (d) Detected corner points are tracked in the stereoAnaglyph (red-cyan stereo pair) (e) Filtered points tracked in bound to Epipolar constraint (f) Sparse 3D point cloud generated from the tracked feature points by triangulation.</i>	13
Figure 5.	<i>Feature point detection, tracking and triangulation (a, b) Stereo pair acquired using the Zed camera (c) Corner Feature points detected in the left frame of stereo pair (d) Detected corner points are tracked in the stereoAnaglyph (red -cyan stereo pair) (e) Filtered points tracked in bound to Epipolar constraint (f) Sparse 3D point cloud generated from the tracked feature points by triangulation.</i>	15
Figure 6.	<i>Result of Point Cloud registrations (a) Sparse clouds from corner feature detection at 0.001 threshold (b) Dense clouds at uniqueness threshold of 60 (a considerably high value).</i>	16
Figure 7.	<i>Flowchart of the proposed approach</i>	17
Figure 8.	<i>Distance measures in 2D case between two curves P and Q.</i>	19
Figure 9.	<i>Point Cloud registration failure of the dataset with moving person at parameters: Inlier ratio=0.8, point-to-plane error metric and point cloud merging at grid factor of 0.0135 m³ (a) Static environment i.e. before person walks in to the scene (a) Dynamic environment</i>	21
Figure 10.	<i>Point Correspondence Selection</i>	22

Figure 11.	Confidence gain during merging of points.....	23
Figure 12.	Projection of points to image plane for confidence reduction (a) illustration of the camera viewing the global cloud (b) projected points onto the image plane from the perspective view.....	25
Figure 13.	Extraction and clustering of Dynamic points(a) Map of the environment containing both static and dynamic/unstable points (b) Extracted dynamic and/ or unstable points (c)dynamic points extracted from within the bounds of static map (d) clustering of points and removal of small clusters (e) visualized clusters pertaining to possible dynamic objects	27
Figure 14.	Mask obtained by projecting the clustered dynamic points	28
Figure 15.	Mixture model for distribution components 1 and 2	29
Figure 16.	Obtaining motion mask from GMM with moving scene (a) previous frames transformed to the current frame 't' (b) mask obtained when training images were not geometrically transformed (c) mask obtained when training images were geometrically (d) mask obtained for the moving person using the proposed approach	32
Figure 17.	Segmentation of the dynamic object using binary masks (a) Mask obtained by projecting the clustered 3D dynamic points onto image plane (b) Masked obtained using GMM based motion detection(c) Segmented dynamic object from 2D images.....	33
Figure 18.	Test case 1; first column shows the original images, second column shows the registered map and the third column presents the segmented moving object.....	38
Figure 19.	Video test case 2, where first column shows the original images, second column shows the registered map and the third column presents the segmented moving object.	40
Figure 20.	Video test case 3. The video was captured in an outdoor dynamic environment. The first column shows the original images, second column shows the registered map.....	42
Figure 21.	Video test case 4. The test data was recorded at Tietotalo, Tampere University of Technology. The first column shows the original images, second column shows the registered map.	45

LIST OF SYMBOLS AND ABBREVIATIONS

SLAM	Simultaneous Localization and Mapping
CML	Concurrent Mapping and Localization
VO	Visual Odometry
GPS	Global Positioning Systems
EKF	Extended Kalman Filter
FastSLAM	Factored Solution to SLAM
MonoSLAM	Monocular Simultaneous Localization and Mapping
SFM	Structure from Motion
ARAP	As Rigid As Possible
COP	Centre of Projection
R^n	n dimensional real coordinate space
c	Optical Centre
f	Focal Length
K	Intrinsic matrix
R	Rotational Matrix
t	Translational vector
t_{max}	Threshold time
KLT	Kanade-Lucas tracker
SGM	Semi-Global Matching
NoFrames	Indicating Number of frames a 3D point has been in view of camera
ICP	Iterative Closest Point
GMM	Gaussian Mixture Model

1. INTRODUCTION

One of the research aims at the junction of Robotics and Computer Vision is to enable robots to autonomously explore unknown, unstructured and possibly dynamic environments. The intuitive approach to solving this problem is to build and maintain a map of the environment and localize the robot within that map. This approach is generally termed as Simultaneous Localization and Mapping (SLAM). When one or many cameras are utilized as the sensing mechanism, then the term Visual SLAM is utilized. During exploration of the environment, if only the camera moves and all the underlying objects in the environment are stationary, then the environment is assumed to be static and ideal for the approach. This assumption can also be held when some moving objects appear for just a few frames of the image sequence and that they reside in only a small portion of the image. On the contrary, if the moving objects cover a large part of the image or if they are in the view of the camera for a long span of time, then it can be easily classified as a dynamic environment. Even though it may seem trivial to handle the dynamic object as an accessory to the static environment, unfortunately, that is not the case. The introduction of a moving object to a static scene often has strong impact on the performance of traditional SLAM approaches.

Furthermore, the dynamic behavior of the environment cannot be ruled under a single description. The extent of dynamic change in the scene and the resulting effect depend entirely on the intended use of the system. Some researchers are interested in studying the slowly varying conditions of the environment such as luminosity. An outdoor environment could be examined for extended periods under flexible light and weather conditions. These changes can induce considerable variation in the appearance of the scene. However, the effects are only observable in long-term mapping since the changes either occur gradually or out of the current view of the camera. Moreover, they are explored only in the next round of exploration. Therefore, the environment in general remains static and the same approach of static environment can be utilized.

An intense dynamic environment, where the objects in view are in continuous motion pose a more difficult challenge. Consider a mobile robot which is building a dense map of an outdoor environment. The unaccounted moving vehicles and pedestrians present in such a scene can considerably disrupt the localization and mapping process. Nonetheless, solving such problems is crucial for both research and real-world applications

The primary aim of this thesis is to develop a Visual SLAM based approach that can effectively perform in an intense dynamic environment. The framework is tested on real-world datasets with two indoor and one outdoor case scenarios. These tests demonstrate

the ability of the method to build and update a map in a true dynamic environment. For a system to work unsupervised in a real dynamic environment, it is essential to maintain a level of autonomy and independence, therefore, no prior knowledge about the motion model (for camera or dynamic objects) and template (for dynamic objects) is considered. The system extracts information about 3D points pertaining to dynamic objects during the proposed framework and segments the points from the static map. The study emphasizes on building a map of the explored environment under dynamic circumstances, hence, moving objects are retained in the map as long as they are viewed in the scene. Moreover, motion based segmentation is performed using the fused information from 3D map and Gaussian Mixture Models (GMM), for further analysis of the dynamic objects in 2D image space. However, the analysis of the dynamics objects and large-scale mapping is not in the scope of this study.

The thesis is organized as follows. Chapter 2 presents the related work for SLAM in both static and dynamic environments. Furthermore, it describes the theoretical background in stereo camera geometry. Chapter 3 details the approach proposed in this study for Visual SLAM in dynamic environment. Next, Chapter 4 tests the efficiency of the method on various dataset. Finally, Chapter 5 provides conclusion drawn based on this study, where limitations of the proposed approach and the potential future research is also briefly discussed.

2. BACKGROUND

2.1 SLAM and Visual SLAM

In 1985, Chatila and Laumond [1] proposed to localize and map an observer in a parallel fashion. The idea proved to carry on and it was later known as Concurrent Mapping and Localization (CML) [2] or Simultaneous Localization and Mapping (SLAM). The latter term has now been generally adopted in the literature.

SLAM is not a specific sequence of steps but rather an approach whereby an observer (vehicle, robot or simply sensor with a processing unit) creates a globally consistent map of an environment and at the same time localizes itself within that map at each instant.

As aforementioned, in order to build a map of the environment, it is necessary that a suitable sensing device is selected. The sensor should be capable of perceiving the environment and provide accurate measurements. The most common sensors utilized for SLAM are lasers [3], global positioning systems (GPS) [4], sonar [5] and cameras [6]. Each of these sensing devices has its own strengths and inherent limitations. Sonar and Laser are capable of providing very dense and accurate information pertaining to the structures in the environment. However, their use becomes troublesome in a highly-cluttered setting or for object recognition application. Furthermore, good quality sonar and laser range finders are mostly expensive, and their structure is both heavy and large, thus making their use difficult for airborne or light weight robotic framework. Meanwhile, GPS sensor lacks the ability to provide information about the structural details of the environment. It can only be used to create a 2D map when functioning alone. Furthermore, GPS does not work effectively in narrow streets, under water, tunnels and sometimes indoor. However, GPS has been used in many SLAM implementations as a supplementary sensor where its information is fused with the primary sensor, thus aiding significantly to localization problem and making the implementation robust [7].

On the other hand, cameras prove to be of significant aid for a variety of SLAM implementations. This is due to the fact that camera based systems are capable of acquiring a wide range of information and further have a multitude of data manipulation methods. A SLAM implementation that employs camera as the primary sensing mechanism is known as visual SLAM [8]. Visual SLAM offers the added benefit of utilizing the innate capabilities of a camera such as color, texture and visual intensity observations for high level tasks like reconstruction [9], detection and recognition [10] of people, objects, and places. Moreover, a camera has the added benefit of being comparatively lighter, cheap and power efficient to other sensors. However, the use of camera introduces its own set of problems to the equation. These problems may arise due to several factors such as blurred images, light changes, lack of textural information and transparent objects in the scene.

Nonetheless, these problems can be mitigated by appropriate camera specification selection or other effective counter measures.

The initial research on using visual cues for mapping was conducted in the 2000s [11, 12] using stereo configuration. Other tried implementing the approach by using a different camera configuration. In [13], Pillia et al. used a monocular camera to implement SLAM in an indoor environment with the aim of recognizing objects with a minimum number of cameras. However, monocular camera poses the problem of scale ambiguity for cases where exact measurements might seem beneficial after reconstruction. Other camera configurations include multi-camera rigs that may either have overlapping or distinct views [14, 15], cameras with wide-angle lens [16], and omnidirectional camera [17]. In recent years, the use of RGB-D cameras has proved valuable for indoor SLAM implementations [18]. These cameras provide both color information and the depth map of the view which assist to complement together for easier data processing and result verification. However, all these advantages are lost when outdoor conditions are considered, especially under sunlight.

Nonetheless, a stereo configuration with large overlapping views still has its relevancy to SLAM due to its accuracy and simplicity in use. The landmarks visible in the overlapping region of the views can be accurately converted to their scaled real 3D positions using *triangulation* procedure [19]. One of the most effective and robust implementation is presented by Engel et al. in [20] that utilizes stereo camera for mapping a large scale outdoor environment in real-time with a standard CPU.

Although many solutions have been proposed and implemented for Visual SLAM, there still exist a few limitations to these techniques. As a result, many implementations produce a map with a large accumulated error and some may even fail the process. These limitations or problems arise due to the fact that few assumptions are made about the environment and/or video acquisition process, that render the implementation useless when these assumptions are not met in the real world. The three core assumptions are:

- 1- Camera motion is smooth and the appearing features in the views are consistent. This assumption often fails in cases when the camera is attached to a quadcopter, humanoid robot or held by a person exploring the environment. The nature of motion is erratic in this case and can lead to erroneous mapping and localization for scenes with a repetitive texture [21].

- 2- All the acquired images are sharp enough for the salient features to be observed. However, this assumption fails when the camera is moving sharply, the objects in the scene moves rapidly or camera turns out of focus. This problem may lead to a total system failure since the required number of salient features are not successfully detected. In 2008, Mei and Reid studied the problem of tracking visual features in on blurred images in real-time [22].

3- The objects in view are stationary and composed of rigid elements. However, this is never the case for real world experimentation. An outdoor environment mostly contains moving objects such as vehicles. Moreover, these objects may not necessarily be rigid e.g. pedestrians moving through the scene. Most of the implementations are not able to accommodate for moving objects and hence fails or generate erroneous maps due to incorrect associations. Some implementations are able to Map and localize successfully in a dynamic environment by considering the moving objects as noise as discarding them from the system. However, these implementations do not retain any information about the moving objects or the dynamics of the scene.

In relevance to frequent occurrence, the first and third problem are crucial and needs to be accommodated for. Much work has been done in regard to improving the localization and pose estimation by introducing probabilistic methods such as Extended Kalman Filter (EKF), Factored Solution to SLAM (FastSLAM). These methods have significantly compensated for the problem and perform effectively under varying circumstances. However, SLAM in a dynamic environment has remained a challenge for a long time. In the next section, we will discuss the State-of-the-art techniques for Visual SLAM in a dynamic environment.

2.2 State-of-the-art for VSLAM in Dynamic Environment

As mentioned before, most SLAM implementations hold steadfast to the assumption that the environment is static or stationary. This assumption was necessary at the initial time, to progressively develop efficient variants of SLAM. The classical SLAM approaches have matured enough that it has little room for improvement. Nonetheless, the problem for SLAM or Visual SLAM, especially in a dynamic environment, is far from solved.

Many recent studies still build upon the assumption of static environment to solve VSLAM problems. Davison et al. [23] proposed a camera tracking system called monoSLAM (monocular Simultaneously Localization and Mapping) to localize and map a newly explored environment. The implementation utilized EKF to effectively calculate the camera pose in real time. Afterward, Newcombe and Davison [24] suggested using Structure from Motion (SFM) for calculating the ego-motion and later, to reconstruct a model of the environment. In addition to the aforementioned studies, the work presented in [25], [26], and [27] hold the postulate that the environment to be explored is static.

In [28], Winston Churchill and Paul Newman propose a technique called lifelong exploration for building map of an outdoor environment based on the accumulated changes. If the localization process fails to register the scene to its previous depiction in the map, a fresh perspective of the environment is registered based on the Visual Odometry (VO). In another study, the dynamics are changed in an offline manner and an update of the map

is required [29]. A static environment is mapped iteratively, where part of the environment is only changed. The purpose of the study was to examine the long-term mapping of an environment that shows a slight change in dynamics. Nonetheless, both these scenarios exhibit a common trait. The environment under study changes gradually and can still be mapped with the classical SLAM approaches.

One way of handling a dynamic environment is to strictly focus on the static points of the environment and discard all the unstable points in order to preserve the map. Konolige and Bowman [30] proposed a vision-based pose-graph SLAM to map an environment with both actively moving objects (e.g. moving people) and passively moving objects (e.g. moved furniture). The approach views the environment as a neighborhood at that time instant. Exemplars are selected for a neighborhood by the proposed least-recently-used-view deletion algorithm. The algorithm helps to limit the exemplars per neighborhood to a minimum figure in order to explain the neighborhood's visual variation with a least set of data. Though this approach is able to localize and map in a dynamic environment, it cannot extract any information about the dynamics of the objects in the scene. Since, all unstable points i.e. noise and/or the points originating from dynamic elements, are removed from the process.

Aguiar et al. proposed a method to attain more information about the dynamic elements in the environment in [31]. The multi-view camera approach utilizes eight cameras in order to track a person. Later, the data is used to reconstruct a high quality spatio-temporally model that is consistent in its texture, shape, and motion. Zollhofer et al. [32] proposed another approach to achieve equivalent results. The implementation uses a single RGB-D camera to reconstruct a non-rigid body in real-time. It adopts a technique called extended non-linear As Rigid As Possible (ARAP), in order to register the RGB-D data to an object template. However, [31] and [32] follow many other implementations in the manner that it requires an initial template/ model of the moving and/or deforming object. The model is then deformed along the process by fitting the 3D points associated with the local regions and registering it rigidly. These implementations deviate from the original essence of the SLAM and focus on merely reconstructing a detailed model of a non-rigid object. Furthermore, the object of interest is merely one and modeled within limited spatial extent. In addition, the implementations tend to fail at tracking and registration when the data points are noisy, sparse or occluded.

Recently, Keller et al. proposed a methodology to overcome the limitations of the aforementioned studies [33]. A Point-Based Fusion method was proposed to localize and map an environment in real-time. A 3D model is reconstructed afterwards while vividly differentiating the dynamic and the static objects in the scene. A commercial RGB-D camera (Kinect /PMD Camboard) was used for the experimentation. The approach takes the outliers into consideration and assign them a confidence value instead of discarding them. The confidence values are raised and lowered with time based on data associations, which later determines if the points are static or dynamic. For Segmentation of dynamic objects,

the individual few dynamic points are used as seed in order to segment the entire moving object from the depth maps. In contrast to previous methods, the implementation can work with a larger spatial extent. It has been tested in indoor environment. Nonetheless, the use of commercial RGB-D camera poses a limitation of indoor use only. These sensors perform poorly in an outdoor environment and very little accurate data is obtained in such a case. For Kinect V2, the maximum range in outdoor environment reduces to 1.9 meters under favorable conditions [34]. Furthermore, only two-thirds of the data obtained is reliable in outdoor conditions. The working range can fall to 0.8 meters in case of sunlight [34].

In this study, we propose a variant of the Keller et al. [33] to tackle dynamic scenes using a stereo camera. The general framework of the approach is maintained; however, a number of necessary alterations have been introduced to handle the different nature of data. The preference to stereo camera over the commercial RGB-D camera is mainly to tackle the problem of outdoor environment exploration.

2.3 Stereo Camera Geometry and Calibration

2.3.1 Camera Model

In our work, we adopt cameras that work on the principle of Pinhole camera model also known as perspective camera model, a widely used and acknowledged approach. Pinhole model provides us with a mathematical relation between the points at 2D image plane and the reprojected 3D points in world coordinates. The transformation is a two-step operation where a mapping exists between 3D world coordinates and 3D camera coordinates (\mathbb{R}^3 to \mathbb{R}^3), followed by projection from 3D camera coordinates to 2D image points (\mathbb{R}^3 to \mathbb{R}^2) [35].

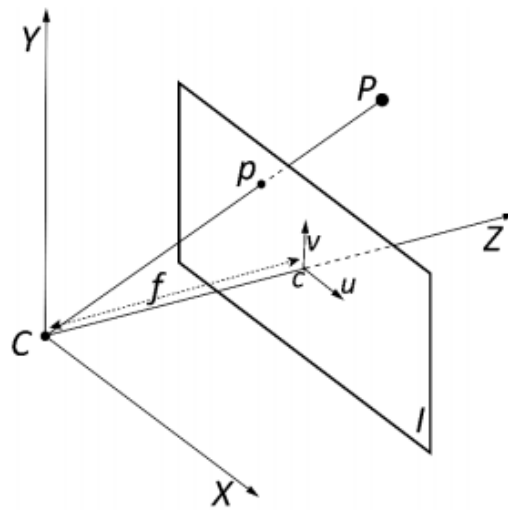


Figure 1. Pinhole camera model. A Point P in 3D space is mapped to a 2D point p on image plane I by the ray connecting P with the center of projection C .

Let us describe the basic parameters of a pinhole camera that will be later utilized to explain the derivatives of these parameters. Figure 1 shows a ray originating from the origin of camera coordinate, here known as the center of projection (COP). A straight line from the center of projection, which is perpendicular to the image plane I is known as the Principal axis, while the point at which principal axis intersects the image plane I is known as the optical center or principal point ($c = c_x, c_y$). The distance between the optical center and the COP is the focal length. In the mentioned figure, a 3D point $P = (X, Y, Z)$ is mapped to the image plane I of coordinates (u, v) . This interrelationship is explained using the concept of similar triangles, and the consequent equation is obtained to be:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{f}{Z} \begin{pmatrix} X \\ Y \end{pmatrix}. \quad (1)$$

The equation above assumes that the origin of the image plane is exactly at the principal point c , however, in practice this exactness is not retained, and by definition the origin of the images I mostly located at the lower or upper left corner. Thus, the equation above is altered to obtain equation 2.

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{f}{Z} \begin{pmatrix} X \\ Y \end{pmatrix} + \frac{c_x}{c_y}. \quad (2)$$

In matrix form we obtain the formation shown in equation 3, which proves more practical during calculations

$$Z \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \underbrace{\begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}}_K \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \quad (3)$$

Here, K is the Intrinsic matrix or camera calibration matrix as it contains the intrinsic camera parameters f and c . The matrix followed by K is a homogenous transformation composed of the Rotational matrix R and translational matrix t . Here R is set to be an identity matrix while the translational matrix is 0. The above equation enables us to transform points from image point coordinates (u, v) to 3D points in camera coordinate system and vice versa. However, we still have to transform the 3D points from the camera coordinate system to world coordinate system. This transformation is achieved as:

$$P_{cam} = R * P_{world} + t, \quad (4)$$

where the 3D rotation matrix $R \in R^{3 \times 3}$ and 3D translation $t \in R^3$ are known as the extrinsic parameters. Finally, by combining the obtained relations we can attain a mapping relationship between points in the image plane and the corresponding 3D points in world coordinate frame.

$$w \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = K [R|t] P_{world} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} [R|t] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \quad (5)$$

Two or more image sensors along with their lenses combine to form a stereo camera. The concept mimics the human's ability of binocular vision in order to perceive three-dimensional information about the scene. The distance between the lenses is known as intra-axial distance or baseline distance. Stereo vision helps to obtain 3D information from multiple views of a scene. At least one pair is necessary to obtain the 3D information by estimating the relative depth of points. The reconstruction of 3D points from disparity maps obtained from the stereo pairs will be discussed in later sections.

2.3.2 Camera Calibration

The estimation of lens and imaging sensor parameters is known as geometric camera calibration, also called camera resectioning [36]. These camera parameters are the intrinsics K , extrinsics $[R | t]$, and distortion coefficients (radial and tangential). Tangential distortion results when the lens and the image plane are not parallel. While, Radial distortion results when light rays bend more near the edges of a lens compared to its optical centre. Small lenses show greater distortion. The theoretical origin of these camera parameters was discussed in section 2.3.1. These parameters are essential for correcting lens distortion, measuring object size in real world scale and finding camera location in the scene. These parameters form the basis of many machine vision applications such as 3D reconstruction, localization, and mapping.

In order to estimate the camera parameters, we need to form a relation between 3D points in world coordinate and its 2D points on the image plane. Typically, these points are obtained by finding feature points in the images of a calibration pattern e.g. checkerboard. For checkerboard calibration pattern the most suited feature points are corner points. The correspondences between 2D and 3D are used to compute the camera parameters. The accuracy of the estimated camera parameters is evaluated by checking the reprojection error.

2.3.3 Epipolar Geometry & Image Rectification

Epipolar geometry is the intrinsic projective geometry of stereo vision. In a stereo vision case, when two cameras view a scene from their distinct position with fixed baseline, some essential geometric relations exist. These relations map a 3D point to a projection on the image plane under some geometric constraints [35]. These relationships were explained in the previous sections; however, the compulsory geometrical constraint will be explained here.

The motivation for defining the geometry and constraints is to alleviate the search for corresponding point matching in the stereo images. Figure 2.a shows a point X lying in a 3D space which when projected back on to the camera image planes finds itself at 2D position x and x' . The points x and x' and their camera centres C and C' are coplanar, and lie on the same plane π . Let's assume, we know position of X in the first camera plane only, which is x . Then from the aforementioned conditions, we can deduce that x' must lie on the intersection of plane π with the second camera image plane. This range of possibilities exists on the line that passes through e' and l' , as shown in figure 2.b. This greatly benefits us in terms of stereo correspondence search where we only need to search for the correspondences on the horizontal Epipolar line.

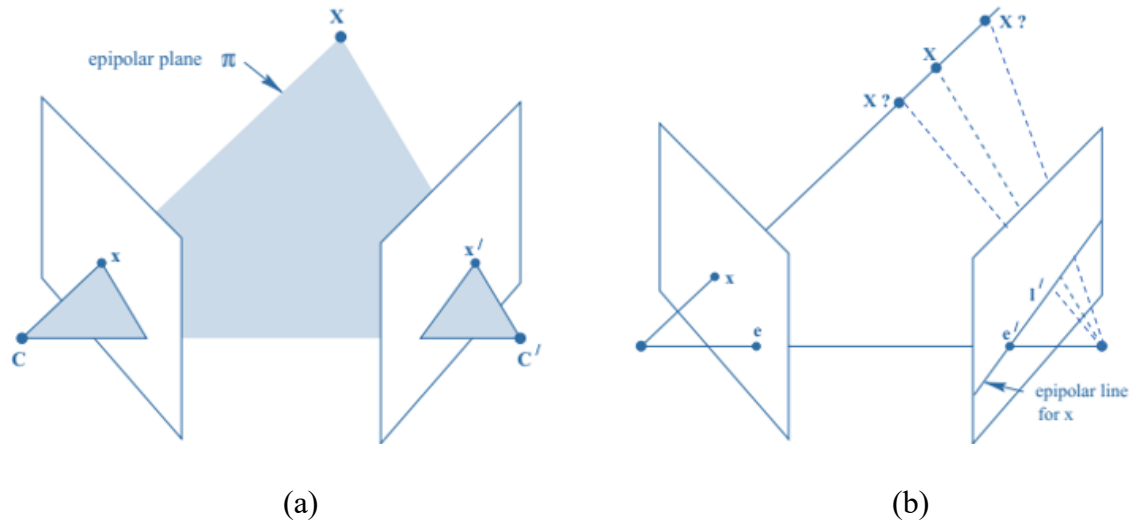


Figure 2. Points following the constraints of Epipolar geometry (reproduced from [37])

However, real images do not directly follow Epipolar geometrical constraint by default. This requires an additional step of rectification, where the images are projected onto a common image plane.

Hence a rectified image must satisfy the following two properties:

- Corresponding points lie on the Epipolar lines and all Epipolar lines are parallel to the horizontal axis.
- The Corresponding points on images have same vertical coordinates.

For rectification of images used in this work, computer vision toolkit's function was utilized. Figure 3 shows the result of image rectification.

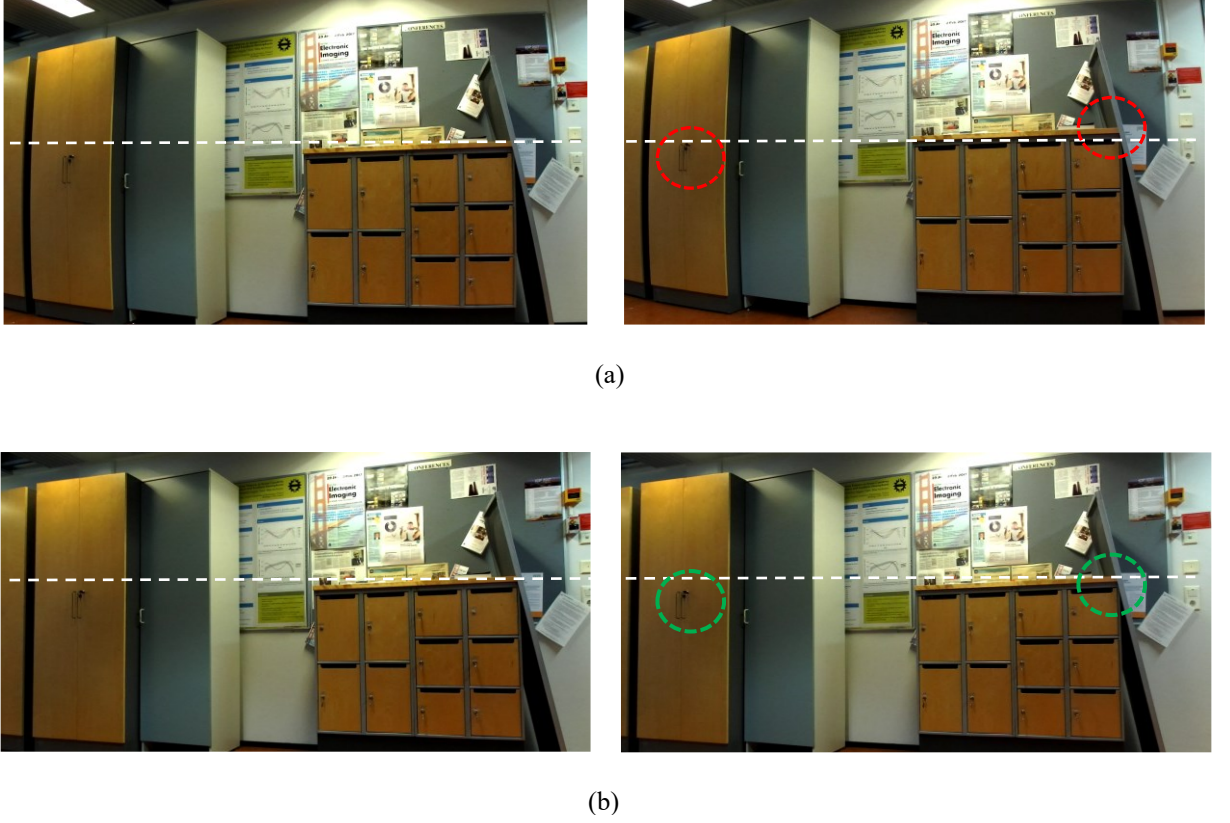


Figure 3. Stereo pair Rectification (a) Original images acquired using the Zed camera; the straight white line passes through same image coordinates (b) Rectified pair; the same visual points in left and right pair lie on the white line which depicts the Epipolar line.

The effects of image rectification can be observed from the above figure. The images were rectified using the camera parameters obtained during the calibration process. Figure 3.a shows the original images of an indoor environment obtained using a stereo camera. It can be observed from the unrectified images the visual features from the left and right frame do not lie on the same vertical coordinate. The objects encircled in red move significantly in the vertical direction between both views. On the other hand, Figure 3.b shows a better result after rectification. The same objects are now encircled in green, and it can be seen that the encircled objects are at the same distance in the left and right view from the white line, which serves as an Epipolar line for the points that it passes through.

2.3.4 Salient Feature detection and triangulation

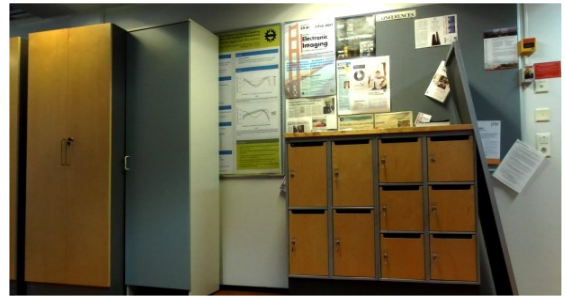
Many types of effective interest point detection approaches have been introduced such as Harris, FAST, SURF, CENSURE, and SIFT. Each of these and other existing methods has its own advantages and pose own set of limitations. For this work, we adopted the use of Harris Corner Detector during our experiments with sparse cloud generation. Harris Corner Detector is still among the widely used interest point detectors [38]. Furthermore, it has been previously utilized in a multitude of Visual SLAM implementations due to its

performance, as in [39]. Haris Corner Detector offers rotation invariance along with high detection repeatability, localization accuracy, and robustness to the varying environment [38]. Therefore, it is practical to assess our approach using this classical detector as a correspondence matching system. The salient features detected for the scene in Figure 4.a and b are shown in Figure 4.c

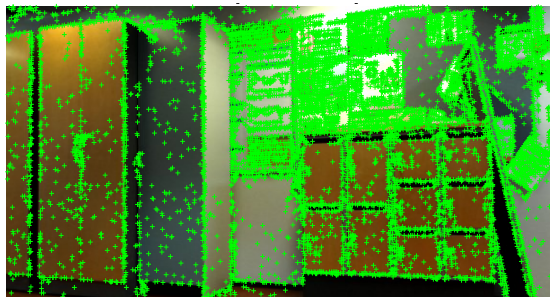
Once the salient points are detected, these are tracked in the corresponding stereo pair. Generally, two main approaches exist for registration of features. Some researchers use block matching method [40, 41] while others focus on the well-developed techniques based on Kanade-Lucas tracker (KLT) [42]. For this work, MATLAB's built-in vision tracker was utilized which is based on KLT. This point based tracker performs effectively in short-term tracking, the result of tracking of the feature points are shown in Figure 4.d. The tracking shows that some features still do not follow the Epipolar constraint, hence, those points are removed by restricting their vertical distance. The filtered salient features that follow the Epipolar constraint are shown in Figure 4.e. These salient feature points are used to obtain the 3D point cloud by a process called triangulation, also known as reconstruction. Triangulation uses the camera extrinsic and intrinsic parameters to locate the point in 3D space using the equations provided in Section 2.3.1.



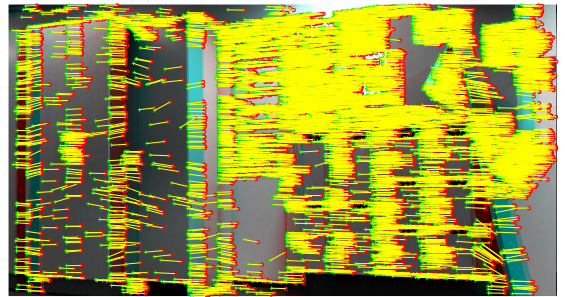
(a)



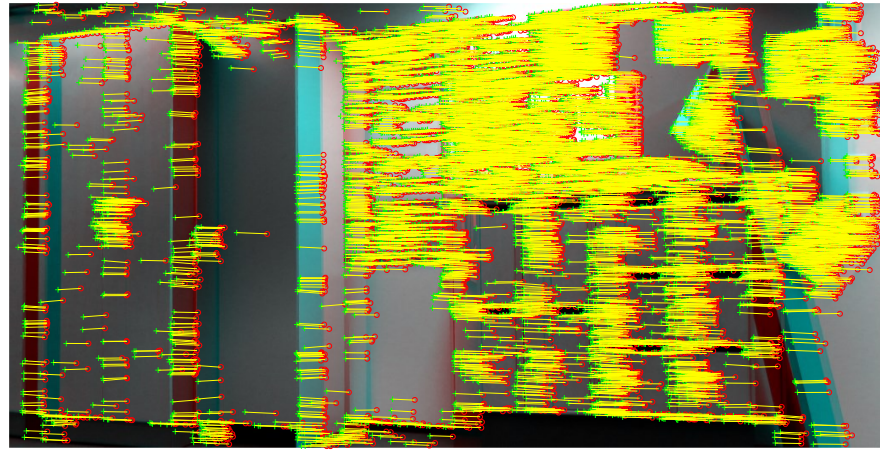
(b)



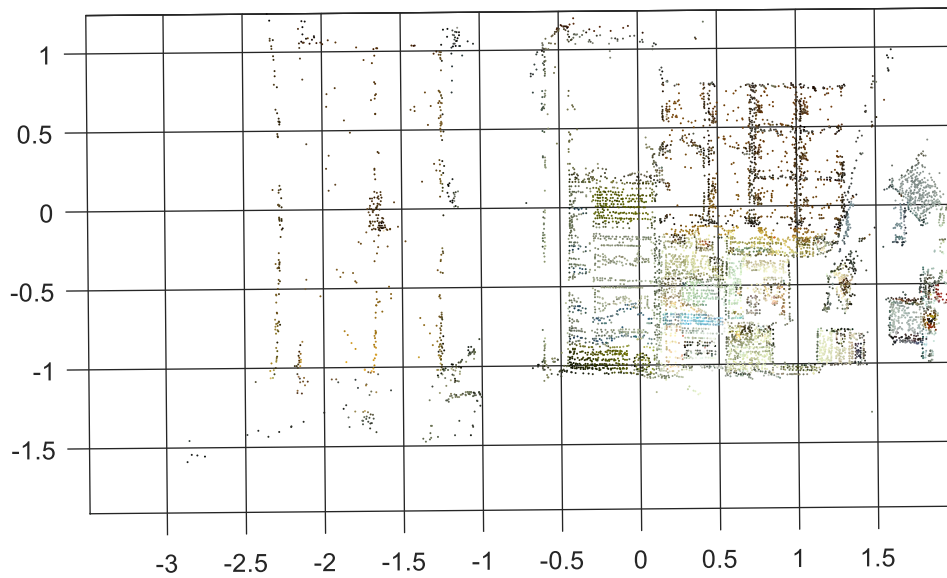
(c)



(d)



(e)



(f)

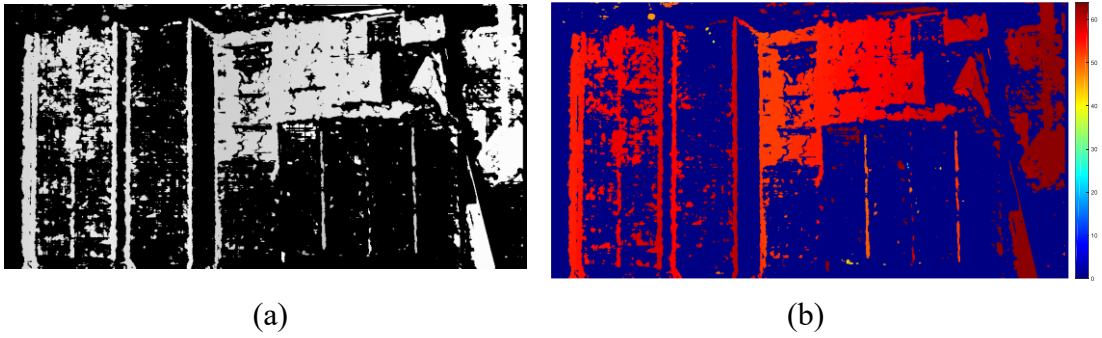
Figure 4. Feature point detection, tracking and triangulation (a, b) Stereo pair acquired using the Zed camera (c) Corner Feature points detected in the left frame of stereo pair (d) Detected corner points are tracked in the stereoAnaglyph (red-cyan stereo pair) (e) Filtered points tracked in bound to Epipolar constraint (f) Sparse 3D point cloud generated from the tracked feature points by triangulation

2.3.5 Dense Disparity Estimation

Feature matching based reconstruction may provide us with more accurate disparity estimates and allow recovery under large displacement, however they cannot be used in many stereo applications requiring dense disparity estimates such as 3-D environment reconstruction. Dense disparity estimation poses its own set of difficulties such as photometric variation and depth discontinuities.

The disparity estimates in this work were computed using MATLAB computer vision toolkit. The function can compute the disparity map using either Block Matching or Semi-Global Matching (SGM) [43] algorithm. The later was chosen due to its added functionality that it enforces similar disparity on neighboring blocks. This constriction results in a more complete disparity map compared to the Block Matching algorithm [44]. Furthermore, it provides a good compromise between computational speed and global optimality. Figure 5.a shows the disparity map computed for the stereo pair shown in Figure 5.a and b. The same disparity map is shown using a color map (jet) for visual inspection in Figure 5.b. The disparity range during computation was set between 0 to 64 and the uniqueness threshold was set to 60. The uniqueness threshold helps to retain pixels with reliable disparities and discards unreliable points.

The disparity map was used to reconstruct a dense 3D point cloud using the camera parameters obtained and explained in earlier sections. The direct result of 3D reconstruction is shown in Figure 5. c. The huge range of the point cloud in all axis is mainly due to the erroneous position estimation. These points can be discarded as they are completely useless and would considerably damage the point cloud registration process. The points are removed limiting the perceived depth of the point cloud to a practical and reliable distance i.e. 6-8meters. For most of the experiments, the distance was limited to 6.5 meters and the result of this truncation is shown in Figure 5. d. The point cloud obtained is dense for areas that have been accurately reconstructed. However, in our work, we do not require such dense point clouds, hence the point clouds are downsampled to a uniform distribution using a grid filter. The filter averages the physical properties of the points in 2 cm cubic range. The downsampling significantly aids to reduce memory and computation time complexity for SLAM. Figure 5.e shows the obtained downsampled point cloud.



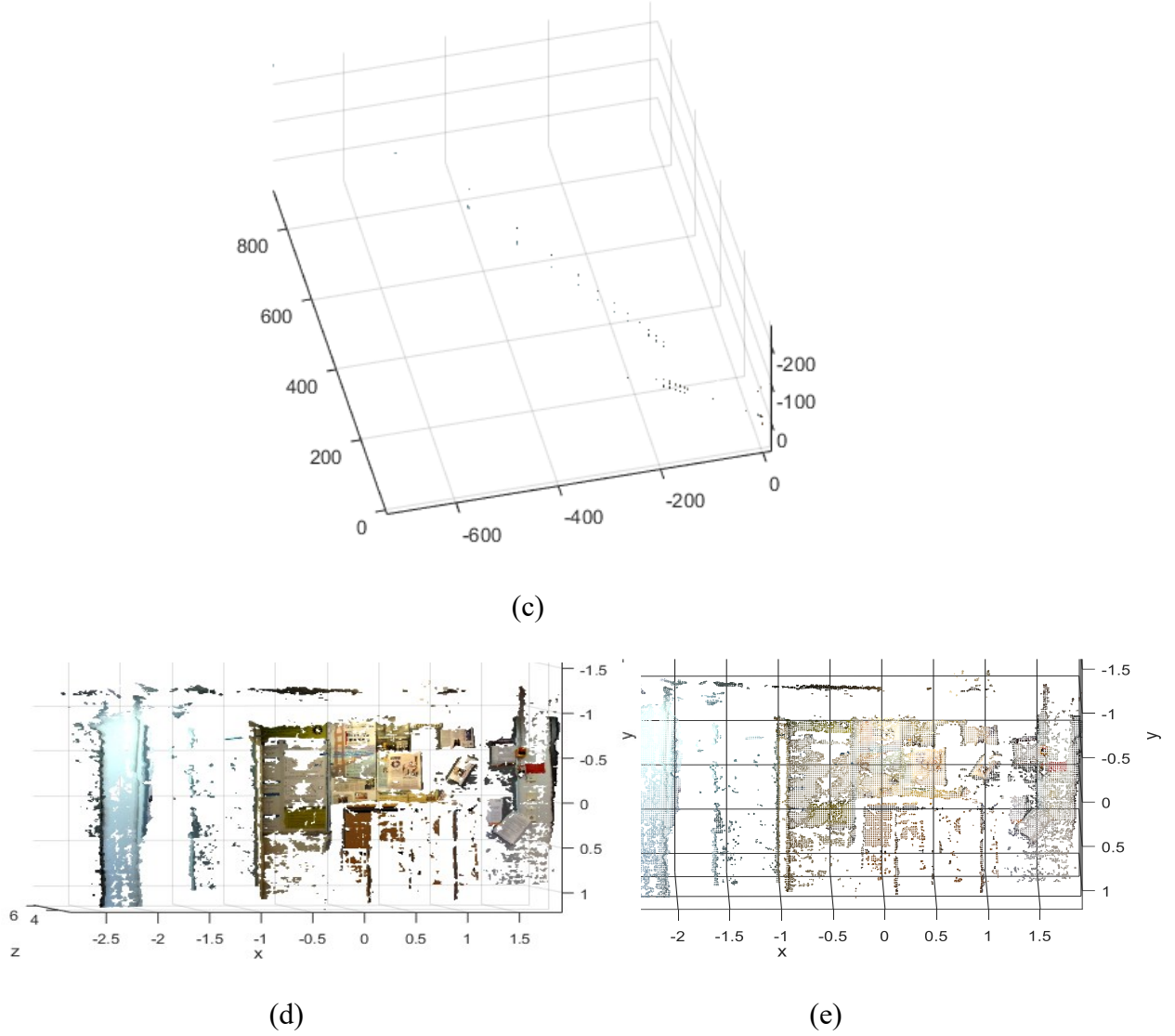


Figure 5. Feature point detection, tracking and triangulation (a, b) Stereo pair acquired using the Zed camera (c) Corner Feature points detected in the left frame of stereo pair (d) Detected corner points are tracked in the stereoAnaglyph (red -cyan stereo pair) (e) Filtered points tracked in bound to Epipolar constraint (f) Sparse 3D point cloud generated from the tracked feature points by triangulation

2.3.6 Comparison of Feature matching and Dense Stereo Estimation

The goal of this work is to both localize the camera/observer and map the environment with enough data that useful information can be extracted for further processing. In the scope of this work, the goal is to extract enough 3D points of the dynamic objects in the scene that they can be effectively used to make decisions.

Initially, salient features were detected and tracked to obtain 3D map of the environment. After successive registration step, the map is dense enough to provide useful information about the environment. This was valid only for cases when the camera did not move

quickly and scenes were explored for relatively long time. Furthermore, the sparsity had the innate drawback, that the features obtained were not uniformly distributed as seen in Figure 6.a. Some area might have a large number of points packed densely, while other areas might have few points distributed irregularly at longer distances. This effect was even more visible for moving object in the scene, which made it considerably difficult to process the 3D points while maintaining enough data points to reflect the entirety of the object. On the other hand, the results from dense stereo estimation gave us more reliable results in terms of uniformity. High threshold was selected in order to keep accurately generated 3D points. Even after significant downsampling of 3D cloud, a considerable number of points are retained with uniform distribution. During registration of successive point clouds, the map becomes dense and effectively represent the real-world environment which can be observed in Figure 6.b. Hence, the use of dense stereo estimation was adopted for viable results in a dynamic environment.

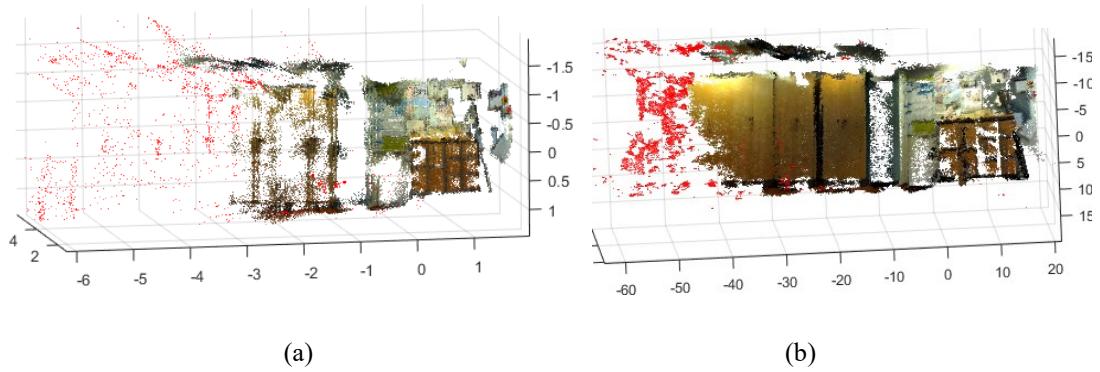


Figure 6. Result of Point Cloud registrations (a) Sparse clouds from corner feature detection at 0.001 threshold (b) Dense clouds at uniqueness threshold of 60 (a considerably high value)

3. PROPOSED METHOD

3.1 Framework

In this subsection, we will briefly present the proposed framework using the flowchart shown in Figure 7. A comprehensive explanation of these steps is provided in other sections.

In the proposed approach, a disparity map is estimated based on the stereo pair using the computed stereo parameters. 3D point cloud is reconstructed from the corresponding disparity map. The point cloud obtained is uniformly downsampled in order to ease the computational load. These steps have been explained in the previous section along the camera geometry.

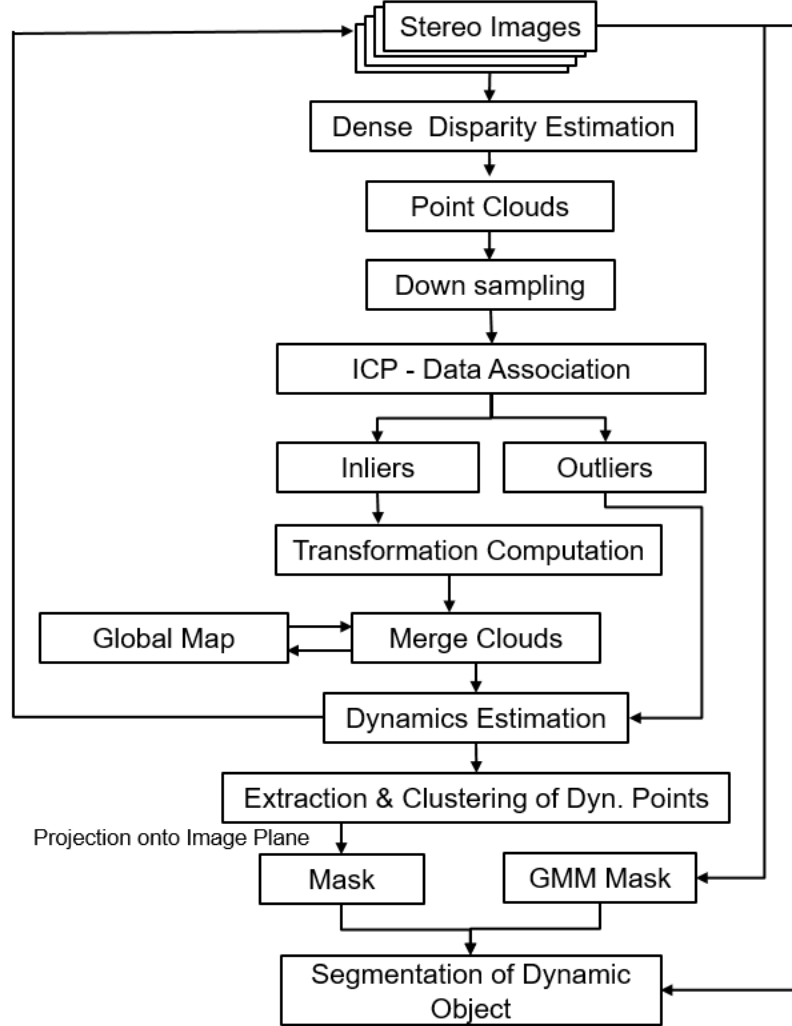


Figure 7 Flowchart of the proposed approach

The point cloud, PtC_k , obtained from stereo pair k is registered to a global map by finding the transformation between PtC_k and PtC_{k-1} (at time $k-1$). The transformation is computed based on correspondences (inliers) between the point clouds using Iterative Closest Point (ICP) algorithm. The non-correspondences (outliers) do not contribute to transformation computation, however, they are utilized to differentiate the dynamic part of the scene. The dynamic points are extracted and clustered from the global map and corresponding image points are segmented using a two mask binary masks obtained from Gaussian Mixture Model (GMM) and the projection of the 3D points onto the image). All these steps would be discussed in detail in the following subsections.

3.2 Global Cloud Model

The process of Visual SLAM allows us to generate a map of the environment under observation. This map can be stored in different form depending on the choice of the developer and ease of use. In this work, we maintain a global cloud after the registration and stitching of the individual point clouds to the global coordinate frame. The global cloud variable is a self-contained model that includes all its relevant processed information. This variable is readily passed into other functions and the nature of the variable is maintained in the output.

The global cloud has 5 distinct parameters namely Location, Color, Normal, Confidence, and NoFrames (indicating Number of frames). As evident from their names, the first three parameters define the physical properties of each 3D point in the Global Cloud variable. The Location is a $N \times 3$ matrix which identifies the xyz position of the N 3D points in space while Color stored the RGB color of the 3D point. Normal is a similar $N \times 3$ matrix which specifies the normal vector of the 3D point. Confidence is a unique parameter introduced regarding a 3D point which will serve as the base criterion for classifying a point as being static or dynamic. If a point gains a total confidence of greater than 1, it is classified as a reliable static point otherwise the points is considered as unreliable which could be part of either moving object or simply noise. The final property defines for how many frames the point has been in the scene and has been viewed by the camera. It helps to keep track of points and remove unstable points after they have been in the scene for some time.

3.3 Point Cloud Alignment and Data Association

Once the 3D point cloud for a scene is obtained, it has to be registered to the Global Cloud. This step transforms the second point cloud to the reference coordinate system defined by the first point cloud which in this case is the Global Cloud. One of the widely-used techniques for point cloud registration is the Iterative Closest Point (ICP) algorithm. ICP was originally introduced in [45] to find the transformation between two point clouds

where one would serve as the reference surface. Finding this transformation includes calculation of the rotational matrix \mathbf{R} and the translational vector \mathbf{t} between the second point cloud and the reference cloud. The transformation is computed by minimizing the squared error between corresponding points in the two clouds. The generalized formulation for ICP can be explained with Figure 8 and the following equations.

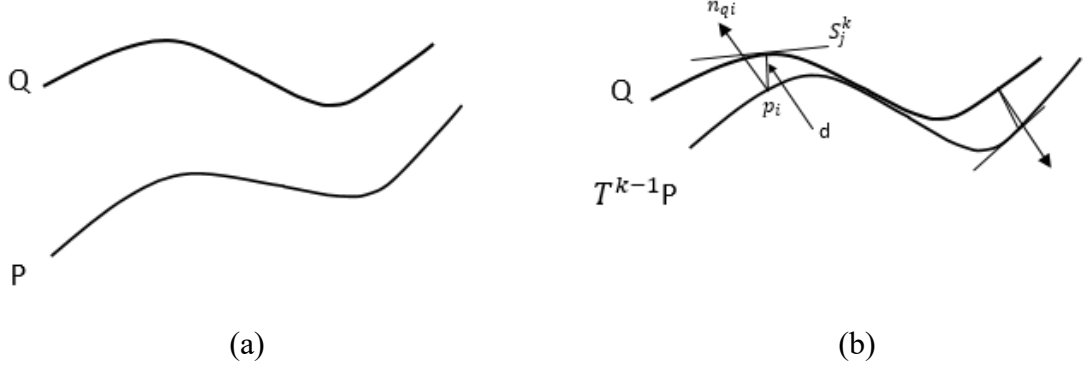


Figure 8 Distance measures in 2D case between two curves P and Q .

Two curves P and Q have to be aligned by finding the transformation between them. Let us select N pairs of corresponding points, $p_i \in P$ and $q_i \in Q$, ($i = 1 \dots N$), known as control points from two views. The transformation T can be found by minimizing the error function:

$$e = \sum_{i=1}^N \|Tp_i - q_i\|^2 \quad (6)$$

The information about correct correspondence can be difficult to obtain for large number of points. Therefore, a more practical approach is to minimize the distance between the points on one surface against the other. Equation 6 takes the following form

$$e = \sum_{i=1}^N \|Tp_i - q'_j\|^2, \text{ where } q'_j = q | \min_{q \in S_j} \|Tp_i - q\|. \quad (7)$$

Here, S_j depicts the tangent plane of Q at point q_j . However, we do not know where the corresponding point q_j is. The corresponding points could be moved closer using a transformation T_0 . In an iterative case the previous transformation T_{k-1} are used for accumulative approximation. A generalization of the above equation is:

$$e^k = \sum_{i=1}^N d_s^2(T^k p_i, S_j^k) \quad (8)$$

Where d_s is the signed distance from a point to the plane and $T^k = T \cdot T^{k-1}$. This minimization constrains the direction in which the distance is reduced between the point and the plane.

The algorithm follows the following steps, iteratively.

- 1-Select a set of N control points $p_i \in P$ and compute their surface normal n_{pi} .
- 2-At each iteration k , the following steps are repeated until convergences.

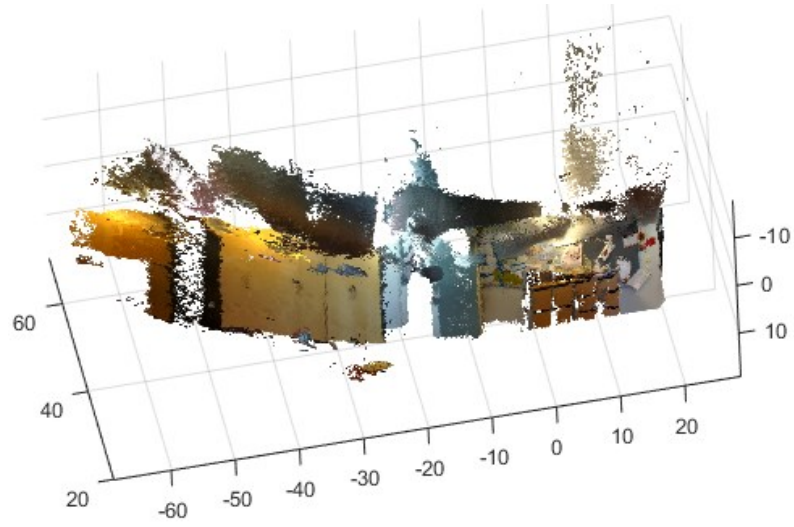
For each control point p_i ,

- a- Apply the transformation T^{k-1} to both the point and its normal to obtain p'_i and n'_{pi} .
- b- Find the intersection between q_i^k of the surface Q and the normal line defined by n'_{pi} for p'_i .
- c- The tangent plane S_i^k is computed for the surface Q at q'_i .

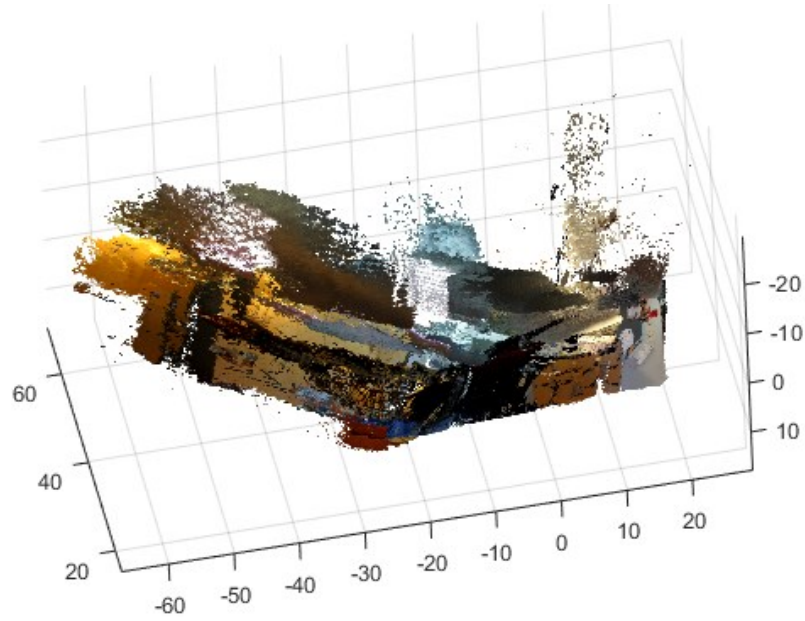
Find the transformation T that would minimize e^k in Equation 8.

Multiple types of error minimization approaches have been proposed namely Point-to-Point, Point-to-Plane, Point-to-Curve, Curve-to-Curve, Line-to-Line, and Line-to-Curve error metrics [46]. In this work, we use Point-to-Plane error minimization, since it provides a sturdier result even in the presence of a degree of noise.

Generally, all the points in the cloud contribute to the calculation of transformation with identical weightage. However, there might be points that will not find its optimal correspondence and might get associated to wrong points. This computed transformation would still give a minimum least-squared error but it would be erroneous in a practical sense. Hence a criterion was adopted that selects the point which should contribute to the calculation of the homogeneous transformation matrix, these points are called *inliers*. On the other hand, the points that are left out are called *outliers*. Outliers are mostly considered as noise and are discarded from the process. MATLAB's ICP implementation requests the user to provide an estimate of the ratio of inliers that are to be expected for computing the transformation. However, the approach is not appropriate for real world data sets where arbitrary noise and change in environments are possible. For example, a walking person is introduced into the scene, the points pertaining to the dynamic object (person) are not static and keep changing their position, hence they cannot contribute to the calculation of transformation matrix, which implies that the number of outliers increased. As a result, the implementation fails with incorrect registration alongside immense noise in the map. This effect can be observed from Figure 9 which shows the mapping of the previous corridor image sequence using MATLAB's point cloud registration and stitching example. Even in a static environment, the mapping is not considerably accurate, the registration is tilted and points and surfaces are generated at a depth greater than the distance of the wall from the observer. This mediocre performance is due to the fact that it cannot accommodate for the varying number of inlier and does not actively remove noise from the scene global cloud. Furthermore, the performance becomes worse as soon as the person walks into the camera view.



(a)



(b)

Figure 9. Point Cloud registration failure of the dataset with moving person at parameters: Inlier ratio=0.8, point-to-plane error metric and point cloud merging at grid factor of 0.0135 m³ (a) Static environment i.e. before person walks in to the scene (a)Dynamic environment

The approach proposed in this study uses a customized version of the MATLAB's ICP implementation. Instead of fixed number of inlier selection, a distance based threshold is set for inlier selection. This distance threshold comes into effect during data association step. This can be explained with the example shown in Figure 10. The points on the lower lines must be transformed and aligned to the line above it. The dark grey points have found their corresponding association and are therefore considered inliers. While the light

grey points have no correspondences, and are considered outliers to the system. Furthermore, the dark grey points joined by the light line are also considered outliers since the Y do not fulfill the distance threshold criterion and cannot be considered reliable correspondences for a rigid transformation.

The inliers are then used to compute the transformation matrix by minimizing least squared error, iteratively. Unlike other approaches, the proposed method in this study discourages discarding the outliers directly after registration. These outliers contain implicit information about dynamic objects in the scene. The extraction of relevant points from these outliers and removal of noise would be discussed in the proceeding sections.

Like any other gradient descent method, ICP expects a good starting point from the user since it also serves as the reference data. Furthermore, the images acquired should provide overlapping views to some degree so that the points from new cloud could associate to the points in Global Cloud. The proposed approach is able to work effectively with slightly overlapping views due to its flexible selection of the number of inliers. However, consistency in data is desirable under such circumstances so that the points could attain enough confidence to be termed reliable.

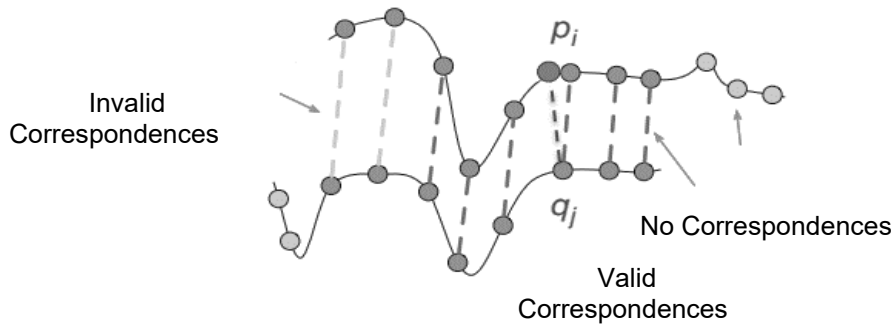


Figure 10. Point Correspondence Selection

Once an appropriate transformation is obtained for the new point cloud against the reference cloud. The new cloud is transformed to the reference coordinate frame using the homogenous matrix $[R | t]$. Hence, this operation continues as the map builds up over the previous point clouds and stored in the Global Cloud. Additionally, camera localization is achieved by transforming the initial position of the camera, $[I | 0]$ where I is the identity matrix, along the registration process to the reference coordinate system. The accumulated transformation provides us with the localization of the observer, where the observer, in this case, is the hand-held camera.

3.4 Merging and Confidence Gain

The addition of new 3D points to the processing stream adds load to the memory and processing capacity. In order to restrain the growth of the amount of points, some criterions must be set to limit the addition of 3D points. Here we use 3D point merging and

removal as the means of restraining the exponential growth of points. In this section, we will only discuss the merging of 3D points and its resultant confidence gain. Previously during association of points, we have successfully formed a relationship between points of a new point cloud to a reference cloud (Global Cloud). Now a point may either associate itself to one point or many points as shown in Figure 10, where point q_{ij} has been associated to two points. Both these points serve as inlier for transformation computation, however, only the closest point is merged physically, which in this case is p_{ij} . This merging is performed so that the duplicate of the same 3D point may be removed. During merging step the Position, RGB colour and the Normal vector of the closest points (p_{ij} and q_{ij}) are averaged for the resultant merged point. Furthermore, the previous confidence of both these points are added with a bonus of 0.1 and the NoFrames value is increased by 1, since this point has been observed in one more scene. However, the other associations of q_{ij} are not averaged physically. It can be assumed that the other associated points q_{ij} are points from the same object obtained at different position due to different surface sampling. Therefore, we do not merge the physical properties of such points and these points are added to the Global Cloud with their original physical properties (position, colour and normal). Nonetheless, the confidence is increased by merging its confidence with the confidence of points q_{ij} along with an added confidence obtained through a Gaussian distribution as shown in Figure 11. The peak value is set to be 0.1 and the standard deviation is set to be half of the distance threshold used during ICP correspondence selection. Hence, the further the point moves away from q_{ij} , the lower confidence gain it gets during merging step. Lastly, the NoFrames value is increased by 1 as for the other point.

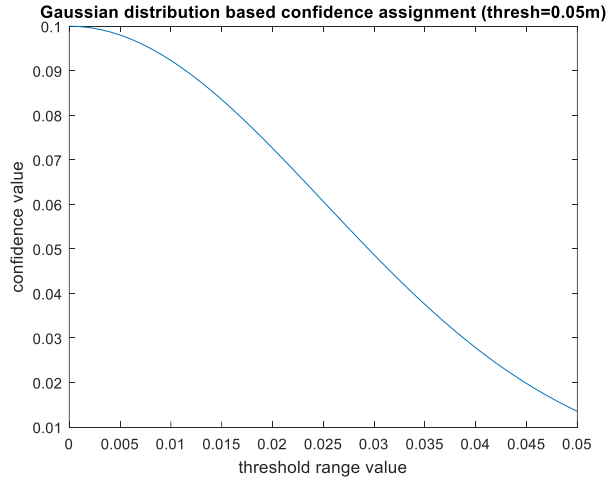


Figure 11. Confidence gain during merging of points

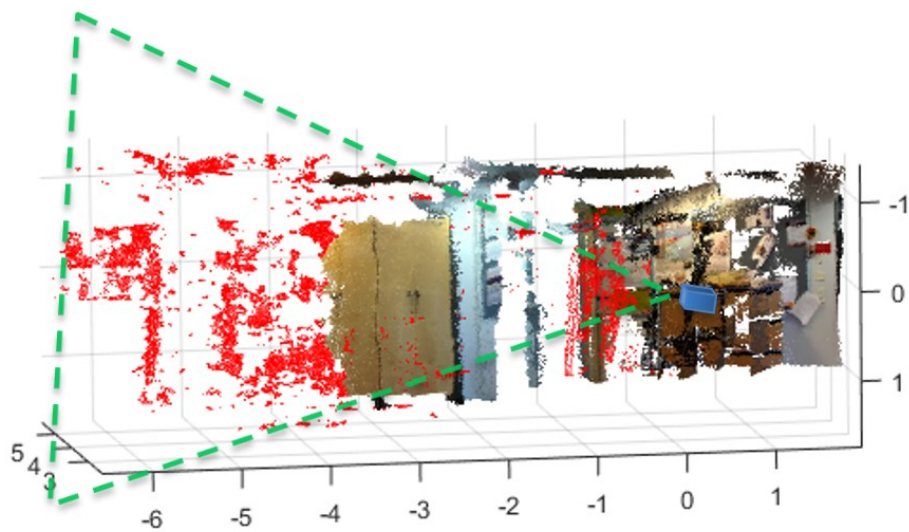
The outliers, on the other hand, are treated slightly differently than the inliers. Outliers are not always noise and can be generated due to the object being out of the scene between the consecutive image sequence, or the process failing to reconstruct the 3D points accurately at the desired position. Hence, they are not removed from the process at this stage and are added directly into the Global Cloud with a confidence of 0 and NoFrame value

of 1. This is done for the purpose that these outliers could possibly serve as inliers in the following sequence of point cloud registration.

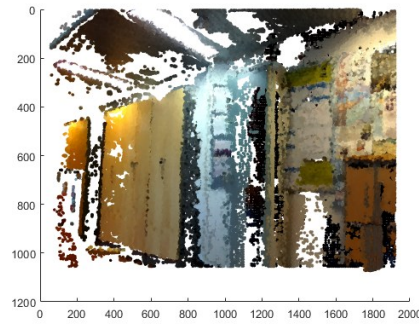
3.5 Confidence Reduction

In the previous step, confidence gain helped us to stabilize the points and convert the points from dynamic/ unstable status to static/ stable. In order to handle a dynamic environment, a mechanism must be designed to convert the static point back to dynamic. For example, a parked car was previously mapped along the environment where all the points from the scene were static. Now, if the vehicle in the scene starts to move, then the 3D points associated to the vehicle must change in confidence and should be classified as dynamic points.

This effect is obtained by continuously reducing the confidence of points from the Global Cloud that are in the view of the camera by a value of 0.01 at each iteration, since it has already been established that the points in view would associate themselves with the newly observed environment. Hence, the points that find association would increase in confidence during the merging step, while at the same time decrease by a fixed value of 0.01. Since the gain would be greater than the loss in confidence, therefore such points would remain static. On the other hand, those points that do not find association, would only loss confidence and, as a result they would change from static status to dynamic/unstable status. Figure 12 illustrates a situation where the far end of the Global Cloud/ environment is being viewed by the camera. The points in perspective view are obtained by projecting the global cloud onto the image plane with the help of accumulated transformations. Only the points in front of the camera that fall onto the image plane after projection are considered for the reduction of confidence.



(a)



(b)

Figure 12. Projection of points to image plane for confidence reduction (a) illustration of the camera viewing the global cloud (b) projected points onto the image plane from the perspective view

3.6 Removal of Unstable points

Until now we focused on building the map with the addition of new measurements. Now we need to improve the registration process by removing the unstable points. This is a crucial step which is essential for the process to work in a dynamic environment. Some constraints are defined in this step in order to facilitate the removal of unstable points. The maximum confidence that a point can accumulate is 1.25. This value has been chosen empirically while keeping the gain (0.1) and reduction (0.01) into consideration. All the points that have been added to the map i.e. the Global Cloud are evaluated at each iteration. Points with confidence less than 1 are extracted as a set of unstable points, which may either contain noise or points from the dynamic object. From these extracted points, only those points are discarded from the map that have been present for more than some threshold time t_{max} . In most of the experiments, the t_{max} was set equivalent to 5 frames. Any unstable point that has not gained enough confidence during 5 iterations is discarded. Hence, this step discards noisy data points and retain only those points in the map that we are confident of. As a result, the registration process is less prone to error. On the other hand, points pertaining to a moving object in the scene are continuously updated, since new points from recent frames are added while the previous points are removed. 5 frames is sufficiently quick for the scene to update the points when the camera is acquiring images at 30 fps.

3.7 Extraction and Clustering of Dynamic Points

The Global Cloud can accommodate 3D points from the static environment and unstable points from the moving objects and/or static objects that occur with less consistency during the registration process. It is essential to differentiate and isolate the points pertaining to dynamic objects from rest of the point cloud. This result is achieved through a sequence of steps that will be discussed here in.

The initial extraction of dynamic points and their clustering would be explained using the same scene that has been utilized in earlier topics of this thesis. Figure 13.a presents an environment with static objects placed in a corridor that is being mapped. A person passes by in the corridor and the points obtained from the person are shown in red in the middle of the static part of the map.

Both the person and the points from the far end of the corridor have low confidence. The points from the far end of the corridor are acquired with inconsistency and are not able to gain confidence quickly. Hence we cannot be confident if these points are static, dynamic or noise. In the first step, we extract out the points with confidence less than 1 and obtain the point cloud shown in Figure 13.b. Since we are sure about the accuracy of the static part of the map, in our analysis we will only take the dynamic points within the bounds of the static part of map into consideration. For the points at the far end of the corridor, we have to wait for the area to be explored further by the observer. Therefore, in the next step, the points that are in the physical bounds of the static portion are maintained, while the other points are discarded as they are not fit for analysis. The obtained points are shown in Figure 13.c.

Now that we have sufficiently accurate data to deal with, we can form clusters based on the proximity of the points. In this study, the points were clustered using MATLAB's built-in function *clusterdata* [47]. The algorithm forms links between data points in space based on the squared Euclidean distance between them and forms a new cluster at the *cutoff* of 1.5. These parameters were set through experimentation and provide good result, irrespective of the dynamics of the scene for consistent point density. Clusters with a low number of points (200 or less) are removed and therefore a refined result is obtained as shown in Figure 13.d and e. The clusters obtained should, however, be further observed. Although we have acquired the points pertaining to the moving person in form of a single cluster, we can also see three other large clusters present at the left end. These three clusters are the remains of the unstable points from the far end of the corridor since the boundary joining the stable and unstable points would never be a crisp difference. The borderline is irregular within the bounds of the static portion of the map and would be a by-product of the extraction process. Hence further process is required to differentiate between the moving object which is truly dynamic in nature and the false dynamic clusters obtained due to inconsistency.

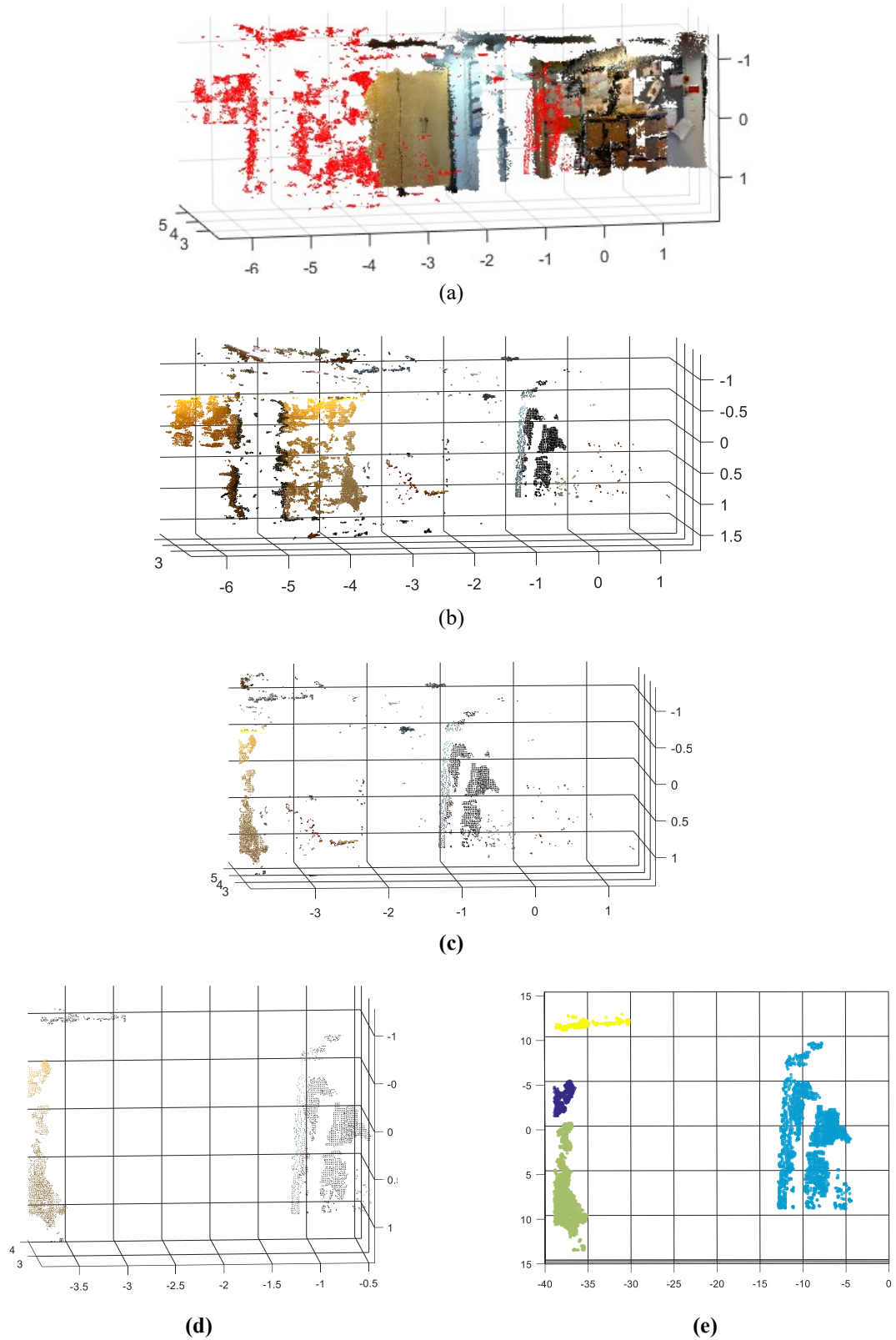


Figure 13. Extraction and clustering of Dynamic points (a) Map of the environment containing both static and dynamic/ unstable points (b) Extracted dynamic and/ or unstable points (c) dynamic points extracted from within the bounds of static map (d) clustering of points and removal of small clusters (e) visualized clusters pertaining to possible dynamic objects

3.8 Segmentation of Dynamic objects (2D images)

In the previous section, we had successfully extracted and clustered the instantaneous dynamic points from the Global Cloud. However, we still must verify the true positive detection (actual moving points) and isolate it from the false positive (noise, inconsistent points). For this purpose, the aid of 2D processing was acquired due to its twofold valuable outcomes at this point. Firstly, the 2D image processing provides an enormous variety of standard implementations to study the motion of objects. These implementations can prove useful and are easy to tune. Secondly, we can easily acquire the actual moving object from the original 2D images as consequence of this verification step. A correlation between the 2D segmented object and the associated 3D points can be built by keeping track of the processed data. As part of the proposed approach, we suggest the use of masks obtained through two different methods for the segmentation of the moving object. The process of acquiring the masks and segmenting the object is explained in the following subsections.

3.9 Mask Obtained from clustered points

The presence of segmented 3D clusters makes it easy to obtain an initial mask. This can be done by projecting the 3D points onto the image plane using the correct extrinsics $[R | t]$. Each cluster is projected separately and a 2-D convex hull is acquired for the X and Y coordinates of the projected points. This results in a solid filled mask of the points instead of just the borderline. Similarly, all the other clusters are projected and their masks are augmented to obtain a mask for all the clusters. The resultant mask is shown in Figure 14. The problem of segmentation has progressed a single step, however, the issue of verification remains.



Figure 14. Mask obtained by projecting the clustered dynamic points

3.10 Motion Mask obtained using GMM

For the purpose of verifying our moving objects, we use Gaussian Mixture Model (GMM) to detect motion in the scene. GMM is a background modeling approach that is able to model the background scene and correctly detect any moving objects in form of foreground. This method was originally introduced in 1999 by Stauffer and Grimson [48], which has later been improved and adopted for a variety of applications such as video surveillance in recent years.

The generalized concept of GMM can be explained using Figure 15. Two gaussian distributions represent some data. This data cannot be accurately represented by a single component/ distribution. Therefore, a mixture of these distributions is adopted. The mixture model is defined by a weighted sum of gaussians.

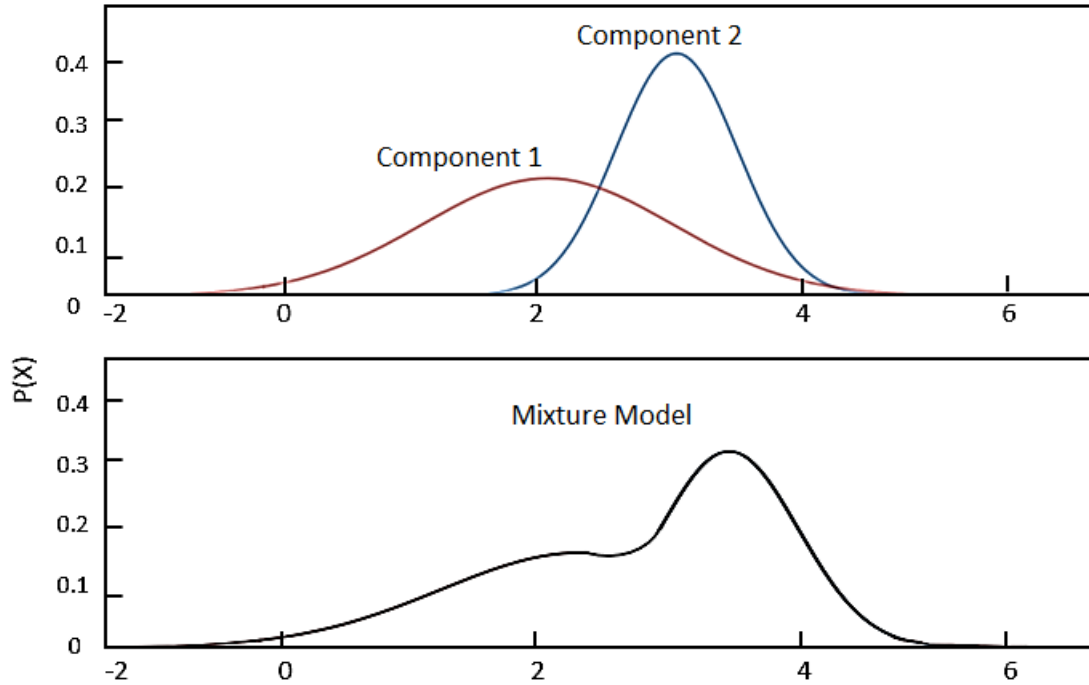


Figure 15 Mixture model for distribution components 1 and 2.

The algorithm treats each pixel independently and characterizes it by a mixture of K Gaussians mixture. The pixel values in an image can be considered over time series and are termed as ‘pixel process’. The gray scale values are represented by a scaler while a colour pixel takes a vector form. At time t , we are aware of the history of a particular pixel (x_o, y_o) , given by

$$[X_1, ..., X_t] = \{ I(x_o, y_o, i) : 1 \leq i \leq t \}, \quad (9)$$

here, I is the image. The probability that a particular pixel might have an intensity at time t is given as:

$$P(X_t) = \sum_{i=1}^K w_{i,t} * \eta(X_t, u_{i,t}, \Sigma_{i,t}). \quad (10)$$

Here, the number of distributions are denoted by K . $w_{i,t}$ is the estimated weight parameter of the i^{th} distribution i.e. what fraction of the data this distribution accounts for. $u_{i,t}$ is the mean value and $\Sigma_{i,t}$ is the covariance matrix of the i^{th} Gaussian distribution in the mixture at time t . The Gaussian probability density function η is given as:

$$\eta(X_t, u_{i,t}, \Sigma_{i,t}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2} (X_t - u_t)^T \Sigma^{-1} (X_t - u_t)}, \quad (11)$$

here, $\Sigma_{k,t} = \sigma_k^2 I$ is the covariance matrix of the i^{th} component. In order to estimate the background model, the first B distributions are considered based on the order of their fitness value $w_{k,t}/\sigma_k$. The update of the weights $w_{k,t}$ and the selection of B distributions are given as:

$$w_{k,t} = (1 - \alpha)w_{i,t-1} + \alpha(M_{k,t}), \quad (12)$$

$$B = \operatorname{argmin}_b (\sum_{k=1}^b w_k > T). \quad (13)$$

where α is the learning rate and $M_{k,t}$ is a binary value which is 1 for the model that matched and 0 otherwise. The threshold T is the minimum portion of the background that should be considered for the model.

The Gaussian components are sorted based on their weights and variances. The formulation proposes that a pixel relating to the background would represent high weight and a weak variance, since the background is static and therefore its value is essentially constant. For detecting the foreground i.e., the moving object, each pixel is classified based on the response of the Gaussian distributions. A pixel that cannot be modeled by the defined /trained background Gaussians is termed as a foreground pixel. Since this approach considers the process pixelwise, it is prone to isolated noise in the images.

The MATLAB implementation of GMM follow the proposals of [49, 50] and provides an easy to tune implementation. However, it should be kept in mind that the GMM approach basically works if the camera is static, and so does the MATLAB's implementation of GMM. Furthermore, MATLAB's documentation recommends training the GMM model with a set of at least 150 images and then further testing should be conducted.

In order to apply GMM on images captured with a moving camera, we propose some pre-processing steps in this study. The result from the proposed approach provides a good compromise between accuracy, robustness, and segmentation of the foreground. Since the camera is moving, the model should be continuously trained on the images from the

newly explored environment. Moreover, the training is conducted with few recent images to keep the novelty of scene. In addition, the training images must be geometrically transformed to the current scene, so that the assumption of a static camera is maintained for the GMM.

If the approach is followed without geometrically transforming the previous images to the current scene for training, the results are not good. GMM erroneously detects the static objects to be moving and segments out their boundaries as foreground. The reason for obtaining just the boundaries is that due to few training images, only the borderlines depicts significant motion in the images. The result of a similar process without geometric alignment of the images is shown in Figure 16.b for the scene in Figure 16.a at time t . The training of the model was performed with 3 latest images. However, when the images are geometrically warped from time $t-2$ and $t-1$ to t and then fed to the training of the model, a significant improvement can be seen. The result obtained is shown in Figure 16.c, where none of the stationary objects are detected.

The image transformation was achieved by finding salient features in the neighboring frames of a video sequence and tracking them in the frame at time t . From here on we can compute the affine image transformation between the previous and current frame using the MATLAB's *estimateGeometricTransform* function based on the point correspondences. Once the transformation is computed, the previous frame is warped to the current frame so that the corresponding points overlap each other. The warping of image to high accuracy is only possible if the images translation is very low.

Another addition to the approach is not to limit the number of training samples from the moving camera to a constant. It is rather useful to accept and reject training images based on the motion that is within some threshold. During experimentation, we found that if the resultant translational motion between the previous images and the image at time t is less than 30 pixels, then it should be accepted as it can be accurately transformed to the current image. Hence, if the camera is acquiring images at 30 fps, then a considerable number of training images can be attained even with a moving camera.

The above-explained approach, when applied to the video sequence for detecting the moving person, provides a peripheral detection of the person. This is due to the fact that the training images already include the moving person, hence GMM is not able to make a clear distinction between the foreground and the background. Therefore, as the person moves, the front of the person occludes the background in the direction of motion while the background in the other direction slowly becomes visible. Nonetheless, the result is accurate with no false detections. The mask obtained for the moving object is shown in Figure 16.d.

The approach is computationally expensive; however, it provides reliable results for detecting dynamic objects position from a moving camera.

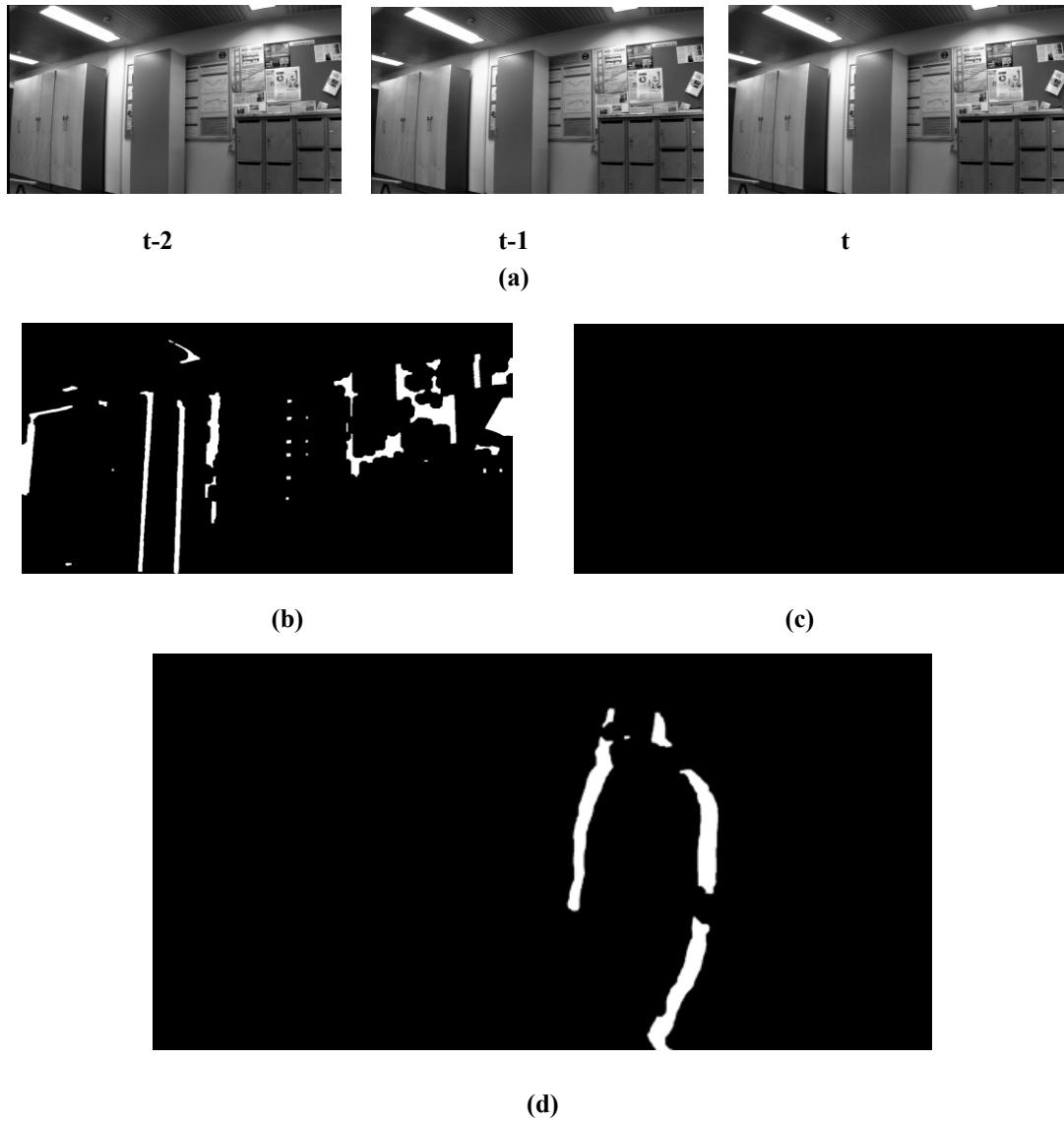


Figure 16. Obtaining motion mask from GMM with moving scene (a) previous frames transformed to the current frame 't' (b) mask obtained when training images were not geometrically transformed (c) mask obtained when training images were geometrically transformed (d) mask obtained for the moving person using the proposed approach

3.11 Segmentation using combined masks

Now that we have the masks obtained from two different methods, a conclusive decision can be made about the actual dynamic object. We propose to use the blobs in the resultant mask of GMM as markers for segmenting out the actual dynamic objects from the first mask (obtained from projection). Since, under such dynamic circumstances where both the camera and the objects in the scene move, only a peripheral area of the moving object can be accurately detected using GMM without having many incorrect detections. The segmentation obtained using the proposed sequence of steps is shown in Figure 17. The segmented object is not finely extracted in this process; nonetheless, it meets the objective

of this work where the dynamic environment was successfully registered using SLAM and the moving object in the scene was identified and isolated for further analysis.

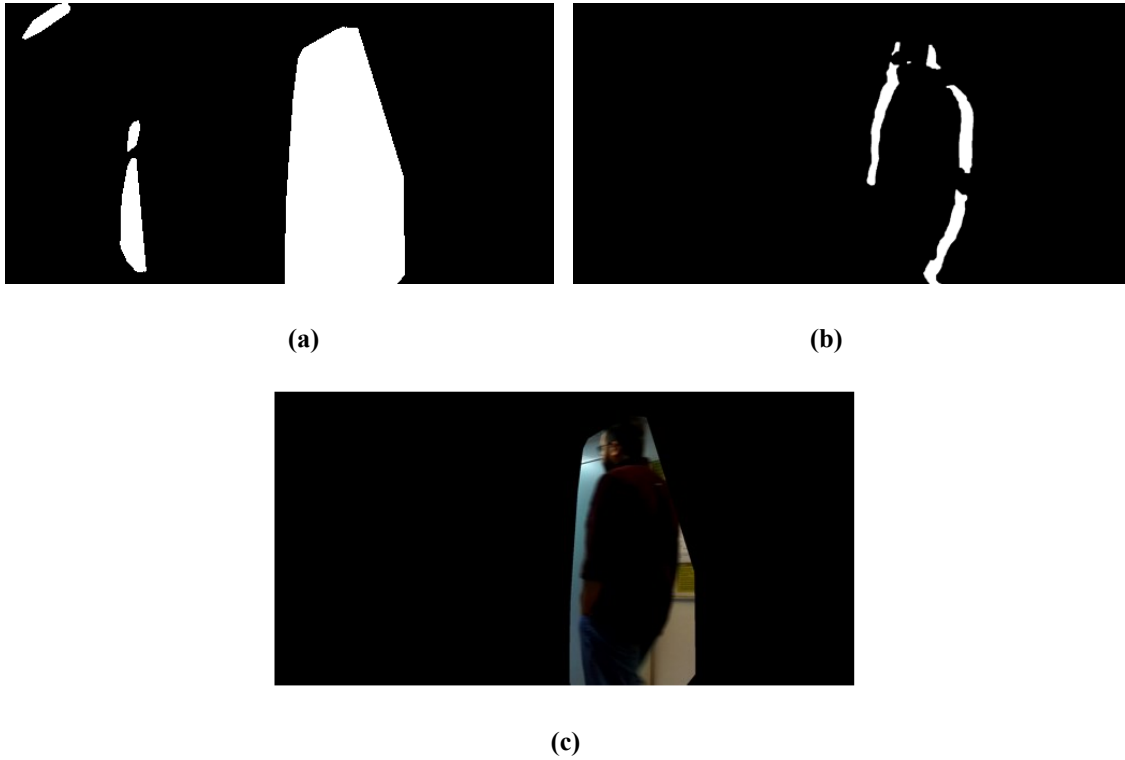


Figure 17. Segmentation of the dynamic object using binary masks (a) Mask obtained by projecting the clustered 3D dynamic points onto image plane (b) Mask obtained using GMM based motion detection (c) Segmented dynamic object from 2D images

4. EXPERIMENTAL SETUP AND RESULTS

In this section, we would present camera setup and analyse the results obtained for the proposed approach of Visual SLAM. Each test case is directed to examine the ability of the algorithm in a different environment. Since the primary goal of the study is to develop a method that can robustly perform in a dynamic environment, therefore, two of the test cases mimic a highly dynamic indoor environment. The third case is directed to test the ability of the approach to map a medium scale outdoor environment.

For experimentations conducted throughout this work, we used a Zed stereo camera [51]. The camera follows Pinhole camera model, with two cameras fixed at a baseline distance of 12 centimeters. The specifications of the Zed camera are notable. The effective range for this camera is 0.5 to 20 meters, with the capability of working both indoor and outdoor [51]. However, the performance deprecates with the increase in distance, therefore it was beneficial to limit our interest to a maximum of 6.5 meters in order to acquire valid and consistent data. This figure was explored empirically during this work.

For calibrating the Zed camera, we utilized the Computer Vision System Toolbox™ calibration algorithm which in turn adopts the work proposed by Jean-Yves Bouguet [52]. The parameters obtained after calibration for the Zed stereo camera are provided in Table 1. These parameters are specific to the device used and might vary slightly due to tolerances adopted in measurements during manufacturing of camera components.

Table 1. Zed camera calibrated parameters

Properties	Left Camera (L)	Right Camera (R)
Radial Distortion	[-0.1609, -0.0095, 0.0298]	[-0.1638, 0.001, 0.0182]
Tangential Distortion	[-3.3259 e-04, -6.2056 e-04]	[-2.1356 e-04, -2.8499 e-04]
Estimate Skew	0	0
Intrinsic Matrix (K)	[1.399 e+03, 0, 0; 0, 1.4008 e+03, 0; 1.0501 e+03, 6.322 e+02, 1]	[1.3997 e+03, 0, 0; 0, 1.4009 e+03, 0; 1.081 e+03, 5.81 e+02, 1]
Focal Length (fx, fy) px	[1.399 e+03, 1.4008 e+03]	[1.399 e+03, 1.4 e+03]
Principal Point (cx, cy) px	[1.0501 e+03, 6.3224 e+02]	[1.081 e+03, 5.81 e+02]
Mean Reprojection Error (px)	0.1961	0.1998
Translation of R Camera from L (mm)	[-1.216 e+02, -0.0957, -1.52]	
Rotation of R Camera from L (mm)	[0.999, -0.0019, -3.839 e-04; 0.0019, 0.999, -0.006; 3.9564 e-04, 0.006, 0.999]	

4.1 Case 1: Object introduction to scene and removal

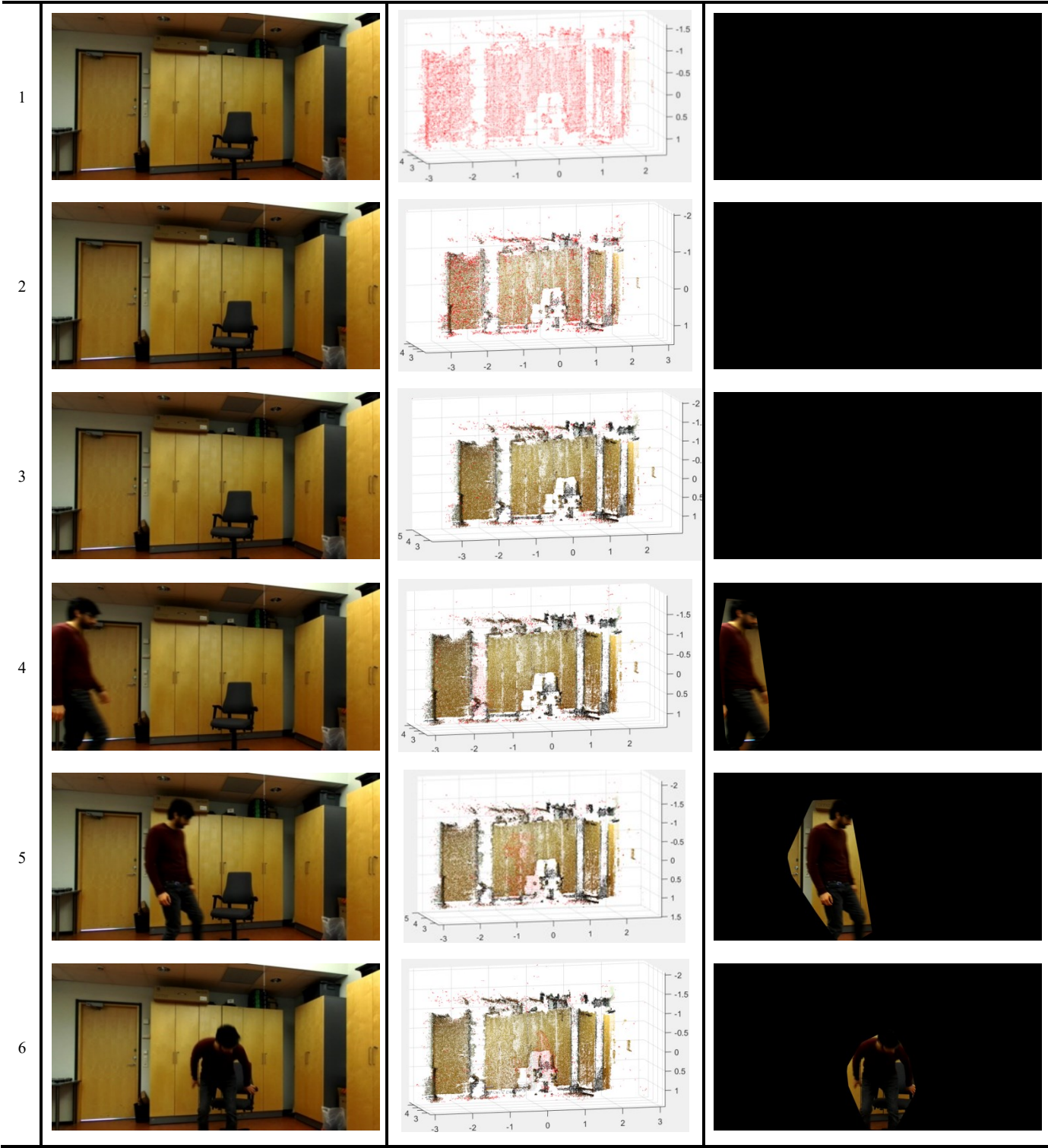
This dataset serves as the first test case to check the robustness of the implementation. The first test is directed to check how the algorithm performs when only the dynamics of the environment are changed. The indoor scene is recorded with the Zed camera fixed at a position. The video is acquired at a frame rate of 30 fps. The parameters used to process the sequence are provided in Table 2. The excerpt from the video along with the processed results can be seen in Figure 18. The first column has excerpt from the original image sequence followed by a map built for the environment in the second column. The third column shows the dynamic object that has been segmented, from the corresponding image, during its motion.

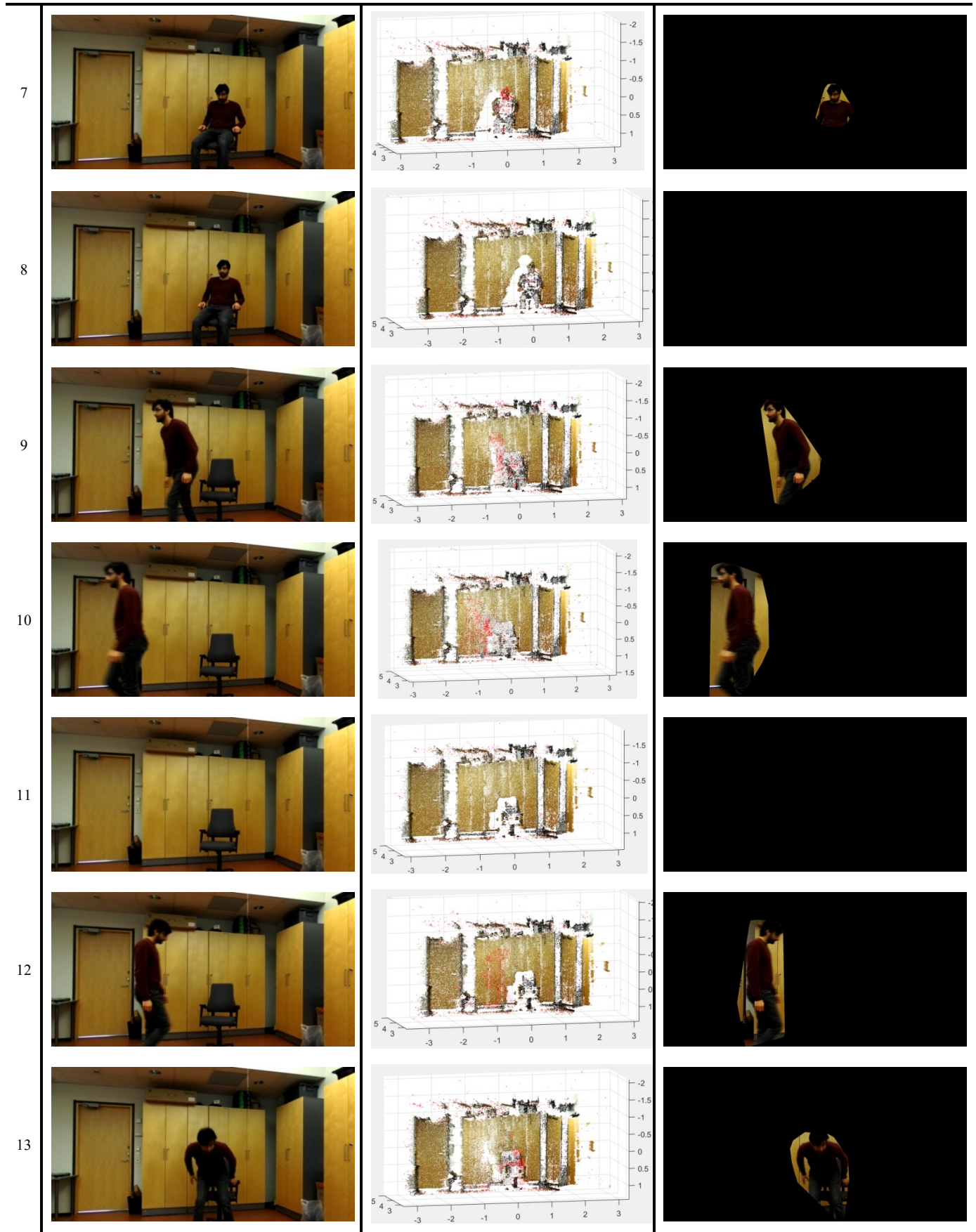
The image sequence starts with a static environment where it's corresponding downsampled point cloud is shown in red color. This shows that that the environment hasn't gathered enough confidence in the start to be classified static. Nonetheless, the segmentation verifies that the environment is static since no motion is observed in the GMM masks. The points quickly gain confidence during registration with the point clouds from the subsequent images. This period is highly affected by the framerate of camera acquisition and takes a few iterations. The points that have gathered confidence greater than 1 are termed static and obtain their original color. As the person walks into the static scene, 3D points pertaining to the person are dynamic and shown in red and at the same time, the person is segmented in the third column. The map successfully updates and is not affected by the moving 3D points. Once the person sits on the chair and does not show any considerable movement, he becomes part of the static environment and is registered into the map (8th row). In the following frames, the person moves out of the scene again. The corresponding 3D points of the person become dynamic since they cannot find associations and they are removed. While new points from the current frames help to continuously update the map. When the person moves out of the view, the map attains the initial static form (11th row). The action is repeated to verify the response of the system.

Table 2. Parameters used for the registration of test dataset 1.

Parameters	Value	Unit
Acquisition Rate	30	Frames per second
Point Cloud Generation Parameters		
Uniqueness threshold	60	constant
Disparity Range	0-64	pixel
Physical Z bound	6	meter
Down sampling	0.04	meter, Volumetric grid step
Registration Parameters		
Correspondence Threshold	0.05	meter
Merging Threshold	0.05	meter
Confidence Gain	0.1 (max), Gaussian distribution	constant

Penalty Reduction	0.01	constant
Threshold time for noise removal (t_{max})	5	frames





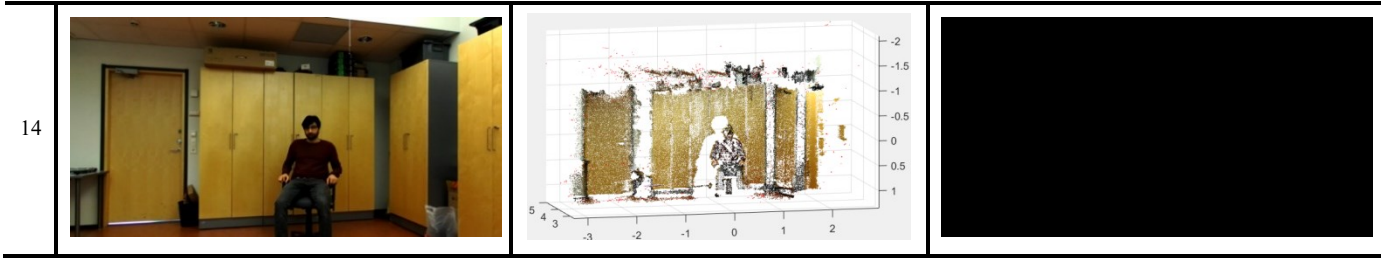


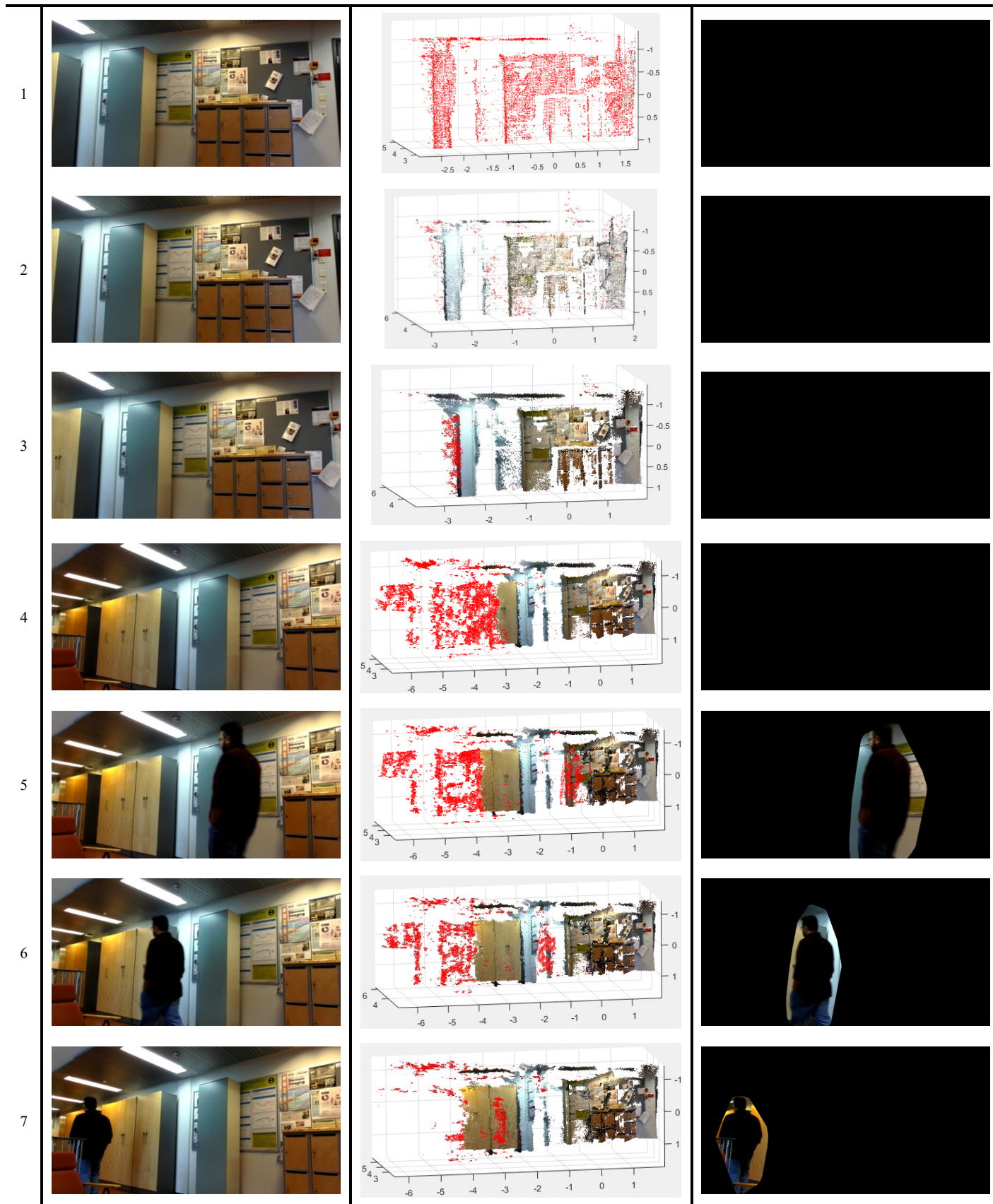
Figure 18. Test case 1; first column shows the original images, second column shows the registered map and the third column presents the segmented moving object.

4.2 Case 2: Passer-by in a moving scene

The second test case is a continuation of the first case, where camera motion is introduced into the scene. The same parameters are utilized for image acquisition as for Case 1. The parameters used to process the sequence are provided in Table 3. The excerpt from the video along with the processed results can be seen in Figure 19. In this test case, we map a corridor while a person passes through the scene. It can be observed from the results that the map is successfully updated along the moving person. The dynamic points pertaining to the moving person are clearly visible in front of the static part of the map, while unstable dynamic points are present at the far end of the map. The unstable/dynamic points at the far end are due to their inconsistency in the individual point clouds, as explained in Section 3.7.

Table 3. Parameters used for the registration of test dataset 2.

Parameters	Value	Unit
Acquisition Rate	30	Frames per second
Point Cloud Generation Parameters		
Uniqueness threshold	60	constant
Disparity Range	0-64	pixel
Physical Z bound	6.5	meter
Down sampling	0.02	meter, Volumetric grid step
Registration Parameters		
Correspondence Threshold	0.03	meter
Merging Threshold	0.03	meter
Confidence Gain	0.1 (max), Gaussian distribution	constant
Penalty Reduction	0.01	constant
Threshold time for noise removal (t_{max})	5	frames



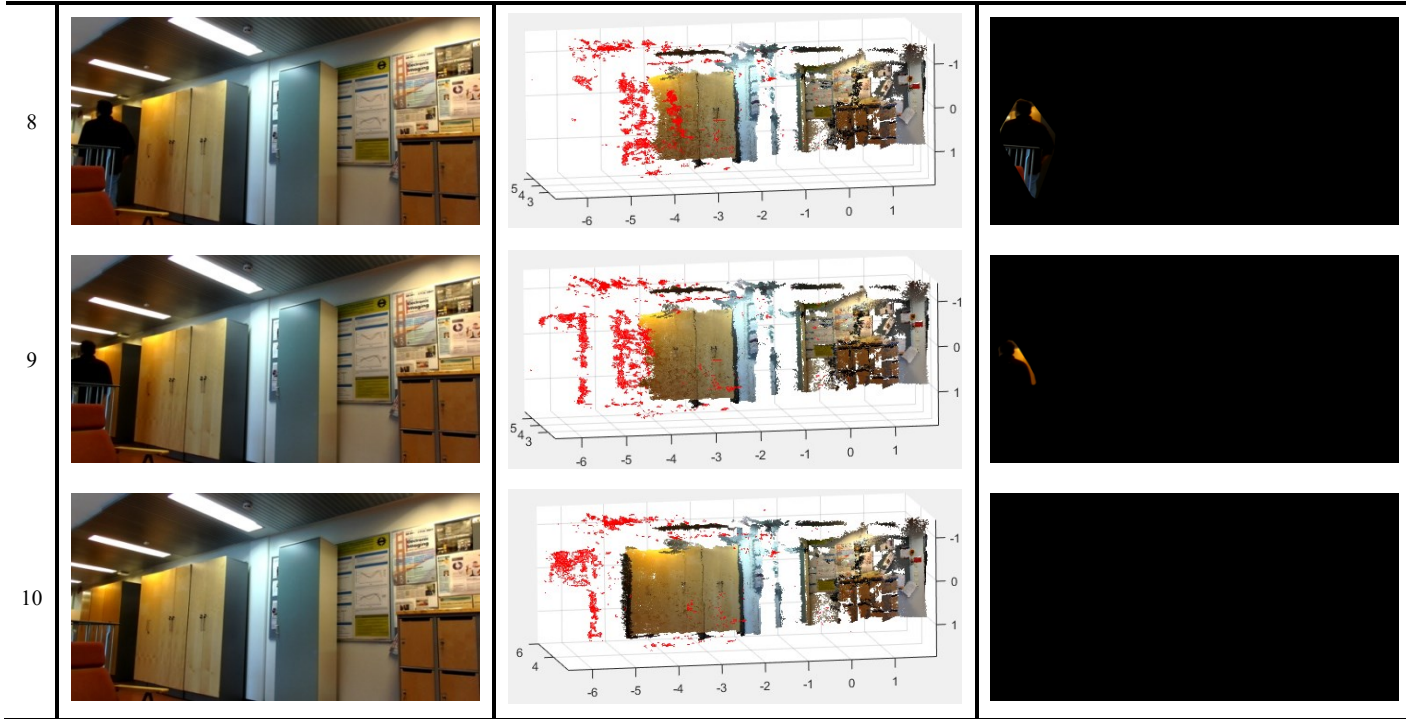


Figure 19. Video test case 2, where first column shows the original images, second column shows the registered map and the third column presents the segmented moving object.

4.3 Case 3: Outdoor environment mapping

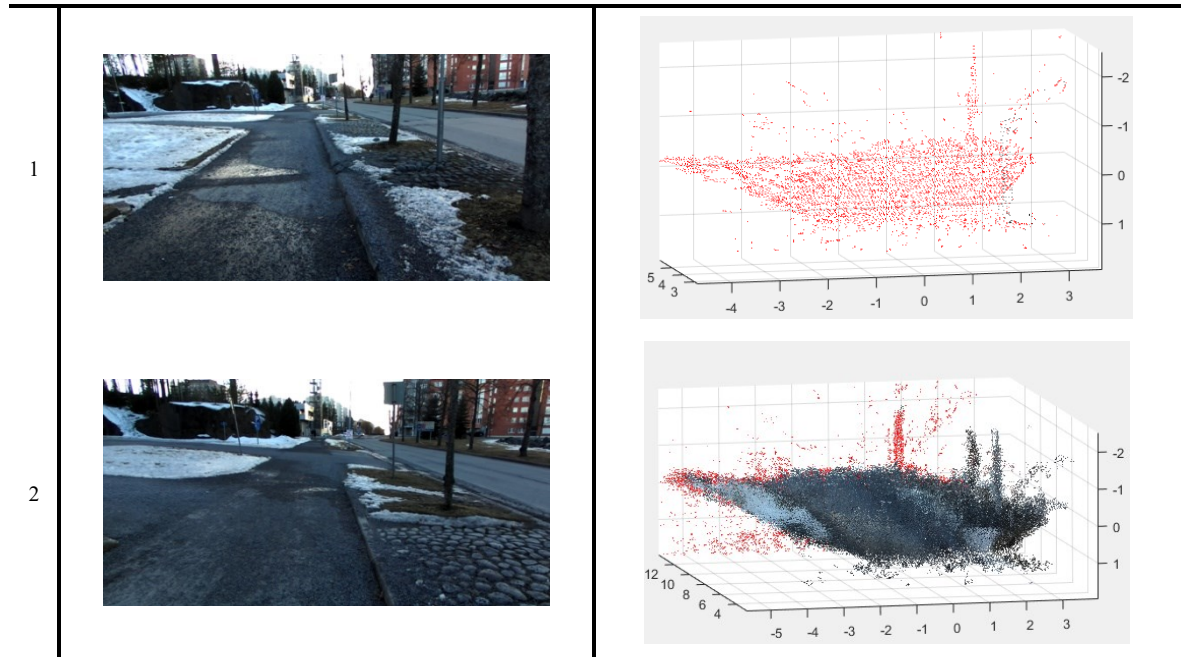
The third case scenario tests the traditional ability of SLAM to map over a longer range in an outdoor environment while maintaining the ability to compensate for any dynamics introduced into the scene. The primary purpose of this dataset was to check the robustness of the implementation. The video was recorded in front of the Tampere University of Technology with the hand-held Zed camera. The dataset provides several challenges to test the system. The video was recorded on a cloudy day in winter. The frame rate of image acquisition was limited to 10. The hand-held camera along with a low framerate resulted in sharp changes in the scene between the frames. The scene was recorded for a total length of 45 meters.

The parameters used to process the sequence are provided in Table 4. Some important excerpts from the video along with the processed results can be seen in Figure 20. The order of data the same as mentioned for the previous test cases. It can be observed that the environment introduces more noise into the registration process as compared to the previous datasets. Nonetheless, the point clouds are registered sequentially and the map builds up (1st and 2nd row). The images in the 3rd row of Figure 20 shows two pedestrians walking in and out of the scene. It can be observed from the map at the corresponding stage that the pedestrians are registered as dynamic objects in the map and the moving objects did not affect the registration process. The pedestrians are then discarded from the map as they move out of the scene. In the subsequent frames (4th and 5th row), a tree

comes into the range of the camera. The tree is initially detected as a dynamic object but it is soon added to the map as a stationary object. This test case effectively demonstrates the ability of the proposed approach to perform in an outdoor dynamic environment.

Table 4. Parameters used for the registration of test dataset 3.

Parameters	Value	Unit
Acquisition Rate	10	Frames per second
Point Cloud Generation Parameters		
Uniqueness threshold	30	constant
Disparity Range	0-64	pixel
Physical Z bound	6	meter
Down sampling	0.08	meter, Volumetric grid step
Registration Parameters		
Correspondence Threshold	0.2	meter
Merging Threshold	0.2	meter
Confidence Gain	0.1 (max) , Gaussian distribution	constant
Penalty Reduction	0.01	constant
Threshold time for noise removal (t_{max})	10	frames



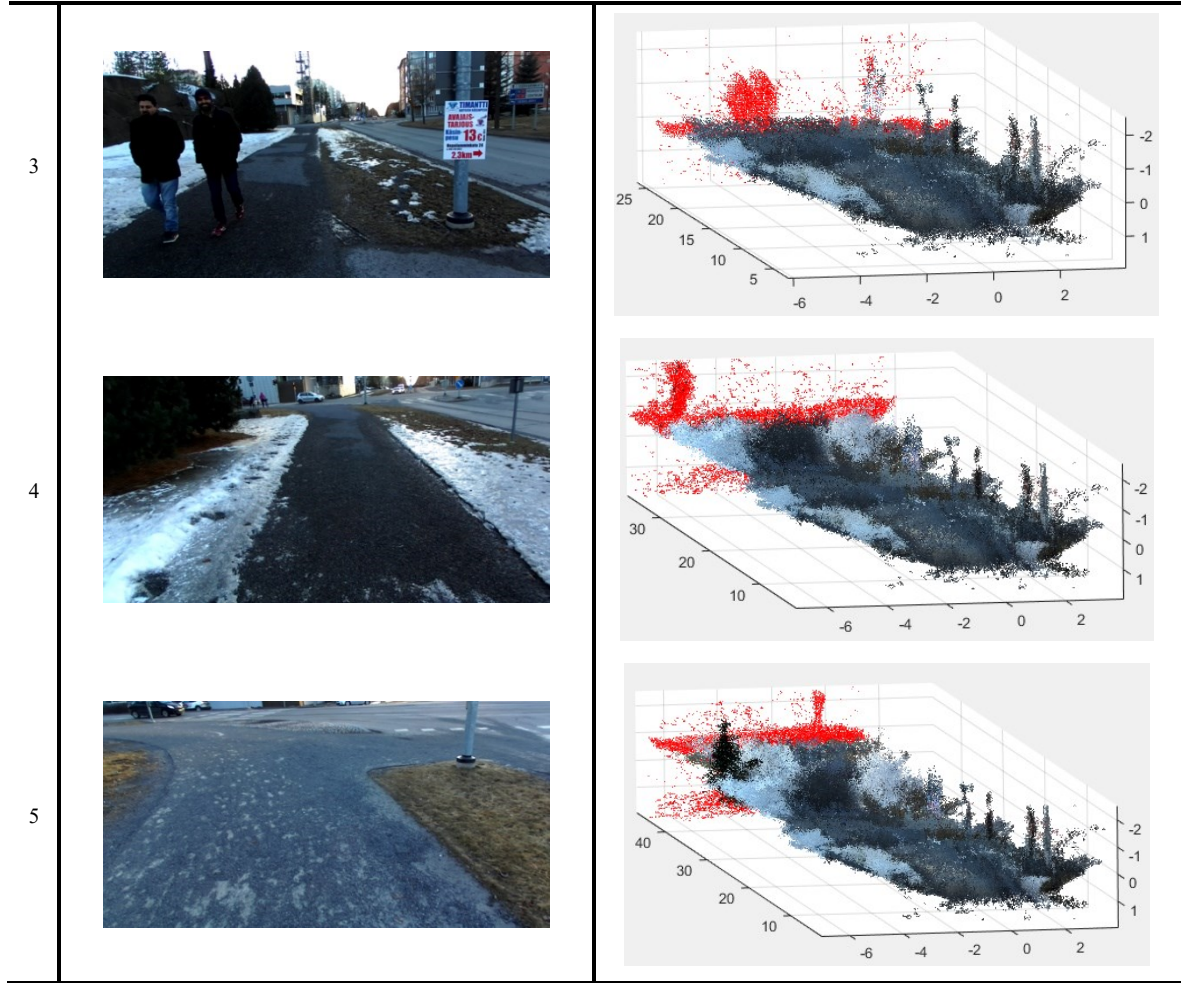


Figure 20. Video test case 3. The video was captured in an outdoor dynamic environment. The first column shows the original images, second column shows the registered map.

4.4 Case 4: Narrow Corridor with Semi-Transparent surfaces

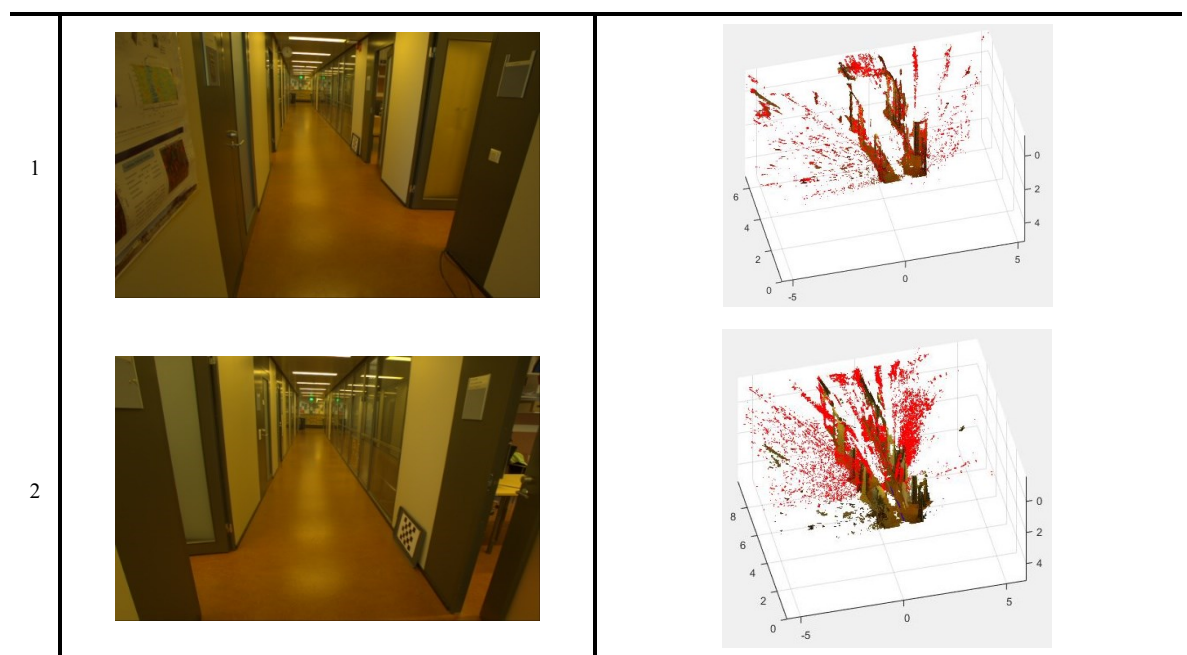
This test dataset was generated to analyse the ability of our system to tackle narrow pathways with less distinct structures. The corridor offers repeating spatial structures which are challenging for the ICP to register. Furthermore, the semi-transparent surfaces (glass doors and room windows) in the hallway generate a great amount of noise in the point clouds, making the registration process more difficult.


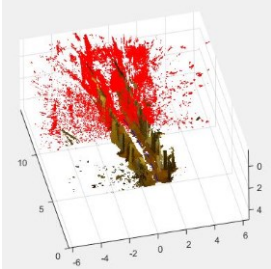

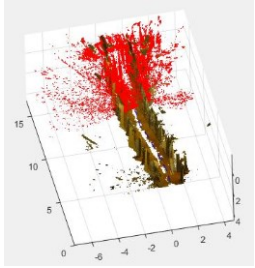
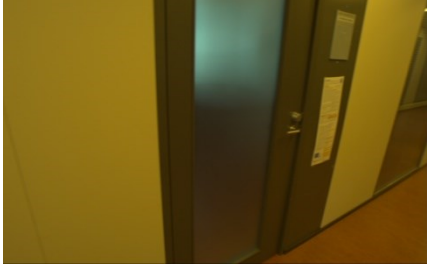
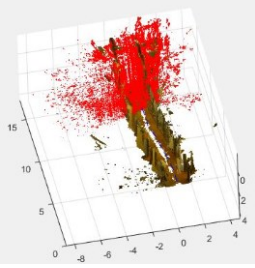
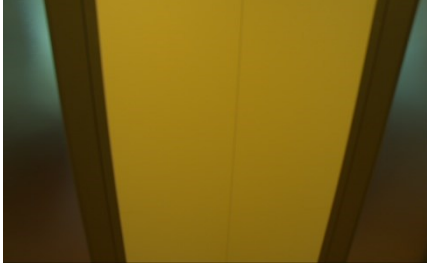
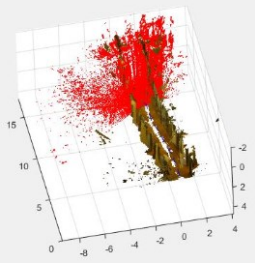
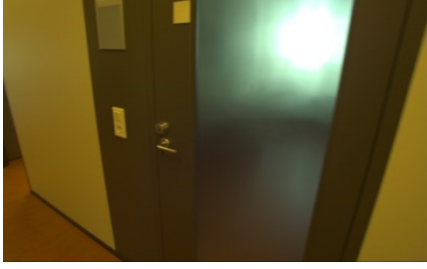
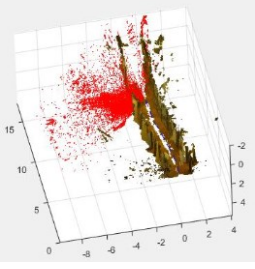

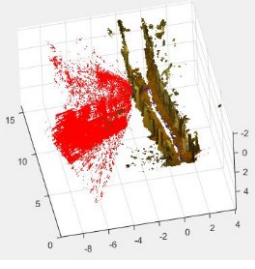
The parameters used to process the sequence are provided in Table 5. Some important excerpts from the video along with the processed results can be seen in Figure 21. The order of data is the same as mentioned for the previous test case. It can be observed throughout the sequence that the point cloud contains a considerable amount of noise. The images in the 3rd row of **Figure 21** illustrate that the semi-transparent and transparent surfaces introduce a lot of noise into the registration process. Nonetheless, the corridor is mapped accurately till the rotation of the camera. During the 180° rotation to turn back (observed in row 4 to 7), the proximity of the camera to the wall restricts the view. The

point clouds generated from the images during the rotation have very few points. Moreover, the presence of the semi-transparent glass door produced a noisy and unstructured point cloud. The result is a 90° rotation instead of 180° . Afterwards, the registration process continues straight (observed in row 8 to 10) in the direction of motion i.e. forward. Upon the second rotation at the end of corridor (observed in row 11 to 13), a similar partial rotation is registered instead of a complete 180° turn due to similar causes. This registration failure was inevitable with just the use of ICP. Such problems can be avoided by either supplementing the trajectory computation with another means of transformation computation such as IMU or using predictive filtering methods to compensate for the brief inaccurate transformation estimation provided by the ICP.

Table 5 Parameters used for the registration of test dataset 4.

Parameters	Value	Unit
Acquisition Rate	10	Frames per second
Point Cloud Generation Parameters		
Uniqueness threshold	15	constant
Disparity Range	0-240	pixel
Physical Z bound	6	meter
Down sampling	0.02	meter, Volumetric grid step
Registration Parameters		
Correspondence Threshold	0.035	meter
Merging Threshold	0.035	meter
Confidence Gain	0.1 (max), Gaussian distribution	constant
Penalty Reduction	0.01	constant
Threshold time for noise removal (t_{max})	10	frames



3		
4		
5		
6		
7		
8		

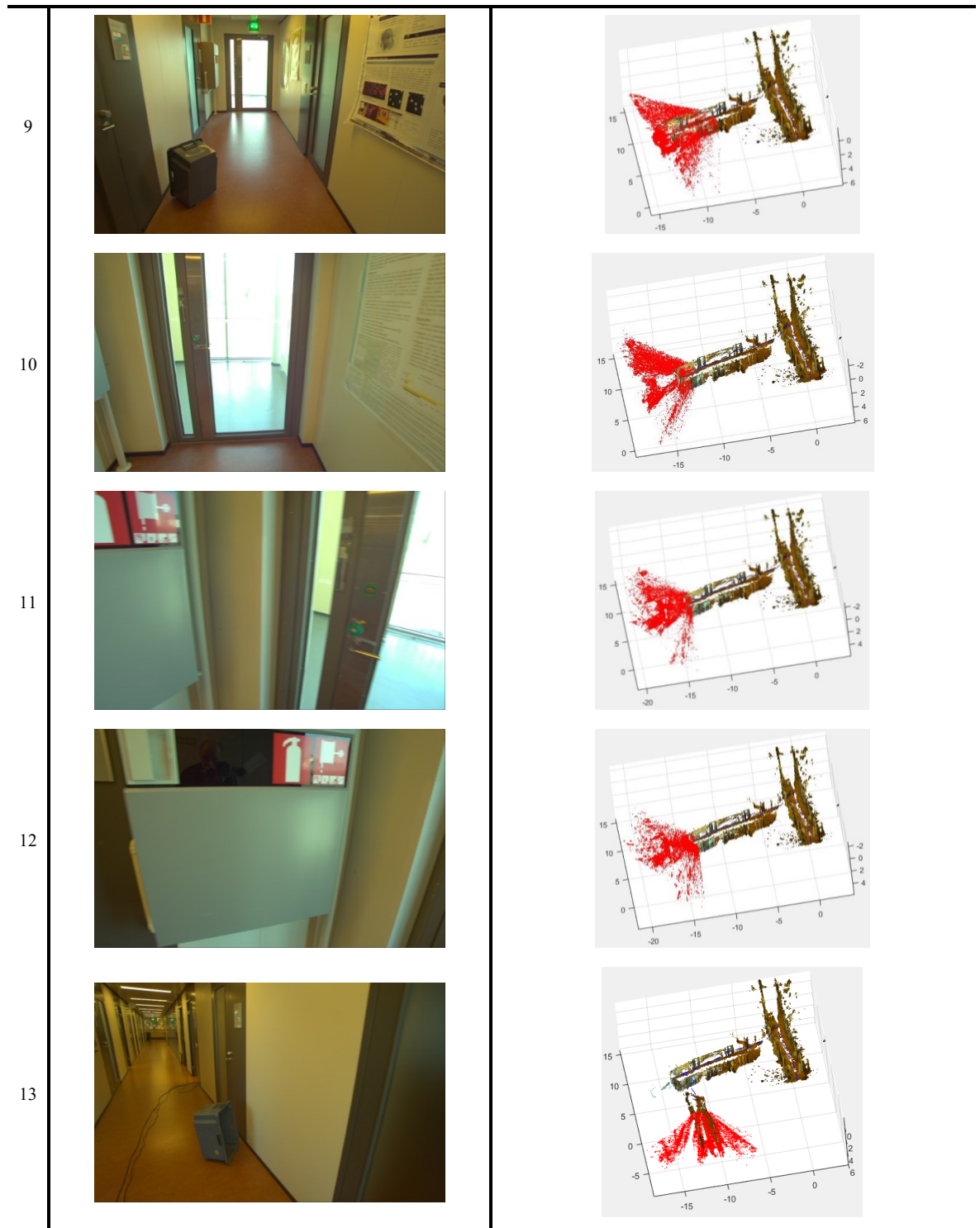


Figure 21. Video test case 4. The test data was recorded at Tietotalo, Tampere University of Technology. The first column shows the original images, second column shows the registered map.

5. CONCLUSIONS

This thesis presents an effective approach to implement stereo based Visual SLAM in an active dynamic environment. The system performs without any prior knowledge or model initialization of the environment or its part. The approach exploits the intrinsic characteristics of the ICP algorithm to develop a robust framework. The outliers of the data association step during the registration process are considered the key contributor to the study of the dynamics of the objects in the scene. A general measure of confidence assigned to each 3D point is used to differentiate the points as static and dynamic. During the study, the main objectives were to (1) generate and update a semi-dense map of the active dynamic environment (2) isolate the dynamic object in the global map (3) segment the moving objects in the 2D images. All the mentioned objectives were successfully accomplished. The results from the experiments show the effectiveness of the system in realistic environments. The test cases were recorded both indoor and outdoor along with additional complexities including camera motions, camera jitter, light conditions and object motion.

Each of the four test cases incrementally introduces challenges for the registration process. The first two cases specifically focus on the dynamic objects in an indoor environment. The datasets were recorded with a hand-held static and moving camera, respectively. Both the scenes were successfully mapped along with accurate segmentation of the moving object. The third test case composes of an outdoor dataset recorded on a cloudy winter day. The video was recoded at 10fps with a hand-held camera, resulting in a choppy image sequence. The scene in this test case offered little distinct spatial structures for the ICP, and therefore, resulted in a noisy map that seems to be compressed at times. Nonetheless, the sequence was successfully registered. The last test case proved to be most challenging and pointed out the limitations of the proposed approach. The presence of many semi-transparent surfaces in the narrow corridor resulted in the generation of noisy point cloud for the image pairs. Moreover, the walls when viewed in close-up during the rotation of the camera provided little texture to generate a structurally sound point cloud with enough 3D points. The combination of semi-transparent surfaces and close-up view of the wall resulted in point clouds that were weak candidates for ICP algorithm. The proposed approach failed to register the point clouds from the images sequence, accurately. For such cases, the system should be aided with complementary sensing mechanism e.g. IMU or predictive filtering algorithms such as Extended Kalman Filtering.

In our experiments, the system was successfully tested on framerates between 10-30. However, 30fps is more desirable for implementations following only visual cues. This is mainly due to the fact that the camera pose difference between the consecutive frames

significantly reduces. As a result, less blur is observed during camera motion, the transformation computation speeds up, and registration accuracy increases.

In its current state, our system requires the initialization of the threshold value. This threshold value determines which points are close enough to be classified as inliers during data association. The same threshold value is then used for merging of the neighboring close points. Although, the threshold value is intuitive in nature and can be easily selected with few trials, an automatic selection methodology would eliminate this tedious parameter tuning. Further improvement could be made in terms of real-time processing capability. The implementation was done in MATLAB, the datasets were recorded and tested offline. This arrangement, especially motion segmentation, seems to be computationally exhaustive for MATLAB and takes a considerable amount of time for processing.

REFERENCES

- [1] Chatila R, Laumond J. Position referencing and consistent world modeling for mobile robots. In Proceedings of 1985 IEEE International Conference on Robotics and Automation on 1985 Jan (pp.138–145).
- [2] Andrade-Cetto J, Sanfeliu A. Concurrent map building and localization with landmark validation. Object recognition supported by user interaction for service robots. In Proceedings of the 16th IAPR international conference on pattern recognition on 2002 Aug 11 2, (pp. 693–696). IEEE
- [3] Nüchter A, Lingemann K, Hertzberg J, Surmann H. 6D SLAM—3D mapping outdoor environments. *Journal of Field Robotics*. 2007;24(8-9):699-722.
- [4] Thrun S, Montemerlo M, Aron A. Probabilistic Terrain Analysis For High-Speed Desert Driving. *Robotics: Science and Systems II*. 2006.
- [5] Tardos J, Neira J, Newman P, Leonard J. Robust Mapping and Localization in Indoor Environments Using Sonar Data. *The International Journal of Robotics Research*. 2002;21(4):311-330.
- [6] Bogdan Rusu R, Sundareshan A, Morisset B, Hauser K, Agrawal M, Latombe J et al. Leaving Flatland: Efficient real-time three-dimensional perception and motion planning. *Journal of Field Robotics*. 2009;26(10):841-862.
- [7] Schleicher D, Bergasa L, Ocana M, Barea R, Lopez M. Real-Time Hierarchical Outdoor SLAM Based on Stereovision and GPS Fusion. *IEEE Transactions on Intelligent Transportation Systems*. 2009;10(3):440-452.
- [8] Younes, Georges, Daniel Asmar, and Elie Shammas. A survey on non-filter-based monocular Visual SLAM systems. *CoRR*. 2016.
- [9] Handa A, Whelan T, McDonald J, Davison A. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *IEEE International Conference on Robotics and Automation (ICRA)* on 2014 May 31 (pp. 1524 – 1531). IEEE.
- [10] Ahn S, Choi M, Choi J, Chung W. Data Association Using Visual Object Recognition for EKF-SLAM in Home Environment. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* on 2006 Oct 9 (pp. 2588 – 2594). IEEE.
- [11] Se S, Lowe D, Little J. Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks. *The International Journal of Robotics Research*. 2002;21(8):735-758.

- [12] Olson C, Matthies L, Schoppers M, Maimone M. Rover navigation using stereo ego-motion. *Robotics and Autonomous Systems*. 2003;43(4):215-229.
- [13] Pillai, Sudeep, and John Leonard. "Monocular slam supported object recognition." *arXiv preprint arXiv:1506.01732* (2015).
- [14] Kaess M, Dellaert F. Probabilistic structure matching for visual SLAM with a multi-camera rig. *Computer Vision and Image Understanding*. 2010;114(2):286-296.
- [15] Carrera G, Angeli A, Davison A. SLAM-based automatic extrinsic calibration of a multi-camera rig. In *International Conference on Robotics and Automation* on 2011 May 9 (pp. 2652 – 2659). IEEE.
- [16] Davison A, González Y, Kita N. Real-time 3D SLAM with wide-angle vision. In *5th IFAC/EURON symposium on intelligent autonomous vehicles*.2004.
- [17] Scaramuzza D, Siegwart R. Appearance guided monocular omnidirectional visual odometry for outdoor ground vehicles. *IEEE Transactions on Robotics* .2008; 24(5): 1015–1026.
- [18] Huang A, Bachrach A, Henry P, Krainin M, Maturana D, Fox D et al. *Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera*. Springer Tracts in Advanced Robotics. 2016;235-252.
- [19] Hartley R, Sturm P (1997) Triangulation. *Computer Vision and Image Understanding*.1997; 68(2): 146–157
- [20] Engel J, Stuckler J, Cremers D. Large-scale direct SLAM with stereo cameras. In *International Conference on Intelligent Robots and Systems (IROS)* on 2015 Sep 28 (pp. 1935 – 1942). IEEE.
- [21] Pupilli M, Calway A. Real-Time Visual SLAM with Resilience to Erratic Motion. In *Computer Society Conference on Computer Vision and Pattern Recognition* on 2006 June 17 (pp. 1244–1249). IEEE.
- [22] Mei C, Reid I. Modeling and generating complex motion blur for real-time tracking. In *IEEE Conference on Computer Vision and Pattern Recognition* on 2008 June 23 (pp.1-8). IEEE.
- [23] Davison A, Reid I, Molton N, Stasse O. MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2007;29(6):1052-1067.

- [24] Newcombe RA, Davison AJ. Live dense reconstruction with a single moving camera. In Conference on Computer Vision and Pattern Recognition. IEEE Computer Society on 2010 June 13 (pp. 1498–505).
- [25] Meilland M, Comport A. On unifying key-frame and voxel-based dense visual SLAM at large scales. In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2013.
- [26] Newcombe R, Lovegrove S, Davison A. DTAM: Dense tracking and mapping in real-time. In International Conference on Computer Vision on 2011 Nov 6 (pp. 2320-2327).
- [27] Kerl C, Stuckler J, Cremers D. Dense Continuous-Time Tracking and Mapping with Rolling Shutter RGB-D Cameras. In International Conference on Computer Vision (ICCV) on 2015 Dec 7 (pp. 2264-2272). IEEE.
- [28] Churchill W, Newman P. Experience-based navigation for long-term localisation. *The International Journal of Robotics Research*. 2013;32(14):1645-1661.
- [29] Walcott-Bryant A, Kaess M, Johannsson H, Leonard J. Dynamic pose graph SLAM: Long-term mapping in low dynamic environments. In International Conference on Intelligent Robots and Systems on 2012 Oct 7-12 (pp.1871 - 1878). IEEE.
- [30] Konolige K, Bowman J, Chen J, Mihelich P, Calonder M, Lepetit V et al. View-based Maps. *The International Journal of Robotics Research*. 2010;29(8):941-957.
- [31] De Aguiar E, Stoll C, Theobalt C, Ahmed N, Seidel H, Thrun S. Performance capture from sparse multi-view video. *ACM SIGGRAPH 2008 papers on - SIGGRAPH '08*. 2008;27(3):1.
- [32] Zollhöfer M, Theobalt C, Stamminger M, Nießner M, Izadi S, Rehmann C et al. Real-time non-rigid reconstruction using an RGB-D camera. *ACM Transactions on Graphics*. 2014;33(4):1-12.
- [33] Keller M, Lefloch D, Lambers M, Izadi S, Weyrich T, Kolb A. Real-Time 3D Reconstruction in Dynamic Scenes Using Point-Based Fusion. In International Conference on 3D Vision on 2013 June 29 (pp.1-8).
- [34] Fankhauser P, Bloesch M, Rodriguez D, Kaestner R, Hutter M, Siegwart R. Kinect v2 for mobile robot navigation: Evaluation and modeling. In International Conference on Advanced Robotics on 2015 July 27 (pp. 388 – 394).
- [35] Hartley R, Zisserman A. Multiple view geometry in computer vision. 1st ed. Cambridge: Cambridge University Press; 2003.

- [36] Calibrator C, App S. What Is Camera Calibration? - MATLAB & Simulink - MathWorks United Kingdom [Internet]. Se.mathworks.com. 2017 [cited 10 June 2017]. Available from: <https://se.mathworks.com/help/vision/ug/camera-calibration.html#buvr2qb-1>
- [37] Hartley R, Zisserman A. Multiple view geometry in computer vision. Cambridge: Cambridge University Press; 2003.
- [38] Gil A, Mozos O, Ballesta M, Reinoso O. A comparative evaluation of interest point detectors and local descriptors for visual SLAM. Machine Vision and Applications. 2009;21(6):905-920.
- [39] Davison A, Kita N. 3D simultaneous localisation and map-building using active vision for a robot moving on undulating terrain. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition on 2001 Dec 8-14 (pp. 384-391. IEEE.
- [40] Schon T, Karlsson R, Tornqvist D, Gustafsson F. A framework for simultaneous localization and mapping utilizing model structure. In 10th International Conference on Information Fusion on 2007 July 9-12 (pp. 1-8).
- [41] Klein G, Murray D. Parallel Tracking and Mapping for Small AR Workspaces. In 6th IEEE and ACM International Symposium on Mixed and Augmented Reality on 2007 Nov 13-16 (pp.1-10).
- [42] B. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In International Joint Conference on Artificial Intelligence (IJCAI) on 1981 August 24-28 (pp. 674–679).
- [43] Hirschmuller H. Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition on 2005 June 20-24 (pp. 807-814).
- [44] Calibrator S. Disparity map between stereo images - MATLAB disparity - MathWorks United Kingdom [Internet]. Se.mathworks.com. 2017 [cited 10 June 2017]. Available from: <https://se.mathworks.com/help/vision/ref/disparity.html#bt9av8y>
- [45] Besl P, McKay N. A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1992;14(2):239-256.
- [46] Pomerleau F, Colas F, Siegwart R. A Review of Point Cloud Registration Algorithms for Mobile Robotics. Foundations and Trends in Robotics. 2015;4(1):1-104.

- [47] Agglomerative clusters from data - MATLAB clusterdata - MathWorks United Kingdom [Internet]. Se.mathworks.com. 2017 [cited 10 June 2017]. Available from: <http://se.mathworks.com/help/stats/clusterdata.html>
- [48] C. Stauffer and W. E. L. Grimson. Adaptive background mixture model for real-time tracking. In Proceedings of IEEE Int'l Conference on Computer Vision and Pattern Recognition, pages 246–252, 1999.
- [49] C. Stauffer and W.E.L. Grimson. Learning patterns of activity using real-time tracking. IEEE Transaction on Pattern Analysis and Machine Intelligence, 22(8):747–757, Aug. 2000.
- [50] P. Kaewtrakulpong, R. Bowden, An Improved Adaptive Background Mixture Model for Realtime Tracking with Shadow Detection, In Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems, AVBS01, Video Based Surveillance Systems: Computer Vision and Distributed Processing (September 2001)
- [51] ZED Stereo Camera [Internet]. Stereolabs.com. 2017 [cited 10 June 2017]. Available from: <https://www.stereolabs.com/>
- [52] Bouguet, J. Y. Camera Calibration Toolbox for Matlab. Computational Vision at the California Institute of Technology. Camera Calibration Toolbox for MATLAB.