UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

FAKULTÄT FÜR
ELEKTROTECHNIK,
INFORMATIK UND
MATHEMATIK

3D Motion Analysis for Mobile Robots

Zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

der Fakultät für Elektrotechnik, Informatik und Mathematik
der Universität Paderborn
vorgelegte Dissertation
von

MS.Eng. Mohamed Salah El-Neshawy Shafik

Paderborn

Referentin: Prof. Dr.-Ing. Bärbel Mertsching
Korreferent: Prof. Dr. Peter Schreier

Tag der mündlichen Prüfung: 20.12.2012
Paderborn, 2013
Diss. EIM-E/288

Dedication

Declaration

I hereby declare that I have completed the work on this PhD dissertation with my own efforts and no part of this work or documentation has been copied from any other source. It is also assured that this work is not submitted to any other institution for award of any degree or certificate.

Paderborn, November 17, 2013

Mohamed Shafik

Kurzfassung

Die Segmentierung von Bewegung hat sich zu einem der schwierigsten Probleme im maschinellen Sehen entwickelt. Verfahren zur Detektion bewegter Objekte sowie zur Schätzung der Bewegungsparameter unterstützen die Verarbeitung dynamischer Szenen beträchtlich. Ein 3D-Bewegung im maschinellen Sehen resultiert aus räumlich-zeitlichen Veränderungen der Pixelinformationen. Der Nachweis solcher Unterschiede zwischen zwei oder mehreren aufeinander folgenden Bildern ist der erste Schritt zur Bestimmung der Bewegung. Daher hängt die Schätzung der Bewegungsparameter zusätzlich zur Segmentierung von Genauigkeit der Detektion ab. Die Berechnung einer einzigen 3D-Bewegung aus einem Fluss von 2D-Bildern durch das Finden der optimalen Koeffizienten in einer 2D-Signal-Transformation hat seine Effizienz unter Beweis gestellt. Allerdings, im Falle mehrerer 3D-Bewegungen, leidet die resultierende Segmentierung unter mehreren Nachteilen, wie die innere Verwechselung zwischen Translation und Rotation und dem Problem der degenerierten Bewegungen, vor allem, wenn das Eingabe-Bewegungsvektorfeld sehr verrauscht ist. Auf der anderen Seite schlagen solche Techniken fehl, wenn 3D-Bewegungen teilweise überlappen.

Diese Arbeit präsentiert eine schnelle Schätzung der Bewegungsparameter, die zu einer signifikanten Verringerung der Rechenzeit des "3D-Motion-Segmentation" Ansatzes sowie einem verringerten mittleren Fehler der geschätzten Parameter auch bei starkem Rauschen führt. Darüber hinaus wurde ein Salienz-basierter Ansatz für die Schätzung und Segmentierung von 3D-Bewegungen aus mehreren bewegten Objekten mittels 2D Bewegungsvektorfeldern entwickelt. Eine Klassifizierungsmodul wurde implementiert, um die globale Bewegung der Kamera zu definieren und um typische Probleme der Wahrnehmung autonomer mobiler Roboter zu lösen, wie Bildrauschen, Verdeckung und Berücksichtigung der Eigenbewegung. Weiterhin schlagen wir eine schnelle biologisch motivierte Schätzung von 3D-Bewegungsparametern

vor. Die Ergebnisse belegen, dass die vorgestellten Verfahren eine erfolgreiche Erkennung und Bewertung von vordefinierten 3D-Bewegungsmustern und insbesondere Bewegungen in die Richtung eines Roboters erlauben. Sie sind damit ein wichtiger Meilenstein in Richtung einer erfolgreichen Vorhersage von Kollisionen.

Abstract

Motion segmentation has evolved into one of the most challenging problems in computer vision. The process of detecting moving objects as well as the estimation of their motion parameters provides a significant source of information to better understand dynamic scenes. A 3D motion in terms of computer vision results from the spatio-temporal change of pixel information. The detection of such differences between two or more consecutive frames is the first step in determining the related motion. Therefore, the estimation of the motion parameters in addition to the segmentation process depends on the accuracy of the detection process. Computing a single 3D motion from a 2D image flow by finding the optimal coefficient values in a 2D signal transform has proven its efficiency. However, in the case of multiple 3D motions, the resulting segmentation suffers from several drawbacks, such as the inherent confusion between translation and rotation and the problem of degenerated motions especially if the input motion vector field (MVF) is very noisy. On the other hand, such techniques failed to handle spatially overlapping 3D motion vector fields (3D transparent motion).

In this work, we present a fast approach to estimate the motion parameter coefficients, which results in a significant reduction of the computational time of the 3D motion segmentation approach as well as a decrease in the mean error of the estimated parameters even with highly noisy MVF. Furthermore, a saliency-based approach for estimating and segmenting 3D motions of multiple moving objects represented by 2D motion vector fields (MVF) was developed. A classification module has been implemented to define the global motion of the mounted camera in order to overcome typical problems in autonomous mobile robotic vision such as noise, occlusions, and inhibition of the ego-motion defects of a moving camera head. Moreover, we propose a fast depth-integrated 3D motion parameter estimation approach which takes into consideration the perspective transformation and the depth information to accurately

estimate biologically motivated classifier cells in the 3D space using the geometrical information of the stereo camera head. The results show a successful detection and estimation of predefined 3D motion patterns such as movements toward the robot which is a vital milestone towards a successful prediction of possible collisions.

Acknowledgements

First of all, praise and thanks be to Allah who enabled me to reach this level of academic achievement. Secondly, I am very grateful to my parents whose endless love and prayers made it come so far. I would like to thank my precious wife for her continued patience and support.

I am extremely thankful to my supervisor Professor Bärbel Mertsching whose guidance and support in all helpful aspects during the time of my PhD kept things advancing.

I would like to express my deepest appreciation to all those who provided me the possibility to complete this work. My colleague Dirk Fischer deserves a special thanks and appreciation for his continuous technical support and keeping the machines up and running without a break.

Contents

1 Introduction

Computer vision holds a special position in developing important applications such as robotics, surveillance and transportation. Among these systems, active vision has the ability to interact with dynamic environments by operating on sequences of images and altering its focal point of attention to scan the scene which provides the ability to detect and track several moving targets. The use of active vision systems on mobile robots significantly changes the way computer vision can be used. Such systems can actively control camera parameters according to the required situation such as orientation, focus and zoom, especially for mobile robots navigating in unknown environments. On the other hand, it has been shown that motion information plays an important role in visual tasks as diverse as control of eye movements, depth perception, object segregation, estimation of ego-motion and time-to-collision.

As the computation power has been increased since the beginning of the motion analysis studies, more techniques and approaches has been used for the motion estimation such as regularization, robust statistics and Markov random fields. Over the last decade, the increased interest in the field of motion segmentation has lead to expanding its applications to many areas of machine vision e. g. object tracking [WS02, HKW08], activity surveillance [AWK+05, MCK09], image and video compression [KA02, LZL+07], and object recognition [Hun05, TMD09]. In active vision systems, the scope of these applications can be more complex but they can help in development of autonomous tools useful such as survivor rescue systems, security guard robots, and adaptive systems for driver assistance.

1.1 Motion Analysis in Active Vision

Motion is the change in the relative position of objects. The navigation of an autonomous vehicle through dynamic environments requires a good sense of motion. Hence, the dynamic model of the environment has to be maintained in order to update the existing information. Mounting active vision systems on mobile robots could provide real-time feedback of the current traffic conditions which allow them to interact with a rapidly changing dynamic environment (fig. 1.1 shows an example of a mounted active vision system on a mobile robot from our lab (GETbot)). In order to achieve such targets, a 3D motion analysis research has to overcome several challenges concerning the detection and recognition of multi-moving objects within the concepts of image understanding.

There are many challenges in 3D motion analysis in dynamic scenes. First, the implemented algorithms must be able to absorb changes in the 3D pose and also tolerate noise in the input images. Secondly, the vision system should be able to detect and classify any additional features that may appear in the observed scene such as a textured background or occluded objects. Implementing the capability to deal with object motions in active vision systems improves the ability to understand complex 3D motions of multiple objects in dynamic environments. In this context, the motion detection process can be considered as a part of a general object recognition module. Such integration is vital to distinguish between object movements and artifacts that could affect the pixels value such as an illumination change.

1.2 Formulation of the Problem

Accurate interpretation of the 3D motion parameters [1] of moving objects is the key to better understand dynamic scenes. The input to the 3D motion parameters estimation module is the 2D optical flow which relies on the change of the spatio-temporal

[1] For more information about 3D motion parameters, refer to section 2.4.

Figure 1.1: An active vision system mounted on a mobile robot from our lab (GETbot).

information of pixels. Computing a single 3D motion from a 2D image flow by finding the optimal coefficient values in a 2D signal transform suffers from ambiguous interpretations concerning 3D motion especially motions in the z direction. On the other hand, one of the main challenges facing the segmentation of 3D multi-moving objects in an active vision system is to partition the MVF within a reasonable computation time. This especially proved to be difficult when moving objects are partially visible and are not spatially connected. Hence, it is important to detect, estimate, and segment the MVF independently from a predefined spatial coherence such as object contours generated from image segmentation approaches. Such methods are dependent on a group of features which could be affected by the continuous environment

change in a dynamic scene, e. g. the results of the color-based segmentation approaches could be affected by illumination changes. The following sections will explain in more details the mentioned challenges starting with the concept of transparent motion then the importance of predicting future collisions and the ego-motion.

1.2.1 Concept of Transparent Motion

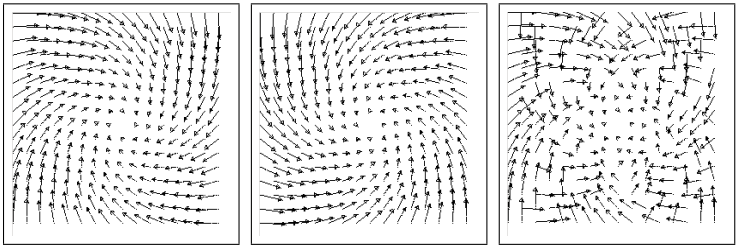
One of the fundamental processes in the computation of 3D motion is the grouping of velocity signals into surfaces (layers) as in the case of motion transparency [DDT$^+$06, SV99]. A special case of layered motion where visual motion is caused by the movement of a small number of objects at different depths in the scene is the transparent motion, which is usually caused by reflections seen in windows and picture frames. Natural images in general may contain reflected and transmitted light [SAA00] where local moving elements appear to be a superposition of two or more spatially overlapping layers when the camera is moving. Hence, the challenge for modeling 3D motion transparency is raised in order to demonstrate how two different motion signals can appear perceptually co-localized in the same space. Furthermore, the 3D motion parameters estimation process requires a multi-valued representation for each point in the image or the co-localization of more global surface descriptors as shown in fig. 1.2 which represents examples of overlapped 3D motions in life and fig. 1.3 where two synthetic 3D motion are group together to give the impression of lacy overlapping surfaces despite the connectivity of the object.

1.2.2 Prediction of Collision

The detection and avoidance of obstacles are very important for mobile robot navigation systems. Using visual sensors instead of strictly range-finding sensors has the advantage of providing a higher density of information. Objects that span a small region of pixels could theoretically be detected in an image, but would almost be missed by laser or sonar depending on the resolution of the range sensor. Some of

(a) (b)

Figure 1.2: Examples of overlapped 3D motions representing the concept of transparent motion. (a) Two swarms of starlings moving in the opposite direction of each other [Win]. (b) Pedestrians crossing the road in opposite directions [Miu].



(a) (b) (c)

Figure 1.3: Synthetic MVF representing the concept of transparent motion. (a) A 3D motion representing translation and rotation in the z axis. (b) A 3D motion represents the same translation in the direction of z axis with opposite rotation about the z axis. (c) Random combination of both MVFs representing the concept of transparent motion.

the vision-based collision detection approaches standing out from the ground the floor region of an image assuming that the floor remains consistently identifiable and consistent over the entire environment [YLC10] or continually adapt the robot's model of the floor [Kum09]. Other approaches rely on features instead of working on the pixel level to approximate real-world locations and trajectories of objects based on their varying location in a series of images assuming that objects are rigid [CG09] which is not always the case where non-rigid objects exists often in autonomous scenarios.

The principal problem of non-rigid objects motion analysis and collisions detection lies in the geometrical assumption of objects based on the segmentation of unreliable features such as color or intensity variations. Such assumption demand smoothing mechanisms to handle non-regular information which affects the estimated 3D motion parameters. The computation of motion characteristics such as velocity, acceleration, displacement vector, etc. is based on object edges or principal corners depends on the quality of the interpretation of object shape and the accuracy of the differential optical flow (more details are represented in section 2.3.3).

1.2.3 Ego-Motion

Self-localization is a key capability for autonomous mobile robots where hardware sensors such as joint encoders and accelerometers are generally used. The main drawback of such sensors are the limitation in certain environment, e. g. the wheels slips over wet ground which make the wheel odometry is unreliable. Hence, the use of visual sensors for motion estimation in such cases provides a better alternative. In order to estimate the ego-motion, the 3D motion parameters have to be estimated from the generated 2D optical flow assuming that there are no significant objects motion in the scene. Such assumption is valid in some applications such as aerial imagery when the ego-motion causes large displacements between consecutive frames [BJG10]. On the other hand, estimating the ego-motion for mobile robots based on the detection and tracking of extracted image features such as in [MOK$^+$10] may suffers from the aperture problem in low textured images. Furthermore, such features mus belongs to

static objects in order to correctly estimate the ego-motion. Otherwise, the computed ego-motion is highly distracted and in some cases it is completely wrong.

1.3 Motion Segmentation and Motion of Segments

Motion estimation has been developed as a major aspect of estimating the three dimensional nature and structure of a scene, as well as the 3D motion of objects and the observer relative to the scene. The generation of a motion vector fields is basically a correlation problem which tries to find the correspondence of a certain feature such as color or edges spatially between two or more consecutive frames. Hence, the generated MVF inherits the main drawbacks of the correlation process such as the ambiguity problem. As one way to overcome such a problem, some assumptions have been integrated to find a reasonable flow field estimate such as flow smoothness which explicitly forces neighboring pixels in the image to have a similar optical flow. Another way to deal with the problem is to group the neighboring pixels which are similar in a certain homogeneity criterion in one segment then estimate the motion of the whole segment by finding its corresponding segment in the other frame. However, the output of this technique contradicts the concept of transparent motion in case that the segmentation criterion is not taken into consideration the depth information of overlapping layered motions. Furthermore, the change of image features in a dynamic environment, e. g. by illumination change results in segmentation errors such as segments size which in turn lead to false estimation of the 3D motion parameters. On the other hand, some approaches segment the generated motion vectors into a set of 3D motions where motion parameters are used as a homogeneity criterion for the segmentation process despite the spatial-connectivity of the motion vectors. Table 1.1 shows a summarized comparison between motion segmentation and motion of segments approaches.

Table 1.1: Summarized comparison between the motion segmentation and motion of segments approaches

	Motion Segmentation	Motion of Segments
Input	Motion vector field (MVF)	Image segments
Output	Set of 3D motions	3D motion parameters for each segment
Advantages	Handle transparent motion Handle high noisy MVF	Fast computation Suitable for objects tracking approaches
Drawbacks	Computationally expensive Not suitable for object tracking	Very sensitive to the segmentation errors Prior information such as spatial coherence is required

1.4 Thesis Outline

In this thesis, we handle the 3D motion segmentation analysis in a new perspective: Biologically inspired motion recognition is involved to deal with spatially overlapped moving elements (fig. 1.4 represents the system architecture of the proposed 3D motion analysis for an active vision system). Hence, the challenge for modeling 3D motion transparency [DDT$^+$06] is raised in order to demonstrate how two different motion signals can appear perceptually co-localized in the same space. Furthermore, another challenge facing the segmentation of 3D multi-moving objects in an active vision system is the segmentation of an incoherent MVF into partitions in reasonable computation time. Therefore, it is important to detect, estimate, and segment the MVF independently from a predefined spatial coherence such as object contours generated from image segmentation approaches.

A general overview about the basic concepts related to the 3D motion analysis is presented in chapter 2. Meanwhile, a thorough review of the literature from different areas of knowledge involved in the work on this project is provided in chapter 3. An enhanced approach for estimating 3D motion parameter coefficients from the gener-

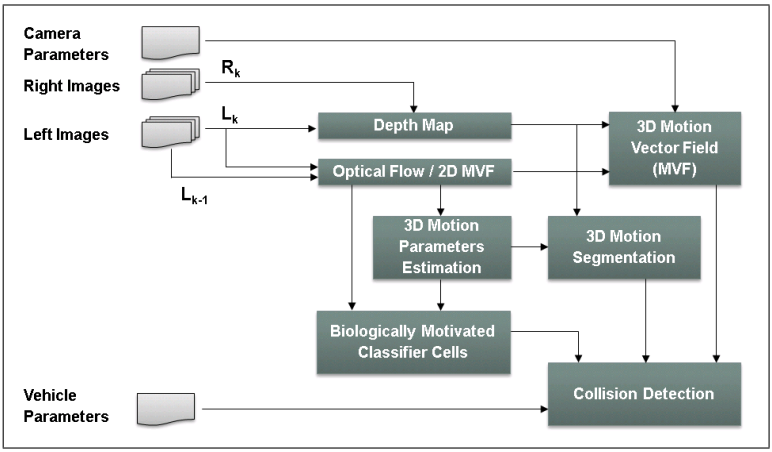


Figure 1.4: System architecture of the proposed 3D motion analysis for an active vision system

ated MVFs is presented in chapter 4 which successfully overcomes the drawback of Daugman's transform [Dau88] of finding the derivative of the error of an estimated parameter with respect to each of the 3D parameter coefficients. A 3D saliency-based motion segmentation approach is explained in chapter 5 while chapter 6 represents the 3D depth-integrated motion estimation and visualization approach. The results of experiments carried out using the developed approaches under different dynamic scenarios are presented in chapter 7 while chapter 8 summarizes the achievements and indicates the issues that requires further work in this direction of research.

2 Basic Concepts

2.1 Visual Motion Estimation

Mainly, there are three classes of visual motion estimation algorithms: gradient based techniques which operate using image derivatives, frequency domain techniques which analyze the image sequence in the frequency domain and token based techniques which track some image tokens between frames. All these techniques share the principle of utilizing the brightness constancy assumption (BCA). They assume that the intensity of light reflected from a point on an object does not change over time so that all changes in the image intensity pattern are due to motion. Thus, before considering specific techniques, it is necessary to analyze the brightness constancy assumption.

2.1.1 Brightness Constancy

The light intensity (brightness) captured by a camera at a particular pixel is generally proportional to the amount of light reflected from the corresponding point in the environment. The amount of reflected light depends on the reflectance property of the surface and the prevailing illumination. Meanwhile, the brightness constancy assumption requires fixed illumination or reflectance otherwise it will fail [KV05]. The brightness of a static object caused by a diffuse light source remains stable, otherwise it will be changed. On the other hand, the movement of the camera will cause the brightness of a point to be changed except the rotation about the lens axis. Furthermore, in case that a shadow of an object lies on another object in a dynamic environment, it will cause distraction for the motion estimation algorithms.

2.1.2 Gradient Based Motion Estimation

As represented in [KV05], the brightness constancy assumption is the first step to estimate motion based on the gradient:

$$\frac{dI(x,y,t)}{dt} = 0 \tag{2.1}$$

where $I(x,y,t)$ is the spatio-temporal image intensity function. Using the chain rule for differentiation we obtain the total derivative:

$$v_x I_x + v_y I_y + I_t = 0 \tag{2.2}$$

where (v_x, v_y) is generated optical flow representing temporal derivative of position. The equation could be rewritten without the coordinates and subscripts indicate the partial derivatives with respect to the subscript variable.

$$\nabla I \cdot \boldsymbol{v} + I_t = 0 \tag{2.3}$$

where $\nabla I = (I_x, I_y)$ is the spatial intensity gradient, $\boldsymbol{v} = (v_x, v_y)^T$ is the image velocity or optical flow at pixel (x,y) at time t and I_t is the temporal intensity derivative (more details are represented in section 2.3.1 and 2.3.2). The goal of the gradient-based optical flow is to find the velocities that minimize the square of this constraints. Such constraint are important for the relation between the optical flow and the intensity derivatives where the velocities are constrained to belong to a parallel line to the intensity gradient. In order to obtain a unique solution for the motion at a point, further constraints must be applied as described later in section 2.3.

2.1.3 Background Subtraction and Surveillance

As one of the main applications that benefits from the detection and estimation of motion is surveillance. In general, a surveillance camera is stationary and the detection

process uses simple image difference techniques. However, such an approach doesn't provide velocity information and error prone if the illumination change rapidly. The following section will highlight in more detail the detection of moving objects using a static camera.

2.2 Motion Detection from a Static Camera

The basic concept behind motion detection is to follow image differences from frame to frame in an image sequence in order to discriminate the background from moving foreground objects. The approach used in this regard examines each pixel of an image if it corresponds to a moving object by a relaxed threshold image difference approach where the background model B_t has to be updated with each image frame I_t to handle the illumination variation [MZK01]:

$$B_{t+1} = \alpha I_t + (1 - \alpha) B_t \tag{2.4}$$

where α regulates the dependency of the background model to the illumination variation and usually is kept small otherwise moving objects will have artificial "tails" behind them. The detection process is applied to two consecutive frames from an input image sequence. The result is a binary image that shows the spatial position of changed pixels values. Fig. 2.1 represents an image sequence from PETS dataset [PET] and the binary result of motion detection.

2.2.1 Region-Based Motion Detection

In order to segment an image into a list of regions or labels using region-based methods, neighboring pixels of initial seed points has to be validated according to a certain criteria. Once a neighboring pixel has been detected and labeled as the new initial seed pixel, the validation process continue to the neighboring pixels as well. As the

(a) (b)

Figure 2.1: Result of motion detection on a sequence of real images. (a) Input sequence from PETS Data set [PET]. (b) Resulting binary image of motion detection between two consecutive frames.

validation process iteratively continue to examine all unallocated neighboring pixels, the segment size is increased. The region is growing until there are no more valid neighboring pixels within a search window w to be included to the region. On the other hand, increasing the size of the search window w will lead to including more valid pixels in the neighborhood as shown in fig. 2.2. The process is iterated on, in the same manner as general data clustering algorithms. Fig. 2.3 represents the result of segmenting the detected motion using the region growing algorithm applied to an image sequence (PETS dataset) [PET] . Segmenting the image using a region-based segmentation algorithm such as region growing may suffer from over-segmentation. Hence, its better to use boundary segmentation models such as “Snake Active Contour” or “Geodesic Active Contour” specially when the purpose of the segmentation is tracking moving objects in a sequence of images (more details are represented in the following section).

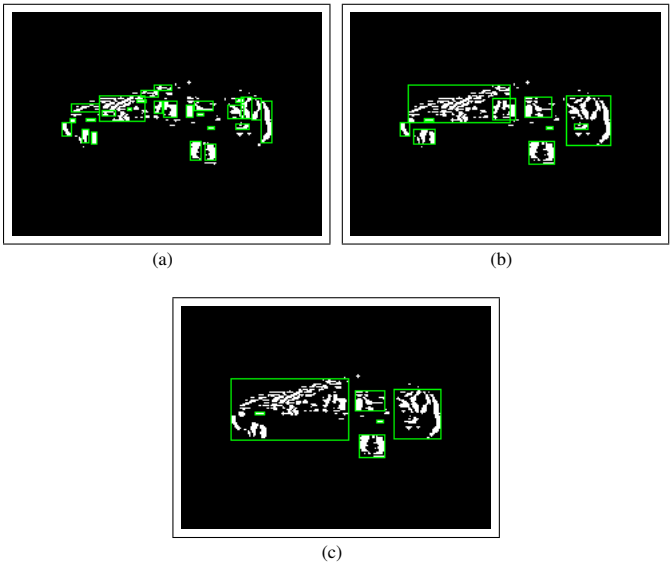


Figure 2.2: Segmentation using a region growing algorithm. (a) With a search window size of $w := 3 \times 3$. (b) $w := 5 \times 5$. (c) $w := 7 \times 7$.

2.2.2 Contour-Based Motion Detection

Active contours or (snakes) are an image segmentation and object boundary detection approach that minimizes the energy of a contour. The energy function consists of internal and external forces [CKS97, NTA06, GME10]. The external force drives the contour nodes towards the inside of the contour until it reaches an object boundary where the external energy supposed to be minimal. Hence, the snake contour is bent and shaped according to the object boundary. The snake model starts from an energy function integrated along a curve $C(pn) = \{x(pn), y(pn)\}$, where the curve nodes $pn \in [0, 1]$. The energy function includes an internal and external term [LGP+02].

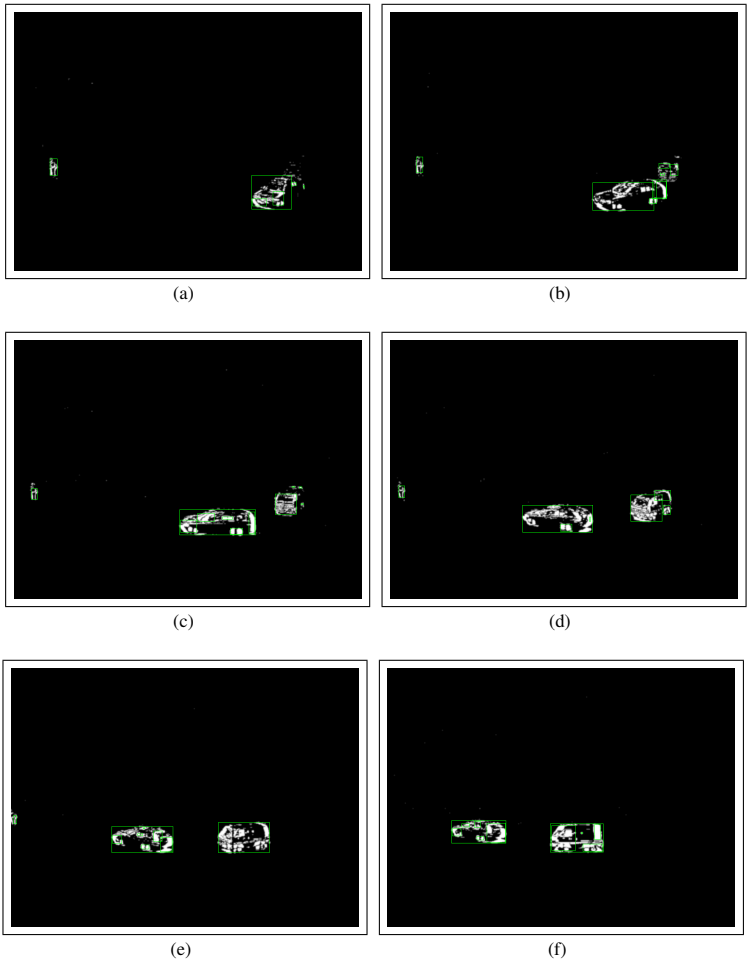


Figure 2.3: Result of segmenting the detected motion from an image sequence (PETS dataset) with a search window size $w := 13 \times 13$. (a-f) Results of the region growing segmentation algorithm after an interval of 24 frames each.

$$E = \int_0^1 [\alpha \cdot E_{int}(C(pn)) + \beta \cdot E_{ext}(C(pn))]\, dp \tag{2.5}$$

where E_{int} is the internal energy which represents physical properties of the contour, while E_{ext} is the applied external energy which relates to the image data (e. g. intensity). The influence of both energies are regulated by α and β. On the other hand, internal and external energies could be defined as combination of other energies, e. g. the elastic energy represents the internal energy.

$$E_{elastic} = K_1 \sum_{i=1}^{N} (L(i, i-1))^2 \tag{2.6}$$

where N is the number of contour points, $L(i, i-1)$ is the distance between two contour points and K_1 is a regulation parameter for the applied forces on the contour point. Hence, the applied forces on both x and y directions are defined as:

$$\begin{aligned} F_{elastic}(x_i) &= 2K_1((x_{i-1} - x_i) + (x_{i+1} - x_i)) \\ F_{elastic}(y_i) &= 2K_1((y_{i-1} - y_i) + (y_{i+1} - y_i)) \end{aligned} \tag{2.7}$$

The contour points loose their energy once they detect an edge point. As the external force drives the snake to shrink and move towards object boundary as shown in fig. 2.4.

One of the main advantages of the adaptive active contour models over the region based segmentation is taking into consideration the spatial relation between the segment size and nearby segments, i. e., the proximity feature of the Gestalt principles. As an example, if two relatively large objects are separated by a certain distance they will be grouped by one contour. On the other hand, if the same distance are used to separate small objects they will be segmented. Fig 2.5 represents the evolution of an active contour over two small and large segments separated from each other with the same distance, while fig. 2.6 represents the result of segmenting the detected motion from an image sequence (PETS dataset) region growing algorithm.

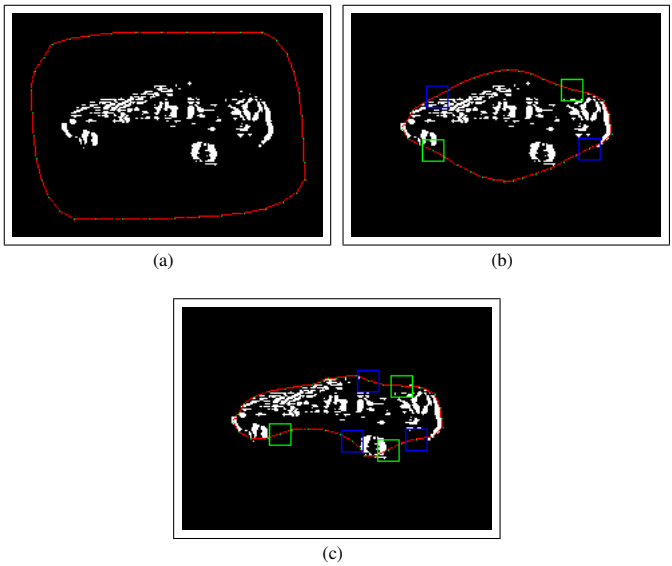


Figure 2.4: Evolve of the snake active contour algorithm over a part of the detected motion from the "PETS dataset". (a) Initial position of the snake where the green dots represents the contour nodes and connected by red lines. (b) The first and the last node in a contour segment that reaches an object boundary is highlighted by a green and a blue square respectively. (c) The snake in its final state.

However, the evolution of the active contour could be stopped by a local minima. Such a problem could be solved at the prize of computational time by using simulated annealing approaches.

2.3 2D Motion Vector Fields

Many recent robotics applications are based on the estimation of the optical flow such as object tracking, 3D scene structure and visual odometry [LK81]. In this section the

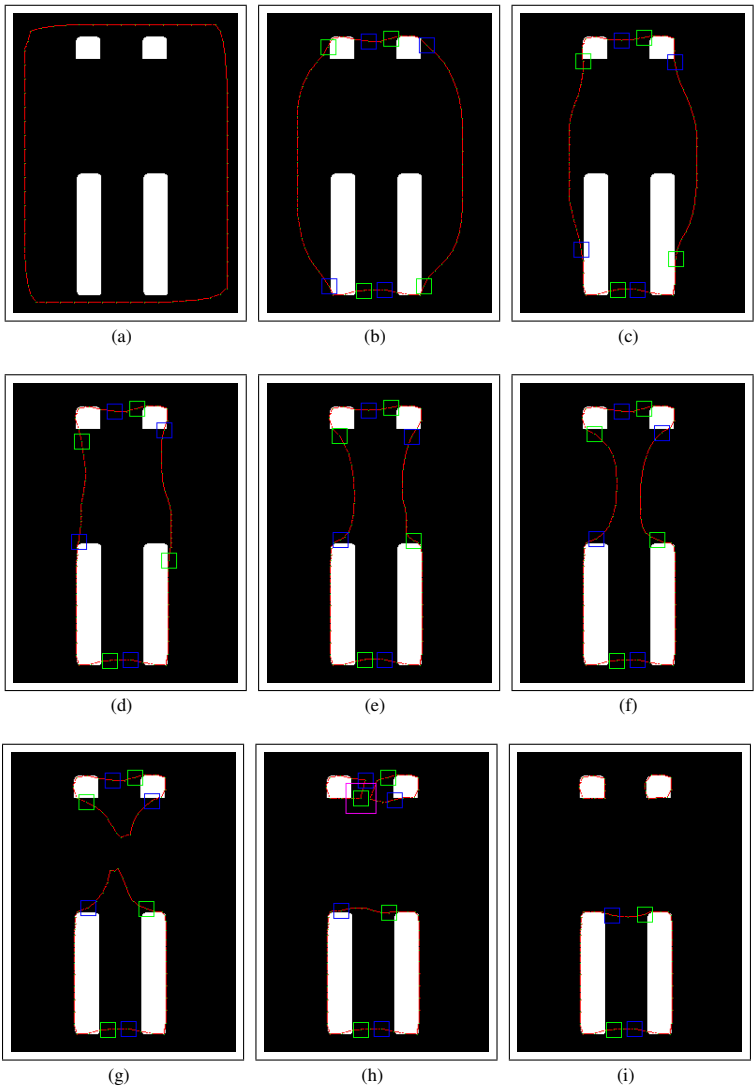


Figure 2.5: Results of the snake active contour algorithm over two small and two large segments separated from each other with the same distance.

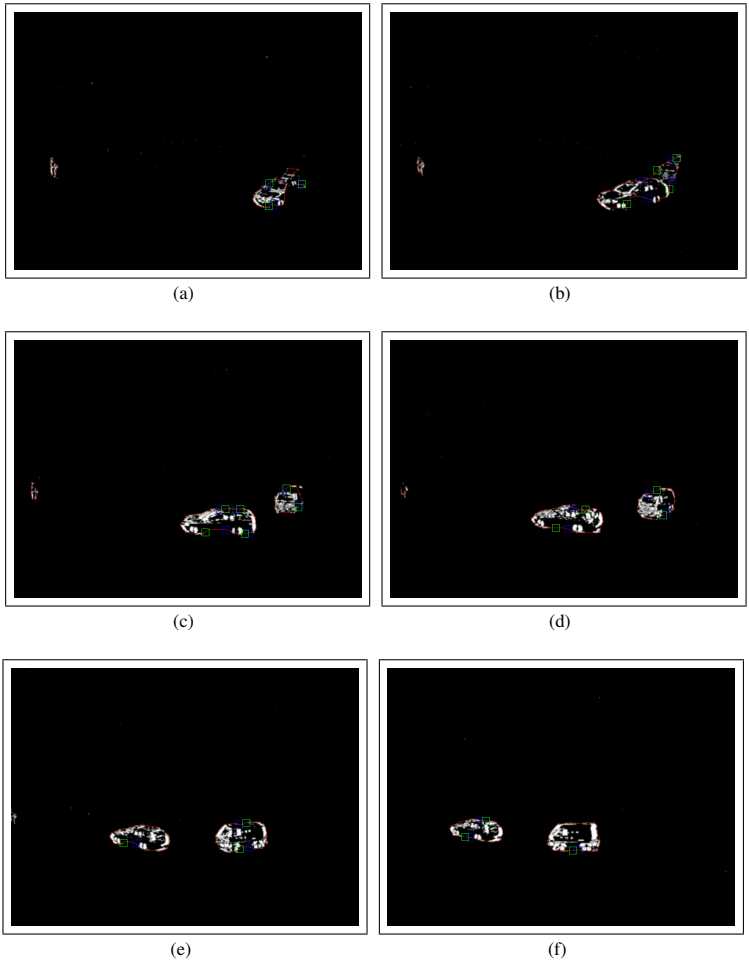


Figure 2.6: Result of segmenting the detected motion from an image sequence (PETS dataset). (a-f) Results of the adaptive active contour segmentation algorithm.

2D and 3D motion constraint equation (as presented in [BT05]) will be introduced. The main difference between the 2D and the 3D optical flow is that in 2D it measures the motion of a pixel between adjacent images while in 3D it measures the motion of the volume voxel between adjacent volumes. Furthermore, both 2D and 3D motions cause temporal changes in image intensity assuming that there are no other reasons. In general, this assumption is usually true but there are many exceptions. The motion constraint equation are the basis of the differential optical flow as explained in the following subsections.

2.3.1 2D Motion Constraint Equation

The 2D motion constraint equation which could be interpreted as the gradient based motion estimation is based on the brightness constancy assumption represented in section 2.1.2.

The pixel point $I(x, y, t)$ is moving spatially by $\delta x, \delta y$ in a time interval δt to $I(x + \delta x, y + \delta y, t + \delta t)$. Hence, $I(x, y, t)$ and $I(x + \delta x, y + \delta y, t + \delta t)$ holds the same intensity information:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t) \tag{2.8}$$

This assumption is true for small local translations assuming that $\delta x, \delta y, \delta t$ are not too big. Thus, the first order of Taylor series expansion can be performed for $I(x, y, t)$ in equation (2.8) to obtain:

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t + \xi \tag{2.9}$$

where ξ are the higher order terms, which could be ignored and removed. Using eq. no. (2.8) and (2.8) we have:

$$\frac{\partial I}{\partial x}v_x + \frac{\partial I}{\partial y}v_y + \frac{\partial I}{\partial t} = 0 \tag{2.10}$$

where $v_x = \frac{\delta x}{\delta t}$ and $v_y = \frac{\delta y}{\delta t}$ are the x and y components of image velocity (optical flow) and $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ and $\frac{\partial I}{\partial t}$ are image intensity derivatives at (x, y, t) which could be written as:

$$I_x = \frac{\partial I}{\partial x}, \;\; I_y = \frac{\partial I}{\partial y} \;\; and \;\; I_t = \frac{\partial I}{\partial t} \tag{2.11}$$

The relation between the x and y components of the optical flow (v_x, v_y) and the intensity derivatives (I_x, I_y, I_t) are:

$$(I_x, I_y) \cdot (v_x, v_y)^T = -I_t \tag{2.12}$$

which is often presented in dot product form.

$$\nabla I \cdot \boldsymbol{v} = -I_t \tag{2.13}$$

where $\nabla I = (I_x, I_y)$ is the spatial intensity gradient and $\boldsymbol{v} = (v_x, v_y)^T$ is the image velocity or optical flow at pixel (x, y) at time t. Eq. 2.13 is the same as the gradient based motion estimation eq. 2.3. The 2D motion constraint equation $\nabla I \cdot \boldsymbol{v} = -I_t$ has two unknowns which resulted from the aperture problem. The aperture problem appears when local image intensity structure is not sufficient to measure full image velocity.

2.3.2 3D Motion Constraint Equation

Similar to the 2D motion constraint equation, it is possible to use the first order Taylor series expansion. since $I(X, Y, Z, t) = I(X + \delta X, Y + \delta Y, Z + \delta Z, t + \delta t)$ [1] it is clear that:

$$I_X V_X + I_Y V_Y + I_Z V_Z + I_t = 0 \tag{2.14}$$

where V_X, V_Y, V_Z are the 3D optical flow components and I_X, I_Y, I_Z and I_t are the 3D spatio-temporal derivatives. Equation 2.14 could also be rewritten as:

$$\nabla_3 I \cdot \boldsymbol{V} = -I_t \tag{2.15}$$

where $\nabla_3 I = (I_X, I_Y, I_Z)$ is the 3D spatial intensity gradient, I_t is the temporal intensity derivative and $\boldsymbol{V} = (V_X, V_Y, V_Z)^T$ is the 3D velocity (see e. g. [BT05] for more details).

2.3.3 2D Optical Flow

After measuring the spatio-temporal intensity derivatives, all velocities normal to the local intensity structures are integrated into full velocities using least squares method in local techniques and regularization in the global approaches. On the other hand, it is assumed that all objects are rigid and there are no specularities in the scene. Under this assumptions, the 2D optical flow is representing an approximation to the ideal projection of 3D motion on an image. The projected velocity $\boldsymbol{V}$ of a 3D point $\boldsymbol{P}$ on a spatio-temporal path $\boldsymbol{K}(t)$ could be written as:

$$\boldsymbol{V} = \frac{d\boldsymbol{K}(t)}{dt} = \left(\frac{dX(t)}{dt}, \frac{dY(t)}{dt}, \frac{dZ(t)}{dt} \right)^T \tag{2.16}$$

[1] To be consistent with literature, capital letters X, Y and Z are representing 3D coordinates.

The projected 2D point $\boldsymbol{p}(t)$ on the image plan with a focal length f in a standard perspective projections as shown in fig. 2.7 will be:

$$\boldsymbol{p}(t) = (x(t), y(t))^T = \left(\frac{fX(t)}{Z(t)}, \frac{fY(t)}{Z(t)} \right)^T \tag{2.17}$$

the instantaneous 2D velocity $\boldsymbol{v}$ is:

$$\begin{aligned} \boldsymbol{v}(t) &= \left(\frac{dx(t)}{dt}, \frac{dy(t)}{dt} \right)^T \\ &= \frac{f}{Z(t)} \left(\frac{dX(t)}{dt}, \frac{dY(t)}{dt} \right)^T - \frac{f dZ(t)}{Z^2(t) dt} (X(t), Y(t))^T \end{aligned} \tag{2.18}$$

for more details about the perspective projection refer to section 6.2.

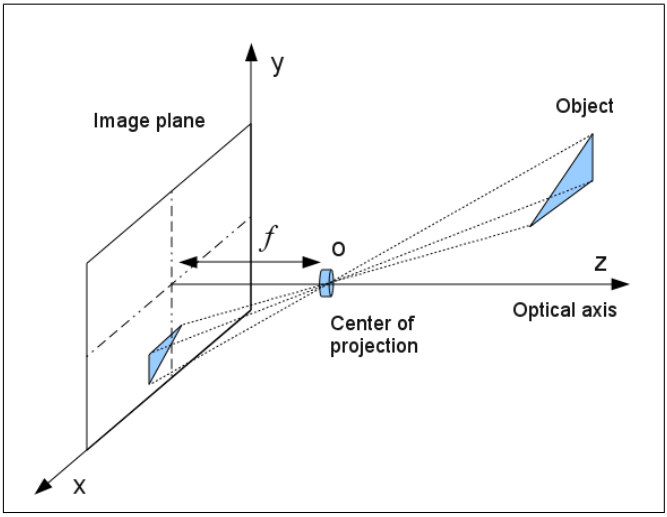


Figure 2.7: Standard perspective projection.

2.4 3D Motion Interpretation

3D motion interpretation of an image flow has become an important problem in computer vision. Early publications such as [FH84] discussed the estimation of general 3D motion parameters of a rigid body from two or more consecutive image frames. Longuet-Higgins and Prazdny [LHP80] introduced equations for computing the 3D egomotion in stable scenes. They suggest that the 3D motion interpretation problem is a matter of solving a system of equations for six motion parameters. A linear optimization approach has been introduced in [Adi85] with an assumption that the optical flow is accurately available.

In the case of interpreting an optical flow, the elementary signals are 2D vector fields of infinitesimal generators of a 3D Euclidean group. The infinitesimal motion of a rigid body, i. e., a 3D vector field can be expressed as a linear combination of six component 3D vector fields. The computation of a 3D motion from a 2D image flow or a motion template finds the optimal coefficient values in a 2D signal transform. The ideal optical motion $\boldsymbol{v}_{opt}$ caused by a motion of a point (x, y, dp) on a rigid visible surface with a distance from the origin $dp = \rho(x, y)$, is

$$\boldsymbol{v}_{opt}(x, y) = \sum_{i=1}^{6} c_i \boldsymbol{e}_i(x, y) \tag{2.19}$$

where $\rho(x, y) > 1$ is a positive function defined on the image plane and $\boldsymbol{e}_i(x, y)$ represents the six infinitesimal generators in form of 2D vector fields [TSL$^+$91,MJM02]. For translation:

$$
\begin{aligned}
\boldsymbol{e}_1(x,y) &= \begin{pmatrix} \rho^{-1}(x,y)\sqrt{1+x^2+y^2} \\ 0 \end{pmatrix} \\
\boldsymbol{e}_2(x,y) &= \begin{pmatrix} 0 \\ \rho^{-1}(x,y)\sqrt{1+x^2+y^2} \end{pmatrix} \\
\boldsymbol{e}_3(x,y) &= \begin{pmatrix} -x\rho^{-1}(x,y)\sqrt{1+x^2+y^2} \\ -y\rho^{-1}(x,y)\sqrt{1+x^2+y^2} \end{pmatrix}
\end{aligned}
\tag{2.20}
$$

and for rotation :

$$
\begin{aligned}
\boldsymbol{e}_4(x,y) &= \begin{pmatrix} -xy \\ 1+y^2 \end{pmatrix} \\
\boldsymbol{e}_5(x,y) &= \begin{pmatrix} 1+x^2 \\ xy \end{pmatrix} \\
\boldsymbol{e}_6(x,y) &= \begin{pmatrix} -y \\ x \end{pmatrix}
\end{aligned}
\tag{2.21}
$$

After the projection of the partial velocities to the image plane (using pinhole-camera mapping), six motion templates will be obtained depending on the object-depth Z, image position (x, y) and the camera focus f. By setting the unknown depth to $Z = 1$ and the unknown focal length to $f = 1$, we can establish relative velocity estimations which yields the templates depicted in Fig. 3.1.

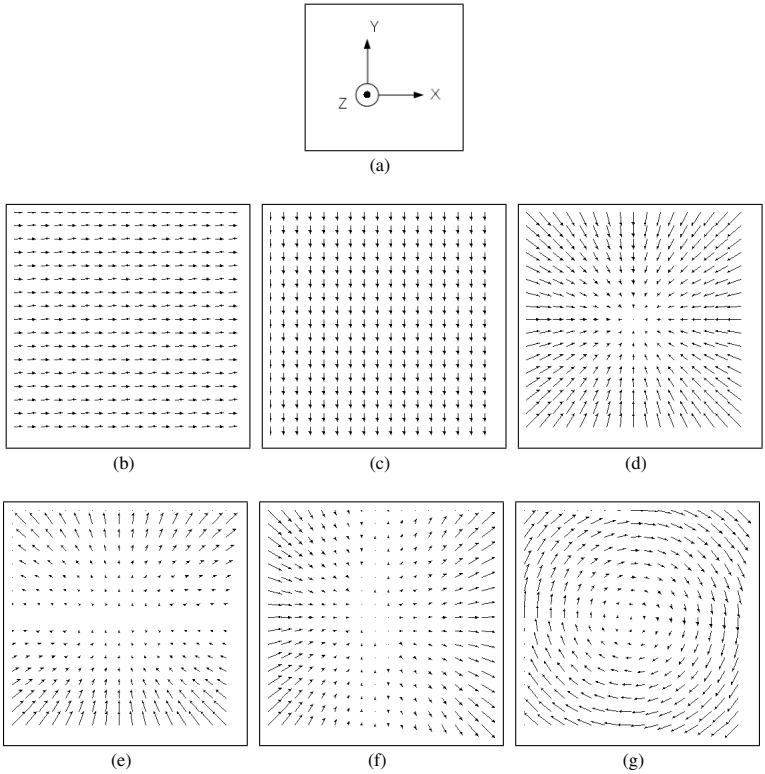


Figure 2.8: Motion templates for the translation and rotation obtained from the projection of the instantaneous velocities of the motion model to the image plane. (a) The coordinate system. (b-d) Translation in the X, Y, Z axes respectively. (e-g) Rotation around the X, Y, Z axes respectively.

3 Related Literature

3.1 Introduction

Over the last decade, robotics has advanced rapidly into applications which require increasing dexterity and dynamic response. The increased interest in the field of motion segmentation has lead to expanding its areas of application to include e. g. object tracking [WS02, HKW08], activity surveillance [AWK+05, MCK09], image and video compression [KA02, LZL+07], and object recognition [Hun05, TMD09].

The remainder of this chapter is organized as follows: section 2 introduce the concept of the biologically motivated classifier cells, while section 3 and 4 gives an account of the motion segmentation approaches based on dense optical flow and 3D shape construction respectively. Finally, section 5 summarizes the chapter.

3.2 Biologically Motivated Classifier

Detecting and estimating 3D motions in an image sequence generally requires a bottom-up approach which is very useful in the context of exploration for autonomous mobile robots. On the other hand, some scenarios especially in dangerous environment requires a top-down approach for the detection and estimation of a certain 3D motion which could represent a possible collision e. g. the movement of a pedestrian in the direction of a car. Therefore, we present in this section a biologically inspired top-down approach for the detection and estimation of a specific 3D motion. In neurophysiology, neurons in the medial superior temporal cortex (MST) in the mammalians brain are sensitive to global patterns of 3D motion such as rotation, translation and expansion [How12]. In this module, models of motion-sensitive cells for the preferential direction will be constructed in order to measure the response of

the sensitive cells to a corresponding motion. The generation of the corresponding map requires calculating a radial symmetric weight function for each cell which is in term of computation time very expensive. Developing a fast connection weight function based on the depth information reduce the time consumption dramatically without the use of a GPU power as in [WRC08] where a biologically motivated classifier and feature descriptors are designed for execution on single instruction multi data hardware using the programmable GPU.

In [MJM02], a model neuron for the detection of motion templates named *c-cell* was introduced. The *c-cell* activation function represents the instantaneous velocity in the preferential direction of the cell. The motion vector $\boldsymbol{v}(x, y)$ at a point (x, y) is computed by a motion parameters c_i and a motion template $\boldsymbol{e}_i^L$ which represents the six infinitesimal generators of a 2D vector field (translation in X, Y, Z and rotation about X, Y, Z axis).

$$\boldsymbol{v}(x, y) = c_i \boldsymbol{e}_i^L(x, y) \tag{3.1}$$

solving the previous equation for the motion parameter c_i:

$$c_i(x, y) = \frac{\boldsymbol{v}^T(x, y) \cdot \boldsymbol{e}_i^L(x, y)}{|\boldsymbol{e}_i^L|^2} \tag{3.2}$$

the *c-cell* of point $\boldsymbol{p}_0$ is defined by:

$$c_i(\boldsymbol{p}_0) = \frac{1}{|\boldsymbol{S}|} \sum_{\boldsymbol{p} \in \boldsymbol{S}} \boldsymbol{v}^T(\boldsymbol{p}) \frac{\omega(\boldsymbol{p}_0 - \boldsymbol{p}) \boldsymbol{e}_i^L(\boldsymbol{p})}{|\boldsymbol{e}_i^L|^2} \tag{3.3}$$

where $\boldsymbol{S}$ is a concatenated vector of the detected MVs in the image and $\boldsymbol{p} := (x, y), \boldsymbol{p}_0 := (x_0, y_0)$. A radial symmetric weight function $\omega(\boldsymbol{p}) = \frac{1}{2\pi\sigma^2} exp(-\frac{1}{2} \frac{|\boldsymbol{p}|^2}{\sigma^2})$ is used to create a local neighborhood around $\boldsymbol{p}_0$ and separate the multi-object motions based on the size of the receptive field and the σ parameter.

Similar to the *c-cell*, [MJM02] introduced an activation function named ξ-*cell* to measure how well is the correspondence of a *c-cell* to a preferential motion:

$$\xi_i(\boldsymbol{p}_0) = \frac{1}{|\boldsymbol{S}|} \sum_{\boldsymbol{p} \in \boldsymbol{S}} |\varpi(\boldsymbol{p}_0 - \boldsymbol{p})(\boldsymbol{v}(\boldsymbol{p}) - c_i(\boldsymbol{p}_0)\boldsymbol{e}_i^L(\boldsymbol{p}))| \tag{3.4}$$

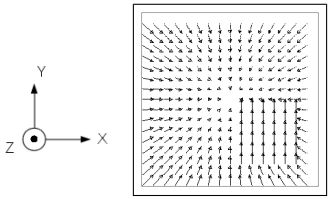
where $c_i(\boldsymbol{p}_0)$ is the estimated *c-cell*, and $\varpi(\boldsymbol{p}_0 - \boldsymbol{p})$ is the updated connection weight function between a point $\boldsymbol{p}_0$ and its neighbor point $\boldsymbol{p}$. Fig. 3.1 shows the response measurement of the sensitive cells tuned to six motion templates $\boldsymbol{e}_i^L(\boldsymbol{p})$ to a corresponding synthetic motion.

We used the depth information generated from a stereo algorithm to enhance the connection weight function $\varpi(\boldsymbol{p})$:

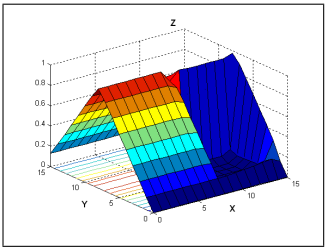
$$\varpi(\boldsymbol{p}_0 - \boldsymbol{p}) = \begin{cases} 1 & \forall \ \boldsymbol{p} \in \Im(\boldsymbol{p}_0) \\ 0 & otherwise \end{cases} \tag{3.5}$$

where $\Im(\boldsymbol{p}_0)$ is the segment label of point $\boldsymbol{p}_0$. The new connection weight function enhances the overall computation time as well as it overcomes the blurring effect of the $\omega(\boldsymbol{p})$ function especially at the edges of an object. Moreover, considering only the points that belong to the same depth level improves the estimation process overcoming the ambiguous interpretation problem.

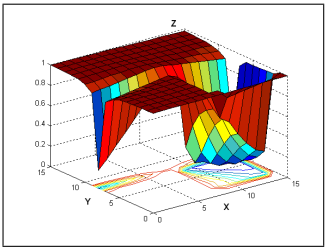
This approach allows the separation of the MVF into arbitrarily predefined motion channels with the *c-cells* encoding the velocity and the ξ-*cells* the error for each channel. As MVFs generally are ambiguous without additional information, the interpretation of a motion can deviate from the actual object motion. Therefore, the solution presented here benefits from the computation of non-exclusive interpretations which preserve as much information as possible for higher-level components within a more complex system design.



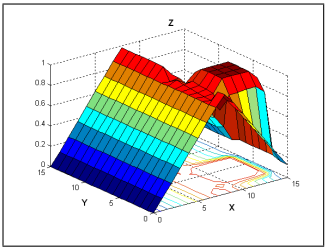
(a) (b)



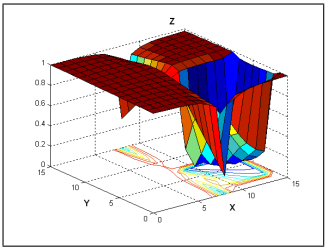
(c) Translation in X



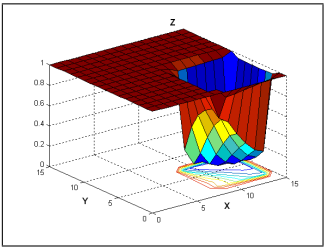
(d) Rotation about X



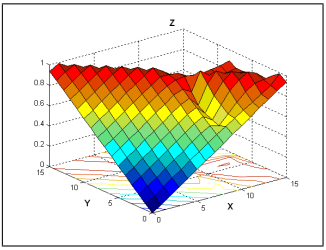
(e) Translation in Y



(f) Rotation about Y



(g) Translation in Z



(h) Rotation about Z

Figure 3.1: Response measurement of the sensitive cells adapted from [MJM02]. (a) Coordinate system. (b) Input MVF. (c-h) The precision $1 - \xi_i(\boldsymbol{p}_0)$ describing how well each location fits the corresponding motion template of cell c_i.

3.3 Motion Segmentation Based on Dense Optical Flow

The majority of motion segmentation approaches are generally based on estimating dense optical flow. The optical flow field was assumed to be piecewise smooth to account for discontinuities caused by occlusion and object boundaries [BJ96, OB98], or separate the image flow into different regions by looking for flow discontinuities [BA91]. Unfortunately, the lack of precision across edges of the most popular motion estimation methods makes them less useful for recovering the exact shape of moving objects. This section gives an account of different approaches to segment a dense optical flow, starting with the expectation maximization technique, then the multi-body factorization algorithm and finally with the random sample and consensus approach.

3.3.1 Expectation Maximization Approaches

For many estimation problems, the expectation-maximization algorithm (EM) is used. [Wei97a] introduced a short tutorial to describe the expectation (E) and the maximization (M) step used in motion segmentation as shown in the rest of this section. In order to segment a set of data points such as two lines that were generated by multiple processes using the EM algorithm, the two lines parameters and the assignment of the points to the correct generating process have to be estimated. The basic structure of an EM algorithm starts with random parameter values for two input models, then iterates until parameter values converge. In the expectation step, points are assigned to the model that fits it best. While in the maximization step, the parameters of the models are updated using points assigned to it.

Motion Segmentation using EM algorithm

Some of motion segmentation approaches based on dense optical flow represent the motion vector field in layers [AS95, DP91]. While the main target is to compute the motion parameters for each layer, each pixel has to assign to the correct layer first.

Using regularized radial basis functions (RBFs) [Wei97b] improved the overlapping layers approach to utilize flexible motion fields. The layered representation methods often use expectation-maximization (EM) techniques [JF01, RR97]. Integrating information of large areas in an image in the EM motion segmentation approaches enhance there robustness. On the other hand, an optimum results depends on good initialization [TSA01, FAH+08]. Some approaches enhance the initialization of the EM algorithm by obtaining the 2D motion parameters using K-means [WA93] or normalized cuts [SM98]. However, such techniques are suffering from the aperture problem.

Some approaches such as [SHP08] use hierarchical clustering of Hidden Markov Models (HMMs) for learning motion behavior in order to detected abnormal behavior in input image sequences. The implemented track clustering algorithm uses an agglomerative HMM clustering technique within the expectation maximization (EM) approach to determine the HMM parameters. However, in order to compute the optimal number of states and to estimate the parameters in each HMM, some assumptions about the data have to be available. Recent work such as in [MAM11] introduced a new estimator called generalized projection based M-estimator (gpbM). The estimator determine each inlier structure iteratively to estimate multiple heteroscedastic inlier structures. However, the inline structure assumes the moving points belongs to a rigid object. Hence, the result of motion segmentation will be affected severely if transparent motion exists i. e., overlapped 3D motions.

3.3.2 Multi-Body Factorization Approaches

Since Tomasi and Kanade (1992) [Tom92] introduced a factorization technique based on orthographic projection to recover structure from motion using features tracked through a sequence of images. Factorization methods have become very popular due to their simplicity. [WW11] introduces a short introduction for the multi-body factorization approach. In dynamic environment where many objects move simultaneously, motion features of different objects could be extracted and sorted according

to the object in a tracking matrix. As the motion features stored in the tracking matrix belongs to different objects, it is required to segment the objects based on the observation of the tracking matrix where interaction between features is measured. The main drawback in [WW11] is the sensitivity of noises. As a solution to the mentioned problem [CKI97] minimizes the total energy of the shape interaction matrix iteratively and [Kan01] integrates model selection and least-median fitting using dimension correction for the segmentation process. Despite the fact that this method gives the 3D structure of the object and the motion of the camera, it assumes that the features belong to the same object i. e., it does not perform segmentation. It can deal only with a single rigid object and it is very sensitive to noise.

Many approaches have been proposed in the field of motion segmentation following the same idea of forcing the rank constraint. These methods are based on using the dimensionality of the subspace in which the image trajectories lie to perform the motion segmentation [KK01, MZMI02, VS03]. The problem is solved using subspace constraints on an input matrix containing the location of a number of points in many frames. They use algebraic factorization techniques to calculate the segmentation of the points into objects in addition to the objects' motion and their 3D structure. Multi-body factorization algorithms use the full temporal trajectory of every point, and therefore, as a main advantage, are capable of segmenting objects whose motions cannot be distinguished using only two frames [GW04]. However, in terms of computation speed, their performance is still far from satisfactory.

3.3.3 RANSAC Based Approaches

The main advantage of the RANSAC algorithm (Random Sample And Consensus) [FB81] is the ability to estimate model parameters in the presence of high number of outliers (noises) which increases its robustness. The input data set to the RANSAC approach assumed to be defined by a parameterized model. The algorithm starts by iteratively selecting a group of the input data set randomly and then validates the hypothesis that those data are inliers and representing the required model. Once the parameters of the fitted model are estimated, the rest of the data set are examined

by the model. When a data point fits the model i. e. the fitting error is less that a predefined threshold, it is added to the inliers (consensus set). Afterward, the model is again estimated from all the inliers and then fitting errors will be estimated to evaluate the model. The process is repeated iteratively and the iterations number could be either fixed or computed [Der10].

RANSAC based motion segmentation approaches such as [MMI06] solve the 3D motion segmentation problem by successive computation of dominant motions using methods from robust statistics. These methods fit a single motion model to all the image measurements using random sample consensus (RANSAC) [FB81]. During the iterative process, the correct estimated measurements of the motion model are removed from the data set and RANSAC is re-applied to the remaining points to obtain a second motion model.

In a comparison of 3D motion segmentation algorithms for affine models [TV07] using a benchmark of 155 motion sequences, a Local Subspace Affinity (LSA) algorithm [YP06] introduced as a general framework for motion segmentation of feature trajectories, has generally shown a better performance than its competitors (The Generalized Principal Component Analysis (GPCA) [VH04], the Multi-Stage Learning (MSL) [SK04], and the RANSAC algorithm). The framework of [YP06] presents the segmentation problem as a linear manifold finding solutions under affine projections. However, the algorithm is robust only in cases where the outliers are not dominant in number. As a solution for the main drawback of the RANSAC approaches where only one model for a particular data set could be processed, [JC10] used RANSAC to process only small set of correspondences in a post processing step to a mixture of Dirichlet process (MDP) in their motion segmentation approach. However, the problem of degenerated (dependent) motion is not addressed and may fail in finding overlapping multi-model data set as in the case of transparent motion.

3.4 Motion Analysis Based on 3D Shape Construction

Recently, many works concentrates on studying the geometry of dynamic scenes by modeling dynamic real world 3D objects [RBW07, YW09] where the projected surface of a 3D object model and the data of a previously estimated 3D pose are used to construct 3D shapes to be integrated in the segmentation process. The constructed 3D model are used to estimate the rigid motion of objects by determining the 3D pose of the objects. Estimating the 3D pose of objects depends on the accuracy of fitting the extracted features from the 3D model such as the projected object surface and the corresponding 2D object contour in the image .

3.4.1 3D Pose Estimation

In [HKW08] a spatio temporal model for estimating 3D poses using a trinocular camera sensor has been proposed. The algorithm avoids typical delays in the filtration of pose estimation process by providing the derivative of the temporal pose instantaneously. However, initializing the parameters of the model is still required. [GRS06, HRT+09] suggested a texture model based method for 3D pose estimation where the influence of the features is automatically adapted during tracking while local descriptors and contours are used for the matching process. This approach has shown its ability to deal with a rich textured and non-static background as it has shown robustness to shadows, occlusions, and noise in general situations overcoming the drawbacks of the single features. However, the use of several cameras from different angles is necessary for the estimation of 3D object positions which is not the case for a single mobile robot.

[BB06] developed an image likelihood function using the Wandering-Stable-Lost framework and the annealed particle filter. The prior 3D model information of the body is used to improve the accuracy of the pose estimation and predict any possible self occlusion. It suggests that when background subtraction is unreliable, an adaptive appearance model for the limbs is essential in order to stabilize the tracking results. [BRC+06] determines position, orientations and the joint angle of the object. These

techniques compromise the reduction of the high dimension search space, the density estimation, or smoothness assumption on the motion patterns. They use an industrial marker-based training samples for the estimation of a nonparametric Parzen density in order to converge the solution by the learned density. The algorithm selects the most probable solution according to the prior state in case that provided information from the input image is not enough fro a unique solution. However, the use of markers could be considered as a drawback in case of dynamic autonomous systems. On the other hand, [BRM+09] integrates the retrieved motion with 3D tracking techniques for capturing marker-less human motion. The use of prior motions to stabilize the tracking based on the results of the classification process means that misallocated priors may then worsen the tracking error.

3.4.2 3D Modeling from Stereo Images

Another application for motion segmentation and 3D modeling [YA07] for consecutive sequences of 3D models (frames) represented as 3D polygon mesh conducted the motion segmentation by analyzing the motion parameters using extracted feature vectors, while each 3D model contains information about the coordinates of vertices, connection between joints and their color. The 3D structure of a model can be extracted using stereo images [SA03, LW08, HS09] by estimating the acquired depth information. However, 3D reconstruction from stereo approaches may suffer from strong illumination change such as the sudden existence of unbalanced light source which may happen occasionally in an unknown dynamic environment.

Recent approaches such as [YK10] reconstructs the 3D target object by tracking the position of a target object in a scene to voxelize the accurate 3D human model, while classification and recognition of human 3D motions and actions requires a Multiple-Kernel based Support Vector Machine. Nevertheless, such an approach requires input images from multiple viewpoints simultaneously which is not applicable for a single mobile robots.

3.4.3 Stereo Active Vision

In a taxonomy proposed by Scharstein and Szelinski [SS02] classification of stereo algorithms has been conducted. The major categories are local methods and global methods. Global methods attempt to minimize an energy function across the entire image area, while local methods minimize a matching cost function for computing the correspondence between the stereo input frames using an aggregation window. In general, local algorithms are suitable for real-time applications but may suffers from crossing depth discontinuities and the aperture problem if the size and the shape of the aggregation window was not defined properly. As a result of such problems, object boundaries are blurred and the texture-less regions are very noisy. On the other hand, global methods such as Dynamic Programming (DP) [LSY06], Belief Propagation (BP) [KSK06] and Graph Cut (GC) [KZ01] make explicit smoothness assumptions on the disparity map. DP approaches assume that the relative ordering of pixels on a scan-line between two frames remains the same (monotonicity assumption) which may cause errors in the depth estimation of narrow foreground objects. As most of the global algorithms, BP and GC approaches gives encouraging results by enforcing the optimization in two-dimensions on the prize of the computational speed. Other stereo approaches based on the minimization of an energy function over a subset of the input image are considered in between local and global algorithms. Their minimization strategy is based on Semi Global Block Matching (SGBM) algorithms [Hir06], Dynamic Programming or Scan-line Optimization (SO) techniques [MTS07] and recently on line segmentation [DL06].

Recent stereo algorithms have significantly advanced the state-of-the-art in terms of quality. However, in terms of speed, they are computationally expensive and takes up to several minutes to compute a disparity map ([TMS+08] gives a performance evaluation of cost aggregation strategies proposed for stereo matching). Some applications such as (autonomous mobile robots, augmented-reality and automatic vehicle guidance) require real-time performance for the generation of the depth map. Hence, the importance of real-time stereo algorithms increases as in [FLV05] where an adapted recursive formulation is proposed to reduce the computing cost of SAD

cost function of a local approach which in turn inherits the ambiguity problem from the local algorithms. As a solution to overcome this problem, they implement a post processing filter application at the last phase of the algorithm which is considered as an overhead to the computation time. On the other hand, [YEA08] overcomes the ambiguity problem in low textured areas by replacing estimates in texture-less regions with fitting planes. The algorithm starts with window-based multi-view stereo matching followed by the application of consistency fusion module. Afterword, a plane-fitting phase is applied by using color segmentation, where a plane is adjusted for each segment. In order to enhance the overall computation time, some approaches integrate the computation power of the GPU. The use of a GPU has been introduced before in global approaches with hierarchical BP [YWY+06] and DP based on adaptive cost aggregation [WLG+06]. As the use of a GPU due to hardware constraints is not applicable on some platforms, solving the low texture problem using an effective variable support based on image segmentation within the SO framework has been addressed in [MTS07]. While the result is promising, the performance is far from being real-time (i. e. some minutes). The computational time has been improved by using line segment techniques and tree dynamic programming as in [DL06]. The segmentation module there contains three steps: computing the initialization marks, repositioning marks, and removing isolated marks. In order to extract linear planes, a parameter estimation approach is used for fitting planes on sparse correspondence. Afterward, dynamic programming is used on the constructed tree to minimize the energy function. The algorithm has performed well on an Intel Pentium IV 2.4 GHz processor (processing time for "tsukuba" [SS02] is about 160 ms). However, the requirement of enforcing the monotonicity inherited from the DP techniques still cause the thin foreground objects problems.

3.5 Constraints of Alternative Systems

Yet, some of these 3D motion estimation and segmentation approaches require a pre-defined 3D model or prior segmentation information [SWE+08]. Such requirements may considered as a vital drawback in the autonomous robotic field where

prior information about the objects 3D model in unpredicted scenarios and model geometry couldn't be available. Moreover, they did not address the multi-moving non-rigid objects problem where several objects could be occluded in different depth levels [KCC10]. Another aspect that should be taken into consideration is the computation speed as active vision applications require fast algorithms to act realistic in such dynamic environment. Hence, In order to overcome such drawbacks, we propose in our work a motion segmentation approach which is capable of handling transparent motion in a reasonable computation speed, proposing a saliency-based 3D motion segmentation approach integrating a real time segment based stereo algorithm, and detecting 3D motion patterns in a biologically inspired approach. On the other hand, some of the 3D motion analysis systems based on the generated depth from stereo information are limited by the use of external hardware and geometrical information as shown in the rest of this section.

3.5.1 Forward Collision Detection

In [NVO+08] a forward collision approach has been introduced for urban traffic environment using the depth information from a stereo camera. However, the system integrates the 3D reconstruction information from a "TYZX" hardware board [TYZ]. The reconstructed 3D points is used to form primary coarse objects to extract the required geometrical information which are used in tracking the constructed 3D coarse objects. A combined radial border scanning algorithm has been used to extract the delimiters of objects based on the generation of the top view projection and the contour extraction. The object delimiters data provides the necessary information required for the forward collision module to handle partially occluded objects. The system introduced a 3D polyhedron model for the drivable tunnel based on the generated information of the elevation map and the car relative velocity. The external hardware-dependent vehicle parameters such as the steering angle, yaw rate and car speed define the geometrical shape of the drivable tunnel. Hence, the output of the forward collision module is dependent on the object delimiters, tracked objects and the drivable tunnel model. However, external hardware-dependent ego-car mechani-

cal and movement parameters has been used in the system as shown in fig. 3.2. The use of such external hardware dependent information limits the usability of the system which is considered one of the main constraints.

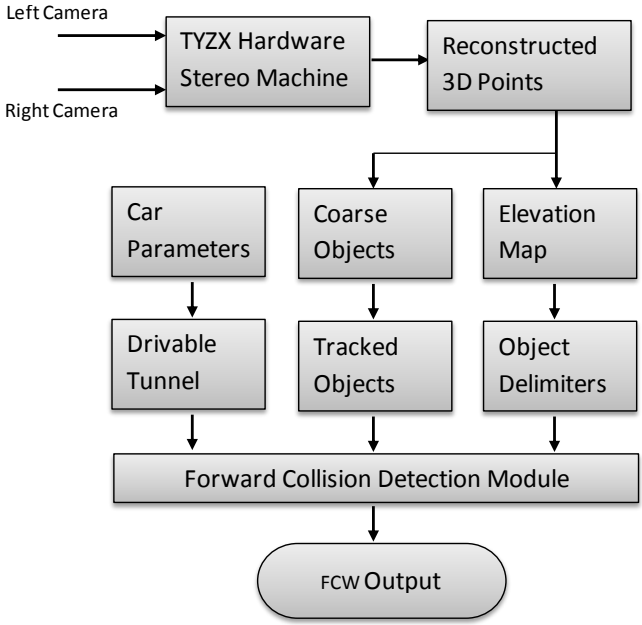


Figure 3.2: Forward collision detection system architecture introduced in [NVO+08]

3.5.2 6D Vision

A 3D variational optical flow integrating the temporal smoothness using Kalman filter assuming a linear motion model has been introduced in [RMW+10]. The Kalman filter integrates a measurement vector $\boldsymbol{m}_t$ generated by a feature extraction module.

$$\boldsymbol{m}_t = \begin{pmatrix} \boldsymbol{p}_t \\ d_{isp}(\boldsymbol{p}_t) \end{pmatrix}, \qquad \boldsymbol{p}_t = \boldsymbol{p}_{t-1} + \boldsymbol{v}(\boldsymbol{p}_{t-1}) \tag{3.6}$$

where $\boldsymbol{v}(\boldsymbol{p}_{t-1})$ is the estimated optical flow of the previous feature position $\boldsymbol{p}_{t-1}$, while $d_{isp}(\boldsymbol{p}_t)$ is the disparity value at the $\boldsymbol{p}_t$ position. The state vector of the Kalman filter $\boldsymbol{\xi} = (X, Y, Z, \dot{X}, \dot{Y}, \dot{Z})^T$ defines the 3D position and velocity of the feature point. The 6D vision approach uses the previous feature position $\boldsymbol{p}_{t-1}$ generated by the feature tracker module instead of the projection of the filtered state $\boldsymbol{\xi}_{t-1}$ in order to avoid the low pass filtering effect. The problem of such approach comes when new feature points appear or disappear. Hence, [RMW+10] introduced a filtered dense optical flow and stereo named Dense6D based on a U-D factorization algorithm. The Dense6D approach associate with every discrete pixel $\boldsymbol{p}_{t-1}$ a Kalman filter $\kappa_{t-1}(\boldsymbol{p}_{t-1})$ and a sub-pixel component $\boldsymbol{sp}_{t-1}(\boldsymbol{p}_{t-1})$. The position and the sub-pixel components are updated by

$$\begin{aligned} \boldsymbol{p}_t &= [\boldsymbol{p}_{t-1} + \boldsymbol{sp}_{t-1}(\boldsymbol{p}_{t-1}) + \boldsymbol{v}(\boldsymbol{p}_{t-1}) + 0.5\,\boldsymbol{px}] \\ \boldsymbol{sp}_t(\boldsymbol{p}_t) &= [\boldsymbol{sp}_{t-1}(\boldsymbol{p}_{t-1}) + \boldsymbol{v}(\boldsymbol{p}_{t-1}) + 0.5\,\boldsymbol{px}] \bmod 1\,\boldsymbol{px} - 0.5\,\boldsymbol{px} \end{aligned} \tag{3.7}$$

where $\boldsymbol{px} = (1, 1)^T$. In order to overcome the problem of false initialization, the covariances of the surrounding filter has been taken into consideration. However, the result of the filtering approach will be highly distracted in case of large optical flow displacements. On the other hand, in order to achieve real time performance they used the GPU and FPGA unit for parallel implementation. Moreover, they compensate the error generated from the ego-motion of vehicle using the external inertial sensor data.

3.5.3 Obstacle Detection in Complex Scenarios

The obstacle detection method in [PN10] integrates the local 3D point information such as the density, the neighborhood area and depth for generating an occupancy

grid framework as shown in fig. 3.3. The input stereo images are used in the generation of the dense depth maps while the left image sequence is used to generate the optical flow. The system fuses the range and motion information extracted from the optical flow and road-obstacles separation modules to detect dynamic obstacles and their motion orientation. While the system is named real time, the performance improvement is due to the use of the GPU in the generation of the optical flow and the use of "TYZX" accelerated hardware system [TYZ] in the 3D scene construction. Furthermore, the vehicle ego-motion is estimated using the external car sensors information such as the speed and the yaw rate.

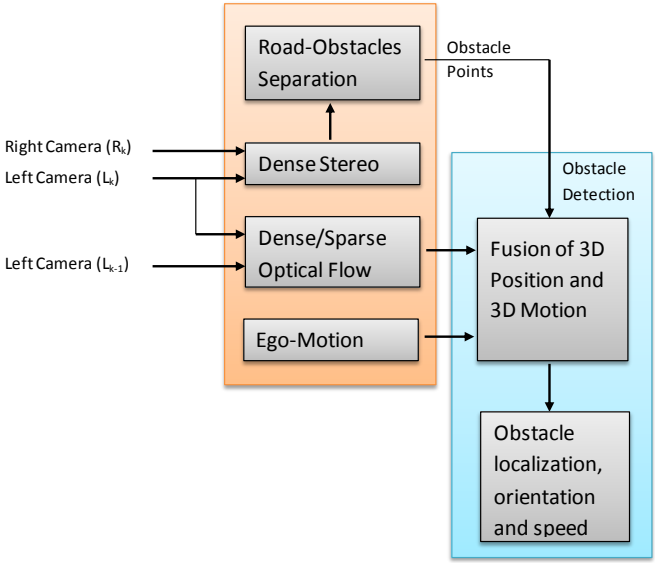


Figure 3.3: Obstacle detection in complex scenarios system architecture introduced in [PN10]

4 3D Motion Parameter Estimation

The basic idea of the proposed algorithm is to enhance the computational speed of the motion segmentation approach represented in [MJM02] by improving the 3D motion parameter estimation process. The segmentation approach initializes the segmentation process with the whole motion vector field (MVF) as one segment. The objective is to obtain a state where only MVs belonging to the same 3D motion are connected. The estimated motion parameters at a point $\boldsymbol{p}_m$ is influenced by other MVs depending on their connectivity to the same 3D motion. Hence, the process of motion parameters estimation is repeated N times for each iteration, where N is the total number of detected MVs. Therefore, enhancing the computational speed of the motion parameters estimation process leads to a significant speed-up in the segmentation approach.

4.1 Daugman's Neural Network

Interpreting optical flow as introduced in [TSL$^+$91] includes a 2D signal transform similar to that described by Daugman [Dau88]. Daugman employed a network of neuron-like units with a specified learning rule. According to the architectural design, the stabilized connection weights are the best least-mean-squares approximation to the Gabor parameters. Daugman's transform finds the derivative of the estimation error with respect to each of the Gabor parameters using a gradient descent method in order to iteratively approximate the solution.

In the case of interpreting an optical flow, the elementary signals are 2D vector fields of infinitesimal generators of a 3D Euclidean group.

The error function $E(w)$ is defined as the difference between the ideal optical motion $\boldsymbol{v}_{opt}(x,y)$ and the sensed optical motion $\boldsymbol{v}(x,y)$ for each small patch of image flow [TSL+91]:

$$E(w) = \sum_{(x,y)\in w} |\boldsymbol{v}(x,y) - \boldsymbol{v}_{opt}(x,y)|^2 \tag{4.1}$$

where $\boldsymbol{v}(x,y), (x,y) \in w$ is an image flow in a window w with m points, $w = \{\boldsymbol{p}_j, j = 1, ..., m\}$. A least-square-error solution is a set of coefficients $c_i, i = 1, ..., 6$ (see section 2.4) which minimizes the error $E(w)$, i. e., $dE(w) = 0$. The derivative of an error $E(w)$ with respect to c_i is given as

$$\begin{aligned} D_{c_i} &= \frac{\partial E(w)}{\partial c_i} \\ &= 2 \sum_{(x,y)\in W} [\boldsymbol{v}^T(x,y) \cdot \boldsymbol{e}_i(x,y)] - 2 \sum_{(x,y)\in W} \left[\left(\sum_{k=1}^{6} c_k \boldsymbol{e}_k(x,y) \right)^T \cdot \boldsymbol{e}_i(x,y) \right] \\ &= 2 \sum_{(x,y)\in W} [\boldsymbol{v}(x,y) - \boldsymbol{v}_{opt}(x,y)]^T \cdot \boldsymbol{e}_i(x,y) \end{aligned} \tag{4.2}$$

D_{c_i} is set equal to zero to solve the equation for the coefficients c_i. This approach has been improved in [MJM02] by including a recursive term $\alpha \cdot \Delta c_{(k-1)_i}$ into the learning rule

$$c_{k+1} = c_k + \Delta c_k \;\; with \;\; \Delta c_{k_i} = -\frac{1}{2} \frac{\partial E}{\partial c_i} + \alpha \cdot \Delta c_{(k-1)_i} \tag{4.3}$$

where α is a constant learning rate, which yields a noticeable speed-up at gradual slopes.

4.2 Enhanced 3D Motion Parameters Estimation

This part describes the functionality of the proposed algorithm in [SM08b]. It discusses the drawback in Daugman's algorithm. According to which the change in a single estimated parameter i. e. c_k is affected by the estimation of other parameters. This would generate an error especially in the scenarios where an input MVF describes the motion generated by one of the parameters in a motion template. The proposed method approaches the aftermentioned problem by making use of global minimum search criterion for each parameter in a MVF, which is applicable from the first iteration step $k = 0$. It is quite possible that each parameter in the estimation process may require different number of iterations m i. e. $m \in \{0, 1, 2, ..., N_m\}$ for particular $k \in \{0, 1, 2, ..., N_k\}$ to be $c^i_{k_m}$ where $i \in \{1, 2, ..., 6\}$, N_m and N_k are predefined threshold values for maximum iterations number. The root mean square error (RMSE) $E^i_{k_m}(c)$ is calculated between the input and the estimated motion vector as

$$E^i_{k_m}(c) = \frac{1}{|\boldsymbol{S}|}\sqrt{\sum_{(x,y)\in S} |\boldsymbol{v}(x,y) - \boldsymbol{v}_{est}(x,y)|^2} \tag{4.4}$$

where $\boldsymbol{v}(x, y)$ is a vector component of input MVF and $\boldsymbol{v}_{est}(x, y)$ is the vector component of the estimated MVF. $\boldsymbol{S}$ is a concatenated vector of the detected MVs in the image. Afterwords, the change in error $\Delta E^i_{k_m}$ between two successive iterations is being calculated as

$$\Delta E^i_{k_m}(c) = E^i_{k_m}(c) - E^i_{k_{m-1}}(c) \tag{4.5}$$

The above parameter $\Delta E^i_{k_m}$ is significant in devising a set of learning rules which determines the stop criterion during the motion parameters estimation process.

We start with the computation of a particular parameter coefficient $c^i_{k_{m+1}}$ as follows:

$$c^i_{k_{m+1}} = c^i_{k_m} + \Delta c^i_{k_m} \tag{4.6}$$

The convergence of $c^i_{k_{m+1}}$ is dependent on the value of $\Delta c^i_{k_m}$ which depends on the value RMSE $E^i_{k_m}(c)$ as given in

$$\Delta c^i_{k_m} = -\tfrac{1}{2}\frac{\Delta E^i_{k_m}(c)}{\Delta \overline{c}^i_{k_m}} + \alpha_i \Delta c^i_{k_{m-1}} \tag{4.7}$$

where

$$\Delta \overline{c}^i_{k_m} = \begin{cases} 1 & if \;\; c^i_{k_m} = c^i_{k_{m-1}} \\ \dfrac{c^i_{k_m} - c^i_{k_{m-1}}}{|c^i_{k_m} - c^i_{k_{m-1}}|} & if \;\; c^i_{k_m} \neq c^i_{k_{m-1}} \end{cases} \tag{4.8}$$

and α_i is an adjustable learning force parameter. Let us assume that at the start of estimation process when $k = 0$, $c^i_{k_m} = 0$ as a default value.

$$\begin{aligned} &\Rightarrow c^i_{k_{m+1}} = \Delta c^i_{k_m} \\ &\Rightarrow \Delta c^i_{k_m} = -\tfrac{1}{2}\Delta E^i_{k_m}(c) \quad \forall \quad \begin{aligned} \Delta c^i_{k_{m-1}} &= 0 \\ \Delta \overline{c}^i_{k_m} &= 1 \end{aligned} \end{aligned} \tag{4.9}$$

It can be seen that $\Delta c^i_{k_m}$ is proportional to $\Delta E^i_{k_m}(c)$ at the first step. This means $\Delta E^i_{k_m}(c)$ will be positive for the first iteration under the assumption that the default input MVF is a blank template i. e. a MVF generated from the motion parameter vector $\mathbf{c} = (0, 0, 0, 0, 0, 0)$. Now if $\Delta E^i_{k_m}(c)$ is increasing, this means that the $c^i_{k_m}$ is not a negative value. Hence, we will seek $c^i_{k_m}$ within the positive values. In order to speed up the seek process, we will consider the value of the estimated $\Delta c^i_{k_m}$ obtained in the first step (4.9) in order to skip redundant computations. Afterwords, we will test the $\Delta E^i_{k_m}(c)$ again. In case it is increasing, we have not reach a global minimum. Although, $c^i_{k_m}$ may reach a local minimum which could further reduce

the RMSE $E^i_{k_m}(c)$. However, this is would not be an optimum solution. This point actually highlights the main difference between Daugman's algorithm and the new methodology. According to which, the new algorithm will not consider the value of $c^i_{k_m}$ obtained at the first iteration $k = 0$ in estimating the other coefficients. The learning rule has been changed to be:

$$c^i_{k_{m+1}} = \begin{cases} c^i_{k_m} + \Delta c^i_{k_m} & \\ c^i_{k_m} - 2\Delta c^i_{k_m} & \text{if } (\Lambda = 0) \\ c^i_{k_0} & \text{if } (\Lambda = 1 \wedge k = 0) \end{cases} \tag{4.10}$$

where Λ is a testing criterion to check the validity of the error convergence in a particular direction

$$\Lambda = \begin{cases} 0 & \text{if } (\Delta E^i_{k_m}(c) \geq 0 \ \wedge \ \Delta \overline{c}^i_{k_m} < 0) \\ 1 & \text{if } (\Delta E^i_{k_m}(c) \geq 0 \ \wedge \ \Delta \overline{c}^i_{k_m} \geq 0) \end{cases} \tag{4.11}$$

For primary motion templates, each template has been generated using only one coefficient and the other coefficients being equal to zero. This leads to the fact that in order to estimate the right value for that coefficient in a fast way, the other coefficients should be zeros. So for the first iteration, as we seek if the MVF is one of the those primary motion templates, we assume correctly constructed MVF will be generated using only one coefficient. Therefore, we check for each $c^i_{k_m}$ if it reaches a global minimum or not, independent from other coefficients.

4.3 Results on 3D Motion Segmentation Approach

The new developed 3D motion parameters estimation algorithm introduced in [SM08b] yields an overall computation time enhancement as shown in fig. 4.1 which demonstrates the improvement in computation time of the motion segmentation approach

with respect to the computational time needed for segmenting a MVF of size (128x192) compared to the results obtained by [MJM02].

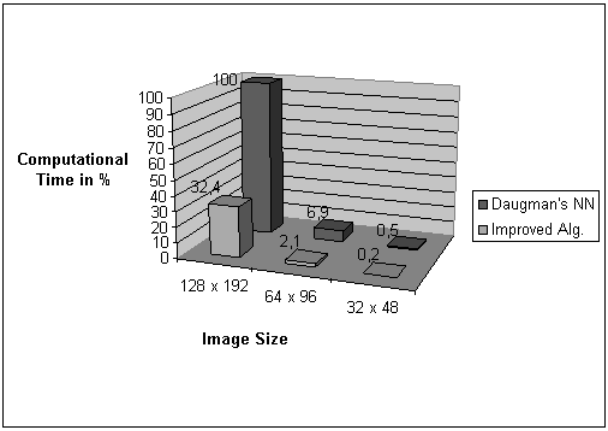


Figure 4.1: Reduction of the computational time achieved by the improved algorithm in [SM08b] needed for segmenting a MVF of size (128×192).

4.4 Chapter Summary

We have presented a fast approach to estimate the motion parameters coefficients, which results in a significant speed up compared to the estimation process from primitive motion patterns as it enhances the reduction of the mean error of the estimated parameters even with highly noised MVF. The proposed algorithm will leave a great influence in reducing the computational time of motion segmentation approaches which implies the need for fast processing methods.

5 3D Saliency-Based Motion Segmentation

In order to emphasize the contribution of the proposed approach in this chapter, the difference between estimating the 3D motion parameters and 3D motion segmentation algorithm has to be recognized. This chapter represents an enhanced 3D motion segmentation approach which integrates the improved 3D motion parameters estimation algorithm introduced earlier in the previous chapter.

In comprehensive systems of multi-object motion analysis in robotic vision, the interpretation of multiple moving objects becomes very important. There are two main challenges facing the segmentation of 3D multi-moving objects in an active vision system. The first is to segment an incoherent MVF into partitions in reasonable computation time, and the second is to overcome the ego-motion problem from the movement of a mobile robot or a camera head.

In the context of the first problem, our active vision system is exposed to some rescue scenarios where objects could be partially visible and not connected. Hence, its important to segment the MVF independently from a predefined spatial information such as object contours generated from image segmentation approaches. Such methods are dependent on a group of features which could be affected by the continuous environment change in a dynamic scene, e. g., the results of the color-based segmentation approaches could be affected by illumination changes.

The 3D motion segmentation approach in [MJM02] is conceptually able to handle transparent motion despite the pixel-connectivity of objects where motion parameters are used as a homogeneity criterion for the segmentation process. Other approaches in this context assume that each segment represents a rigid and connected object such as [GW04] where 2D non-motion affinity cues (such as spatial coherence) are incorporated into 3D motion segmentation using the Expectation Maximization (EM) algorithm. In the Expectation step, the mean and covariance of the 3D motions are

calculated using matrix operations, and in the Maximization step the structure and the segmentation are calculated by performing energy minimization. [SM06] also assumes that each segment represents a single rigid body motion in space. The segmentation process is based on an estimate of the optical flow consistent with a single rigid motion in each segmented region. The method which allows both viewing system and viewed objects to move iterates three steps until convergence. The first step is the evolution of closed curves via level sets, and then comes the computation of essential parameters of rigid motion by linear least squares in region of segmentation, while the third step is the estimation of optical flow consistent with a single rigid motion. In [HC05], a set of feature points tracked across a number of frames is obtained in order to segment 3D motions of multiple moving rigid objects. The feature points whose motion is consistent with a given motion matrix are determined using seed selection and coherence measure mechanisms that provide a map which is segmented by region growing algorithm. However, using the spatial coherence in the mentioned works, requires prior information of the object geometry. Such information is mainly based on a predefined assumption of spatial constraints or detecting certain groups of features such as in [PB06] which is in the case of our autonomous system are not available. In addition, implementing such constraints leads to image segmentation rather than segmenting the generated MVF based on its motion parameters. Similar in concept the work done in [WGP09] where prior geometric information (bounding box) is required for tracking selected image segments (manually delinated contour for each object in the first frame) based on the object's index and relative depth information using a single pairwise Markov random field (MRF). In this context the work presented in [WNL08] is also based in the detection of image segments rather than the segmentation of the computed MVF where a part hierarchy detector is defined for the required object class e. g. pedestrians and learned by boosting shape information from local image features. In [TP07] instead of tracking predefined image segments, a texture-based back-ground subtraction is used to detect objects and a unifying distance measure algorithm is build to utilize the tracking and classification module. The motion information is used in the previous approaches as an extra cue for tracking image segments besides other features such as color histogram and texture. While other approaches use the motion information to detect and track the

objects behaviors as represented in a visual surveillance survey of object motion and behaviors [HTW+04].

The second challenging problem (estimation of the camera ego-motion) has been subjected in early work such as [KKR+97] which addresses the problem of capturing, calibrating, and estimating 3D ego-motion of a monocular camera including the camera position in a known 3D environment. The intrinsic and extrinsic camera parameters of a real camera are estimated using an automated landmark-based camera calibration method which requires prior knowledge of the virtual environment. In [SMS00] a single camera has been used for computing the ego-motion of the vehicle relative to the road. A probability density function for each image patch is computed, then the probability functions from all patches are combined where prior motion estimates give low weight to patches that are not related to the road. The motion model has been reduced to 3 essential parameters, which eliminates the ambiguity between rotations and translations but also limits the representation of a six degree of freedom 3D motion. Recent work such as [SFG+07] has handled the same problem using a stereo-vision system where feature points (basicly road lane markings) are matched between pairs of frames and linked into 3D trajectories. However, the estimated parameter is only the vehicle velocity. In [SO06], in order to estimate the vehicle ego-motion, static regions must be extracted first which are dependent on the road plane.

In this chapter, a new algorithm is proposed to enhance the computational speed of the motion segmentation approach presented in [MJM02] which is very expensive computationally due to its vector-based mechanism. The new algorithm assumes a limited number of motions in two or more consecutive frames. Hence, the new approach attaches the most salient motion according to its vector numbers to the first segment, then the next salient motions in a fast iterative process using an enhanced motion parameters estimation algorithm [SM08b]. Moreover, the new approach is able to deal with the ego-motion problem resulting from the movement of the mobile robot by addressing the most salient motion resulting from the segmentation of the generated MVF under certain constraints as the global motion of the scene.

5.1 Filtering of 2D Input MVFs

In case of using a sequence of real images, a new challenge has been raised due to the large number of generated VFs. In order to reduce the number of processed vectors, the input data could be represented in different scales. Scaling the input image itself will result in big losses of input information, while scaling the generated MVF will produce better result as shown in fig. 5.1 using input sequence from PETS Dataset [PET].

5.2 Vector-Based Motion Segmentation

The motion segmentation approach in [MJM02] sets a connection weight function $\varsigma(\boldsymbol{p}_m, \boldsymbol{p}_n) \in [1, 0]$ between all MVs to be $\varsigma(\boldsymbol{p}_m, \boldsymbol{p}_n) := 1$. The weight function is iteratively updated for each pair of image points. For an image point $\boldsymbol{p}_m$, the update process starts by estimating the motion parameters $c(\boldsymbol{p}_m)$ using the following error function derived from equation 4.1

$$E(c(\boldsymbol{p}_m)) = \frac{1}{|\boldsymbol{S}|} \sum_{\boldsymbol{p} \in \boldsymbol{S}} \varsigma(\boldsymbol{p}_m, \boldsymbol{p}) |\boldsymbol{v}(\boldsymbol{p}) - \boldsymbol{v}_{opt}(\boldsymbol{p})|^2 \tag{5.1}$$

The motion parameters $c(\boldsymbol{p}_m)$ are influenced by other MVs depending on their connectivity to the same 3D motion. A generated residual VF describes the error between the generated MVF and the actual input field (for more details refere to section 2.4).

$$\boldsymbol{f}_m(\boldsymbol{p}) = \sum_{i=1}^{M} c_i(\boldsymbol{p}_m) \boldsymbol{e}_i(\boldsymbol{p}) - \boldsymbol{v}(\boldsymbol{p}) \tag{5.2}$$

For a pair of points $(\boldsymbol{p}_m, \boldsymbol{p}_n)$, the error vectors $\boldsymbol{f}_m(\boldsymbol{p}_m)$ and $\boldsymbol{f}_m(\boldsymbol{p}_n)$ are compared by evaluating a deviation measure $\Delta f(\boldsymbol{p}_m, \boldsymbol{p}_n)$ and $\Delta f(\boldsymbol{p}_n, \boldsymbol{p}_m)$. The weight function is updated by the following equation

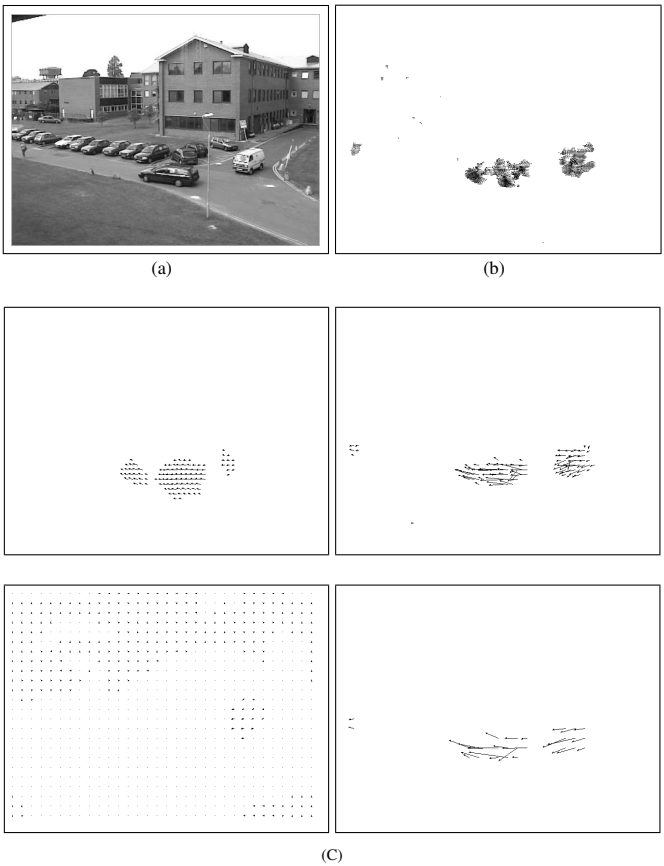


Figure 5.1: Representation of computed MVFs at different scales. (a) Input sequence from PETS Dataset [PET]. (b) Resulting MVF. (c) Left: MVFs generated from scaling the input images. Right: MVFs resulted from scaling the generated MVF. From up to down, image sizes: $64 \times 96, 32 \times 48$ respectively.

$$\varsigma(\boldsymbol{p}_m, \boldsymbol{p}_n) \equiv \varsigma(\boldsymbol{p}_m, \boldsymbol{p}_n) - \alpha \frac{1}{2}(\Delta f(\boldsymbol{p}_m, \boldsymbol{p}_n) + \Delta f(\boldsymbol{p}_n, \boldsymbol{p}_m)) \tag{5.3}$$

The update process is iteratively repeated for each pair until there is no significant change. Fig. 5.2 demonstrates the segmentation process of a synthetic MVF containing two different motions.

5.3 Saliency-Based 3D Motion Segmentation

The segmentation approach is developed to be a saliency-based approach instead of vector-based in order to increase the processing speed which is considered the main improvement of the proposed algorithm over [MJM02] and [SM08b]. In case of the ego-motion problem, in order to detect other kinds of motion while the robot is moving, the global motion should be estimated in such a scene. Other approaches to estimate global motion such as in [SSH05] use 2D affine transformation parameters for this purpose.

The proposed algorithm [SM08a] combines the two goals of segmenting multiple 3D motions, and estimating the global motion of a scene by considering the most salient motion has been segmented, i. e., the first segment $\zeta^{i=1}, i \in \{1, 2, 3, ..., m\}$ is the global motion of that scene in case that the vectors number N_{ζ^i} are above a threshold $\tau_{\zeta_{max}}$.

Before the segmentation approach starts, a noise reduction process is applied to the input MVF in order to limit the estimation process to the valid vectors only. Then, a motion segments class is initialized where every segment contains the motion parameters information $c(\zeta^i)$ of the attached motion. While the segmentation process considers the whole MVF is representing one motion at the first iteration as in [MJM02], the learning rule in the estimation process has been developed from equation 4.3 to be

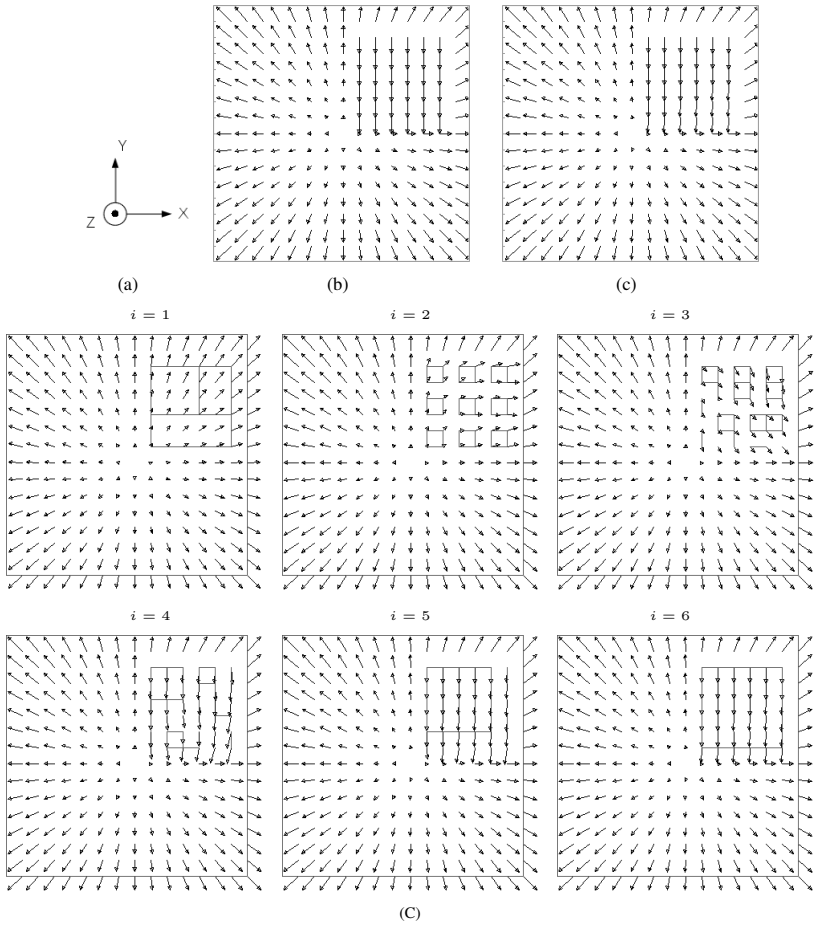


Figure 5.2: Vector-based motion segmentation of two different motions: (a) Coordinate system. (b) Input synthetic MVF. (c) Result of segmentation process. (d) Evolution of results after i iterations.

$$c^i_{k_{m+1}} = \begin{cases} c^i_{k_m} + \Delta c^i_{k_m} & \\ c^i_{k_m} - 2\Delta c^i_{k_m} & \text{if } (\Lambda = 0) \\ c^i_{k_0} & \text{if } (\Lambda = 1 \wedge k = 0) \end{cases} \tag{5.4}$$

where each parameter in the estimation process may require a different number of iterations m i. e. $m \in \{0, 1, 2, ..., N_m\}$ for a particular $k \in \{0, 1, 2, ..., N_k\}$ to be $c^i_{k_m}$ where $i \in \{1, 2, ..., 6\}$. Meanwhile, the convergence of $c^i_{k_{m+1}}$ is dependent on the value of $\Delta c^i_{k_m}$.

$$\Delta c^i_{k_m} = -\tfrac{1}{2} \frac{\Delta E^i_{k_m}(c)}{\Delta \overline{c}^i_{k_m}} + \alpha_i \Delta c^i_{k_{m-1}} \tag{5.5}$$

where

$$\Delta \overline{c}^i_{k_m} = \begin{cases} 1 & if \;\; c^i_{k_m} = c^i_{k_{m-1}} \\ \dfrac{c^i_{k_m} - c^i_{k_{m-1}}}{|c^i_{k_m} - c^i_{k_{m-1}}|} & if \;\; c^i_{k_m} \neq c^i_{k_{m-1}} \end{cases} \tag{5.6}$$

and Λ is a testing criterion to check the validity of the error convergence in a particular direction:

$$\Lambda = \begin{cases} 0 & \text{if } (\Delta E^i_{k_m}(c) \geq 0 \;\; \wedge \;\; \Delta \overline{c}^i_{k_m} < 0) \\ 1 & \text{if } (\Delta E^i_{k_m}(c) \geq 0 \;\; \wedge \;\; \Delta \overline{c}^i_{k_m} \geq 0) \end{cases} \tag{5.7}$$

where $\Delta E^i_{k_m}$ is the change in error between two successive iterations

$$\Delta E^i_{k_m}(c) = E^i_{k_m}(c) - E^i_{k_{m-1}}(c) \tag{5.8}$$

A validation process is applied to each unprocessed vector $\boldsymbol{v}_k^{\varepsilon=0}, \varepsilon \in \{1,0\}$ in order to detect whether it belongs to the same motion or not by measuring the vector difference $\boldsymbol{\vartheta}_f$ between the estimated vector and the actual input vector

$$\begin{aligned} \boldsymbol{\vartheta}_f(k) \quad &= \boldsymbol{v}_k^{\varepsilon=0} - \boldsymbol{v}_{inp}(k) \\ \boldsymbol{v}_k^{\varepsilon=0} &\in \zeta_i \quad if \quad \boldsymbol{\vartheta}_f(k) < \tau_{min}^{\vartheta_f} \end{aligned} \tag{5.9}$$

where $\tau_{min}^{\vartheta_f}$ is the minimum threshold that a vector difference should pass in order to consider an estimated vector $\boldsymbol{v}_k^{\varepsilon=0}$ belonging to the current motion segment ζ^i generated by the motion parameters $c(\zeta^i)$

After the validation process is done, the estimation process is applied after the exclusion of vectors that do not belong to the same motion. Hence, the estimated motion parameters ζ^i is enhanced.

The validation process is repeated until the maximum value of all vector differences does not exceed the minimum threshold $\tau_{min}^{\vartheta_f}$. Afterwards, the estimated motion parameter coefficients will be assigned to the first segment and each estimated vector that belong to the same motion will be marked as processed as shown in fig. 5.3 which demonstrates the segmentation process of a synthetic MVF containing two different motions.

$$\boldsymbol{v}_k^{\varepsilon=0} \rightarrow \boldsymbol{v}_k^{\varepsilon=1} \quad if \quad c(\boldsymbol{v}_k) \equiv c(\zeta^i) \tag{5.10}$$

The segmentation approach will continue with the remaining unprocessed vectors in order to segment other existed motions until either the number of segments reaches a predefined threshold –under the assumption that there is a limited number of motion in two or more consecutive frames– or the last resulted segment size is below a minimum threshold τ_{min}^{ζ}.

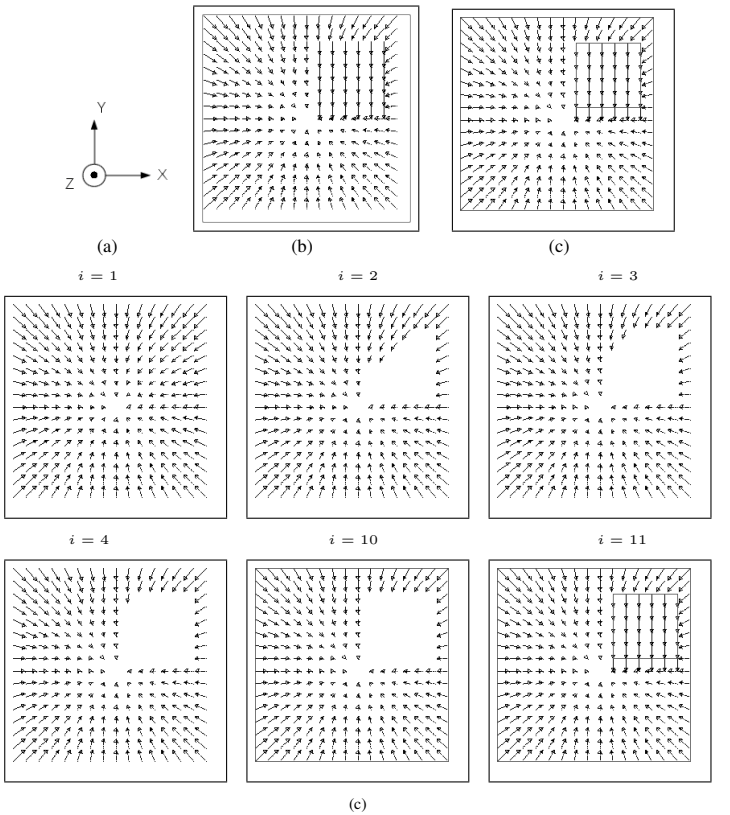


Figure 5.3: Saliency-based motion segmentation of two different motions: (a) Coordinate system. (b) Input synthetic MVF. (c) Result of segmentation process. (d) Evolution of results after i iterations.

5.4 Chapter Summary

In this work, we propose a saliency-based approach for estimating and segmenting 3D motions of multiple moving objects represented by 2D motion vector fields. In order to overcome typical problems in autonomous mobile robotic vision such as noises in the generated MVFs, occlusions, and inhibition of the ego-motion defects of a moving camera head, a classification module has been implemented to define the global motion of the mounted camera. The proposed method achieves valuable reduction in computational time by applying a guided control module which limits the segmentation output to a flexible predefined threshold value (results of the segmentation approach are discussed in chapter 7). The computational enhancement is very important since the output of the motion segmentation approach is implemented in an active vision system.

6 Depth-Integrated 3D Motion Estimation

In this chapter, a new algorithm is proposed [SM11a] to enhance the computational speed of the motion segmentation approach presented in [SM08a] by integrating the depth information in the 3D motion parameters estimation process (see section 6.3). Hence, the search space can be reduced to five dimensions which represent the rotation about the X, Y, and Z axes and translation in the direction of X and Y axes. The geometrical information of the mobile robot and the mounted stereo camera head has been taken into consideration in order to accurately position the motion vectors in the 3D spatial domain. The resulting 3D MVF provide the ability to detect and estimate any predefined motion patterns which is vital in predicting any possible collision not only with the robot but with any objects in the observed 3D environment. The disparity map is generated using a segment-based scan line stereo algorithm presented in [SM09] which is fast and independent of the GPU power.

6.1 Pinhole Stereo Geometry

In order to estimate the metric values of the disparity maps, the distance between the stereo cameras b and the focal length f as shown in fig. 6.1 has to be known.

Stereo algorithms search only in a window of disparities where the range of determined objects is restricted to an interval called Horopter. The horopter defines a curve of 3D points with zero retinal disparity [CS09] i. e. the retinal images have the same distance from the two foveae as shown in fig. 6.2.

The search window can be moved to an offset by shifting the stereo images along the baseline which must be large enough to encompass the ranges of objects in the scene. Hence, the determined depth value d will be:

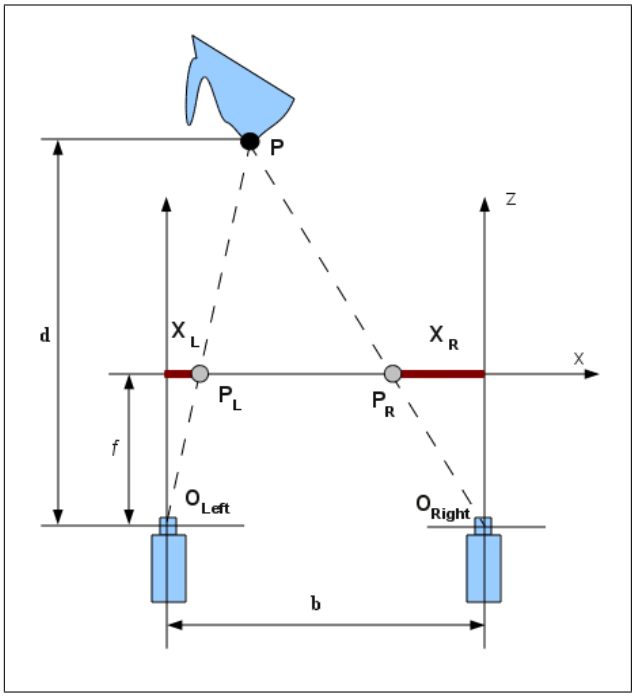


Figure 6.1: Pinhole stereo geometry.

$$d = \frac{bf}{x_r - x_l} \tag{6.1}$$

where $x_r - x_l$ is the metric disparity value. In order to use the disparity value in pixel d_{isp}, a metric to pixel transformer $\frac{1}{k}$ is used in $d_{isp} = (x_r - x_l)\frac{1}{k}$ as well as to transform the metric focal length f to be in pixel $n = f\frac{1}{k}$. The metric depth value could be rewritten as:

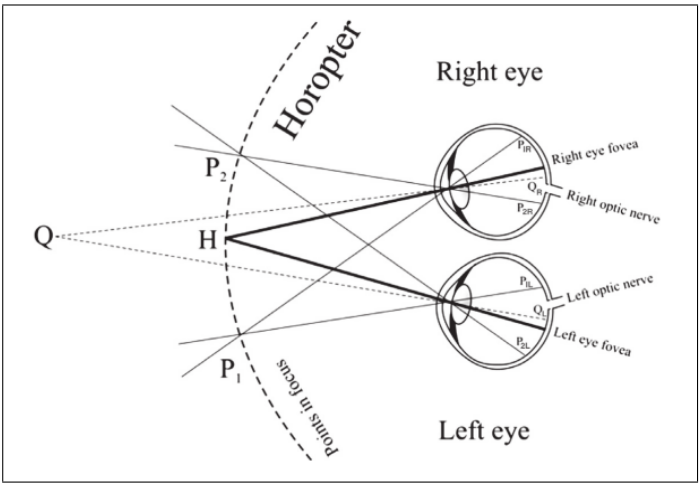


Figure 6.2: The horopter curve and the disparity on the retina where (H) is a point of fixation [CS09].

$$d = \frac{b\,n}{d_{isp}} \tag{6.2}$$

Fig 6.3 demonstrate the relation between the depth value d and different search ranges of disparity window for a constant value of the distance between the stereo cameras b and the focal length f.

6.2 Perspective Projection

A perspective projection represents an objects as it would be seen by an observer positioned at a certain vantage point [CP79]. The center of projection is at the origin o of the 3D reference frame. The focal length f determines the distance between the origin and the image plane which is is parallel to the (x, y) plane along the Z

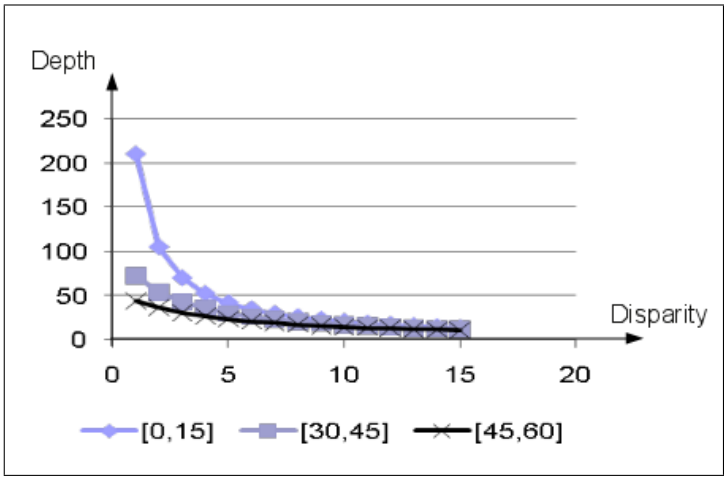


Figure 6.3: The relation between the depth value d and different search ranges of disparity window [0,15], [30,45], and [45,60].

axis [MT96]. The 3D point $\boldsymbol{P}$ projects to the image point $\boldsymbol{p}$ as shown in fig. 2.7. The 2D coordinates of $\boldsymbol{p}$ are (x, y), while (X, Y, Z) are the 3D coordinates of $\boldsymbol{P}$:

$$x = \frac{fX}{Z} \qquad y = \frac{fY}{Z} \tag{6.3}$$

The homogeneous coordinates (in case of full camera calibration while assuming that the focal length $f = 1$) will be:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{6.4}$$

The principal point is not always the origin of the image coordinates in case of real images which shift the world coordinate system from the reference frame. Hence, the Euclidean motion of the 3D coordinates must be integrated into the equation (matrix M). On the other hand, a transformation matrix K is required to handle the scaling difference of the image axes [MT96].

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim K \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} M \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{6.5}$$

The exterior camera parameters which represent the 3D position and pose of the camera are determined by the matrix M, while the interior camera parameters are given by the matrix K which is independent from the camera position:

$$K = \begin{bmatrix} s_x & s_\theta & u_0 \\ 0 & s_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.6}$$

where s_x and s_y are the scaling factors of the $\boldsymbol{x}$ and $\boldsymbol{y}$ axes respectively, s_θ determines the skew between the axes, while the intersection of the principal axis and the image plane are defined by (u_0, v_0) (the principal point) [MT96].

6.3 Integrating Depth for Estimating 3D Motion

In this part, the functionality of the proposed algorithm in [SM11a] will be described. Integrating the depth information in the 3D motion parameters estimation process reduces the search space to 5D where the parameter coefficient of the translation in Z direction c_3^i will equal the depth difference between two consecutive disparity maps:

$$c_3^i = d_i^{t+1} - d_i^t \tag{6.7}$$

where d_i^t is the depth of point $\boldsymbol{p}_{i,t} = (x, y, t)^T$ and d_i^{t+1} is the depth of its correspondence point $\boldsymbol{p}_{i,t+1} = (x + \delta x, y + \delta y, t + 1)^T$ determined by the motion vector $\boldsymbol{v}(\boldsymbol{p}_{i,t})$ generated using a fast variational optical flow approach [BWF+05]. Before the estimation approach starts, a noise reduction process is applied to the input MVF in order to limit the estimation process to the valid vectors only. Then, a motion segments class is initialized where every segment contains the motion parameters information $c(\zeta^i)$ of the attached motion. The segmentation process considers the whole MVF is a global motion at the first iteration.

A validation process is applied to each unprocessed vector $\boldsymbol{v}_k^{\varepsilon=0}$ $\varepsilon \in \{1, 0\}$ in order to detect whether it belongs to the same motion or not by measuring the vector difference ϑ_f between the estimated vector and the actual input vector. The estimation process is re-applied after the exclusion of vectors that do not belong to the same motion.

6.3.1 Real-Time Segment Based Stereo Algorithm

The goal of stereo algorithms is to establish pixel correspondences between the left image I_l and the right image I_r. In order to achieve reasonable results, two geometric constraints are used: first on the imaging systems, i. e. the input stereo images are *rectified* where the epipolar lines are aligned with corresponding scan-lines. And second on the scene, i. e. the *smoothness assumption* where the disparity map is smooth almost everywhere except at the border of objects assuming that scene is composed of smooth structures which in the case of autonomous mobile robots applications is not granted.

The first step of the proposed technique in [SM09] is the line segmentation of the reference image I_r, in which the epipolar line epl_y is segmented into different labels $l_i(epl_y) \in \Gamma$ in the label space Γ based on the Euclidean color differences between

the color of a seed pixel $\boldsymbol{g}_s^{l_i}$ where $\boldsymbol{g} = (R, G, B)^T$ and the color of the neighborhood pixels of the same epipolar line:

$$(x + k, y) \in l_i(epl_y) \qquad \forall \ |\boldsymbol{g}_s^{l_i} - \boldsymbol{g}_k| < \tau_c \tag{6.8}$$

where τ_c is the Euclidean color distance threshold and k is an adjacent segment in a particular epipolar line epl_y. The correspondence problem is formulated as an energy minimization function between segments of the input images.

$$E(d_\Gamma) = argmin(E_{data}^{l_i}(d_\Gamma) + E_{smooth}^{l_{(i,d)\in k}}(d_\Gamma)) \tag{6.9}$$

where $E(d_\Gamma)$ is the estimated disparity map of line segment of label $i \in \Gamma$ for a disparity value d_Γ. The data term $E_{data}^{l_i}(d_\Gamma)$ of the energy function is the matching cost between a segment $l_i(epl_y)$ in the reference image and the opponent segments $l_{i,d}(epl_y)$ in the target image. The smoothness term $E_{smooth}^{l_{(i,d)\in k}}(d_\Gamma)$ encodes the smoothness assumption (see equation 6.12).

Matching Cost and Optimization

In order to reduce the complexity of calculations, a matching cost C_M based on the absolute color difference between the points of the current segment in the reference image and all disparity hypotheses is used to evaluate the data term

$$C_M(q_r, q_{l,d}) = \sum_{c \in \Re} |q_r^c(x, y) - q_{l,d}^c(x + d_{isp}, y)| \tag{6.10}$$

where $\Re$ is the RGB color space, $q^c(x, y)$ $c \in \Re$ is a single color channel value at the point (x, y), while $q_r(x, y) \in I_r$, $q_{l,d}(x, y) \in I_l$, and d_{isp} is the hypothesized disparity value.

The data term is computed from the sum of the matching cost along the segment points

$$E_{data}^{l_i}(d_\Gamma) = \sum_{q=q_s}^{q_r^N} C_M(q_r, q_{l,d}) \qquad \forall\ q_r \in l_i(epl_y) \tag{6.11}$$

where q_s is the starting seed pixel of a line segment and q_r^N is the last point in the same segment.

In order to enhance the optimization performance, we propose an effective and simplified smoothness term within the scan-line optimization (SO) framework.

$$E_{smooth}^{l_{(i,d)\in K}}(d_\Gamma) = \lambda(\ell_{l_i}) \ \cdot\ |d_{l_i} - d_{l_K}| \tag{6.12}$$

where $\lambda(\ell_{l_i})$ is an ascending function to the length of the current segment ℓ_{l_i} used to penalize depth discontinuities. The concept behind the function is to balance the relation between the disparity of a segment and the sum of the matching cost of the segment points. While the matching cost is affected by the length of the segment, only one disparity value is assigned to all the segment points and the best value is chosen within a winner take all (WTA) scheme. Considering the inter-scan-line smoothness resulting from line segmentation leads to overcome the ambiguity problem without the use of a recursive smoothing function as in BP approaches or facing narrow front objects problem as in DP algorithms.

Results of the Proposed Approach

In an effort to reduce the overall computation time, the depth from the stereo approach is applied without a refinement step depending on the enhancement done by the modified smoothing function. Fig. 6.4 represents a qualitative comparison of the proposed algorithm to the ground truth of the Middlebury data-set [SS02, SS03]. The second row depicts the generated depth map without the use of the smoothing function, while the third row represents the result of the stereo approach using the smoothing function. The result shows that when the smoothing function is applied, it provides a better quality. However, the use of the smoothing function increases the

processing time by about 5∼9 ms which is not very critical for real-time application. On the other hand, the result without the smoothing function is affected by noise but this is not very critical to the depth perception.

6.3.2 3D Representation of Motion Parameters

The visualization difference between a projected 3D point into a 2D plane using the equations proposed in [TSL+91] and the 3D homogeneous transformation matrix resulting from multiplying the current 3D spatial position and the perspective matrix must be taken into consideration. Hence, in order to represent a similar visualization of the projected 3D point in the real 3D spatial domain, transformation functions have to be applied to estimate the transformation matrix coefficients (t_X, t_Y, t_Z for translation motion and $\theta_X, \theta_Y, \theta_Z$ for rotation motion) from the pre-estimated 3D motion parameter coefficients of the projected motion c_i (see equation 2.19).

The translation in the X and Y direction will be equal to the pre-estimated 3D motion parameters c_1, c_2, while the translation in the Z direction and the rotation motions involve the perspective information. For a 3D perspective projection, a 3D point in eye space is projected onto the near plane (projection plane) where X_e as shown in fig. 6.5 is mapped to x and calculated using the triangles similarity [Ahn]:

$$\begin{aligned} \frac{x}{X_e} &= \frac{-n}{Z_e} \\ x &= \frac{-nX_e}{Z_e} \end{aligned} \tag{6.13}$$

and similarly for y:

$$y = \frac{-nY_e}{Z_e} \tag{6.14}$$

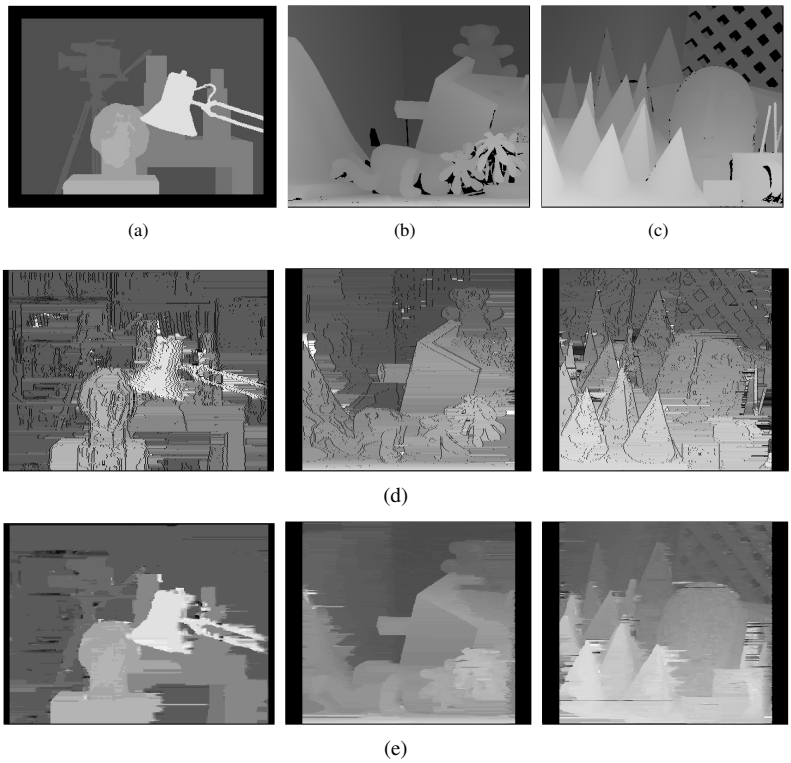


Figure 6.4: Qualitative comparison of the generated depth map: (a-c) Ground truth data for the three images from the Middle-bury data-set (Tsukkuba, Teddy, and Cones). Result of the proposed line segment based stereo algorithm (d) without and (e) with the use of the modified smoothing function.

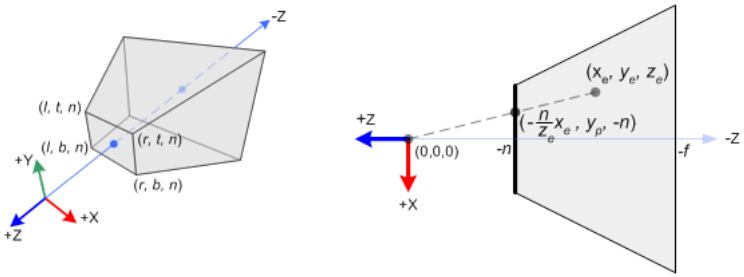


Figure 6.5: Projection in OpenGL of a 3D point in eye space onto the near plane [Ahn].

Translation in X, Y and Z

The following transformation matrix is used to estimate the translation in X direction:

$$\begin{bmatrix} X'_e \\ Y'_e \\ Z'_e \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_X \\ 0 & 1 & 0 & t_Y \\ 0 & 0 & 1 & t_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_e \\ Y_e \\ Z_e \\ 1 \end{bmatrix} \tag{6.15}$$

from the above transformation X'_e is the new value for X_e with $t_Y = 0$ and $t_Z = 0$:

$$X'_e = X_e + t_X \tag{6.16}$$

x' from eq. 6.13 is:

$$x' = \frac{-nX'_e}{Z'_e} = \frac{-n(X_e + t_X)}{Z_e} \tag{6.17}$$

while x' from the estimated 3D motion parameter coefficient of the projected motion c_1 for $Z = 1$ (see eq. 2.19 and 2.20) is:

$$\begin{aligned} \boldsymbol{e}_1(x,y) &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ x' &= x + c_1 \end{aligned} \tag{6.18}$$

and similarly for the translation in y:

$$\begin{aligned} \boldsymbol{e}_2(x,y) &= \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ y' &= y + c_2 \end{aligned} \tag{6.19}$$

From eq. 6.17 and 6.18 :

$$\begin{aligned} x + c_1 &= \frac{-n(X_e + t_X)}{Z_e} = \frac{-nX_e}{Z_e} - \frac{nt_X}{Z_e} \\ x + c_1 &= x - \frac{nt_X}{Z_e} \\ t_X &= -\frac{nc_1}{Z_e} \end{aligned} \tag{6.20}$$

And similarly for the translation in t_Y:

$$t_Y = -\frac{nc_2}{Z_e} \tag{6.21}$$

while for the translation in Z direction, x' and y' from eq. 2.20 will be:

$$\begin{aligned} \boldsymbol{e}_3(x,y) &= \begin{pmatrix} -x \\ -y \end{pmatrix} \\ x' &= x + (-c_3 x_s k) \\ y' &= y + (-c_3 y_s k) \end{aligned} \tag{6.22}$$

where $x_s \in [-1, 1]$ is the normalized value of the x location on the near plane, k is a scaling factor. On the other hand, X'_e and Y'_e from eq. 6.15 with $t_X = 0$ and $t_Y = 0$ are:

$$\begin{aligned} X'_e &= X_e + t_X = X_e \\ Y'_e &= Y_e + t_Y = Y_e \\ Z'_e &= Z_e + t_Z \end{aligned} \tag{6.23}$$

from eq. 6.13, 6.22 and 6.23:

$$\begin{aligned} x - c_3 x_s k &= \frac{-nX'_e}{Z'_e} = \frac{-nX_e}{Z'_e} \\ Z'_e &= -\frac{nX_e}{x - c_3 x_s k} \end{aligned} \tag{6.24}$$

Hence the translation in z direction t_z will be

$$t_z = \frac{-nX_e}{x - c_3 x_s k} - Z_e \tag{6.25}$$

Fig. 6.6 demonstrates the translation in the X, Y and Z axes using the translation parameter coefficient c_i from eq. 2.19 and the transformed translation parameter t_X, t_Y and t_Z from (eq. 6.20, 6.21 and 6.25).

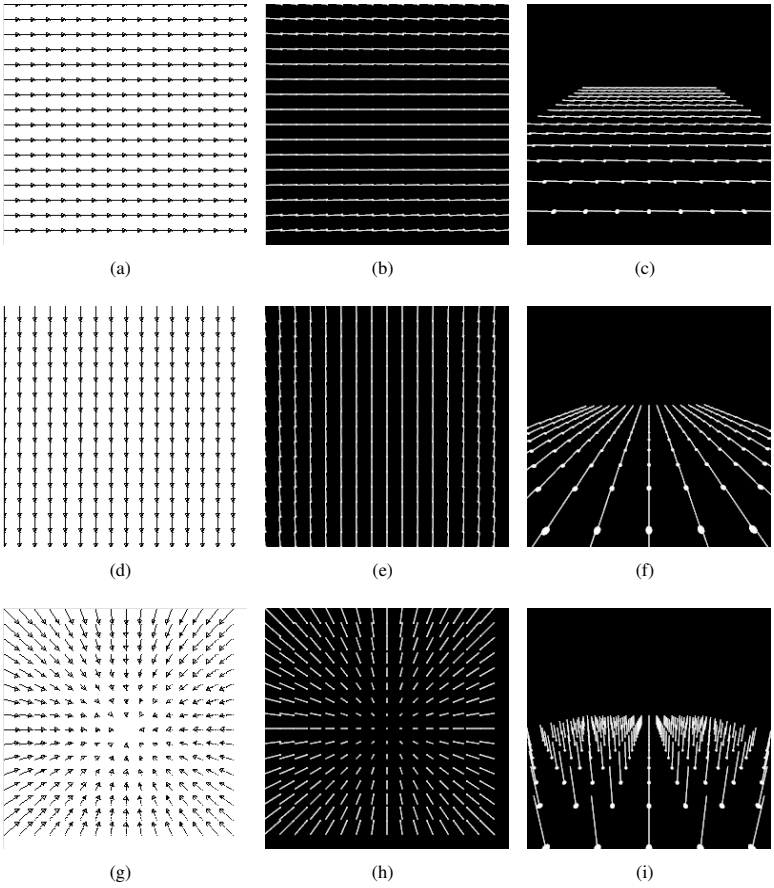


Figure 6.6: Translation in the X, Y and Z axes. (a-c) Translation in the X direction, (a) using the translation parameter coefficient c_1, (b) using the transformed translation parameter t_x, (c) perspective view of (b) using OpenGL. (d-f) Translation in the Y direction. (g-i) Translation in the Z direction.

Rotation about x, y and z

In order to estimate the rotation parameters such as the rotation about the Z axis θ_Z, the following transformation matrix has to be used:

$$\begin{bmatrix} X'_e \\ Y'_e \\ Z'_e \\ w \end{bmatrix} = \begin{bmatrix} \cos\theta_Z & -\sin\theta_Z & 0 & 0 \\ \sin\theta_Z & \cos\theta_Z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_e \\ Y_e \\ Z_e \\ 1 \end{bmatrix} \tag{6.26}$$

X'_e and Y'_e are computed from the above transformation:

$$\begin{aligned} X'_e &= X_e \cos\theta_Z - Y_e \sin\theta_Z \\ Y'_e &= X_e \sin\theta_Z + Y_e \cos\theta_Z \\ Z'_e &= Z_e \end{aligned} \tag{6.27}$$

For the rotation about Z axis, x' and y' from eq. 2.21 will be:

$$\begin{aligned} \boldsymbol{e}_6(x,y) &= \begin{pmatrix} -y \\ x \end{pmatrix} \\ x' &= x - c_6 y_s k \\ y' &= y + c_6 x_s k \end{aligned} \tag{6.28}$$

From eq. 6.13 and 6.27:

$$x' = \frac{-nX'_e}{Z'_e} = \frac{-n(X_e \cos\theta_Z - Y_e \sin\theta_Z)}{Z_e} \tag{6.29}$$

From eq. 6.28 and 6.30:

$$
\begin{aligned}
x - c_6 y_s k &= \frac{-n(X_e \cos\theta_Z - Y_e \sin\theta_Z)}{Z_e} \\
X_e \cos\theta_Z - Y_e \sin\theta_Z &= (x - c_6 y_s k)\frac{-Z_e}{n}
\end{aligned}
\tag{6.30}
$$

Solving the equation using the trigonometric identities yields:

$$
\begin{aligned}
a\,\sin\theta + b\,\cos\theta &= \sqrt{a^2+b^2}\,\sin(\theta+\alpha) = c \\
\theta &= \sin^{-1}\left(\frac{c}{\sqrt{a^2+b^2}}\right) - \tan^{-1}\left(\frac{b}{a}\right)
\end{aligned}
\tag{6.31}
$$

where $a = -Y_e$, $b = X_e$ and $c = (x - c_6 y_s k)\dfrac{-Z_e}{n}$. Hence, the rotation about the z axis is computed from eq. 6.30 and 6.31:

$$
\begin{aligned}
\theta_Z &= \sin^{-1}\left(\frac{(x - c_6 y_s k)\dfrac{-Z_e}{n}}{\sqrt{X_e^2 + Y_e^2}}\right) - \tan^{-1}\left(\frac{X_e}{-Y_e}\right) \\
\theta_Z &= \sin^{-1}\left(\frac{(nX_e + c_6 y_s k Z_e)}{n\sqrt{X_e^2 + Y_e^2}}\right) - \tan^{-1}\left(\frac{X_e}{-Y_e}\right)
\end{aligned}
\tag{6.32}
$$

The same procedure is applied for the estimation of the rotation parameter θ_X:

$$
\begin{aligned}
Z_e \cos\theta_X + Y_e \sin\theta_X &= \frac{-nX_e}{(x - c_4 x_s y_s k)} \\
\theta_X &= \sin^{-1}\left(\frac{nX_e Z_e}{(nX_e + c_4 x_s y_s k Z_e)\sqrt{X_e^2 + Y_e^2}}\right) - \tan^{-1}\left(\frac{Z_e}{Y_e}\right)
\end{aligned}
\tag{6.33}
$$

and for θ_Y:

$$Z_e \cos\theta_Y - X_e \sin\theta_Y = \frac{-nY_e}{(y + c_5 x_s y_s k)}$$

$$\theta_Y = \sin^{-1}\left(\frac{nY_e Z_e}{(nY_e - c_5 x_s y_s k Z_e)\sqrt{X_e^2 + Y_e^2}}\right) - \tan^{-1}\left(\frac{Z_e}{-X_e}\right) \tag{6.34}$$

Fig. 6.7 demonstrates the rotation about the X, Y and Z axes using the rotation parameter coefficient c_i from eq. 2.19 and the transformed translation parameter θ_X, θ_Y and θ_Z from (eq. 6.32, 6.33 and 6.34).

6.3.3 3D Representation of a Motion Vector Field

The representation of a vector in the 3D domain requires the 3D spatial information of its two points $\boldsymbol{P}_1 = (X_e, Y_e, Z_e)^T$ and $\boldsymbol{P}_2 = (X'_e, Y'_e, Z'_e)^T$. The estimated depth value d_i^t for point $\boldsymbol{P}_1$ and the focal length f are used in the eq. 6.13 where $-Z_e = d_i^t$ and $-n = f$. Similar to point $\boldsymbol{P}_1$, $-Z'_e = d_i^{t+1}$ for point $\boldsymbol{P}_2$ are used in eq. 6.14, which yields:

$$\boldsymbol{P}_1 = \begin{pmatrix} x_i \dfrac{d_i^t}{f} \\ y_i \dfrac{d_i^t}{f} \\ d_i^t \end{pmatrix} \qquad \boldsymbol{P}_2 = \begin{pmatrix} (x_i + DX_i)\dfrac{d_i^t}{f} \\ (y_i + DY_i)\dfrac{d_i^t}{f} \\ d_i^{t+1} \end{pmatrix} \tag{6.35}$$

For an accurate 3D representation of the 2D MVs, DX_i and DY_i from eq. 6.35 are functions of the depth information:

$$DX_i = v_x^i + (d_i^{t+1} - d_i^t)x_s, \qquad DY_i = v_y^i + (d_i^{t+1} - d_i^t)y_s \tag{6.36}$$

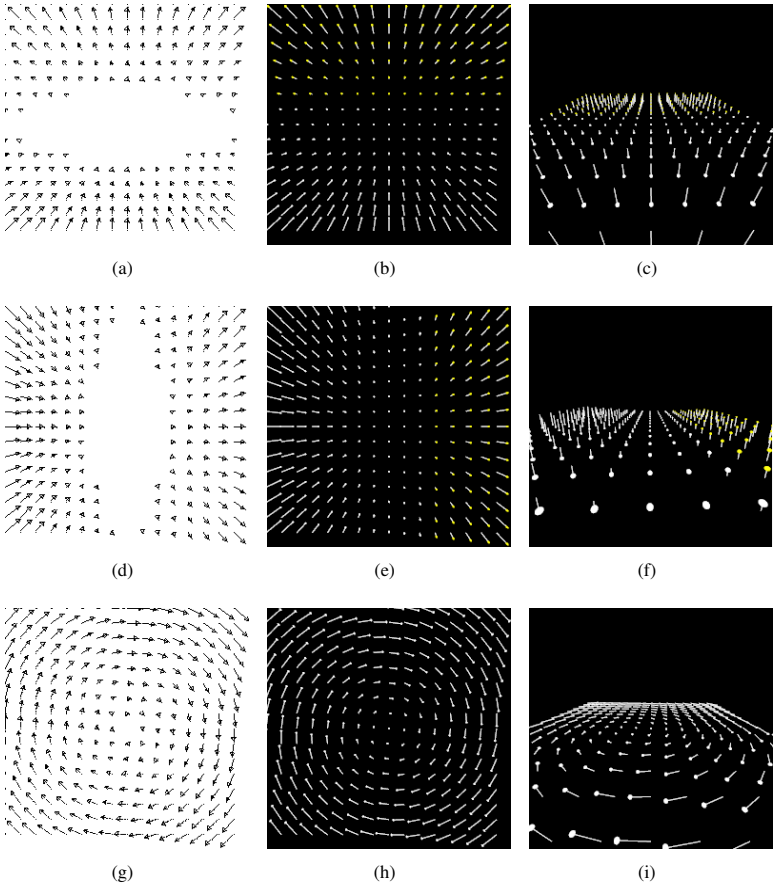


Figure 6.7: Rotation about the X, Y and Z axes. (a-c) Rotation about the X axis, (a) using the rotation parameter coefficient c_4, (b) using the transformed rotation parameter θ_X, (c) perspective view of (b) using OpenGL. (d-f) Rotation about the Y axis. (g-i) Rotation about the Z axis.

where the v_x^i and v_y^i are the 2D generated MV components. Fig. 6.8 represents the error (see eq. 4.4) resulting from using the 2D MV components v_x^i and v_y^i in the estimation of x' and y' values of a 3D motion parameters $c = (1, 0, 1, 0, 0, 0)$ representing translation in the X and Z direction.

6.4 Detection of 3D Motion Patterns

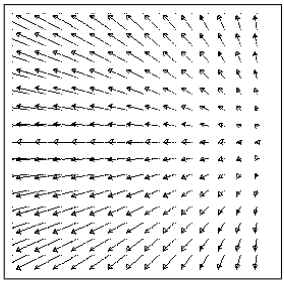
On the other hand, the proposed approach succeeds in detecting a predefined motion pattern as shown in fig. 6.9 where a ball is moving forward in the Z direction towards the robot. The 3D MVs that present the translation in the Z direction (which describes possible objects movements in the direction of the robot) are represented in yellow (for more details see chapter 7).

6.4.1 Collision Detection with the Drivable Tunnel

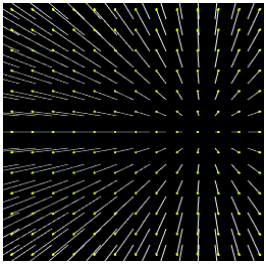
In order to improve the prediction of possible collisions, a drivable tunnel model has been constructed representing the virtual area around the 3D motion path of the vehicle. Fig. 6.10 shows the drivable tunnel model where the color of the tunnel has been scaled from green to red representing the danger of the collision based on the distance to the vehicle. The detection process are based on the 3D motion vectors $\boldsymbol{V} = \boldsymbol{P}_1 + \delta t(\boldsymbol{P}_2 - \boldsymbol{P}_1)$ pointing towards a tunnel plane $\mathbb{k}_n$ in the direction of the vehicle

$$\boldsymbol{V} = (V_x, V_y, V_z)^T = \begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \end{pmatrix} + \delta t \begin{pmatrix} X_2 - X_1 \\ Y_2 - Y_1 \\ Z_2 - Z_1 \end{pmatrix} \tag{6.37}$$

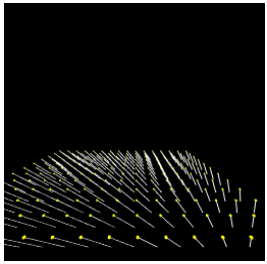
where $\delta t = t_i - t_{i+k}$ and $Z_2 - Z_1 \geq 0$, the danger of the collision is dependent on the distance χ to the tunnel plane $\mathbb{k}_n$ which the 3D motion vector $\boldsymbol{V}$ is intersecting



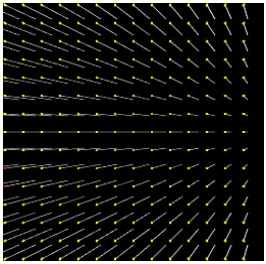
(a)



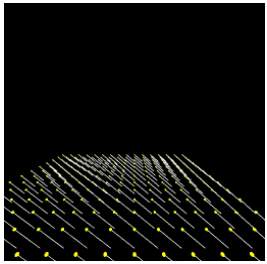
(b)



(c)



(d)



(e)

Figure 6.8: A synthetic 3D motion template. (a) The generated 2D MVF of the motion parameters $c = (1, 0, 1, 0, 0, 0)$ representing translation in the $-X$ and Z direction. (b-c) The incorrect 3D MVF and its perspective view generated using v_x^i and v_y^i values of the 2D MVF. (d-e) The correct 3D MVF generated using DX_i and DY_i values.

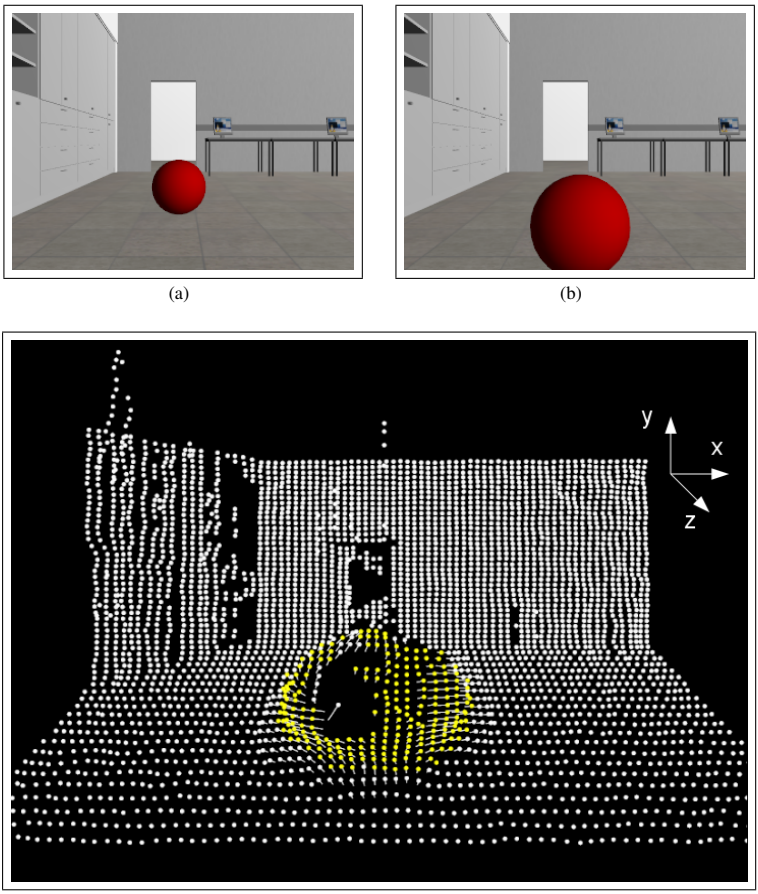
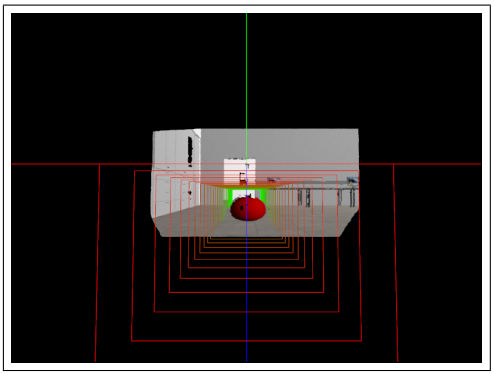


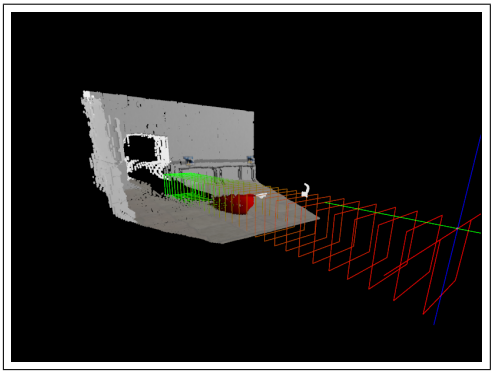
Figure 6.9: Detection of 3D motion patterns. (a) First image in the scene. (b) Last image in the scene. (c) Resulted 3D MVF where yellow MVs represent the translation in the Z direction.

$$\chi = \frac{(\boldsymbol{P}_{\mathbb{k}} - \boldsymbol{P}_1)^T \cdot \boldsymbol{n}}{\boldsymbol{V}^T \cdot \boldsymbol{n}} \tag{6.38}$$

where $\boldsymbol{P}_{\mathbb{k}}$ is a point on the tunnel plane $\mathbb{k}_n$ and $\boldsymbol{n}$ is a normal vector to that plane.



(a)



(b)

Figure 6.10: The drivable tunnel model where the color of the tunnel scaled form green to red represents the danger of the possible collision. (a) Front view. (b) Auxiliary view.

6.5 Chapter Summary

We have presented a fast depth-integrated 3D motion parameter estimation approach which enhanced the overall computation time of a 3D salient-based motion segmentation algorithm (see chapter 7) by reducing the search space of the parameter coefficient to five dimension. In addition, the presented 3D motion parameters representation algorithm has taken into consideration the perspective transformation and the depth information to accurately position motion vectors of the generated depth sequence in the 3D space using the geometrical information of the stereo camera head. Moreover, the proposed approach has successfully detected and estimated predefined motion patterns describes important 3D motions such as movements toward the robot which is very helpful in detecting possible collisions of moving objects with the robot.

7 Results and Evaluation

The proposed motion segmentation approaches are implemented as a complete software framework using object oriented programming techniques with C++. This chapter presents the output generated by the algorithms on various test cases selected to evaluate its performance, for which details are described in sections 7.2, 7.3 and 7.4 respectively. The results were obtained using two different platforms described in section 7.1 for testing the algorithms' performance on images sequence of static camera, controlled virtual environments in a simulation framework, and real-life scenarios using a stereo camera.

7.1 Experimentation Platforms

The first platform is an evaluation framework for single images as well as image sequences. An interactive graphical user interface has been designed using Qt framework (Qt is a cross-platform application and UI framework [Sum10]) and run under ROS (*Robot Operating System*) as shown in fig. 7.1 to evaluate the different approaches of 3D motion analysis and to control the involved parameters of the algorithms.

The second platform is the robot simulation framework (SIMORE) developed in our group [KHS+08, KM10] which allows integration and manipulation of dynamic 3D environments with simulated sensors, actors, and complete robots. The robot can be operated by manual input devices, a graphical user interface and program commands. The interface associated with the simulator is reliable enough so that the control commands can be directly transferred to a real robot platform after successful simulation tests. In addition to the 3D graphics engine, the simulator has a physics engine to guarantee a correct physical behavior of the simulated objects.

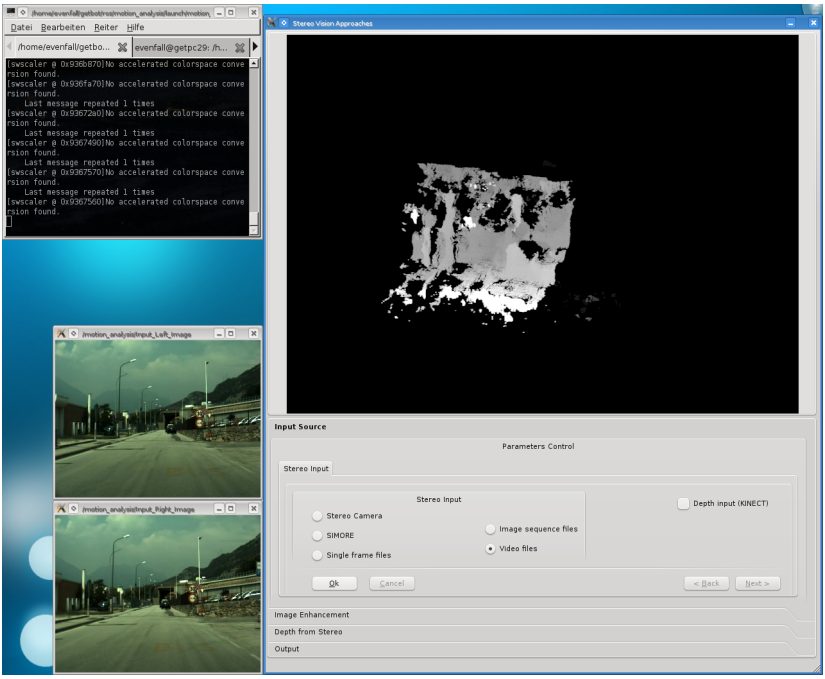


Figure 7.1: Graphical user interface for the evaluation of the proposed algorithms. Real time representation of depth maps within the GUI using stereo video as an input is shown.

This platform helps in testing the algorithms in a three dimensional world with the ability of maneuvering the sensor head as well as the whole robot to estimate the 3D motion of objects with the existence of the ego-motion. The test scenes can be created with scalable complexity and they are utterly reproducible as illumination conditions remain stable and the arrangement of objects remains intact for an arbitrarily long period of time. Moreover, experiments can be conducted uninterruptedly without disturbances from hardware failures and emptying of batteries. Hence the core functionality of the algorithms can be verified and validated through this system. Fig. 7.2

presents a sample virtual environment with a simulated robot maneuvering inside it. The visual input seen through the cameras are also shown.

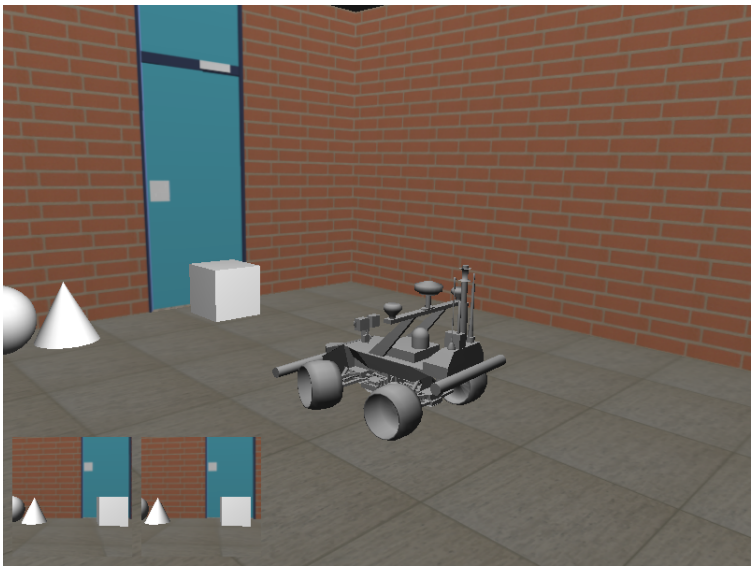


Figure 7.2: Robot simulation framework (SIMORE) with stereo image stream representing the output of the simulated stereo camera head.

7.2 3D Motion Parameters Estimation Results

In order to evaluate the 3D motion parameters estimation algorithm, a graphical user interface has been designed to generate synthetic MVFs by editing the coefficient parameters manually as shown in fig. 7.3. The input data set represented in fig. 7.4,

describes a synthetic MVF generated by different coefficients values and after application of 100% noise to each vector component and random equally distributed removal of MVs (with $\rho = 0.5$).

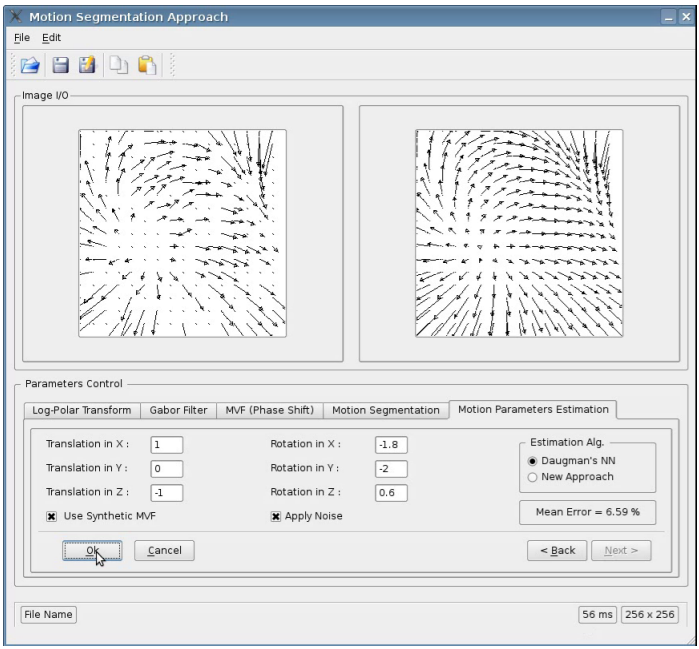
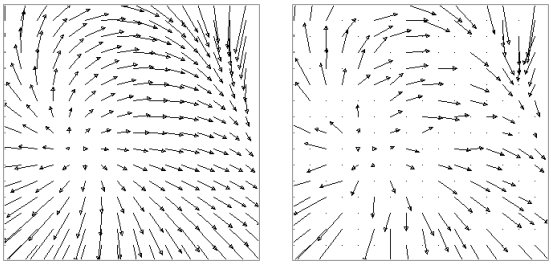
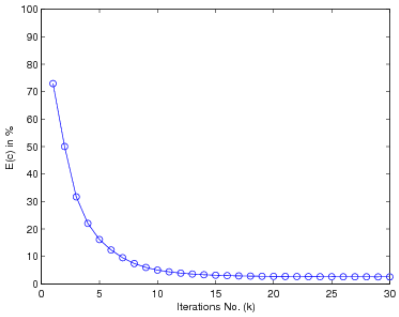


Figure 7.3: Graphical user interface for the evaluation of the proposed algorithms. The result of a 3D motion parameters estimation process is shown with the percentage of the mean error.

In order to investigate the performance of the proposed algorithm correctly, the testing criterion is based on the progression of the mean error of the estimated parameters E_{total} instead of the progression of the mean square error $E(c)$ over the general iteration step k as shown in Fig. 7.4.



(a) (b)

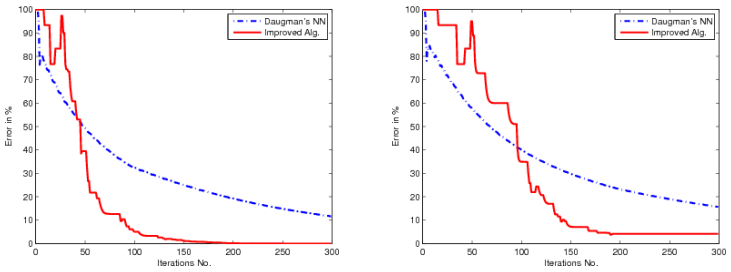


(c)

Figure 7.4: Synthetic MVFs. (a) Generated by $c = (1, 0, -1, -1.8, -2, 0.6)$. (b) After application of noise and MVs removal. (c) Progression of the mean square error $E(c)$ over the general iteration steps k.

$$E_{total} = \frac{1}{6}\sum_{i=0}^{6}\varepsilon_i \quad where \quad \varepsilon_i = \frac{c_{opt} - c_i}{c_{opt}} \times 100$$

Fig. 7.5 and fig. 7.6 demonstrates a comparison between the implemented Daugman's NN in [MJM02] and the proposed algorithm in [SM08b] for the progression of the mean error of the estimated parameters E_{total} over the particular iteration step



(a) (b)

Figure 7.5: Progression of the mean error of the estimated parameters over the particular iteration steps for the implemented Daugman's NN in [MJM02] and the proposed algorithm in [SM08b]. (a) For a synthetic MVF generated by $c = (1, 0, -1, -1.8, -2, 0.6)$. (b) For a synthetic MVF after application of 100% noise to each vector component and random equally distributed removal of MVs (with $\rho = 0.5$).

k_i^m (note that we use here the more precise iteration step k_i^m instead of the general iteration step k as in [MJM02]).

Due to the linearity between the derivative error D_{v_i} and the partial velocity coefficients of the translation in X and Y direction (c_1, c_2), the performance of the implemented Daugman's network in [MJM02] is almost the same as that of the proposed algorithm. On the contrary, the non-linear relation with respect to the translation in Z direction and the rotation in X,Y and Z $(c_3, ..., c_6)$ leads to the need of increased number of iteration steps. This drawback has been overcome by the new algorithm as seen in the results of the first data set. In the second data set, the new approach showed an enhanced performance in reaching a minimum error of $E_{total} < 0.01\%$ for a synthetic MVF and $E_{total} < 0.5\%$ for a significant alteration to the same MVF.

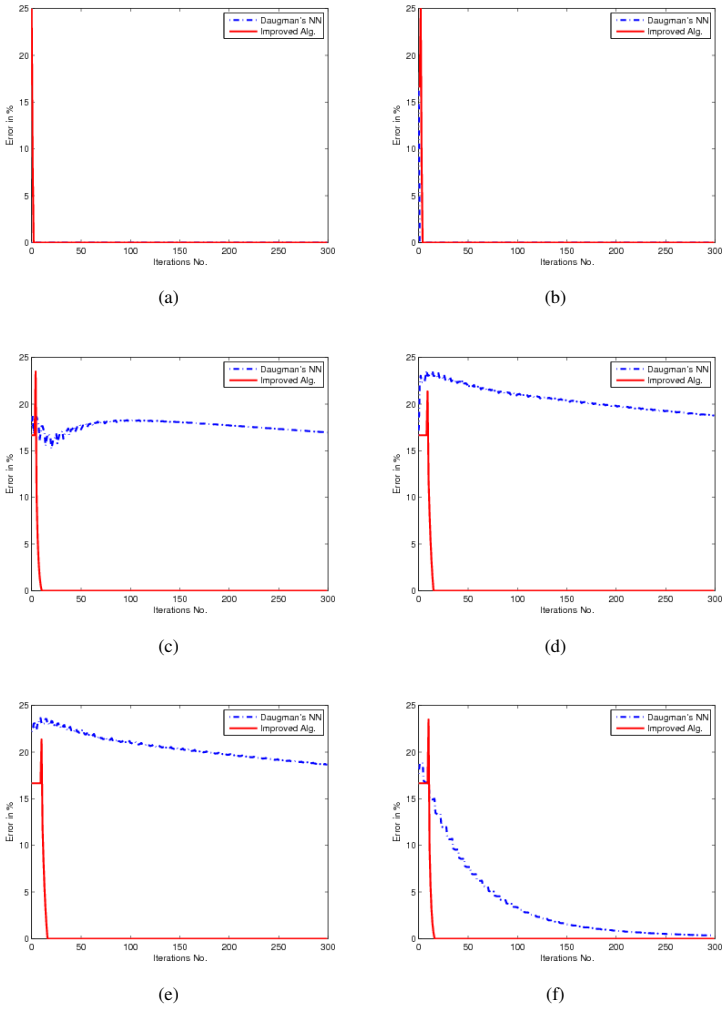


Figure 7.6: Progression of the mean error of the estimated parameters over the particular iteration steps for the implemented Daugman's NN in [MJM02] and the proposed algorithm in [SM08b]. (a-f) For the instantaneous velocity coefficients $c_1, ..., c_6$ respectively.

7.3 Saliency-Based Motion Segmentation Results

In this section, the result of applying the motion segmentation approach to three different data sets will be presented. The first data set represents a synthetically generated MVF. The second data set represents the motion of objects censored by a moving camera on a sequence of simulator framework (Simore), while the third data set describes the motion of multiple objects obtained by a stationary camera.

7.3.1 Synthetic Motion Templates

In the first data set, the segmentation approach is able to deal with noises in a synthetic MVF generated by different coefficient values with random equally distributed removal of MVs as directed in fig. 7.7 where the first image shows the results of segmenting two different motions, while the second and the third images represent the first and the second motion, respectively. In this data set, the most salient motion which is almost the same as the size of the input image can be considered a global motion of input sequence.

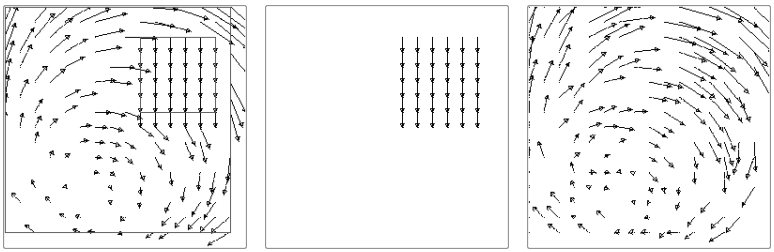


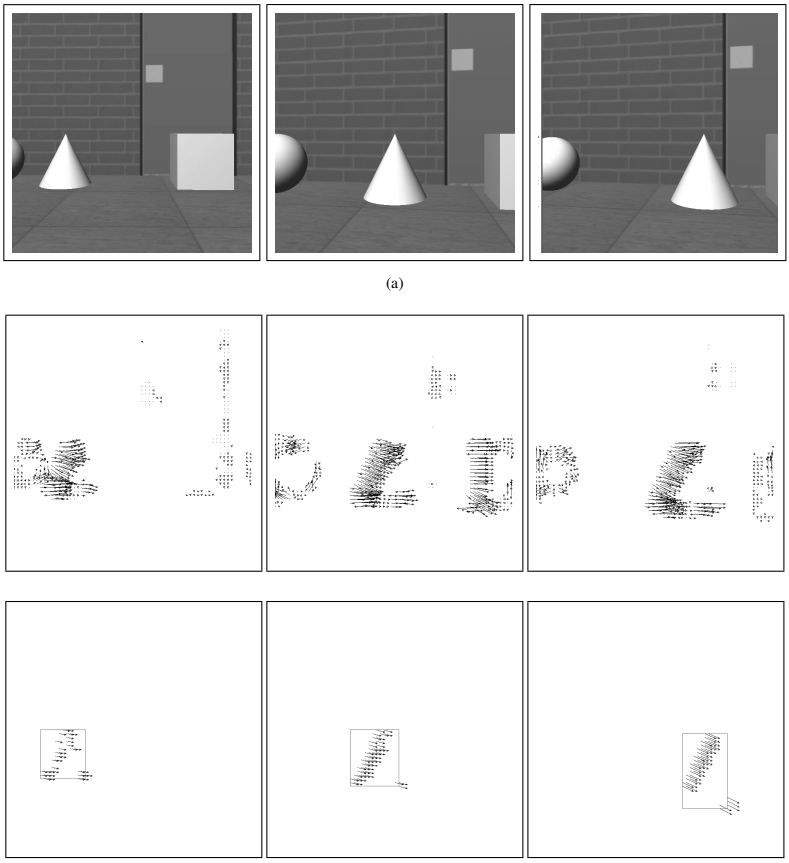
Figure 7.7: Segmentation of two different synthetic motions: (a) Result of the motion segmentation approach, (b) first motion, (c) second motion.

7.3.2 Dynamic Virtual Scene from a Moving Camera

The second data set represents a virtual environment simulating a mobile robot in a simple room which contains multiple moving objects. In this environment the simulated robot is moving forward and steering towards the left in front of a stable cube, a moving cone, and a size changeable ball. The proposed algorithm succeeds to detect the moving cone despite the effect of the ego-motion problem. The segmentation approach has shown the ability to distinguish between the most salient motion and the global motion of the MVF where the most salient motion results from the moving cone at the same time while the robot moves towards the cone as shown in fig. 7.8. The cone is faster than the robot. Hence, the VFs representing the cone have higher values than the rest of the VFs which promote the motion of the cone to be considered the most salient motion. In this case, the global motion will be defined based on the segment bounding window size relative to the image dimension. Hence, it could be used as a reference for estimating the camera ego-motion in the absence of predefined well known land marks which is vital for the extraction of static areas.

7.3.3 Dynamic Real-World Scene from a Static Camera

On the other hand, in the third data set, a sequence of real images taken from PETS dataset [PET] will be used. The segmentation approach was able to segment the movement of the two cars successfully in the first and the second segment as shown in fig. 7.9 as they represent the most two salient motions in the scene. The rest of the vectors have been segmented in afterward which means less salient values. As the segmentation approach uses the motion parameters as a homogeneity criterion, the motion of the third moving object (a person) has been merged with the middle car motion since both objects (the person and the middle car) are moving in the same direction. Due to a threshold for the segment size, the pedestrian is removed by the segmentation process. However, implementing 2D spatial constraints could succeed in separating the moving person, but it may lead to over-segmenting occluded objects. In order to correctly estimate an initial contour for the object geometry, the



(a)

(b)

Figure 7.8: Result of motion segmentation approach on a sequence of simulator framework (Simore). (a) Input sequence from a virtual mobile robot camera of moving cone. (b) Up, generated MVF. Down, representation of the most salient motion.

depth information must be available which in the case of the mobile robot can be obtained from the stereo camera head.

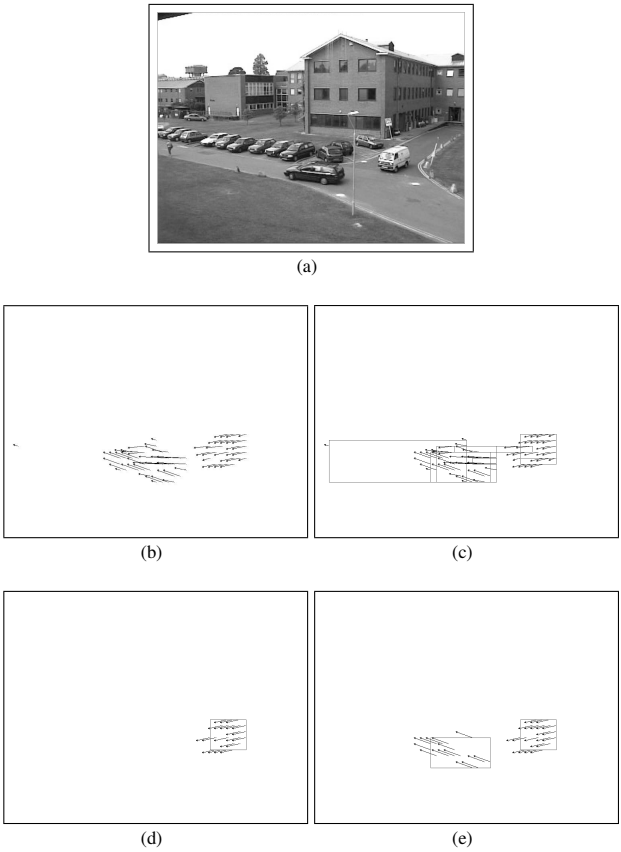


Figure 7.9: Result of motion segmentation approach on a sequence of real images. (a) Input sequence from PETS Dataset. (b) Resulting MVF. (c) Result of motion segmentation with no size limit constraints. (d) First most salient motion (1^{st}segment). (e) Second most salient motion (2^{nd}segment).

7.3.4 Performance Results

Applying the guided size control module has helped in reducing the computation time of the segmentation process in [SM08a] compared to the related segmentation approaches [MJM02, SM08b], which use 3D motion parameter coefficients as a homogeneity criterion in the absence of spatial coherence information, such as in [MJM02] and to an enhanced algorithm after improving the motion parameters estimation process in [SM08b]. The proposed segmentation algorithm in [SM08a] has shown a significant speed-up in the overall computation time for the same segmentation results as shown in fig. 7.10 where the segmentation approaches applied to four data sets, the synthetic generated MVF shown in fig. 7.7, the same synthetic MVF after application of 100% noise to each vector component, the traffic scene of PETS dataset acquired by a static camera (see fig. 7.9), and the dynamic scene of the simulator framework (SIMORE) as shown in 7.8.

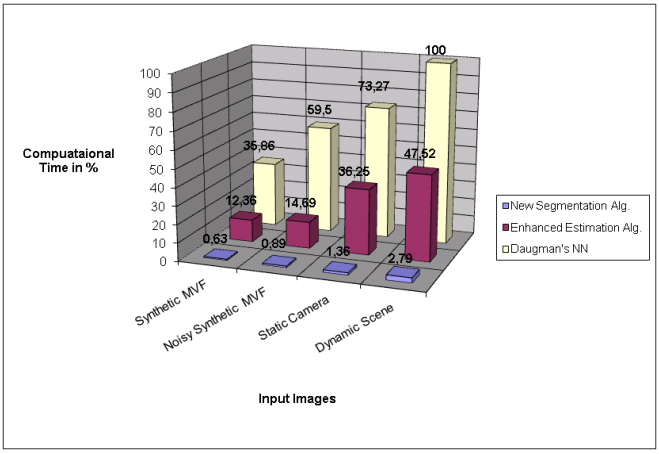


Figure 7.10: Enhancement of the computational time of the new approach of motion segmentation applied to different data sets compared to the result of the segmentation approach in [MJM02] and the improved algorithm in [SM08a].

7.4 Depth-Integrated Motion Segmentation Results

In this section, the result of applying the proposed approach in [SM11a, SM11b] to different data sets will be presented. As in the previous section, the first data set represents a synthetically generated MVF. The second data set represents a sequence of simulator framework (Simore), while the third data set describes real stereo image sequence acquired by a moving car.

7.4.1 Synthetic Motion Templates

The main advantage of the first data set which represents synthetic 3D motion templates is the availability of the ground truth data for the evaluation of the segmentation process. Fig. 7.11 shows the result of the depth-integrated segmentation approach in [SM11a] of two different motions. The first motion consists of the translation in the X and Z direction, while the second motion represents the translation in the Y direction.

In fig. 7.12, the result of 3D motion segmentation of synthetic MVFs representing the concept of transparent motion are shown. The synthetic MVFs are consist of two overlapped 3D motion which are opposite in the rotation about the Z axis. Furthermore, random noise has been applied to each vector component in order to evaluate the reliability. Each raw in fig. 7.12 represents the segmentation result of two different overlapping synthetic MVFs. The first column in the first raw of fig. 7.12 depicts the first 3D motion generated from the motion parameters $c = (0, 0, 1, 0, 0, 1)$ which consists of the translation in Z direction and the rotation about the Z axis. Similarly, the second column in the first raw shows the 3D motion generated from the motion parameters $c = (0, 0, 1, 0, 0, -1)$ which consists of the same translation in the Z direction but with an opposite rotation about the Z axis. The third column represents the overlapping 3D motions, while the forth and the fifth column represent the results of the motion segmentation process.

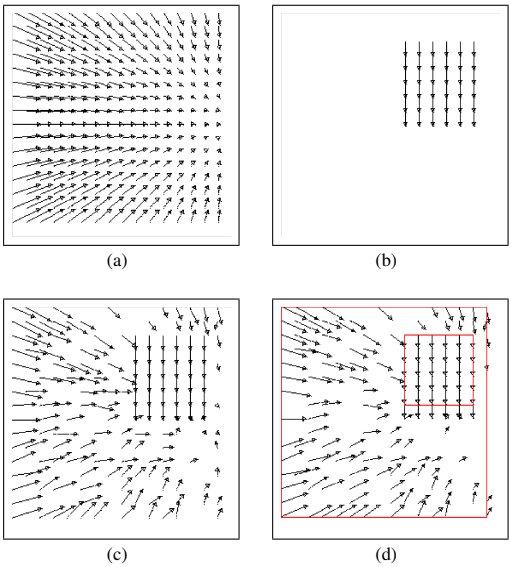


Figure 7.11: Segmentation of two different synthetic motions: (a) first motion, (b) second motion, (c) noisy MVF consists of the two previous motions, (d) result of the motion segmentation approach

The proposed approach has a significant reduction of the total iterations number required for the 3D motion segmentation process which leads to a noticeable computational time improvement. Fig. 7.13 shows the progression of the root mean square error $E_k(c(\boldsymbol{p}_m))$ over the total iteration steps k of the synthetic MVF depicted in fig. 7.11 for the proposed algorithm [SM11a] compared to the segmentation approach in [SM08a]. The behavior of the RMSE progression is dependent on the segmentation process and 3D motion parameters value of the existing 3D motions in the input MVF. The 3D motion parameters estimation process of a complex 3D motion results in wide fluctuation in the convergence curve due to the nonlinear representation of the 3D motion, e. g. the first 3D motion segment in fig. 7.13.

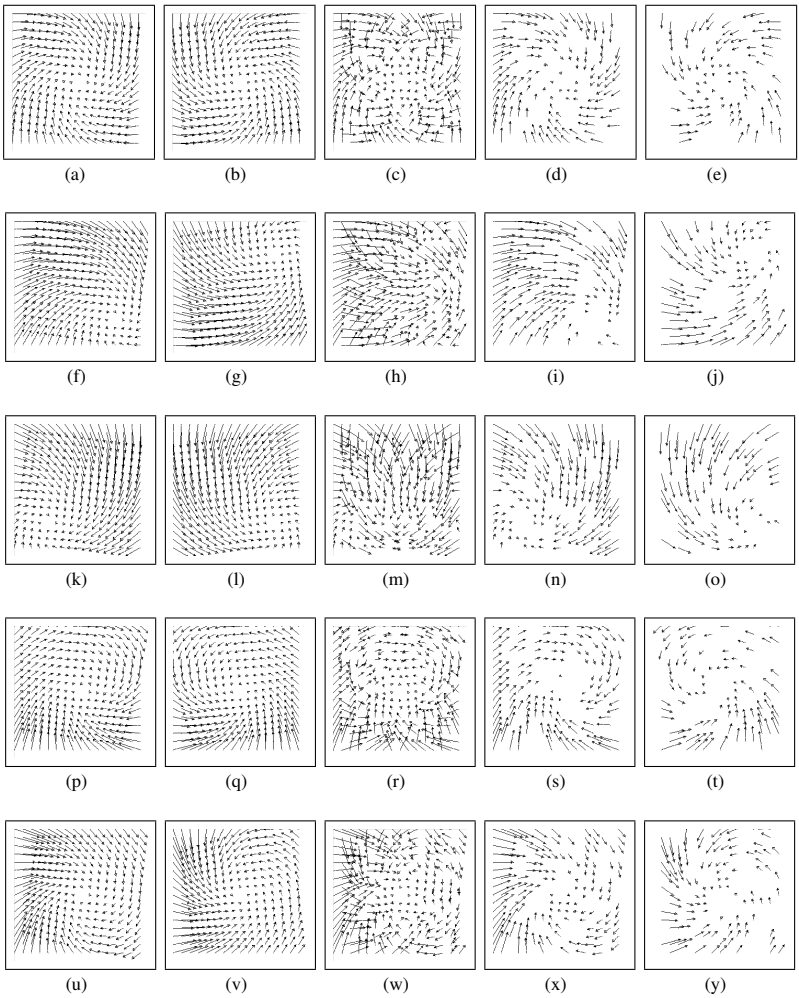
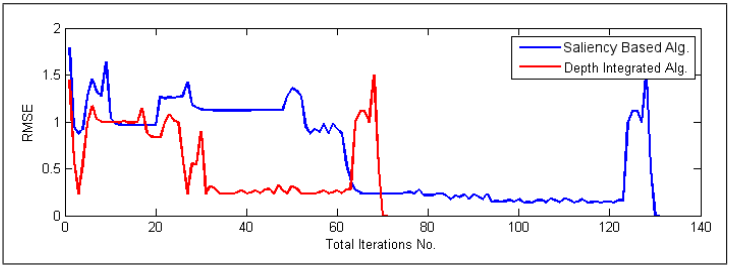
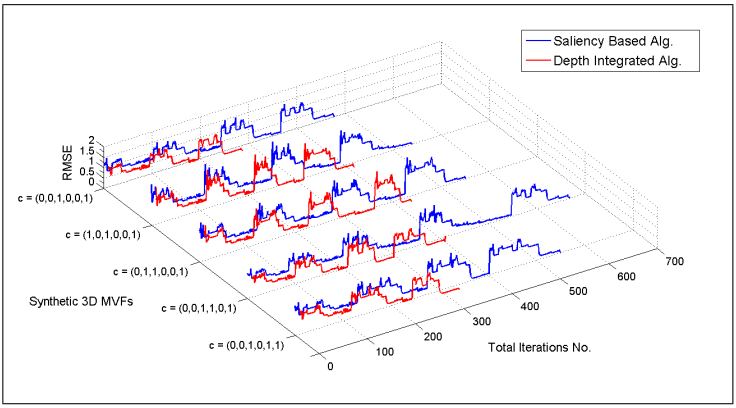


Figure 7.12: Results of the segmentation of two overlapping 3D motions: (a) The first 3D motion of the motion parameters $c = (0, 0, 1, 0, 0, 1)$. (b) Second 3D motion with opposite rotation about the Z axis ($c_6 = -1$). (c) A noisy synthetic MVF consists of the two previous motions. (d) The first resulted segment. (e) The second segment. (f-j) $c = (1, 0, 1, 0, 0, 1)$. (k-o) $c = (0, 1, 1, 0, 0, 1)$. (p-t) $c = (0, 0, 1, 1, 0, 1)$. (u-y) $c = (0, 0, 1, 0, 1, 1)$.

On the other hand, a translation motion either in the X or Y direction such as the second motion segment in fig. 7.11 is progressing fast and forward due to the linear representation of the 3D motion (more details are represented in chapter 6).



(a)



(b)

Figure 7.13: Progression of the root mean square error $E_k(c(\boldsymbol{p}_m))$ over the total iteration steps k of the previously represented synthetic MVFs for the proposed depth-integrated algorithm in [SM11a] compared to the segmentation approach in [SM08a]. (a) For the synthetic MVF of fig. 7.11. (b) For the synthetic MVFs of fig. 7.12.

7.4.2 Dynamic Virtual Scene from a Moving Stereo Camera

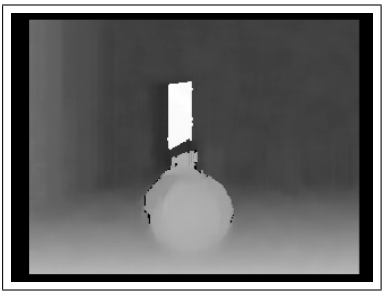
In order to correctly test and analyze the result of the proposed algorithm, a virtual environment simulating a mobile robot in a scalable complex scene is used. The first scenario in this environment a represents a moving ball in front of a mobile robot as shown in fig. 7.14. The generated depth map from the stereo images are used in the 3D MVF representation as shown in fig 7.15. The world 3D coordinate axes are depicted in fig 7.15 where the X axis is represented in red, the Y axis in blue and the Z axis in green which could be considered as a reference for the next 3D MVFs representations.

(a) (b)

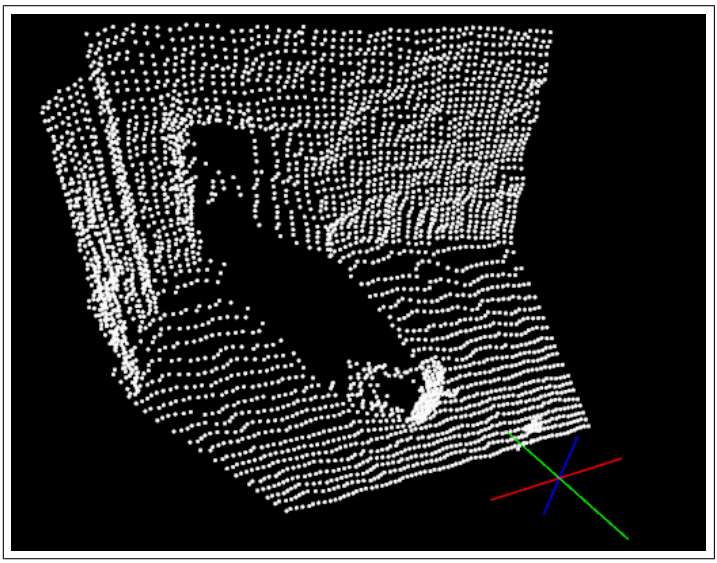
Figure 7.14: Stereo image stream representing the output of the simulated stereo camera head from the robot simulation framework (SIMORE). (a) Left image. (b) Right image.

In this scenario the ball are moving forward towards the robot and then move backward. The 2D optical flow represents the movement of the ball in the Z direction in a range of MVs pointing outwards from the center of the ball and inwards in the reverse movement. Fig. 7.16 represents the generated optical flow for the movement of the ball towards the robot.

In fig. 7.17 the 3D motion pattern which describes the forward movements in the z direction has been detected and represented by yellow vectors. On the other hand,



(a)



(b)

Figure 7.15: Construction of 3D MVF. (a) generated depth map. (b) Constructed 3D MVF.

when the ball are moving backward the MVs are represented by the default white color.

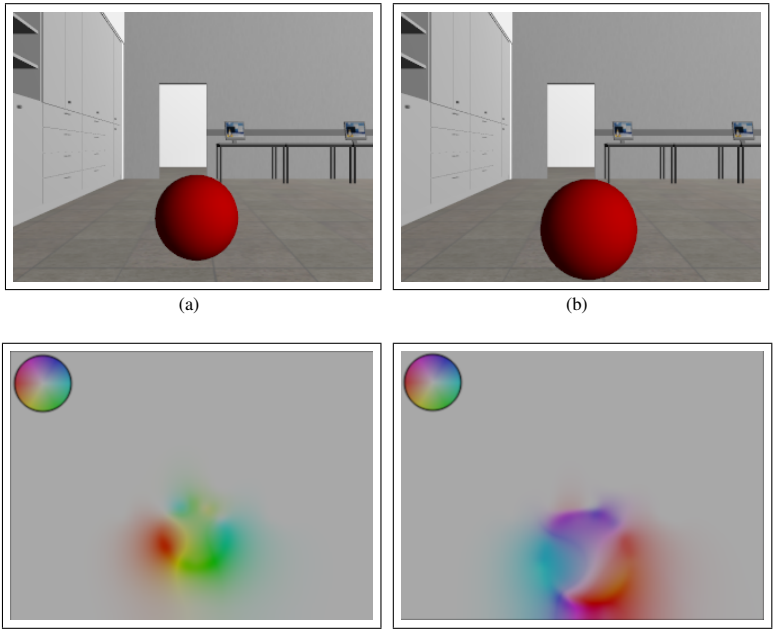
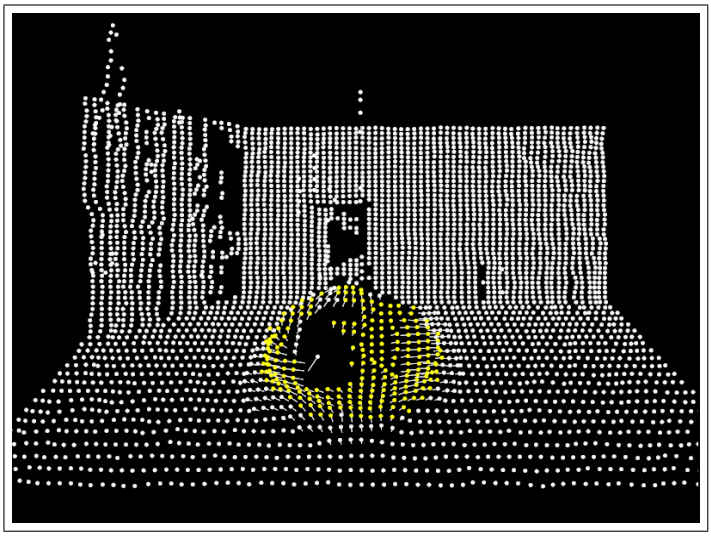
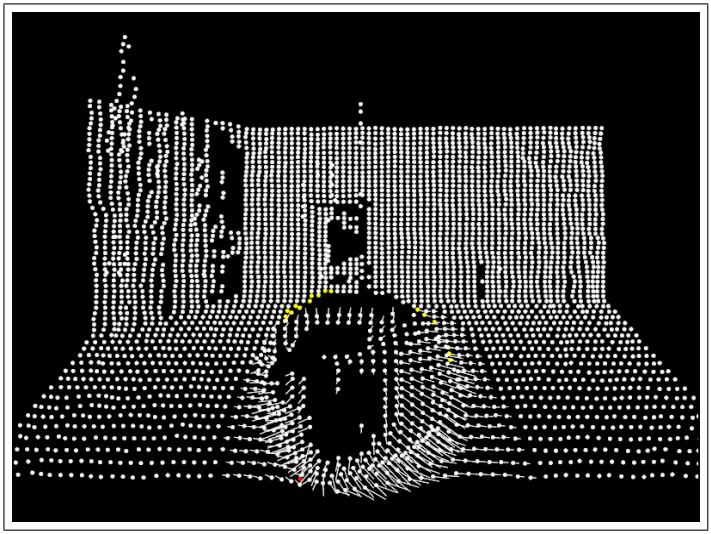


Figure 7.16: Representation of 2D optical flow. (a) The ball moves forward. (b) The ball moves backward. (c-d) The generated optical flow.



(a)



(b)

Figure 7.17: Representation of 3D MVF. (a) The ball moves forward. (b) The ball moves backward.

The complexity of the scene has been increased in the second scenario where the simulated robot is in front of a stable cube, a moving cone, and a size changeable ball as shown in fig. 7.2 while the generated depth maps from the stereo image stream for the first and the last frames in the scene and the 2D optical flow are represented in fig.7.18.

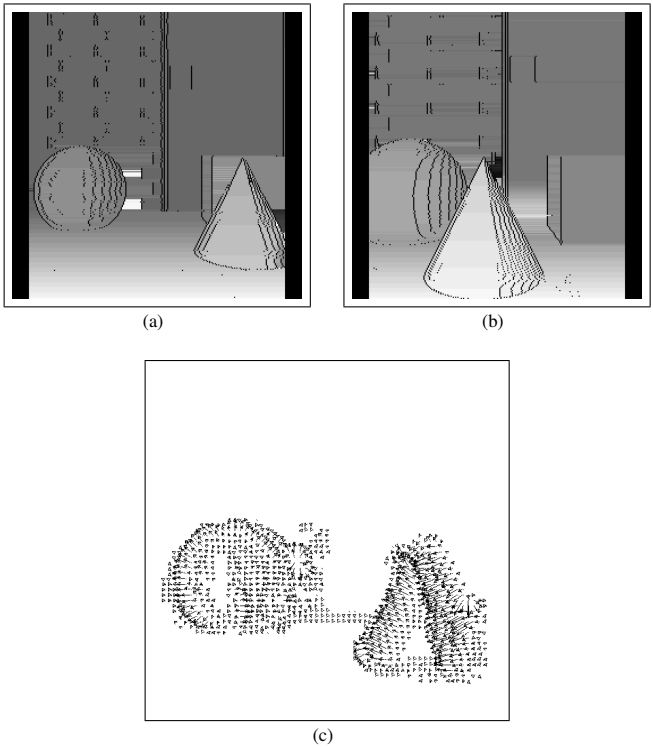


Figure 7.18: Representation of the generated depth maps and optical flow. (a) The depth map of the first image in the scene. (b) The depth map of the last image in the scene. (c) The generated 2D optical flow.

The 3D motion patterns that describes the translation in the z direction has been successfully detected and represented by yellow MVs as in the case of the increasing size of the ball. On the other hand, the cone is moving to the left and therefore the majority of its MVs are still in the default white color. Furthermore, the MVs that points towards the robot area which describes a possible collision with the robot has been represented by red as shown in fig. 7.19.

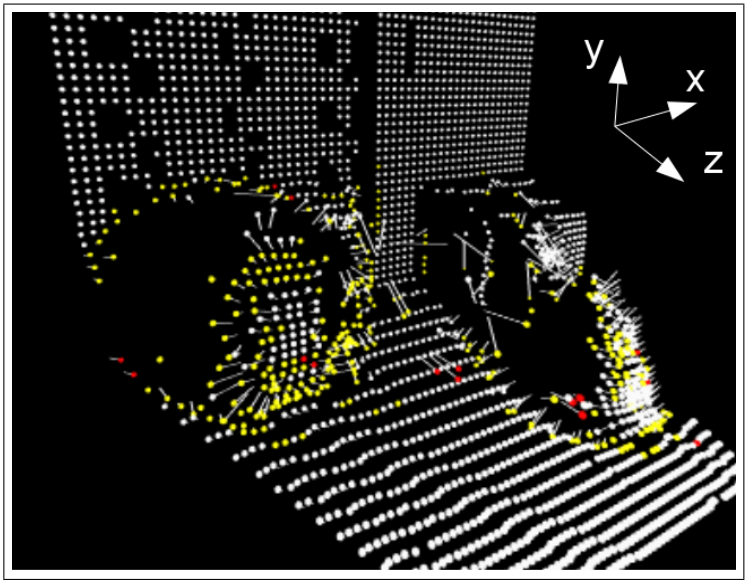


Figure 7.19: Detection of 3D motion patterns in the 3D MVF where the yellow MVs represent the translation motion in the z direction and the red MVs represent the motion towards the robot area which could be a possible collisions with the robot.

7.4.3 Dynamic Scene from a Moving Stereo Camera

This first data set is representing real stereo image sequence acquired from a stereo system mounted on a moving car [KKV+11]. In this scene, a car is entering a roundabout while a man is crossing the street. Fig. 7.20 shows the generated depth images of the roundabout scene using the SGBM algorithms [Hir06].

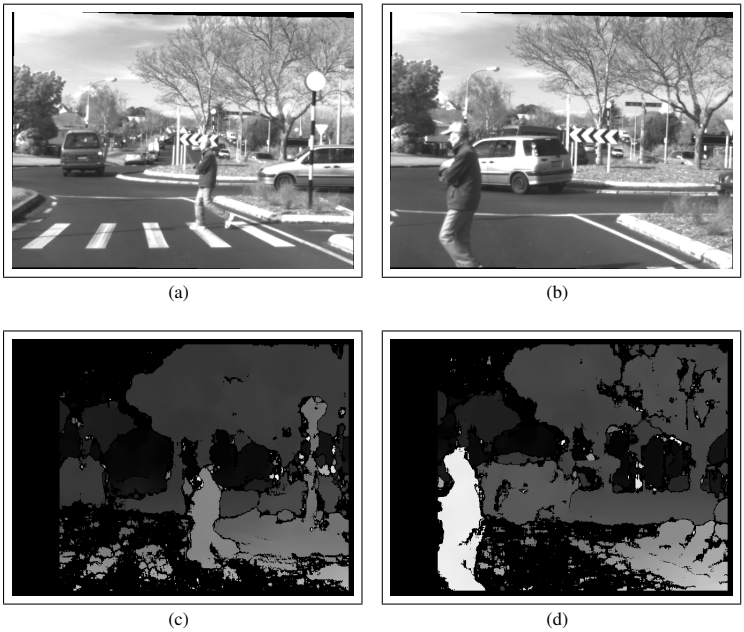


Figure 7.20: Stereo image sequence from the "roundabout" scene [KKV+11] . (a) Left image at the beginning of the sequence. (b) Left image after 40 frames. (c-d) Generated depth maps using the SGBM algorithm [Hir06].

The generated depth images are used in the 3D construction of the scene as shown in fig. 7.21. while the construction of the 3D MVFs requires the generation of the 2D optical flow as well.

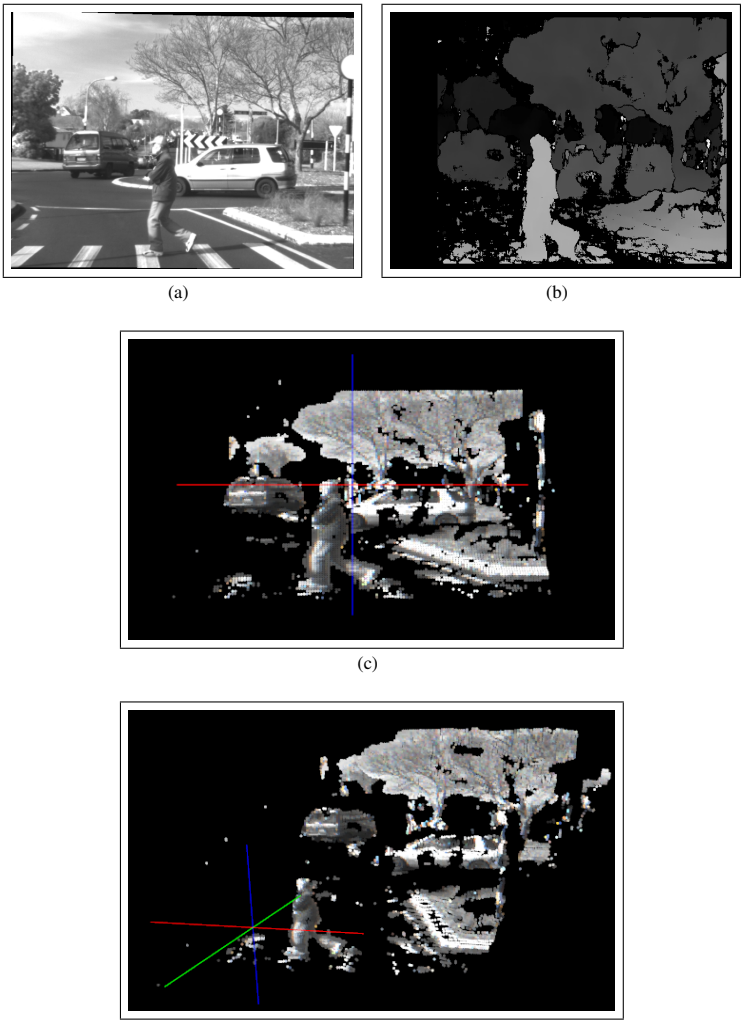


Figure 7.21: 3D construction of the scene. (a) Left image . (b) Generated depth map using the SGBM algorithm [Hir06]. (c-d) The constructed 3D scene.

Fig. 7.22 represents the constructed 3D MVF and the detected translation in the Z direction. Furthermore, MVs which lies within a certain threshold distance from the car ($d \leq \tau_z$) and pointing towards it has been represented by red which describes a possible collision with the car. On the other hand, if the MV is pointing outside the car area then it will be represented by the default white color as shown in fig. 7.22 where a pedestrian has already passed the front of the car area.

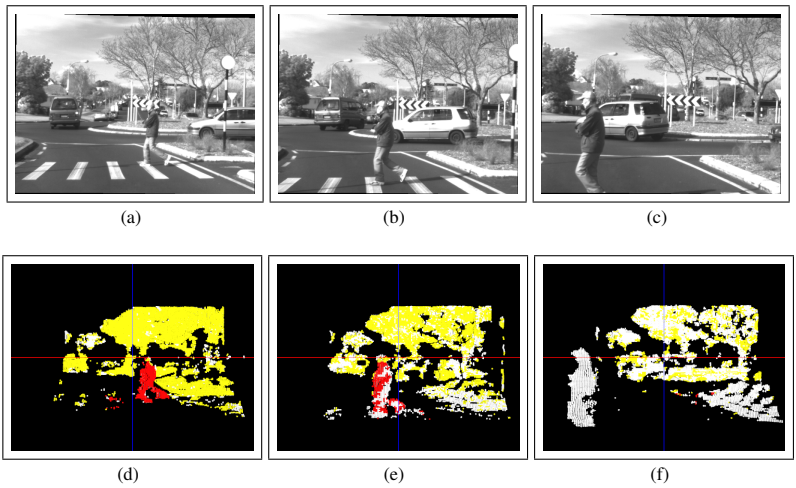
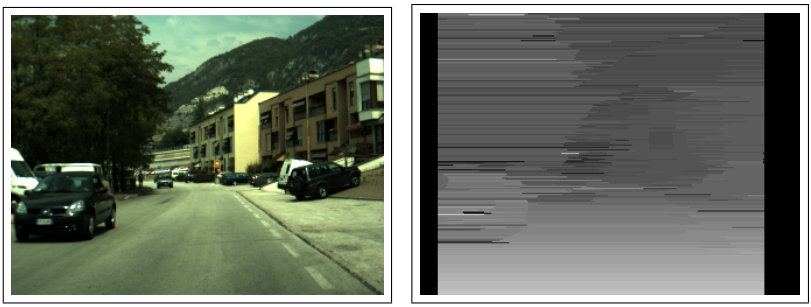


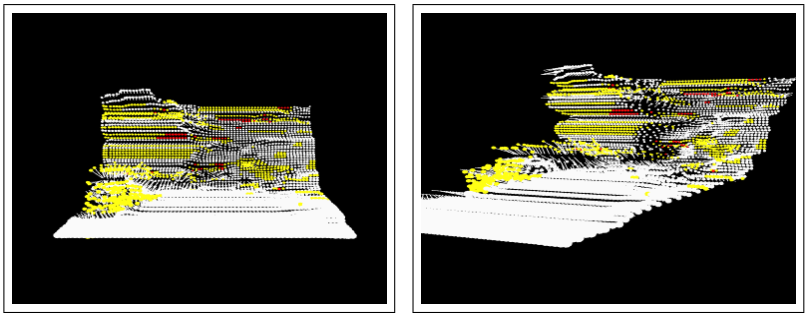
Figure 7.22: Detection of 3D Motion pattern. (a-c) Left images acquired from the mounted stereo camera. (d-e) Constructed 3D MVFs where yellow MVs represent the translation in the Z direction and the red MVs represent near MVs that point towards the car area. (f) MVs belong to the pedestrian are pointing outside the car area.

Similar to the first data set, the second data set is representing stereo image sequence acquired from camera system mounted on a car [DIP]. The proposed approach has successfully modeled the 3D spatiotemporal information from the generated depth maps detecting a predefined motion patterns that present the translation in the Z direction as in fig. 7.23 where the mounted stereo system is moving forward and the

detected possible collision were the upcoming car as well as the tree behind it and some part of the background scene.



(a)



(b)

Figure 7.23: 3D representation of MVFs generated from the DIPLODOC road stereo sequence. (a) Left, an acquired image from the mounted stereo camera. Right, the generated depth map. (b) The result of the 3D MVF representation of the proposed approach.

The 3D MVFs representation is very important to the 3D motion segmentation process, especially where the scene ground is heavily textured which results on generating reasonable amounts of MVs. Such MVs of the scene ground should not interfere with other MVs in the 3D motion segmentation process, otherwise false results will

be generated. The accurate positioning of such MVs gives the ability to easily detect and eliminate them before starting the process of 3D motion segmentation.

On the other hand, in cases where the vehicle are moving relatively fast the ego motion become the most salient motion. Hence the first motion vector field resulted from the motion segmentation approach represent the ego motion of the vehicle. Fig.7.24 represents the most salient motion resulted from the motion segmentation approach [SM11a] and a synthetic motion template representing the resulted 3D motion parameters coefficients of the most salient 3D motion segment from "roundabout" scene [KKV+11]. While fig. 7.25 represents the resulted most salient motion taken from the "DIPLODOC" image sequence [DIP].

Integrating the generated depth information into the 3D motion segmentation process [SM11a] reduced the total iterations number required for the estimation of the most salient 3D motion which leads to a noticeable computational time improvement. Fig. 7.26 shows the progression of the root mean square error $E_k(c(\boldsymbol{p}_m))$ over the total iteration steps k required to estimate the most salient 3D motion depicted in fig. 7.24 and fig. 7.25 for the proposed algorithm [SM11a] compared to the segmentation approach in [SM08a]. Taking into consideration that the progression of the RMSE is affected by the amount of noisy motion vectors exists in the input MVF, the elimination of noisy MVs (Outliers) during the segmentation process may results in converging the error curve to zero. Furthermore, fig. 7.27 shows the histogram of the average end point error Epe between the estimated motion vector (v_{est_x}, v_{est_y}) resulting from the segmentation process and the input MVs (v_x, v_y) for all MVs:

$$Epe = \frac{1}{k} \sum_{i \in k} \sqrt{(v^i_{est_x} - v^i_x)^2 + (v^i_{est_y} - v^i_y)^2} \tag{7.1}$$

where k is the total iteration number, while the result of the histogram is fit to a Gaussian curve to examine the frequency distribution of the proposed approach in [SM11a] compared to the segmentation approach in [SM08a].

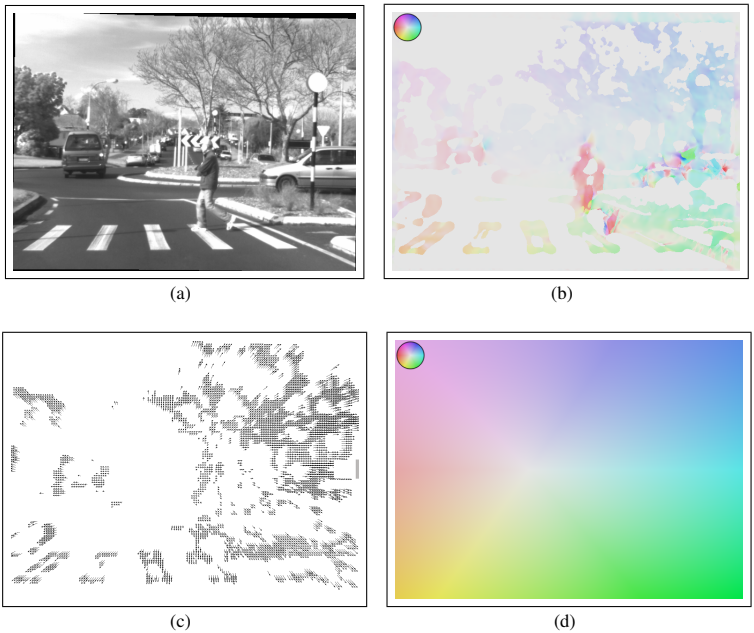


Figure 7.24: The most salient 3D motion resulted from the motion segmentation approach taken from the "roundabout" scene [KKV+11]. (a) Left image. (b) Generated optical flow. (c) the resulted most salient motion. (d) A synthetic motion template representing the 3D motion parameters coefficients of the most salient motion.

7.5 Collision Detection with the Drivable Tunnel

The drivable tunnel model represents the spatio-temporal path of the vehicle in a dynamic environment. The danger of the objects collision with the tunnel are scaled based on the distance to the vehicle from green representing less danger situation to red which represents the high level of danger. The detection process depends on the speed and the direction of the 3D motion vectors that points to the tunnel in the

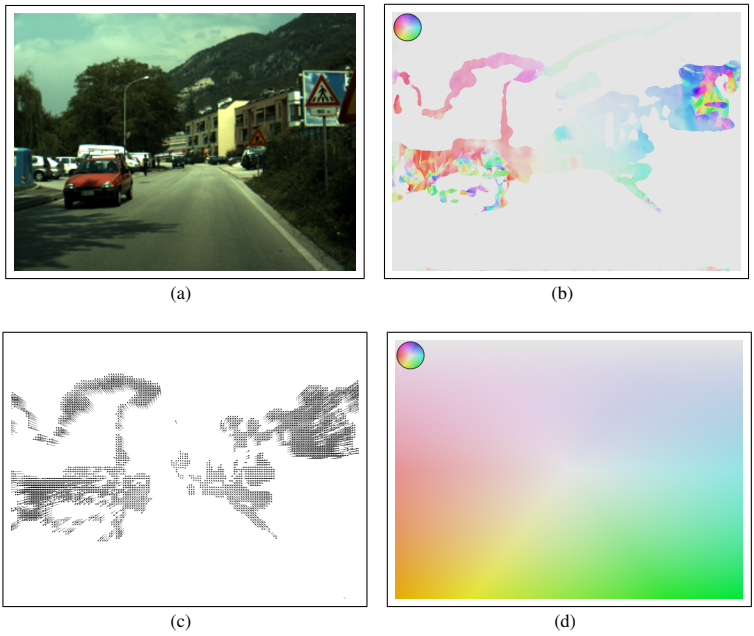
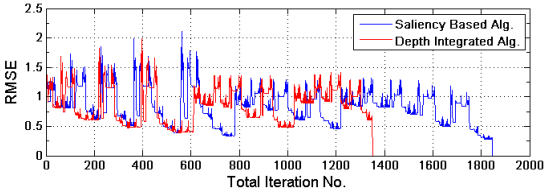
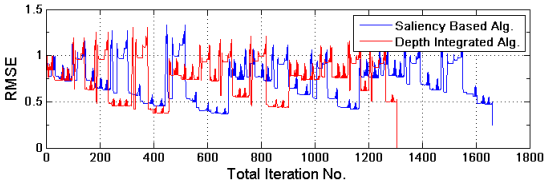


Figure 7.25: The most salient 3D motion resulted from the motion segmentation approach taken from the "DIPLODOC" image sequence [DIP]. (a) Left image. (b) Generated optical flow. (c) the resulted most salient motion. (d) A synthetic motion template representing the 3D motion parameters coefficients of the most salient motion.

direction of the vehicle taking the advantages of the relative difference between the ego-motion of the vehicle and the rest of the 3D motion vectors. Hence, a possible collision is only detected if a 3D motion vector is intersecting a plane of the drivable tunnel after δt time. Fig.7.28 shows the detection of possible collision in the virtual scene represented in fig. 7.14 for a red ball moving towards the robot drivable tunnel then crossing the tunnel. The first image represents the start position of the ball, while the second image represents the 3D motion vectors generated when the ball start



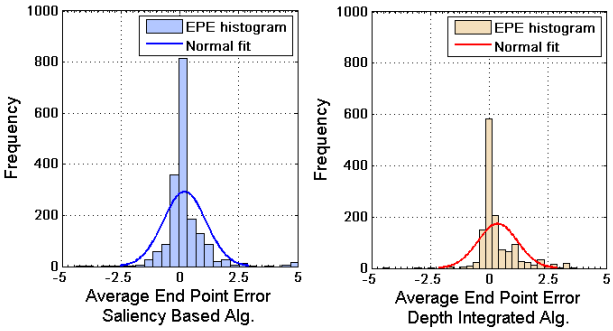
(a)



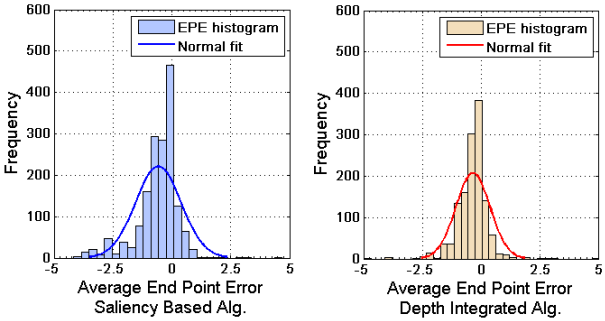
(b)

Figure 7.26: Progression of the root mean square error $E_k(c(\boldsymbol{p}_m))$ over the total iteration steps k for the proposed depth-integrated algorithm in [SM11a] compared to the segmentation approach in [SM08a]. (a) For the most salient 3D motion of the "roundabout" scene depicted in fig. 7.24. (b) For the results of the "DIPLODOC" image sequence [DIP] shown in fig. 7.25.

to move. The resulting 3D motion vector end points after δt time are intersecting with the drivable tunnel and color coded with the same tunnel plane color that the end points of the 3D motion vector are intersecting. Hence, the collision has been detected even the ball is entirely outside of the drivable tunnel. The third image shows the change of the 3D vectors color based on the tunnel plane they are intersecting. In the forth image, the 3D motion vector of the ball are pointing outside the drivable tunnel which means that the ball is moving away from the robot spatio-temporal path. In such a case there is no threat to the robot and it could be safely state that there is no collision even if a part of the ball is still inside of the tunnel.



(a)



(b)

Figure 7.27: Histogram and the normal fit of the average end point error of the resulting most salient 3D motion overall the 3D motion segmentation process in [SM11a] compared to the segmentation approach in [SM08a]. (a) Results of the most salient 3D motion of the "roundabout" scene depicted in fig. 7.24. (b) Results of the "DIPLODOC" image sequence [DIP] shown in fig. 7.25.

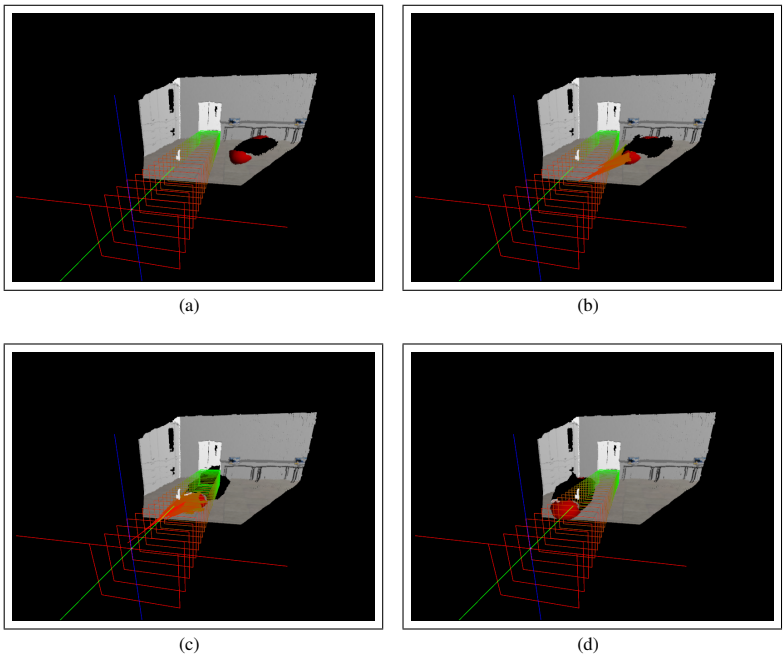


Figure 7.28: Collision detection with the drivable tunnel. (a) Start position of the ball. (b) The ball start moving in the direction of the tunnel. (c) The ball crossing the tunnel. (d) The ball is moving away from the tunnel.

The second data set is the roundabout scene represented in fig. 7.20 where a man is crossing the street while the car is moving forward, while fig. 7.29 shows the the drivable tunnel of the car in different views.

In fig. 7.30 the detection of possible collision is represented first by color coding the original input left images and second by the 3D motion vector field. The first image represents the pedestrian crossing the street, while the resulting 3D motion vectors are intersecting the the drivable tunnel in the low danger part causing a possible collision even the majority part of the pedestrian is outside the drivable tunnel. In the second

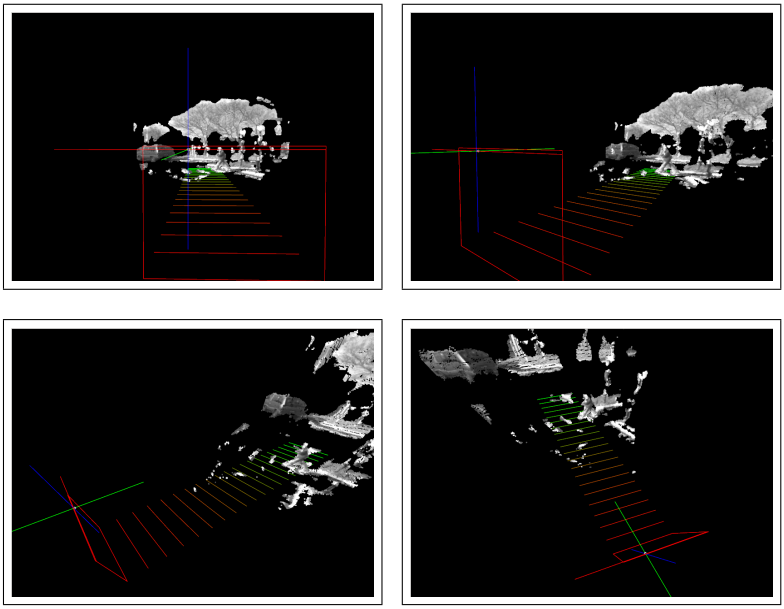


Figure 7.29: Car drivable tunnel model in different views.

and the third images, the pedestrian is moving forward as well as the car causing the 3D motion vectors of the pedestrian to intersect the drivable tunnel in a higher danger level. In the forth image, the 3D motion vectors resulting from the movement of the pedestrian are pointing outside the drivable tunnel which means at the time δt the pedestrian will be completely outside the drivable area.

7.6 Chapter Summary

This chapter has presented results of the proposed 3D motion parameter estimation algorithm, saliency-based and depth-integrated motion segmentation approaches and

the collision detection with the drivable tunnel using two different platforms. Table 7.1 represents an overall comparison of the 3D motion analysis approach with alternative systems introduced in [PN10, NVO+08, RMW+10]. The developed graphical user interface provides the capability of controlling the involved parameters to evaluate the results in the real time. The simulation framework on the other hand integrates the capability of manipulating dynamic 3D environments and scaling the complexity of the dynamic scene as well as integrating with other active vision applications [MFM+10, AM10]. A formal evaluation of the results produced by the proposed algorithms has been discussed along with a comparison with other existing algorithms. Furthermore, quantitative metrics have been applied to judge the validity of results, efficiency and the performance of the proposed approaches.

Table 7.1: Summarized comparison between the proposed 3D motion analysis approach and the alternative systems

	Proposed 3D Motion Analysis Approach	Obstacle Detection in Complex Scenarios – [PN10]	Forward Collision Detection – [NVO+08]	6D Vision – [RMW+10]
Estimation of 3D MVF	*	*		*
Temporal smoothness by KF				*
3D Motion parameters estimation	*			
Obstacle detection and separation		*	*	
Collision Detection	*	*	*	*
Drivable tunnel model	*		*	
Specialized Hardware independent	*			
Handle 3D transparent motion	*			

Figure 7.30: Collision detection with the drivable tunnel. (a) Start position of the pedestrian. (b-c) The pedestrian is crossing the street while the car is moving forward. (d) The 3D motion vectors of the pedestrian are pointing outside the drivable tunnel.

8 Conclusion

This chapter first summarizes the contributions of the new approach discussed in this dissertation and then reviews the achievements made in the field of 3D motion analysis. After that a critical discussion on the theoretical aspects of the proposed 3D motion segmentation algorithm verses the commonly used late motion segmentation is presented. After completion of the work presented here many directions have become visible that need to be investigated further for reaching an optimal model of 3D spatio-temporal motion recognition. The dissertation is concluded with indications of such directions.

8.1 Scientific Contributions

The early effort in the direction of motion analysis was in [SM08b] which is conceptually able to handle transparent motions where two or more 3D motions are grouped together to give the impression of lacy overlapping surfaces despite the connectivity of the object. In this algorithm, the estimated motion parameters serve as a homogeneity criterion for the segmentation approach and the 3D motion can be expressed as a linear combination of six component 3D vector fields. The computation of a 3D motion from a 2D image flow or a motion template finds the optimal coefficient values in a 2D signal transform. The enhanced approach for estimating 3D motion parameter coefficients from the generated MVFs [SM08b] has a great influence on reducing the computation time of the motion segmentation approach. The algorithm successfully overcomes the drawback of Daugman's transform of finding the derivative of an error with respect to each of the 3D parameter coefficients. However, the overall segmentation process still does not satisfy our requirement for fast processing algorithms. In order to speed up the segmentation process, two approaches are suggested.

The first approach is to modify the segmentation algorithm in [SM08b] to be a salient-based segmentation process. In this approach [SM08a], instead of applying the validation criterion to each vector to check whether or not it belongs to the same motion of a certain vector, it examines the unsegmented vector whether or not it belongs to the main dominant motion in a MVF. Limiting the number of segments representing the most dominant (salient) motion resulting from two consecutive frames leads not only to a great reduction in computation time but also provides the most salient motion which in most cases under certain constraints can be considered the global motion of a dynamic scene. Such information is very useful in determining the ego-motion of a camera head mounted on a mobile robot. Due to the iterative nature of the segmentation process, the computation time can be expensive in the case of several moving objects or very high noisy MVFs.

The second approach presents a fast 3D motion parameter estimation algorithm integrating the depth information [SM11a] to enhance the computational speed of the motion segmentation approach presented in [SM08a] by integrating the depth information in the 3D motion parameters estimation process. Hence, the search space has been reduced to be five dimensions which represent the rotation around the x, y, and z axes and translation in the direction of x, y axes. The resulting 3D motion parameters are used to generate and accurately positioning motion vectors of the generated depth sequence in the 3D space using the geometrical information of the stereo camera head. The resulting 3D MVF provide the ability to detect and estimate any predefined motion patterns which is vital in predicting any possible collision not only with the robot but with any objects in the observed 3D environment. The disparity map is generated using a segment-based scan line stereo algorithm presented in [SM09] which is fast and independent of the GPU power (needed for other applications).

8.2 Discussion

From the experience gained during the work in this area of research, analyzing the 3D motion of moving objects in a dynamic scene requires more than just the spa-

tial coherence of objects boundaries resulted from image segmentation. Most of the revised motion analysis algorithms tend to implement such constraints and assumptions in the motion detection process. Hence, the motion segmentation module depends on detection of the moving objects which in turn depends on the quality of the image segmentation. Although the generation of motion vector process depends on the quality of finding the corresponding features in the next frame(s) and in generals suffers from ambiguities, it is still beret than detecting moving objects based on image segmentation. In my opinion, the recognition of an object requires more than grouping similar coherent pixels in one segment. Such a process should be involved within a bigger conceptual frame such as scene understanding which requires a lot of prior information to correctly detect, recognize, then segment an object which in the end could contributes in the estimation of its motion.

On the other hand, in order to estimate and segment moving elements appear to be grouped into two or more spatially overlapping surfaces in a motion segmentation approach, the 3D motion parameters estimation process requires a multi-valued representation for each point in the image or the co-localization of more global surface descriptors. Hence, the motion segmentation algorithm will process the motion vector field as an input to estimate possible 3D motions using the motion parameter coefficients as a homogeneity criterion.

This work presents a fast depth-integrated 3D motion segmentation approach which enhanced the overall computation time of modeling 3D transparency motions. The 3D spatial localization of motion vectors implements the geometrical information of the mobile robot and the mounted stereo camera to modify the perspective transformation for accurate positioning of motion vectors in the 3D space. Moreover, the proposed approach has successfully detected and estimated predefined motion patterns describing important 3D motions such as movements toward the robot which is very helpful in detecting possible future collisions of moving objects with the robot.

8.3 Outlook

A step forward after integrating the depth information in the estimation and segmentation of the 3D motion parameters is to construct a long term spatio-temporal memory to save output of the segmentation process. Such technique will provide the capability to detect and recognize 3D spatio-temporal motion patterns which respond to certain moving behaviors of the existing objects in the dynamic environment.

Bibliography

[Adi85] ADIV, G.: Determining three-dimensional motion and structure from optical flow generated by several moving objects. In: *IEEE Trans. Pattern Anal. Machine Intell* PAMI-7 (1985), pp. 348–401

[Ahn] AHN, Song H.: *OpenGL Projection Matrix*. `http://www.songho.ca/opengl/gl_projectionmatrix.html`, (Last access: August 2012)

[AM10] AZIZ, Z.; MERTSCHING, B.: Survivor search with autonomous UGVs using multimodal overt attention. In: *IEEE International Workshop on Safety, Security and Rescue Robotics 2010*, 2010

[AS95] AYER, S.; SAWHNEY, H.: Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding. In: *IEEE International Conference on Computer Vision, ICCV 95*, 1995, pp. 777–784

[AWK+05] AGUILERA, J.; WILDENAUER, H.; KAMPEL, M.; BORG, M.; THIRDE, D.; FERRYMAN, J.: Evaluation of motion segmentation quality for aircraft activity surveillance. In: *Proceedings of the 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS'05)*. Beijing, China, October 2005, pp. 293–300

[BA91] BLACK, M.; ANANDAN, P.: Robust dynamic motion estimation over time. In: *IEEE Conference on Computer Vision and Pattern Recognition*, 1991, pp. 296–302

[BB06] BALAN, A.; BLACK, M.: An adaptive appearance model approach for model-based articulated object tracking. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR* Bd. 1, 2006, pp. 758–765

[BJ96] BLACK, M.; JEPSON, A.: Estimating optical flow in segmented images using variable-order parametric models with local deformations. In: *T-PAMI*, 1996, pp. 972–986

[BJG10] BLANCO, C. del; JAUREGUIZAR, F; GARCÍA, N.: Robust tracking in aerial imagery based on an ego-motion Bayesian model. In: *EURASIP J. Adv. Signal Process* (2010), feb., pp. 30:1–30:18

[BRC+06] BROX, T.; ROSENHAHN, B.; CREMERS, D.; SEIDEL, H.-P.: Nonparametric density estimation for human pose tracking. In: *DAGM06*. Berlin, Germany, 2006, pp. 546–555

[BRM+09] BAAK, A.; ROSENHAHN, B.; MUELLER, M.; SEIDEL, H.-P.: Stabilizing motion tracking using retrieved motion priors. In: *IEEE International Conference on Computer Vision (ICCV)*, 2009

[BT05] BARRON, J.L.; THACKER, N.A.: Tutorial: computing 2D and 3D optical flow. In: *Tina Memo* (2005), Nr. 2004-012

[BWF+05] BRUHN, A.; WEICKERT, J.; FEDDERN, C.; KOHLBERGER, T.; SCHNORR, C.: Variational optical flow computation in real time. In: *IP* 14 (2005), Nr. 5, pp. 608–615

[CG09] CHAVEZ, A.; GUSTAFSON, D.: Vision-based obstacle avoidance using SIFT features. In: *Advances in Visual Computing* Bd. 5876. 2009, pp. 550–557

[CKI97] COSTEIRA, J.; KANADE, T.; INVARIANTS, M. A.: A multi-body factorization method for independently moving objects. In: *International Journal of Computer Vision* 29 (1997), pp. 159–179

[CKS97] CASELLES, V.; KIMMEL, R.; SAPIRO, G.: Geodesic active contours. In: *International Journal of Computer Vision* 22 (1997), pp. 61–79

[CP79] CARLBOM, I.; PACIOREK, J.: Corrigenda: "geometric projection and viewing transformations". In: *ACM Comput. Surv.* 11 (1979), Nr. 3, pp. 280. – ISSN 0360–0300

[CS09] CYGANEK, B.; SIEBERT, J.P.: *An introduction to 3D computer vision techniques and algorithms*. John Wiley & Sons, Ltd, 2009

[Dau88] DAUGMAN, J. G.: Complete discrete 2-D Gabor transform by neural networks for image analysis and compression. In: *IEEE Transactions on ASSP* 36 (1988), Nr. 7, pp. 1169–1179

[DDT+06] DURANT, S.; DONOSO, A.; TAN, S.; JOHNSTON, A.: Moving from spatially segregated to transparent motion: A modelling approach. In: *Biol. Lett. 2006* 2 (2006), Nr. 1, pp. 101–105

[Der10] DERPANIS, K.: Overview of the RANSAC Algorithm. In: *Image Rochester NY* 4 (2010), pp. 2–3

[DIP] DIPLODOC: *Distributed Processing of Local Data for On-Line Car Services, a DIPLODOC road stereo sequence*. `http://tev.fbk.eu/DATABASES/road.html`, (Last access: August 2012)

[DL06] DENG, Y.; LIN, X.: A fast line segment based dense stereo algorithm using tree dynamic programming. In: *ECCV*, 2006, pp. 201 – 212

[DP91] DARRELL, T.; PENTLAND, A.: Robust estimation of a multi-layer motion representation. In: *Proc. IEEE Workshop on Visual Motion*, 1991, pp. 173–178

[FAH+08] FOSSATI, A.; ARNAUD, E.; HORAUD, R.; FUA, P.: Tracking articulated bodies using Generalized Expectation Maximization. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '08)*, 2008, pp. 1–6

[FB81] FISCHLER, M. A.; BOLLES, R. C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: *Communications of the ACM* 26 (1981), pp. 381–395

[FH84] FANG, J.-Q.; HUANG, T. S.: Solving three-dimensional small-rotation motion equations: Uniqueness, algorithms, and numerical results. In: *Computer Vision, Graphics, and Image Processing* 26 (1984), pp. 183–206

[FLV05] FOGGIA, P.; LIMONGIELLO, A.; VENTO, M.: A real-time stereo-vision system for moving object and obstacle detection in AVG and AMR applications. In: *Proc. of the Seventh Int. Workshop on Computer Architecture for Machine Perception (CAMP)*. Washington, DC, USA : IEEE Computer Society, 2005, pp. 58 – 63

[GME10] GEBHARD, Matthias; MATTES, Julian; EILS, Roland: An active contour model for segmentation based on cubic B-splines and gradient vector flow. In: NIESSEN, Wiro (Hrsg.); VIERGEVER, Max (Hrsg.): *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2001* Bd. 2208. Springer Berlin / Heidelberg, 2010, pp. 1373–1375

[GRS06] GALL, J.; ROSENHAHN, B.; SEIDEL, H.-P.: Robust pose estimation with 3D textured models. In: *IEEE Pacific-Rim Symposium on Image*

and Video Technology (PSIVT'06), Springer, LNCS 4319, 2006, pp. 84–95

[GW04] GRUBER, A.; WEISS, Y.: Multibody factorization with uncertainty and missing data using the EM algorithm. In: *Proc. of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004* 1 (July 2004), pp. 707–714

[HC05] HAJDER, L.; CHETVERIKOV, D.: Robust 3-D segmentation of multiple moving objects under weak perspective. In: *IEEE International Conference on Computer Vision Workshop on Dynamical Vision*. Beijing, 2005

[Hir06] HIRSCHMULLER, H.: Stereo vision in structured environments by consistent semi-global matching. In: *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition* (2006), June, Nr. 2, pp. 2386 – 2393. – ISSN 1063–6919

[HKW08] HAHN, M.; KRUGER, L.; WOHLER, C.: Spatio-temporal 3D pose estimation and tracking of human body parts using the shape flow algorithm. In: *19th International Conference on Pattern Recognition (ICPR '08)*, 2008. – ISSN 1051–4651, pp. 1–4

[How12] HOWARD, I.P.: *Perceiving in depth, volume 1: Basic mechanisms*. Oxford University Press, USA, 2012 (Oxford Psychology Series). – ISBN 9780199764143

[HRT+09] HASLER, N.; ROSENHAHN, B.; THORMÄHLEN, T.; WAND, M.; GALL, J.; H.-P.SEIDEL: Markerless motion capture with unsynchronized moving cameras. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*, 2009

[HS09] HOSAM, O.; SUN, X.: Three-dimensional reconstruction using enhanced shape from stereo technique. In: *IEEE/ACIS International Conference on Computer and Information Science (ICIS '09)*. Washington, DC, USA, 2009, pp. 627–632

[HTW+04] HU, W.; TAN, T.; WANG, L.; MAYBANK, S.: A survey on visual surveillance of object motion and behaviors. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 34 (2004), August, Nr. 3, pp. 334–352

[Hun05] HUNTER, J.E.: Human motion segmentation and object recognition using fuzzy rules. In: *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on* (2005), pp. 210–216

[JC10] JIAN, Y.; CHEN, C.: Two-view motion segmentation with model selection and outlier removal by RANSAC-enhanced dirichlet process mixture models. In: *International Journal of Computer Vision* 88 (2010), Nr. 3, pp. 489–501. – ISSN 0920–5691

[JF01] JOJIC, N.; FREY, B. J.: Learning flexible sprites in video layers. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2001, pp. I:199–206

[KA02] KELLER, Y.; AVERBUCH, A.: Fast gradient methods based global motion estimation for video compression. In: *International Conference on Image Processing (ICIP) 2002*. Rochester, USA, September 2002

[Kan01] KANATANI, K.: Motion segmentation by subspace separation and model selection. In: *Eighth IEEE International Conference on Computer Vision, ICCV 2001* Bd. 2, 2001, pp. 586–591

[KCC10] KIM, J.; CHUNG, M.; CHOI, B.: Recursive estimation of motion and a scene model with a two-camera system of divergent view. In: *Pattern Recognition* 43 (2010), Nr. 6, pp. 2265–2280

[KHS+08] KUTTER, O.; HILKER, C.; SIMON, A.; MERTSCHING, B.: Modeling and simulating mobile robots environments. In: *3rd International Conference on Computer Graphics Theory and Applications (GRAPP '08)*. Madeira, Portugal, January 2008, pp. 335 – 341

[KK01] KE, Q.; KANADE, T.: A subspace approach to layer extraction. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2001, pp. I: 255–262

[KKR+97] KOLLER, D.; KLINKER, G.; ROSE, E.; BREEN, D.; WHITAKER, R.; TUCERYAN, M.: Automated camera calibration and 3D egomotion estimation for augmented reality applications. In: *Computer Analysis of Images and Patterns*, 1997, pp. 199–206

[KKV+11] KLETTE, R.; KRUGER, N.; VAUDREY, T.; PAUWELS, K.; HULLE, M. van; MORALES, S.; KANDIL, F.I.; HAEUSLER, R.; PUGEAULT, N.; RABE, C.; LAPPE, M.: Performance of correspondence algorithms in vision-based driver assistance using an online image sequence database.

In: *IEEE Transactions on Vehicular Technology* 60 (2011), jun, Nr. 5, pp. 2012 –2026

[KM10] KOTTHÄUSER, T.; MERTSCHING, B.: Validating vision and robotic algorithms for dynamic real world environments. In: *Second International Conference on Simulation, Modeling and Programming for Autonomous Robot (SIMPAR)*. 2010 (LNAI 6472), pp. 97–108

[KSK06] KLAUS, A.; SORMANN, M.; KARNER, K.: Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In: *18th Int. Conf. on Pattern Recognition, ICPR 2006* 3 (2006), pp. 15 – 18. – ISSN 1051–4651

[Kum09] KUMAR, S.: Binocular Stereo Vision Based Obstacle Avoidance Algorithm for Autonomous Mobile Robots. In: *Advance computing conference, IACC 2009*, 2009, pp. 254–259

[KV05] KOLODKO, J.; VLACIC, L.: *Motion Vision: design of compact motion sensing solutions for autonomous systems navigation*. The Institution of Engineering and Technology, 2005. – ISBN 978–0–86341–453–4

[KZ01] KOLMOGOROV, V.; ZABIH, R.: Computing visual correspondence with occlusions using graph cuts. In: *Proc. Int. Conf. Computer Vision (ICCV)*, 2001, pp. 508 – 515

[LGP+02] LEFEVRE, S.; GERARD, J.; PIRON, A.; VINCENT, N.: An extended snake model for real-time multiple object tracking. In: *Proc. of Int. Workshop on Advanced Concepts for Intelligent Vision Systems*, 2002, pp. 268 – 275

[LHP80] LONGUET-HIGGINS, H. C.; PRAZDNY, K.: The interpretation of a moving retinal image. In: *Proc. R. Soc.* London, B 208, 1980, pp. 385–397

[LK81] LUCAS, B.; KANADE, T.: An iterative image registration technique with an application to stereo vision. In: *Proc. of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, 1981, pp. 674–679

[LSY06] LEI, C.; SELZER, J.; YANG, Y.: Region-tree based stereo using dynamic programming optimization. In: *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition* 2 (2006), pp. 2378 – 2385. – ISSN 1063–6919

[LW08] LIN, H.Y.; WU, J.R.: 3D reconstruction by combining shape from silhouette with stereo. In: *ICPR08*, 2008, pp. 1–4

[LZL+07] LU, Y.; ZHANG, Z.; LIU, Z.; XU, J.: Efficient motion segmentation for H.264 compressed video. In: *MIPPR 2007* 6786 (2007), Nr. 1, pp. 678638

[MAM11] MITTAL, S.; ANAND, S.; MEER, P.: Generalized projection based M-estimator: Theory and applications. In: *Computer Vision and Pattern Recognition Conference CVPR (2011)*, 2011, pp. 2689–2696

[MCK09] MA, Y.; CISAR, P.; KEMBHAVI, A.: Motion segmentation and activity representation in crowds. In: *Int. J. Imaging Syst. Technol.* 19 (2009), Nr. 2, pp. 80–90

[MFM+10] MUJAHED, M.; FISCHER, D.; MERTSCHING, B.; JADDU, H.: Closet gap based (CG) reactive obstacle avoidance navigation for highly cluttered environments. In: *IEEE/RSJ Internation Conference on Intelligent Robots and Systems*, 2010

[Miu] MIURA, Y.: *Tokyo's Shibuya district*. `http://www.japantimes.co.jp/text/fl20091011x2.html`, (Last access: August 2012)

[MJM02] MASSAD, A.; JESIKIEWICZ, M.; MERTSCHING, B.: Space-variant motion analysis for an active-vision system. In: *Advanced Concepts for Intelligent Vision Systems*. Ghent, Belgium, 2002

[MMI06] MANOR, L.; MACHLINE, M.; IRANI, M.: Multi-body factorization with uncertainty:Revisiting motion consistency. In: *IJCV* 68 (2006), Nr. 1, pp. 27–41

[MOK+10] MATSUHISA, R.; ONO, S.; KAWASAKI, H.; BANNO, A.; IKEUCHI, K.: Image-based egomotion estimation using on-vehicle omnidirectional camera. In: *International Journal of Intelligent Transportation Systems Research* 8 (2010), pp. 106–117. – ISSN 1868–8659. – 10.1007/s13177-010-0011-z

[MT96] MOHR, R.; TRIGGS, B.: Projective geometry for image analysis / International Society for Photogrammetry and Remote Sensing. Vienna, July 1996 (WG 111/2). – Forschungsbericht

[MTS07] MATTOCCIA, S.; TOMBARI, F.; STEFANO, L. D.: Stereo vision enabling precise border localization within a scanline optimization framework. In: *ACCV*, 2007, pp. 517 – 527

[MZK01] MCIVOR, A.; ZANG, Q.; KLETTE, R.: The background subtraction problem for video surveillance systems. In: KLETTE, Reinhard (Hrsg.); PELEG, Shmuel (Hrsg.); SOMMER, Gerald (Hrsg.): *Robot Vision* Bd. 1998. Springer Berlin / Heidelberg, 2001, pp. 176–183

[MZMI02] MACHLINE, M.; ZELNIK-MANOR, L.; IRANI, M.: Multi-body segmentation: Revisiting motion consistency. In: *Workshop on Vision and Modeling of Dynamic Scenes*, 2002

[NTA06] NIETHAMMER, Marc; TANNENBAUM, Allen; ANGENENT, Sigurd: Dynamic active contours for visual tracking. In: *IEEE Trans. Auto. Control* 51 (2006), pp. 562–579

[NVO+08] NEDEVSCHI, S.; VATAVU, A.; ONIGA, F.; MEINECKE, M.M.: Forward collision detection using a Stereo Vision System. In: *4th International Conference on Intelligent Computer Communication and Processing, ICCP 2008*, 2008, pp. 115 –122

[OB98] ODOBEZ, J.-M.; BOUTHEMY, P.: Direct incremental model-based image motion segmentation for video analysis. In: *Signal Processing* 66 (1998), Nr. 2, pp. 143–155

[PB06] PUNDLIK, S. J.; BIRCHFIELD, S. T.: Motion segmentation at any speed. In: *Proceedings of the British Machine Vision Conference*. Scotland, September 2006

[PET] PETS: *Performance Evaluation on Tracking and Surveillance, a benchmark database for visual surveillance, The University of Reading, UK*. `http://www.cvg.cs.rdg.ac.uk/datasets/index.html`, (Last access: August 2012)

[PN10] PANTILIE, C.D.; NEDEVSCHI, S.: Real-time obstacle detection in complex scenarios using dense stereo vision and optical flow. In: *13th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2010, pp. 439 –444

[RBW07] ROSENHAHN, B.; BROX, T.; WEICKERT, J.: Three-dimensional shape knowledge for joint image segmentation and pose tracking. In: *Int. J. Comput. Vision* 73 (2007), Nr. 3, pp. 243–262

[RMW+10] RABE, C.; MÜLLER, T.; WEDEL, A.; FRANKE, U.: Dense, robust, and accurate motion field estimation from stereo image sequences in real-time. In: DANIILIDIS, Kostas (Hrsg.); MARAGOS, Petros (Hrsg.);

PARAGIOS, Nikos (Hrsg.): *Proceedings of the 11th European Conference on Computer Vision* Bd. 6314, Springer, September 2010 (Lecture Notes in Computer Science), pp. 582–595

[RR97] ROWLEY, H. A.; REHG, J. M.: Analyzing articulated motion using expectation-maximization. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (1997), pp. 935–941

[SA03] SEIN, M. M.; ARAKAWA, Y.: Determining the depth measurement for the 3D model reconstruction. In: *SICE Annual Conference 2003*, 2003, pp. 865–868

[SAA00] SZELISKI, R.; AVIDAN, S.; ANANDAN, P.: Layer extraction from multiple images containing reflections and transparency. In: *IEEE Conference on Computer Vision and Pattern Recognition* Bd. 1, 2000, pp. 246 –253

[SFG+07] SOTELO, M.; FLORES, R.; GARCÍA, R.; OCAÑA, M.; GARCÑA, M.; PARRA, I.; FERNÁNDEZ, D.; GAVILÁN, M.; NARANJO, J.: Egomotion computing for vehicle velocity estimation. In: MORENO-DÍAZ, Roberto (Hrsg.); PICHLER, Franz (Hrsg.); QUESADA-ARENCIBIA, Alexis (Hrsg.): *EUROCAST* Bd. 4739, 2007 (LNCS), pp. 1119–1125

[SHP08] SWEARS, E.; HOOGS, A.; PERERA, A.G.A.: Learning motion patterns in surveillance video using HMM clustering. In: *IEEE Workshop on Motion and video Computing (WMVC 2008)*, 2008, pp. 1–8

[SK04] SUGAYA, Y.; KANATANI, K.: Geometric structure of degeneracy for multi-body motion segmentation. In: *Workshop on Statistical Methods in Video Processing*, 2004

[SM98] SHI, J.; MALIK, J.: Motion segmentation and tracking using normalized cuts. In: *IEEE International Conference on Computer Vision*, 1998, pp. 1154–1160

[SM06] SEKKATI, H.; MITICHE, A.: Joint optical flow estimation, segmentation, and 3D interpretation with level sets. In: *Computer Vision and Image Understanding* 103 (2006), Nr. 2, pp. 89–100

[SM08a] SHAFIK, M.; MERTSCHING, B.: Fast saliency-based motion segmentation algorithm for an active vision system. In: *Advanced Concepts for Intelligent Vision Systems (ACIVS 2008)* Bd. 5259. France, October 2008 (LNCS). – ISSN 0302–9743, pp. 578 – 588

[SM08b] SHAFIK, Mohamed; MERTSCHING, Bärbel: Enhanced Motion Parameters Estimation for an Active Vision System. In: *Pattern Recognition and Image Analysis* 18 (2008), September, Nr. 3, pp. 370 – 375. – ISSN 1054–6618

[SM09] SHAFIK, M.; MERTSCHING, B.: Real-time scan-line segment based stereo vision for the estimation of biologically motivated classifier cells. In: *KI 2009: Advances in Artificial Intelligence* Bd. 5803, 2009 (LNAI). – ISBN 978–3–642–04616–2, pp. 89 – 96

[SM11a] SHAFIK, M.; MERTSCHING, B.: Fast depth-integrated 3D motion estimation and visualization for an active vision system. In: *International Conference on Computer Vision Theory and Applications (VISAPP 2011)*. Vilamoura - Algarve, Portugal, March 2011, pp. 97 – 103

[SM11b] SHAFIK, M.; MERTSCHING, B.: Real time stereo-based biologically inspired 3D motion classifier cells. In: *The 6th IEEE Conference on Industrial Electronics and Applications (ICIEA 2011)*. Beijing, China, June 2011, pp. 91–96

[SMS00] STEIN, G.; MANO, O.; SHASHUA, A.: A robust method for computing vehicle ego-motion. In: *IEEE Intelligent Vehicles Symposium*. Dearborn, MI, October 2000

[SO06] SEKI, A.; OKUTOMI, M.: Ego-motion estimation by matching dewarped road regions using stereo images. In: *IEEE Inter. Conference on Robotics and Automation*, 2006, pp. 901–907

[SS02] SCHARSTEIN, D.; SZELISKI, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In: *Int. Journal of Computer Vision* 47 (2002), June, Nr. 1/2/3, pp. 7 – 42

[SS03] SCHARSTEIN, D.; SZELISKI, R.: High-accuracy stereo depth maps using structured light. In: *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition* 1 (2003), June, pp. 195 – 202. – ISSN 1063–6919

[SSH05] SU, Y.; SUN, M. T.; HSU, V.: Global motion estimation from coarsely sampled motion vector field and the applications. In: *IEEE Trans. on Circuits and Systems for Video Technology* 15 (2005), Nr. 2, pp. 232–242

[Sum10] SUMMERFIELD, M.: *Advanced Qt programming: Creating great software with C++ and Qt 4*. Addison Wesley Professional, 2010 (Prentice Hall Open Source Software Development Series). – ISBN 9780321635907

[SV99] SNOWDEN, R.; VERSTRATEN, F.: Motion transparency: making models of motion perception transparent. In: *Trends in Cognitive Sciences* 3 (1999), Nr. 10, pp. 369 – 377

[SWE+08] SCHMUDDERICH, J.; WILLERT, V.; EGGERT, J.; REBHAN, S.; GOERICK, C.; SAGERER, G.; KORNER, E.: Estimating object proper motion using optical flow, kinematics, and depth information. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 38 (2008), Nr. 4, pp. 1139–1151

[TMD09] TOSHEV, A.; MAKADIA, A.; DANIILIDIS, K.: Shape-based object recognition in videos using 3D synthetic object models. In: *CVPR09*, 2009, pp. 288–295

[TMS+08] TOMBARI, F.; MATTOCCIA, S.; STEFANO, L. D.; ADDIMANDA, E.: Classification and evaluation of cost aggregation methods for stereo correspondence. In: *IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*. Florida, USA, December 2008

[Tom92] TOMASI, C.: Shape and motion from image streams under orthography: a factorization method. In: *International Journal of Computer Vision* 9 (1992), pp. 137–154

[TP07] TAKALA, V.; PIETIKAINEN, M.: Multi-object tracking using color, texture and motion. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR '07*, 2007, pp. 1–7

[TSA01] TORR, P.; SZELISKI, R.; ANANDAN, P.: An integrated Bayesian approach to layer extraction from image sequences. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 23 (2001), Nr. 3, pp. 297–303

[TSL+91] TSAO, T.-R.; SHYU, H.-J.; LIBERT, J. M.; CHEN, V. C.: A Lie group approach to a neural system for three-dimensional interpretation of visual motion. In: *IEEE Trans. on Neural Networks* 2 (1991), Nr. 1, pp. 149–155

[TV07] TRON, R.; VIDAL, R.: A benchmark for the comparison of 3D motion segmentation algorithms. In: *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07*, 2007, pp. 1–8

[TYZ] TYZX: *TYZX DeepSea Stereo Camera*. `http://www.tyzx.com/products/DeepSeaG3.html`, (Last access: August 2012)

[VH04] VIDAL, R.; HARTLEY, R.: Motion segmentation with missing data by PowerFactorization and Generalized PCA. In: *IEEE Conference on Computer Vision and Pattern Recognition*, 2004, pp. 310–316

[VS03] VIDAL, R.; SASTRY, S.: Optimal segmentation of dynamic scenes from two perspective views. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2 (18-20 June 2003), pp. 281–286

[WA93] WANG, J.; ADELSON, E. H.: Layered representation for motion analysis. In: *Proc. Conf. Computer Vision and Pattern Recognition*, 1993, pp. 361–366

[Wei97a] WEISS, Y.: Motion Segmentation using EM - a short tutorial. In: *MIT E10-120, Cambridge, MA 02139, USA* (1997)

[Wei97b] WEISS, Y.: Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In: *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 1997, pp. 520–526

[WGP09] WANG, C.; GORCE, M.; PARAGIOS, N.: Segmentation, ordering and multi-object tracking using graphical models. In: *IEEE 12th International Conference on Computer Vision*, 2009, pp. 747 –754

[Win] WINTER, D.: *Starlings murmuration*. `http://www.keepturningleft.co.uk/category/blogs/`, (Last access: August 2012)

[WLG+06] WANG, L.; LIAO, M.; GONG, M.; YANG, R.; NISTER, D.: High-quality real-time stereo using adaptive cost aggregation and dynamic programming. In: *Int. Symposium on 3D Data Processing Visualization and Transmission* (2006), pp. 798 – 805

[WNL08] WU, B.; NEVATIA, R.; LI, Y.: Segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR '08*, 2008, pp. 1–8

[WRC08] WOODBECK, K.; ROTH, G.; CHEN, H.: Visual cortex on the GPU: Biologically inspired classifier and feature descriptor for rapid recognition. In: *CVPRW* (2008), June, pp. 1 – 8

[WS02] WONG, K.; SPETSAKIS, M.: Motion segmentation and tracking. In: *Proceedings of 15th International Conference on Vision Interface*, 2002, pp. 80–87

[WW11] WANG, G.; WU, Q. M.: Introduction to Structure and Motion Factorization. In: *Guide to Three Dimensional Structure and Motion Factorization*. Springer London, 2011 (Advances in Pattern Recognition), pp. 63–86

[YA07] YAMASAKI, T.; AIZAWA, K.: Motion segmentation and retrieval for 3D video based on modified shape distribution. In: *EURASIP Journal on Advances in Signal Processing* 2007 (2007), pp. Article ID 59535, 11 pages

[YEA08] YANG, Q.; ENGELS, C.; AKBARZADEH, A.: Near real-time stereo for weakly-textured scenes. In: *British Machine Vision Conference (BMVC)*. Leeds, UK, 2008

[YK10] YOON, S.; KUIJPER, A.: 3D human action recognition using model segmentation. In: *Image Analysis and Recognition* Bd. 6111. 2010, pp. 189–199

[YLC10] YAMADA, M.; LIN, Chi-Hsien; CHENG, Ming-Yang: Vision based obstacle avoidance and target tracking for autonomous mobile robots. In: *11th IEEE International Workshop on Advanced Motion Control*, 2010, pp. 153–158

[YP06] YAN, J.Y.; POLLEFEYS, M.: A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In: *European Conference on Computer Vision, ECCV06*, 2006, pp. 94–106

[YW09] YANG, S.; WANG, C.: Multiple-model RANSAC for ego-motion estimation in highly dynamic environments. In: *IEEE International Conference on Robotics and Automation (ICRA '09)*, 2009, pp. 3531–3538

[YWY+06] YANG, Q.; WANG, L.; YANG, R.; WANG, S.; LIAO, M.; NISTÉR, D.: Real-time global stereo matching using hierarchical belief propagation. In: *BMVC*, 2006, pp. 989 – 998

List of Tables

List of Figures

List of Abbreviations

BCA	Brightness Constancy Assumption
BP	Belief Propagation
DP	Dynamic Programming
EM	Expectation Maximization
FPGA	Field Programmable Gate Array
GC	Graph Cut
gpbM	Generalized Projection Based M-estimator
GPCA	Generalized Principal Component Analysis
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HMM	Hidden Markov Model
KF	Kalman Filter
KF	Kalman Filter
MDP	Mixture of Dirichlet Process
MSL	Multi-Stage Learning
MV	Motion Vector
MST	Medial Superior Temporal
MVF	Motion Vector Field
NN	Neural Network
RANSAC	Random Sample And Consensus
RBF	Regularized Radial Basis Functions
RMSE	Root Mean Square Error
ROS	Robot Operating System
SAD	Sum of Absolute Differences
SGBM	Semi Global Block Matching
SO	Scan-line Optimization
UI	User Interface

WTA	Winner Take All

List of Symbols

α	Regulation parameter
β	Regulation parameter
Γ	Label space
$\Delta E^i_{k_m}(c)$	Error deviation function between two successive iterations
$\Delta c^i_{k_m}$	Particular motion parameter deviation function
$\Delta \overline{c}^i_{k_m}$	Convergence measurement function of $\Delta c^i_{k_m}$
$\Delta f(\boldsymbol{p}_m, \boldsymbol{p}_n)$	Error vector deviation measurement function
δt	Temporal interval
δX	3D spatial movement in the X direction
δx	2D spatial movement in the x direction
δY	3D spatial movement in the Y direction
δy	2D spatial movement in the y direction
δZ	3D spatial movement in the Z direction
ζ^i	Motion segment
θ_X	Rotation motion around the X axis
θ_Y	Rotation motion around the Y axis
θ_Z	Rotation motion around the Z axis
$\kappa_{t-1}(\boldsymbol{p}_{t-1})$	Kalman filter at a feature position $\boldsymbol{p}_{t-1}$
$\vartheta_{\boldsymbol{f}}$	Vector deviation measure function
Λ	Testing criterion to check the validity of the error convergence
$\lambda(\ell_{l_i})$	Ascending function to the length of the current segment
ξ	Higher order terms of the Taylor series expansion
ξ-cell	Activation function to measure the correspondence of a *c-cell*
$\varpi(\boldsymbol{p}_0 - \boldsymbol{p})$	Updated connection weight function
$\rho(x, y)$	Positive function defined on the image plane
$\varsigma(\boldsymbol{p}_m, \boldsymbol{p}_n)$	Connection weight function between image points $\boldsymbol{p}_m$ and $\boldsymbol{p}_n$

τ	Threshold parameter
χ	Distance to the tunnel plane
$\omega(\boldsymbol{p})$	Radial symmetric weight function
$\mathbb{k}_n$	Tunnel plane
∇I	2D spatial intensity gradient
$\nabla_3 I$	3D spatial intensity gradient
B_t	Image background model
b	Distance between the stereo cameras
$C(p)$	Contour curve
C_M	Matching cost function
c-cell	Activation function to represent the instantaneous velocity
c_i	Motion parameter
$c^i_{k_m}$	Particular motion parameter
D_{c_i}	The derivative of an error function with respect to c_i
DX	Depth information function in the X direction
DY	Depth information function in the Y direction
d	Depth value
d_{isp}	Disparity value
dp	Distance of a projected point on the image plane from the origin
E	Energy function for a contour curve
$E(d_\Gamma)$	Estimated disparity map of line segment
$E^i_{k_m}(c)$	Error function between the input and the estimated motion vector
$E(w)$	Error function for an image window
E_{ext}	Applied external energy of a contour node
E_{int}	Internal energy of a contour node
Epe	End point error function
epl_y	Epipolar line
$\boldsymbol{e}_i(x, y)$	Six infinitesimal generators in form of 2D vector fields
$F_{elastic}(x_i)$	Applied force on x direction
$F_{elastic}(y_i)$	Applied force on y direction
f	Focal length

$\boldsymbol{f}_m(\boldsymbol{p})$	An error vector for image point p
$\boldsymbol{g}$	RGB color vector
$I(x, y, t)$	2D spatio-temporal image intensity function
$I(X, Y, Z, t)$	3D spatio-temporal image intensity function
I_l	Left image of a stereo input
I_r	Right image of a stereo input
I_t	Temporal intensity derivative
I_X	3D spatial intensity derivative at X direction
I_x	2D spatial intensity derivative at x direction
I_Y	3D spatial intensity derivative at Y direction
I_y	2D spatial intensity derivative at y direction
I_Z	3D spatial intensity derivative at Z direction
$\Im(\boldsymbol{p}_0)$	Image segment label of point $\boldsymbol{p}_0$
K	Transformation matrix
$\boldsymbol{K}(t)$	Spatio-temporal path
K_1	Regulation parameter for an applied force on a contour point
$L(i, i-1)$	Distance between two contour points
l_i	Image label
M	3D position matrix
$\boldsymbol{m}_t$	Measurement vector of the Kalman filter
n	Focal length in pixel
$\boldsymbol{n}$	Normal vector to the tunnel plane
$\boldsymbol{P}$	3D Image point
$\boldsymbol{p}$	2D Image point
pn	Curve node
$q^c(x, y)$	Single color channel value at point (x, y)
$\boldsymbol{S}$	Concatenated vector of the detected MVs in the image
s_x	Scaling factor in the x direction
s_y	Scaling factor in the y direction
$\boldsymbol{sp}_t(\boldsymbol{p}_t)$	A sub-pixel component of the Kalman filter
t_X	Translation motion in the X direction

t_Y	Translation motion in the Y direction
t_Z	Translation motion in the Z direction
$\boldsymbol{V}$	3D image velocity or optical flow
$\boldsymbol{v}$	2D image velocity or optical flow
$\boldsymbol{v}_{est}$	2D estimated motion vector
v_{est_x}	The x component of a 2D estimated motion vector
v_{est_y}	The y component of a 2D estimated motion vector
$\boldsymbol{v}_{opt}$	2D ideal motion vector
V_X	The X component of a 3D optical flow
v_x	The x component of a 2D optical flow
V_Y	The Y component of a 3D optical flow
v_y	The y component of a 2D optical flow
V_Z	The Z component of a 3D optical flow
w	Image window
W	Tracking Matrix