Vers un système de vision artificielle opportuniste pour l'analyse de scènes complexes à partir de caméras embarquées

Tan Khoa Mai

université PARIS-SACLAY

NNT : 2018SACLE045

VERS UN SYSTÈME DE VISION ARTIFICIELLE OPPORTUNISTE POUR L'ANALYSE DE SCENES COMPLEXES À PARTIR DE CAMÉRAS EMBARQUÉES

Thèse de doctorat de l'Université Paris-Saclay
Préparée à l'Université d'Évry Val d'Essonne

École doctorale n°580 Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat : Traitement du signal et des images

Thèse présentée et soutenue à Évry, le 13 Décembre 2018, par

Tan-Khoa Mai

Composition du Jury :

Roland Chapuis Professeur, Université Clermont Auvergne, Institut Pascal	Président
Vincent Frémont Professeur, École centrale de Nantes, LS2N	Rapporteur
Pascal Vasseur Professeur, IUT Rouen, LITIS	Rapporteur
Catherine Achard Maître de conférences HDR, UPMC Sorbonne Université, ISIR	Examinateur
Antoine Manzanera Enseignant-chercheur HDR, ENSTA Paristech	Examinateur
Guy Lebesnerais Directeur de recherche, ONERA	Examinateur
Samia Bouchafa-Bruneau Professeur, Université d'Évry Val d'Essonne, IBISC	Directrice de thèse
Michèle Gouiffès Maître de conférences HDR, Université Paris-Sud, LIMSI	Co-Directrice de thèse

Thèse de doctorat

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Vers un système de vision artificielle opportuniste pour l'analyse de scènes complexes à partir de caméras embarquées

Mots clés : flot optique, détection de surfaces planaires, odométrie visuelle

Résumé : L'objectif de ces travaux de thèse consiste à proposer un système de vision pour l'analyse de scènes dynamiques dit « opportuniste » En ce sens, il est orienté vers une tâche applicative précise et profite des connaissances à priori fournies par l'application ainsi que des indices disponibles provenant de la scène perçue (couleur, texture, géométrie) selon leur pertinence.

Cette thèse propose de pousser jusqu'au bout la vision monoculaire basée sur l'analyse d'images issues d'une caméra embarquée sur un véhicule mobile. Dans un premier temps est proposée une nouvelle méthode d'estimation du flot optique, information directement estimable à partir d'une séquence d'images. L'approche se base sur la génération d'une carte de fiabilité pour raffiner le flot optique à travers un processus itératif profitant d'informations disponibles telles que la couleur. La carte de mouvement ainsi obtenue est ensuite exploitée pour une détection rapide des plans 3D principaux. Pour cela, une approche cumulative, appelée *uv*-velocité, exploitant les propriétés géométriques du champ des vecteurs de mouvement, est développée. Elle permet de détecter les surfaces planaires en partant d'hypothèses concernant la nature de l'ego-mouvement. Contrairement à l'approche déjà existante *c*-velocité, la méthode proposée permet une stratégie de vote plus progressive qui prend en compte plus de modèles d'ego-mouvement et plus de modèles de surfaces planaires

Le modèle de mouvement de chaque surface détectée est réintégré à la méthode d'estimation du flot optique qui devient une méthode d'optimisation sous contrainte de validité du modèle planaire afin d'améliorer la précision de l'estimation du flot optique. Par ailleurs, nous montrons dans cette thèse comment un processus d'odométrie visuelle peut tirer profit de la méthode de détection de surfaces planaires.

L'approche d'estimation du flot optique est évaluée en termes de précision et de temps d'exécution sur la base de données Middlebury. En ce qui concerne la *uv*-velocité, la validation est faite aussi bien sur des flots simulés que sur des images de la base de données de KITTI.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France

Title : Towards an opportunistic artificial vision system for analysing complex scenes from onboard cameras

Keywords : optical flow, plane segmentation, visual odometry

Abstract: The thesis intends to develop the bricks of an opportunistic vision system for dynamic scene analysis, an opportunistic system that would be guided by the applicative task, that would benefit from any knowledge and prioris made available by the application, and take profit of all available cues (color, texture, geometry) depending on their quality and relevance. The context of color monocular vision is considered, with a camera embedded on a mobile platform. A dense optical flow technique is first proposed. After a rough estimation, a reliability map is computed and is used for refining the motion map, through an iterative propagation process constrained by local information, starting by the color cues.
This motion map is then analyzed for rough and fast plane segmentation. A cumulative approach called *uv*-velocity has been developed. It allows the fast exhibition of prominent planar surfaces under certain assumption related the ego-motion. Contrary to its predecessor, the so-called *c*-velocity, it allows a more progressive voting strategy, it avoids using sampling, it is not limited to translations of the camera and can detect a wider range of surfaces..
The motion models related to each surface can then be re-injected as a constraint in the estimation of the next optical flow. The raw and fast planar segmentation produced by *uv*-velocity can be used to fasten the estimation visual odometry.
The results of optical flow estimation remain acceptable in terms of precision and execution time (tested on Middleburry dataset) which can be the input for creating the voting space to detect the planes on image. After the simulations and real experiments on KITTI dataset, *uv*-velocity shows its potential to be the polyvalent image registration on plane detection and opportunistic alert for the system.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France

Acknowledgements

First of all, I would like to express my deep gratitude to my advisors, Professor Samia BOUCHAFA-BRUNEAU and Doctor Michèle GOUIFFÈS, for the valuable guidance during the past 3 years. All of the scientific advice and the technical exchanges has always been useful and allowed me to orient my works towards good direction. Your enlightenment not only helped boosting the research but also positively change my point of view on the subject. I especially thank Madame Samia for integrating me into the teaching equip. It was one of the most precious experience I've ever had. Beside, I also appreciate Madame Michèle's support for being proactive in exchanging ideas, sharing documents and correcting reports which are crucial in advancing research. I'm thankful for all of your efforts and your time to improve and correct my writing.

I would like to thank my colleagues in IBISC laboratory, especially Sara, for being a positive community to help me going through up and down life of PhD student. Thank you for all of the activities that we do together, the exchanges in which we share about life, goals and dream.

I also want to thank Université Paris Saclay for allowing me to realize this thesis. It has been the most important period of my research life . I learned a lot of experiences in doing research and teaching as well as knowledge through the formations.

Once again, I want to express my appreciation to AUF (Agence Universitaire de la Francophonie) organization for making my goal comes true. The support was important for me to come and study in France.

The last words is for my family, the big spiritual support for me during this period. Despite the distance, they have always been encouraging me and giving me strength to finish this thesis.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

General Introduction

1.1 Context

Autonomous navigation in unknown environments is a crucial task for mobile robots designed for example for surveillance, map exploration or intervention on dangerous areas. The task generally consists of three sub-tasks, *i.e* mapping, localization and navigation, which are often solved simultaneously. The research on this topic has started around 1979 [GSC79] and has greatly developed until now. The solutions of this task depend strongly on the information feedback to the robot and the environment in which the robot evolves. Beside the odometry sensor (*i.e* encoder, IMU, GPS) to keep track of robot navigation, additional sensors are used to help robot to perceive the environment. Let us cite a few examples below.

• **Infrared sensor** is designed for short range obstacle detection. Due to a limited range of sensitivity (up to 1 meter) and a high level of noise, this kind of sensor is often used as close obstacle detector [CS05]. Its capacity to provide information about 3D world is limited.

• **SONAR** [1] [TNNL02, WC00]. A low cost SONAR sensor can perceive the environment in a short cone ahead. However, its precision is relatively low due to noise introduced by the reflection of the sound signals.

• **Laser/LIDAR** [2] [LNHS05] is much more precise than SONAR. It also has higher range of measure, *i.e.* from several centimeters to about thirty meters. LIDAR can scan the whole 3D maps 360 degrees around the robot, providing a more complete information. However, the cost and power consumption of this sensor is considerably high.

The combination of multiple sensors is also an interesting idea that attracts research to improve the precision of the task [WDEH95]. Beside these sensors, the kind of sensor that

1. SONAR initially means SOund Navigation And Ranging.
2. LIDAR stands for LIght Detection And Ranging.

probably offers the wider range of applications is the camera. It can provide rich information about the environment of the robot with relatively lower energy consumption than the above-mentioned technologies. Like human vision, computer vision can exploit a large amount of information about both the scene and the robot state, which helps to perform various tasks like object recognition [PYLP11], obstacle detection [BBC+14], odometry estimation [ASO10], or scene reconstruction [SCS+10]. In the mapping, localization and navigation tasks, camera can be used independently or in conjunction with some of the above-mentioned sensors. The configuration of the vision system, either stereoscopie or monoscopic, varies from an application to the other.

• **Stereo-vision** [YgWGl12, BBH03] is one of the most common configurations. It provides a good balance between the amount of input information and the processing complexity. The stereo-vision offers the depth on the view field that brings the 3D information about the environment, like LIDAR, but the system is cheaper and lighter.

• **Mono-vision**, in contrast to the stereo-vision, is limited to the 2D image which is the projection of the 3D world. As a consequence, the application for navigation on mono-vision is limited to obstacle recognition [BBC+14] and avoidance [NBCL06]. The localization can be improved with the help of predefined landmarks [OKK11], where the position and orientation of the robot can be deduced from the captured image and the landmarks image. Another way to get the missing 3D information is to use the camera with an integrated depth sensor [FPS14] to help the localization and navigation.

With the numerous sensors and approaches, the research on autonomous navigation is abundant and various. A big part of this research uses the stereo-vision or the combination of the stereo-vision with other sensors. The pure mono-vision approaches are more rare because of the lack of depth information. A huge research for robot navigation by mono-vision is presented in Mono-SLAM [DRMS07], where the task of mapping, localization and navigation is achieved by using only one camera in real-time. However, the application requires known features on the scene at the beginning to compute the scale factor and the depth. The research field of mono-vision continues to provide interesting perspectives, because of the low energy consumption and the low weight of monoscopic sensors, which is an interesting asset for the embedding on a robot platform.

Because of the lack of 3D information, the navigation by pure mono-vision always requires to consider additional assumptions or constraints. In this thesis, we intend to develop a processing chain to support the autonomous navigation task, by achieving planar segmentation and odometry estimation for autonomous ground vehicles. Some assumptions have to be made concerning both the type of ego-motion and the geometric structure of the scene. Assuming the robot evolves in a human-made world consisting of planar surfaces (wall, building, road..), the detection of the main planes of the scene helps the robot decide about its localization, by finding the orientation and relative distance to the different planar surfaces. The map around the robot can be defined up to a scale factor.

The planar segmentation also allows robots to focus on the obstacles located in the view field by eliminating most background pixels (road, wall...). Once the relative position of the robot is determined, the odometry helps to keep track of this localization. After the stages of the initialization and calibration, the real scale factor can be retrieved, leading to the real scale map and odometry value. We call it the *mono-vision opportunistic system* because the system will take profit of the ego-motion and planes to interpret the 3D scene from the single camera. The overall structure of our opportunistic system is shown on figure 1.1. The whole system consists of three main elements :

• **Motion map estimation**. Optical flow is considered here as the key feature for scene interpretation, and analyzed further to extract structure information ;

• **Planar segmentation** by uv-velocity. This voting method allows to estimate the parameters of simplified motion models that relates velocity values to pixel coordinates. Planar segmentation is reduced to parabolas and lines detection in a 2D space. The segmentation is achieved by assigning each pixel to the closest parametric curve ;

• **Odometry estimation** benefits from the previous planar segmentation to quickly compute the homography matrix composed of the orientation and direction of the translation movement.

1.2 Contributions

The context of color monocular vision is considered, with a camera embedded on a mobile platform, for instance a vehicle. A dense optical flow technique is first proposed. In addition to the existing abundant literature related to optical flow (OF) estimation, the proposed technique produces not only the motion map but also its reliability map. This map is produced after a first rough estimation of OF, and is therefore used for refining the motion map, through an iterative propagation process guided by local information such as color cues. This reliability map marks the basic trait of the proposed opportunistic systems since it controls the refinement process and gives a higher priority for most reliable results. Since OF is the primitive information from image sequences, the estimation of the quality helps the downstream process to orient towards the good OF estimation, and eventually to reduce the output errors and calculation time.

The motion map is then analyzed for rough and fast planar segmentation. A cumulative approach called uv-velocity has been developed. It allows the fast exhibition of prominent basic planar surfaces under certain assumptions related the egomotion. This relation is represented by a parametric curve in uv-voting spaces. In short, the planar segmentation task is reduced to the estimation of 2D curve parameters. Contrary to its predecessor, the so-called c-velocity [BZ12], it allows a more progressive voting strategy, it is not limited to translations of the camera and can detect a wider range of surfaces. uv-velocity and

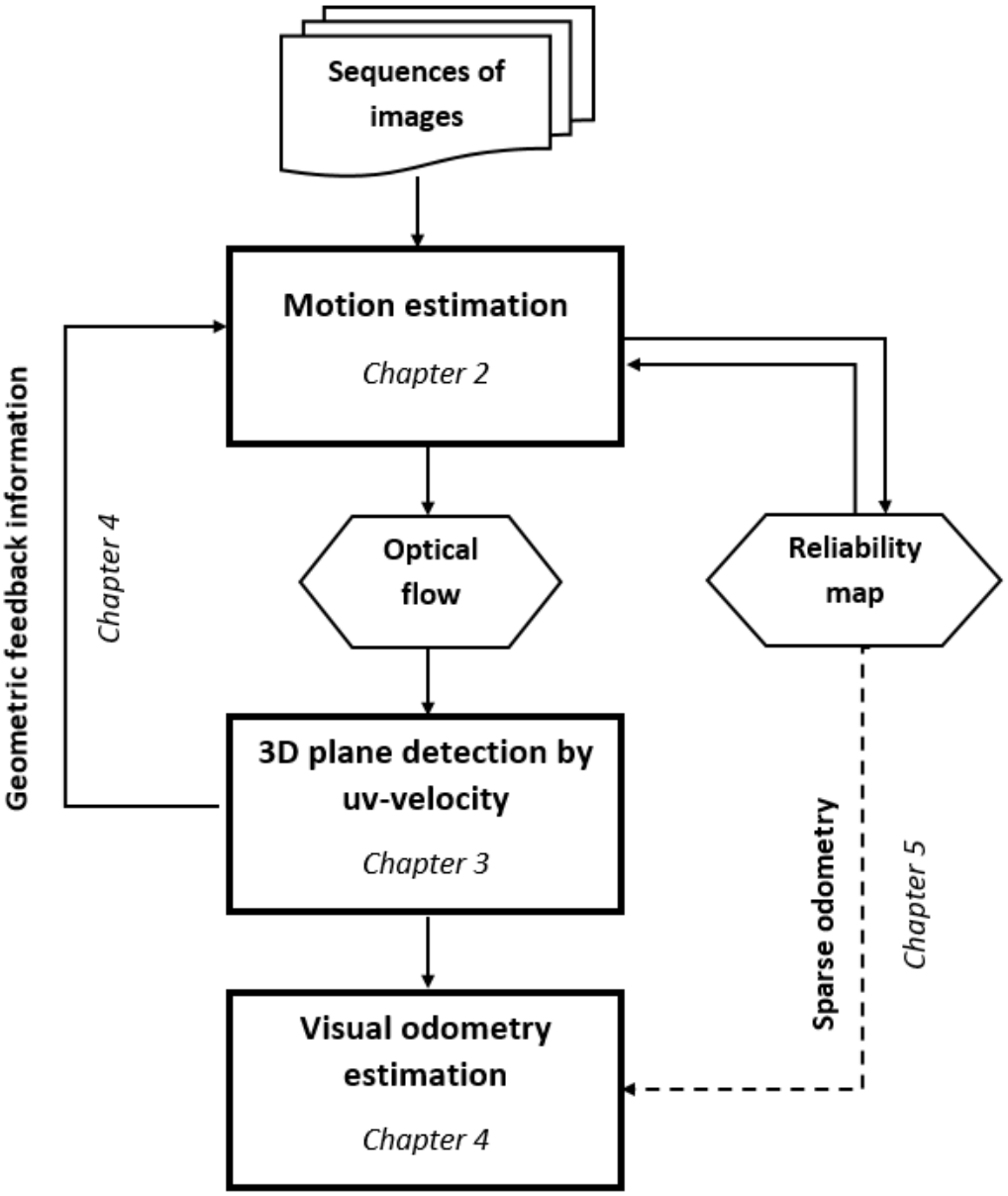


FIGURE 1.1: Overview of our opportunistic system

planar segmentation reveal the basic geometric information on the scene. The curve parameters that define the motion models related to each surface can then be re-injected as a geometric constraint in the estimation of the next OF (computed in the next frame). The raw and fast planar segmentation produced by *uv*-velocity can be used to fasten the estimation visual odometry.

1.3 Thesis outline

Chapter 2 explains our work related to optical flow estimation. After a recall of the main principles of OF estimation for monocular vision, the chapter focuses on the OF quality evaluation. It continues with the explanation of our contributions and results.

In Chapter 3, called **From Structure to motion** focuses on the planes detection from monocular vision, by exploiting the optical flow and its reliability map. After a study on the state-of-the-art techniques, the principles of the c-velocity are presented. This method is the root or our reflections, and has led to our contribution called uv-velocity. The principles of our voting spaces and their analysis are then explained.

Chapter 4, called **From structure to motion** introduces two different works, that are more exploratory. The first one tends to use the geometry information provided by the uv-velocity to improve the next OF estimation of the sequence. The second approach uses uv-velocity as a first stage for visual odometry.

Chapter 5 experiments some variants of the processing chain, by extending the proposed optical flow to a gray level input (instead of color) and by reducing the density of the motion map. The effectiveness of the new *sparse optical flow* is then tested on the subsequent planar segmentation and visual odometry estimation.

Chapter 6 concludes this work and opens a large number of perspectives.

1.4 Publications

• Optical Flow refinement using iterative propagation under color, proximity and flow reliability constraints – Journal of Visual Communication and Image Representation (under second round of review)

• Optical flow refinement using reliable flow propagation – in Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 6 : VISAPP, (VISIGRAPP 2017), pp. 451–458, 2017

• Exploiting optical flow field properties for 3D structure identification – in Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics-Volume 2 : ICINCO, pp. 459464, INSTICC, SciTePress, 2017

Chapitre 2

Optical flow refinement using iterative propagation under color, proximity and flow reliability constraints

2.1 Introduction

Cameras provide rich information that comes from the real 3D world but are also one of the most complex and versatile sensors used in robotics. Images are analyzed to make the robot detect objects as the Human Visual System can do [DT05], distinguish between different regions [DMC15] or be able to classify different scene contexts [LSD15]. When cameras move, for example when they are mounted on a moving robot, the acquired image sequences can provide additional information about the ego-motion of the camera, but also the 3D motion of moving objects as well as 3D scene depth and structure. In case of binocular systems, depth could be estimated precisely thanks to baseline, that is the fixed known distance between cameras. Even if this distance is not known, it could be estimated by achieving a calibration, which is already a well-known problem. In the monocular case, it is possible to recover depth and motion (up to a scale factor) using only one moving camera. The task of recovering this information is called "Structure From Motion", which has been studied extensively from over a century [ÖVBS17]. Beside classical methods providing 3D information of high precision, we define in this chapter the first brick of our opportunistic approach (see figure 2.1).

Our hypothesis is that under certain conditions related to scene structure and to nature of motion, it is possible to make the estimation process more robust (less prone to errors) and more efficient. The first step consists in estimating the optical flow. It could be sparse

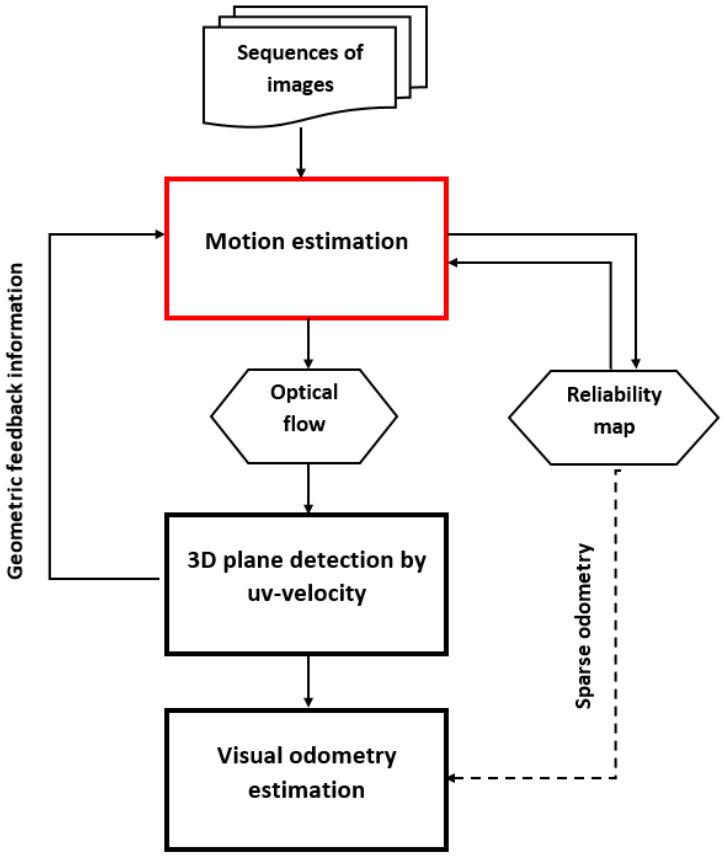


FIGURE 2.1: The developed block in our opportunistic system : motion estimation

or dense. In our case, we choose the latter possibility (or at least semi-dense) because our main structure-from-motion approach is based on a voting process. It means that we need to estimate motion at every pixel to make a voting process relevant from a statistical point of view. Moreover, we will be careful to propose, even in this first step, an estimation method that declines the concept of opportunism in two ways. In other words, the method must be able :

— to achieve a self-evaluation, by estimating the reliability of first estimations
— to evaluate the relevance of the information used (color or distribution of levels in a neighborhood) and to take benefit from it.

There are many motion estimation methods in the literature (over 154 methods on Middlebury site [1]). Motion estimation refers generally to optical flow estimation (the next section provides the definition of optical flow) and groups two categories of approaches : local and global. The global ones are mostly variants of the well-known Horn and Schunck approach [HS81]. The name "global" comes from the way the motion is estimated.

Global methods have proved to be effective and accurate. However, optical flow is estimated at every pixel while minimizing an energy function : it means that all pixels

1. http ://vision.middlebury.edu/flow/eval/results/results-e1.php

are considered with optical flow of same level of relevance. But in fact, there are pixels of which optical flow is more relevant than the others in term of precision. Moreover, knowing the estimation quality at each pixel can be exploited to refine estimates. Starting from these considerations, we propose a new optical flow estimation method that has the ability to evaluate the quality of the estimations and to propagate the "good" estimations, to correct the "bad" ones under color similarity and proximity constraints. Since it is more complicated to evaluate quality of optical flow using a global method, we will start from a local method and make it global by introducing a new propagation process.

The overview of our proposed method is shown in figure 2.2.

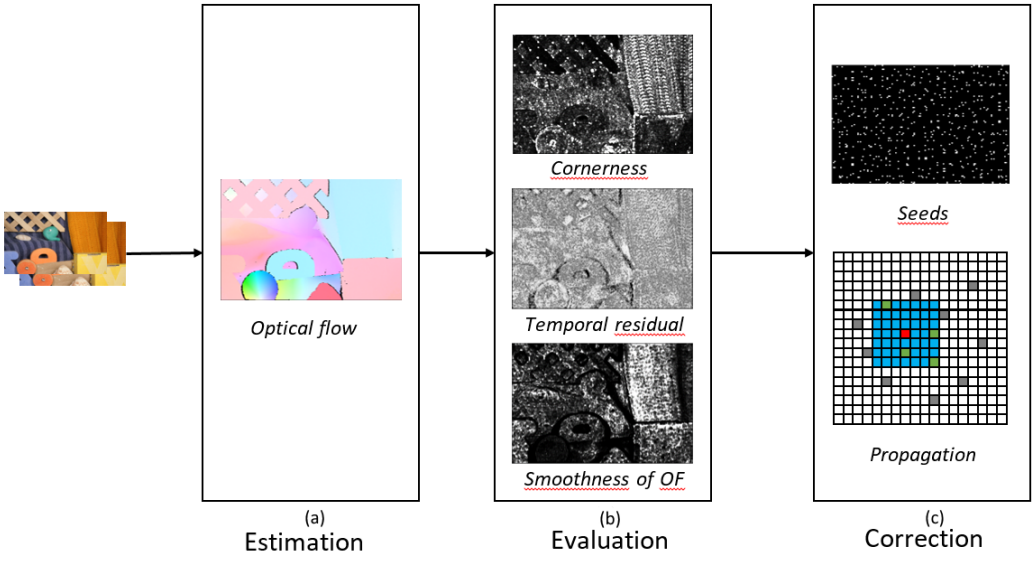


FIGURE 2.2: Overview of our method. (a) The optical flow is estimated. (b) The quality of the optical flow is evaluated. (c) The motion map is corrected iteratively in a propagation process.

In this chapter, we first present global and local optical flow estimation methods in order to point out main differences between these two categories. Then, we propose some measures to evaluate the quality of the optical flow estimation. The chapter continues with the description of our propagation process, which intends to improve optical flow precision. A numerical implementation is also given to detail the approach step-by-step. Finally, we present our experiments to assess the relevance of our quality measure and the effectiveness of our propagation progress on Middlebury dataset.

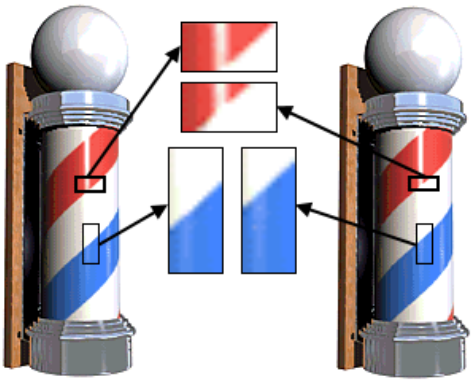


FIGURE 2.3: The barber pole is a typical example that is used to reveal the difference between optical flow and 2D motion. Even if the real 3D motion is around the Z axis, the apparent motion does not correspond to its projection in the image : red patches seem to move from right to left meanwhile blue patches move rather from bottom to top. Moreover, in this illusion, two optical flows can be observed with different observation windows : this is what is commonly called the "aperture problem".

2.2 Motion estimation using differentiation : starting from brightness constancy hypothesis

When the camera captures a moving object, the projection of 3D motion on image plane is called 2D motion while optical flow is the apparent motion, that is the 2D displacement of each pixel from time t to $t + \delta t$. 2D motion and optical flow must not be confused. The well-known barber pole optical illusion is often mentioned to show the difference between real 2D movement and optical flow (see 2.3).

Except in some special cases, optical flow is often considered as an approximation of the 2D motion on image plane. The last decades have seen the emergence of hundreds of new optical flow estimation methods. Many classifications have been proposed to group all the proposed approaches into several categories depending on a given criterion. Among them, one can classify approaches in : frequential [FZB02, SOCM01, CM87] versus spatial [NM02, LL97], sparse [KJ12, LYMD13, NRB14, BDS15, BTS17] versus dense [HW88], local [LK81] versus global [HS81], etc.. Whatever the philosophy, the brightness constancy is mostly the main hypothesis that is used since pixel intensities are the only direct information that can be extracted from images. The hypothesis assumes that pixel intensity does not vary under movement. If $I(x, y, t)$ is the intensity of a pixel at time t and $I(x + dx, y + dy, t + dt)$ its intensity after moving at time $t + dt$, brightness constancy constraint could be written :

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \tag{2.1}$$

Using Taylor expansion and neglecting the terms of order greater than 2, the equation (2.1) becomes :

$$I(x,y,t) = I(x,y,t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt, \tag{2.2}$$

which leads to :

$$\frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt = 0 \tag{2.3}$$

By dividing each term by dt, we have :

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t}\frac{dt}{dt} = 0, \tag{2.4}$$

which can be simplified as :

$$I_x\frac{dx}{dt} + I_y\frac{dy}{dt} + I_t = 0 \text{ where } I_x = \frac{\partial I}{\partial x}, I_y = \frac{\partial I}{\partial y} \text{ and } I_t = \frac{\partial I}{\partial t} \tag{2.5}$$

As a result, the equation is written as :

$$I_x\frac{dx}{dt} + I_y\frac{dy}{dt} = -I_t \tag{2.6}$$

Let us now define : $\mathbf{x} = [x,y]^\top$: a pixel position, $\mathbf{u} = [u,v]^\top = [\frac{dx}{dt}, \frac{dy}{dt}]$: the two components of optical flow on horizontal (for u) and vertical (for v) directions and $\nabla I = [I_x, I_y]^\top$. Then, the brightness constancy equation becomes :

$$\begin{aligned} I_x u + I_y v &= -I_t \\ \nabla I^T.\mathbf{u} &= -It \end{aligned} \tag{2.7}$$

The equation (2.7) is called the brightness constancy constraint or optical flow constraint. Since $\mathbf{u}$ is a vector of two components, a unique solution can not be found directly. Figure 2.4 illustrates this problem on the solution space ($[u,v]^\top$). This is a common problem called "aperture problem" in estimating optical flow (see 2.3).

To overcome the aperture problem, additional equations are needed to find a unique optical flow. One can cite two different strategies : a local method (Lukas-Kandae-Tomasi [LK81]) and a global method (Horn and Schunck [HS81]). Today, these works still influence the research on optical flow estimation. In next section, we first study how global methods estimate the optical flow. Then we explain how to start from a local approach to arrive to a global one using our new propagation process.

2.2.1 Global estimation

Besides the brightness constancy equation, Horn and Schunck [HS81] propose an additional constraint called smoothing constraint which is a kind of regularization in order

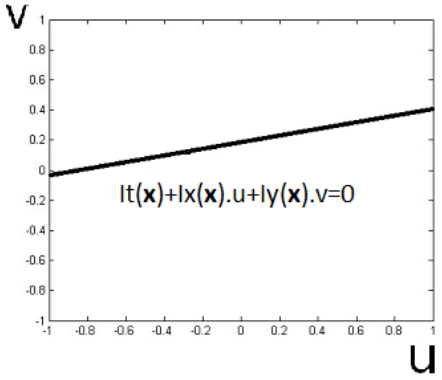


FIGURE 2.4: Illustration of brightness constancy. The optimal solution can be any point (u, v) on this line

to find a unique optical flow solution. The complete energy function on whole image Ω becomes :

$$\underset{(u,v)}{\operatorname{argmin}} E = \int_{\Omega} [I_x(\mathbf{x})u(\mathbf{x}) + I_y(\mathbf{x})v(\mathbf{x}) + I_t]^2 + \lambda(||\nabla u(\mathbf{x})||^2 + ||\nabla v(\mathbf{x})||^2)d\mathbf{x} \tag{2.8}$$

The first term of the energy function is called the data-term E_D :

$$E_D = [I_x(\mathbf{x})u(\mathbf{x}) + I_y(\mathbf{x})v(\mathbf{x}) + It]^2, \tag{2.9}$$

which corresponds to a cost function intended to satisfy the brightness constraint. It is called data-term because it embeds the image derivatives information. The second term is called the regularization modelling E_R :

$$E_R = ||\nabla u(\mathbf{x})||^2 + ||\nabla v(\mathbf{x})||^2 \tag{2.10}$$

It correspond to an hypothesis that is used in most global optical flow methods. In [HS81], the authors assume that optical flow should vary smoothly over the image. It means that the gradient of optical flow should also gradually change.

Optical flow is then considered as the solution that minimizes the energy function E. The parameter λ is a coefficient used to give more or less importance to the regularization with regard to the data term. The minimization is performed over all the pixels in the image. That is why these methods are classified as "global".

Let us detail below the numerical steps to minimize the energy E in order to find the solution. By using a numerical approximation for equation (2.8) and by setting :

$$||\nabla u(\mathbf{x})|| = \sum_{\mathbf{k} \in N(\mathbf{x})} (\alpha(\mathbf{k})u(\mathbf{k}) - \alpha(\mathbf{x})u(\mathbf{x})),$$

where N($\mathbf{x}$) is the neighborhood of pixel $\mathbf{x}$ and α is the weight of each neighbor), we can write the energy function in a different way :

$$\begin{aligned} E(u,v,\mathbf{x}) = &\sum_{\mathbf{x}\in\Omega}[I_t(\mathbf{x}) + I_x(\mathbf{x})u(\mathbf{x}) + I_y(\mathbf{x})v(\mathbf{x})]^2 + \\ &\lambda \sum_{\mathbf{k}\in N(\mathbf{x})} \left\{[\alpha(\mathbf{k})u(\mathbf{k}) - \alpha(\mathbf{x})u(\mathbf{x})]^2 + [\alpha(\mathbf{k})v(\mathbf{k}) - \alpha(\mathbf{x})v(\mathbf{x})]^2\right\} \end{aligned} \tag{2.11}$$

Since the previous function in equation (2.11) is differentiable and convex, in order to find the optimal solution, we just set the partial derivatives of this function upon u and v to zero :

$$\frac{\partial E}{u(\mathbf{x})} = [I_t(\mathbf{x}) + I_x(\mathbf{x})u(\mathbf{x}) + I_y(\mathbf{x})v(\mathbf{x})]\, I_x(\mathbf{x}) - \lambda \sum_{\mathbf{k}\in N(\mathbf{x})} [\alpha(\mathbf{k})u(\mathbf{k}) - \alpha(\mathbf{x})u(\mathbf{x})] = 0 \tag{2.12}$$

$$\frac{\partial E}{v(\mathbf{x})} = [I_t(\mathbf{x}) + I_x(\mathbf{x})u(\mathbf{x}) + I_y(\mathbf{x})v(\mathbf{x})] I_y(\mathbf{x}) - \lambda \sum_{\mathbf{k}\in N(\mathbf{x})} [\alpha(\mathbf{k})v(\mathbf{k}) - \alpha(\mathbf{x})v(\mathbf{x})) = 0 \tag{2.13}$$

The equations (2.12) and (2.13) hold for each pixel at position $\mathbf{x}$. Then, for images of dimension $W \times H$, there are $Q = HW$ pairs of equations.

The system equation made from equations 2.12 and 2.13 could be expressed in a linear form by introducing matrix notations :

$$\mathbf{A}\begin{bmatrix}\mathbf{U}\\ \mathbf{V}\end{bmatrix} = \mathbf{b} \tag{2.14}$$

where the matrices $\mathbf{b}$, $\mathbf{U}$ and $\mathbf{V}$ are defined right below. $\mathbf{U}$ and $\mathbf{V}$ are row matrices of dimensions $Q \times 1$ which contain the optical flow components u, v respectively.

$$\mathbf{b} = \begin{bmatrix} -I_x(\mathbf{x}_1)I_t(\mathbf{x}_1) \\ -I_x(\mathbf{x}_2)I_t(\mathbf{x}_2) \\ \vdots \\ -I_x(\mathbf{x}_Q)I_t(\mathbf{x}_Q) \\ -I_y(\mathbf{x}_1)I_t(\mathbf{x}_1) \\ -I_y(\mathbf{x}_2)I_t(\mathbf{x}_2) \\ \vdots \\ -I_y(\mathbf{x}_Q)I_t(\mathbf{x}_Q) \end{bmatrix} \tag{2.15}$$

$$\mathbf{U} = \begin{bmatrix} u(\mathbf{x}_1) \\ u(\mathbf{x}_1) \\ \vdots \\ u(\mathbf{x}_Q) \end{bmatrix} \tag{2.16}$$

$$\mathbf{V} = \begin{bmatrix} v(\mathbf{x}_1) \\ v(\mathbf{x}_1) \\ \vdots \\ v(\mathbf{x}_Q) \end{bmatrix} \tag{2.17}$$

Matrix $\mathbf{A}$ can be decomposed in the following way :

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{xx} & \mathbf{A}_{xy} \\ \mathbf{A}_{xy} & \mathbf{A}_{yy} \end{bmatrix} + \begin{bmatrix} \mathbf{F} & 0 \\ 0 & \mathbf{F} \end{bmatrix} \tag{2.18}$$

where the three different sub-matrices $\mathbf{A}_{xx}$, $\mathbf{A}_{yy}$ and $\mathbf{A}_{xy}$ have dimensions $Q \times Q$ and $\mathbf{F}$ is the sparse coefficient matrix of same dimensions. Their expressions are given respectively in equations (2.19) to (2.22).

$$\mathbf{A}_{xx} = \begin{bmatrix} I_x^2(\mathbf{x}_1) & 0 & \cdots & \cdots & 0 \\ 0 & I_x^2(\mathbf{x}_2) & 0 & & \vdots \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & I_x^2(\mathbf{x}_Q) \end{bmatrix} \quad (2.19) \qquad \mathbf{A}_{yy} = \begin{bmatrix} I_y^2(\mathbf{x}_1) & 0 & \cdots & \cdots & 0 \\ 0 & I_y^2(\mathbf{x}_2) & 0 & & \vdots \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & I_y^2(\mathbf{x}_Q) \end{bmatrix} \quad (2.20)$$

$$\mathbf{A}_{xy} = \begin{bmatrix} I_x(\mathbf{x}_1)I_y(\mathbf{x}_1) & 0 & \cdots & \cdots & 0 \\ 0 & I_x(\mathbf{x}_2)I_y(\mathbf{x}_2) & 0 & & \vdots \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & I_x(\mathbf{x}_Q)I_y(\mathbf{x}_Q) \end{bmatrix} \quad (2.21)$$

$$\mathbf{FU} = \begin{bmatrix} \lambda \sum_{\mathbf{k} \in N(\mathbf{x}_1)} (\alpha(\mathbf{k})u(\mathbf{k}) - \alpha(\mathbf{x}_1)u(\mathbf{x}_1)) \\ \lambda \sum_{\mathbf{k} \in N(\mathbf{x}_2)} (\alpha(\mathbf{k})u(\mathbf{k}) - \alpha(\mathbf{x}_2)u(\mathbf{x}_2)) \\ \vdots \\ \lambda \sum_{\mathbf{k} \in N(\mathbf{x}_Q)} (\alpha(\mathbf{k})u(\mathbf{k}) - \alpha(\mathbf{x}_Q)u(\mathbf{x}_Q)) \end{bmatrix}; \mathbf{FV} = \begin{bmatrix} \lambda \sum_{\mathbf{k} \in N(\mathbf{x}_1)} (\alpha(\mathbf{k})v(\mathbf{k}) - \alpha(\mathbf{x}_1)v(\mathbf{x}_1)) \\ \lambda \sum_{\mathbf{k} \in N(\mathbf{x}_2)} (\alpha(\mathbf{k})v(\mathbf{k}) - \alpha(\mathbf{x}_2)v(\mathbf{x}_2)) \\ \vdots \\ \lambda \sum_{\mathbf{k} \in N(\mathbf{x}_Q)} (\alpha(\mathbf{k})v(\mathbf{k}) - \alpha(\mathbf{x}_Q)v(\mathbf{x}_Q)) \end{bmatrix} \quad (2.22)$$

Since the dimensions of A $(2Q \times 2Q)$ and b $(2Q \times 1)$ are large, solving the linear equation (2.14) directly $\begin{bmatrix} \mathbf{U} \\ \mathbf{V} \end{bmatrix} = \mathbf{A}^{-1}\mathbf{b}$ is time consuming, especially concerning the computation of $\mathbf{A}^{-1}$. However, we can adapt previous equations in order to solve the system using an iterative process (Jacobi or Gauss-Seidel) and expect to reduce calculation time.

The Horn and Schunck method proposes a first solution to solve the aperture problem by considering an additional regularization term. However, the flow smoothing assumption is not always satisfied in practice particularly at the border between two different motions or in case of noise. In these cases, the proposed energy function tends to over-smooth optical flow, which leads to an imprecise estimation. To solve this problem, Black and Anandan [BA91] propose to change the quadratic error function to a robust error function like the Lorentzian function which is proved to have some good robustness properties and to reduce the contribution of false estimations (outliers in a given neighborhood) :

$$\rho(x) = log\left(\frac{1 + x^2}{2\sigma^2}\right) \quad (2.23)$$

By introducing a Lorentzian formulation, our energy function becomes :

$$E(u,v) = \int \rho_D[I_t + I_x(\mathbf{x})u(\mathbf{x}) + I_y(\mathbf{x})v(\mathbf{x})] + \lambda\{\rho_R(||\nabla u(\mathbf{x})||) + \rho_R(||\nabla v(\mathbf{x})||)\}d\mathbf{x} \quad (2.24)$$

In the Black and Anandan proposition, the brightness constancy and the smoothing constraint are still considered but with the introduction of a robust function in order to solve the over-smoothing problem. The key point of robust functions is that it weights out the outliers in data-terms and regularization term with W_D and W_R respectively (equation 2.26 and 2.27) in the optimization process while in the quadratic error function, outliers have the same weight than inliers.

$$\begin{aligned}
\frac{\partial E}{u(\mathbf{x})} =& W_D(u,v)[I_t(\mathbf{x}) + I_x(\mathbf{x})u(\mathbf{x}) + I_y(\mathbf{x})v(\mathbf{x})]I_x(\mathbf{x}) - \\
& \lambda \sum_{\mathbf{k}\in N(\mathbf{x})} W_R(u,v)[\alpha(\mathbf{k})u(\mathbf{k}) - \alpha(\mathbf{x})u(\mathbf{x})] = 0 \\
\frac{\partial E}{v(\mathbf{x})} =& W_D(x,y)[I_t(\mathbf{x}) + I_x(\mathbf{x})u(\mathbf{x}) + I_y(\mathbf{x})v(\mathbf{x})]I_y(\mathbf{x}) - \\
& \lambda \sum_{\mathbf{k}\in N(\mathbf{x})} W_R(u,v,\mathbf{k})[\alpha(\mathbf{k})v(\mathbf{k}) - \alpha(\mathbf{x})v(\mathbf{x})] = 0,
\end{aligned} \quad (2.25)$$

where

$$W_D(u,v) = \frac{\rho'_D[I_t(\mathbf{x}) + I_x(\mathbf{x})u(\mathbf{x}) + I_y(\mathbf{x})v(\mathbf{x})]}{I_t(\mathbf{x}) + I_x(\mathbf{x})u(\mathbf{x}) + I_y(\mathbf{x})v(\mathbf{x})} = \frac{\rho'_D(e)}{e} \quad (2.26)$$

and

$$W_R(u,v,\mathbf{k}) = \frac{\rho'_R[\alpha(\mathbf{k})v(\mathbf{k}) - \alpha(\mathbf{k})v(\mathbf{x})]}{\alpha(\mathbf{k})v(\mathbf{k}) - \alpha(\mathbf{k})v(\mathbf{x})} = \frac{\rho'_R(e)}{e} \quad (2.27)$$

Let us analyze the behavior of the Lorentzian robust function (equation 2.23) in figure 2.5 for estimating $x = 0$. The terms $W = \frac{\rho'(x)}{x}$ degrades as x goes further from the point $x = 0$, making the influence of the outliers (which favor the other points than $x = 0$) decrease. On the contrary, the weight factor of the quadratic function is constant. As a consequence, it averages all of the information around the estimation point $x = 0$, which leads to an average result.

Although the robust functions have proved to give good estimation at the border of motions, the function is not convex, so an additional iterative scheme is needed to find the optimum solution. Finally, the method consumes a lot of additional time.

From the early works of Horn and Schunck (HnS) and Black and Anandan (BA), numerous global approaches have been proposed with various energy functions and new strategies for defining data and regularization terms. For example, instead of using the robust function, [WPZ+09, WCPB09] use the absolute value in the energy function

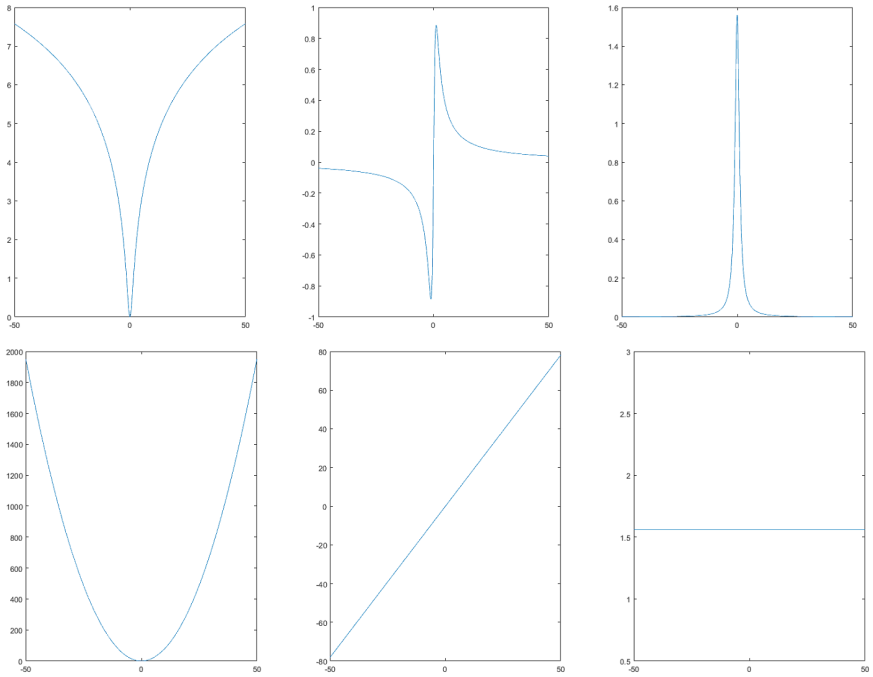


FIGURE 2.5: Illustration of the behavior of the error function $\rho(x)$ (left), the derived function $\rho'(x)$ (middle) and the term $\frac{\rho'(x)}{x}$ (right) of the Lorentzian and quadratic function (from top to bottom respectively)

instead of power 2 and besides the brightness constancy, they use in addition a texture decomposition of the image. [KLL13] adapts many data models into the optimization process since the brightness constancy is not always reliable [XJM10] also use gradient together with brightness as data cost but with edge-preserving regularization term to refine the motion at the high gradient level regions. [SVB13] proposes 3-frames approaches where smoothness of optical flow in time is introduced. [YL15] use the homography model instead of normal smoothness constraints on superpixels on image. [AWS] integrate aniotropic diffusion filter to smooth the optical flow and estimate the large displacements. [YDB12] adapt local orientation instead of horizontal and vertical ones. [BBM09, BM11, CYT11, TZ99, WRHS13, MHG15] use additional feature matching to create another data term for large movements (SIFT, DAISY, SURF). [SRB10, SRB14] segment and estimate optical flow field simultaneously by modelizing the optical flow by affine model and recently [VSR13, VRS14, VSR15, MHG18] exploit stereovision to determine the "Scene Flow" that includes also depth estimation besides motion.

The peculiarity of each method is to add an other criterion or an other hypothesis to solve the problem but they all share the same basis that starts from HnS and BA works. A quick overview of the results obtained by the optical flow methods, made available by the benchmark site Middlebury, shows that the calculation time of global methods is always an issue because of the large number of equations in the system to be solved. Some applications however do not require a dense optical flow estimation. In these cases, local methods are more suitable.

2.2.2 Local estimation

The family of local methods starts from the same assumptions as global methods, they consider however that in a given fixed neighborhood, optical flow is constant. Most popular local approaches come from the work of Lucas-Kanade [LK81] where the optical flow is estimated by minimizing an energy function made up with a sum of squared differences of brightness constancy constraints on a sub region N :

$$\underset{\mathbf{u}(\mathbf{x})}{\operatorname{argmin}} E = \sum_{\mathbf{k}\in N(\mathbf{x})} [I_t(\mathbf{k}) + \nabla I^\top(\mathbf{k})\mathbf{u}(\mathbf{x})]^2 \tag{2.28}$$

By differentiating this equation over $\mathbf{u}(\mathbf{x})$, we have :

$$\frac{\partial E}{\mathbf{u}(\mathbf{x})} = \sum_{\mathbf{k}\in N(\mathbf{x})} [I_t(\mathbf{k}) + \nabla I^\top(\mathbf{k})\mathbf{u}(\mathbf{x})]\nabla I(\mathbf{k}) = 0 \tag{2.29}$$

$$\begin{cases} \frac{\partial E}{u} & = \sum_{\mathbf{k}\in N(\mathbf{x})} [I_t(\mathbf{k}) + I_x(\mathbf{k})u + I_y(\mathbf{k})v] I_x(\mathbf{k}) = 0 \\ \frac{\partial E}{v} & = \sum_{\mathbf{k}\in N(\mathbf{x})} [I_t(\mathbf{k}) + I_x(\mathbf{k})u + I_y(\mathbf{k})v] I_y(\mathbf{k}) = 0 \end{cases} \tag{2.30}$$

The optimal solution is found by solving the linear system of equations :

$$\mathbf{A}\mathbf{u}(\mathbf{x}) = \mathbf{b} \tag{2.31}$$

where

$$\mathbf{A} = \sum_{\mathbf{k}\in N(\mathbf{x})} \nabla I^\top(\mathbf{k})\nabla I(\mathbf{k}) \tag{2.32}$$

and

$$\mathbf{b} = -\sum_{\mathbf{k}\in N(\mathbf{x})} I_t(\mathbf{k})\nabla I(\mathbf{k}) \tag{2.33}$$

Whatever the size of the considered neighborhood, matrix $\mathbf{A}$ is always of dimension 2×2 and $\mathbf{b}$ is of dimension 2×1. Solving $\mathbf{u}(\mathbf{x}) = \mathbf{A}^{-1}\mathbf{b}$ is extremely fast and precise if the two hypotheses are satisfied and the sub region has enough texture to determine the unique optical flow to avoid the aperture problem. Indeed, mathematically, only two points are required, using the brightness constancy constraint. However, we also risk to have two identical equations, which leads to a singularity of matrix $\mathbf{A}$ that makes the inversion $\mathbf{A}^{-1}$ impossible to compute. Therefore, the choice of the window size is critical. When it is too small, the aperture problem is not solved and the flow can not be estimated. When it is too large, the hypothesis of an homogeneous optical flow over the region is no longer valid, leading to inaccurate estimations. In [ST94], the authors introduce a strategy to find "good features to track". The criterion to select these points is detailed in the next Section 2.3.

From the developments of the classical KLT method, many authors try to overcome the difficulties of local methods like in [BA91, BHS98] where the initial quadratic error is replaced by a more robust error function and a weight windows function w on the sub region to reduce the influence of points far from the center (equation 2.34). Like global methods, the window function applied in this case helps removing and reducing the impact of outliers on the estimation to give more robust and accurate optical flow.

$$\underset{\mathbf{u}(\mathbf{x})}{\operatorname{argmin}} E = \sum_{\mathbf{k}\in N(\mathbf{x})} w(\mathbf{k})\rho[(I_t(\mathbf{k}) + \nabla I^\top(\mathbf{k})\mathbf{u}(\mathbf{x}))] \tag{2.34}$$

An another way to solve the energy function is to form the tensor structure where a single matrix represents the spatio-temporal information on a sub-region. A large number of researchers focus on the analysis of the tensor structure. For example, in [FBYJ00, Far00, Far03, BA96], a parametric model of higher order is used by integrating color information. [NG98, BW02, MN01, LCR03] investigate adaptive local neighborhoods while Nagel et al. [MN01] analyze the tensor structure in order to segment the image into regions where flows can be estimated. Brox et al. [BW02] propose to propagate nonlinear structure tensor in order to preserve the discontinuity by reducing the influence of the neighbors for which the gradient magnitude is high.

Calling $\mathbf{v} = [u, v, 1]^\top$, the sum of squared error functions of optical flow constraints to minimize is re-formulated from equations (2.7) and (2.28) to have :

$$\underset{\mathbf{v}(\mathbf{x})}{\operatorname{argmin}} E = \sum_{\mathbf{k}\in N(\mathbf{x})} \left(\begin{bmatrix} I_x(\mathbf{k}) & I_y(\mathbf{k}) & I_t(\mathbf{k}) \end{bmatrix} .\mathbf{v}(\mathbf{x}) \right)^2 \tag{2.35}$$

The associated Euler-Lagrange equation is then written :

$$\mathcal{L}(\mathbf{v}(\mathbf{x}), \lambda) = \sum_{\mathbf{k}\in N(\mathbf{x})} \left(\begin{bmatrix} I_x(\mathbf{k}) & I_y(\mathbf{k}) & I_t(\mathbf{k}) \end{bmatrix} \mathbf{v}(\mathbf{x}) \right)^2 - \lambda(||\mathbf{v}(\mathbf{x})||^2 - 1) \tag{2.36}$$

We can differentiate the (2.36) by $\mathbf{v}$ and λ to find to optimal solution :

$$\frac{\partial \mathcal{L}}{\mathbf{u}} = \sum_{\mathbf{i}\in N(\mathbf{x})} \left(\begin{bmatrix} I_x(\mathbf{k}) & I_y(\mathbf{k}) & I_t(\mathbf{k}) \end{bmatrix}' \begin{bmatrix} I_x(\mathbf{k}) & I_y(\mathbf{k}) & I_t(\mathbf{k}) \end{bmatrix} \mathbf{v}(\mathbf{x}) \right) - \lambda \mathbf{v}(\mathbf{x}) = 0 \tag{2.37}$$

$$\frac{\partial \mathcal{L}}{\lambda} = ||\mathbf{v}(\mathbf{x})||^2 - 1 = 0 \tag{2.38}$$

The matrix $\mathbf{T}(\mathbf{x}) = \sum_{\mathbf{i}\in N(\mathbf{x})} \begin{bmatrix} I_x(\mathbf{k}) & I_y(\mathbf{k}) & I_t(\mathbf{k}) \end{bmatrix}^\top \begin{bmatrix} I_x(\mathbf{k}) & I_y(\mathbf{k}) & I_t(\mathbf{k}) \end{bmatrix}$ in equation

2.37 is called structure tensor $\mathbf{T}(\mathbf{x})$ of an image patch. The equation (2.37) can be re-written as :

$$\mathbf{T}(\mathbf{x})\mathbf{v}(\mathbf{x}) = \lambda\mathbf{v}(\mathbf{x}) \tag{2.39}$$

So according to (2.39), $\mathbf{v}(\mathbf{x})$ has to be the minimum eigenvector of $\mathbf{T}(\mathbf{x})$ to minimize (2.35). Without loss of generality, let us order the eigenvalues as $0 \leq \lambda_1 \leq \lambda_2 \leq \lambda_3$ and call the 3 corresponding eigenvectors $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$. Four cases have to be investigated once the eigenvalues and eigenvectors are computed :

1) $\lambda_1 = \lambda_2 = \lambda_3 = 0$: there is no gradient in the image patch. Hence no motion can be estimated.

2) $\lambda_1 = \lambda_2 = 0, \lambda_3 > 0$: there is not enough spatial structure information on image patch to reliably estimate the optical flow. So, only the normal flow can be recovered (line flow) :

$$\mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix} = -\frac{\mathbf{e}_3(3)}{\mathbf{e}_3(1)^2 + \mathbf{e}_3(2)^2} \begin{bmatrix} \mathbf{e}_3(1) \\ \mathbf{e}_3(2) \end{bmatrix} \tag{2.40}$$

3) $\lambda_1 = 0, \lambda_2 > 0, \lambda_3 > 0$: the spatial and temporal information is sufficiently coherent to precisely estimate the optical flow (point flow) :

$$\mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix} = -\frac{1}{\mathbf{e}_1(3)} \begin{bmatrix} \mathbf{e}_1(1) \\ \mathbf{e}_1(2) \end{bmatrix} \tag{2.41}$$

4) $\lambda_1, \lambda_2 > 0, \lambda_3 > 0$: the spatial and temporal information vary in all directions. Therefore no motion can be estimated.

Between the two ways of finding the optimal solution, experiments show that solving the optical flow field by a gradient descent approach (KLT's choice) with the incremental refinement produces better results than using the structure tensor where the increment refinement considers an image deformed by the previous estimation.

Whatever the choice of the approach to solve the optimization problem, local methods generally face similar problems, namely : the choice of windows size ; the invalid brightness constancy hypothesis in practice ; the imprecision of the estimation when the optical flow is estimated at borders between different motions ; and the drifted estimation when the minimization process is not able to converge.

Next section is devoted to the evaluation of the quality of the optical flow. Let us recall that our strategy is to study precisely the KLT method and its main disadvantages in order to propose a new approach, that should answer some of the main problems highlighted in previous sections.

2.3 Evaluating the quality of the optical flow

Evaluating the quality of an OF estimation is useful when it can be used for further application. Somehow, there is not much methods to evaluate the quality of optical flow estimation and if they exist, the evaluation is done *a priori* like a prediction for the quality of the estimation. The most common criterion is the cornerness measure presented together with KLT method [ST94] under the term "good features to track" :

$$\mathbf{J} = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} => w_{corner} = min(\lambda_1, \lambda_2) \tag{2.42}$$

This corner measure is more meaningful than the gradient alone since it carries information about the structure of the patch around a pixel. Hence, it is used in our work and included in our reliability criterion.

In Section 2.2.2, the development of optimization by structure tensor mentions four cases revealing whether the optical flow can be estimated. Turning out the structure tensor can also allow to predict the quality of the estimation. In [JHG99, LCR03], the authors present a corner confidence measure which is used to adapt the window size of the neighborhood :

$$\alpha = \left(\frac{\lambda_3 - \lambda_1}{\lambda_3 + \lambda_1}\right)^2 - \left(\frac{\lambda_3 - \lambda_2}{\lambda_3 + \lambda_2}\right)^2, \tag{2.43}$$

where $0 < \lambda_1 < \lambda_2 < \lambda_3$ are eigenvalues of structure tensor matrix.

This criterion, compared to the cornerness measure above-mentioned, includes an additional temporal information. So, the measure reveals theoretically more precisely the quality of the optical flow but it requires the computation of the structure tensor, which is not available in the optical flow estimation by KLT method. Therefore, this criterion will be compared independently to the criteria introduced below.

An alternative technique consists in a learning approach. In [KMG08], the authors try to learn optical flow patch models. The motion of each patch is matched to the most similar model using the Mahalanobis distance. Meanwhile, [AHPB13] try to learn the specific feature extracted from images in order to evaluate the confidence of the estimated optical flow. Because of the learning , the reliability is only meaningful for the chosen estimation methods.

Both methods require learning, either the flow distribution in the first case or the customized feature in the second case, which limits the generality of the approach.

In our work, by using the KLT method to estimate the optical flow, we open the door to two additional criteria in order to evaluate the quality of estimation :

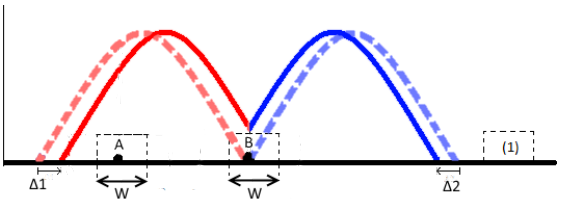


FIGURE 2.6: Illustration of the difference between the distributions of local estimated optical flow on point A and B. The dotted line is the profile of intensity from the previous instant (I_1) while the continuous line represents the same profile after motion in (I_2). The abscissa axis represents the coordinates of the points and the ordinate axis represents their intensity. Two objects in the profile with different movements (Δ_1,Δ_2) collide at point B. The window (1) is the size of neighborhood to estimate the displacement at each point.

2.3.1 The motion local uniformity

As mentioned previously, one of the disadvantages of KLT methods is the bad estimation of large movements. This can be solved partially by using a pyramidal representation. Another difficulty related to optical flow is the estimation at frontiers between two or more different movements. figure 2.6 illustrates this problem in a 1-D signal by showing the intensity profiles of two different objects, before (in dotted lines) and after motion (in solid lines). After the local estimation, the motions around point **A** found in the neighborhood W would be uniform and close to the Δ_1 while the motions around point **B** are scattered between Δ_1 and Δ_2.

Moreover, figure 2.7 illustrates the idea of the variance criterion on two images of the sequence Venus of database MiddleBury and its optical flow map with four points noted (a) to (d) chosen for illustrative purpose. The distribution of estimated movements around these points are shown in the figure 2.8. On the one hand, the points (a) and (b) correspond to areas of uniform motion where the estimated optical flows around that point converge around the brightness constraint. On the other hand, the points (c) and (d) illustrate the optical flow computation for non-uniform motion, with more or less color homogeneousness. The nearer the point is to motion frontiers (figure 2.8c,d), the more scattered the estimations are, and vice-versa (figure 2.8a,b).

To address this issue, the first criterion evaluates the local uniformity of the estimated results. Let S be the set of accounted estimated movements around the point $\mathbf{x}$. The reliability s_{var} of estimated flow at that point is formed by computing the variance of S :

$$s_{var}(\mathbf{x}) = \frac{1}{\sigma_S^2(\mathbf{x}) + \epsilon} \tag{2.44}$$

where $\sigma_S(\mathbf{x})$ is the standard deviation of S and ϵ is a small value used to avoid a zero at denominator. The variance reliability score w_{var} is then calculated by normalizing the

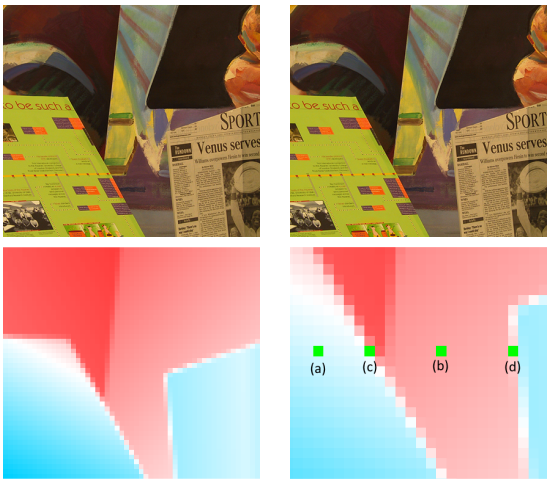


FIGURE 2.7: The image pair (top) and its motion map – color-encoded ground truth – (bottom-left) of the sequence Venus from the MiddleBury database. The bottom-right image zooms on one particular region of ground truth of which 4 pixels are investigated.

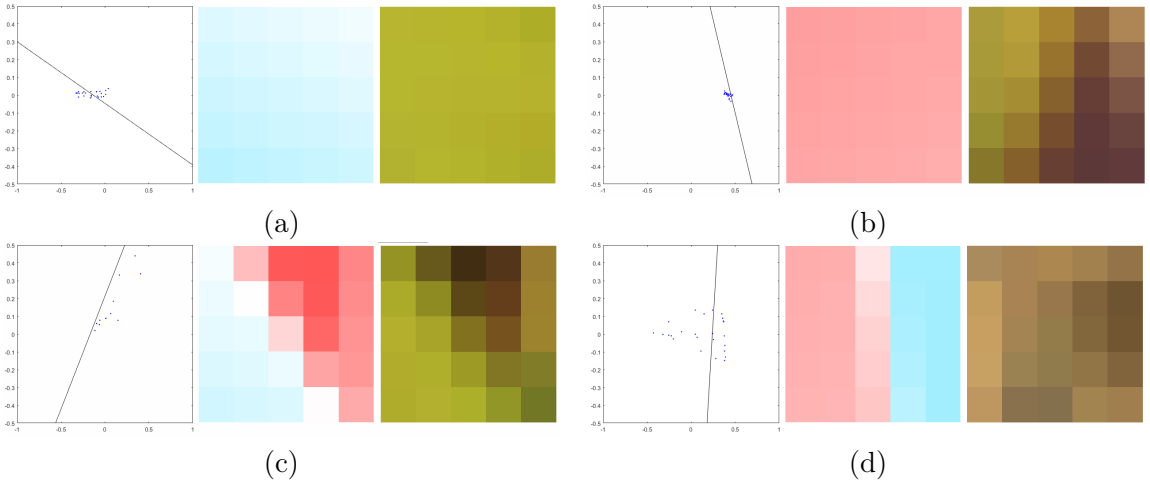


FIGURE 2.8: Distribution of estimated flows (blue dots) in a neighbor 5×5 of each investigated pixel (left). The line is the brightness constraint. The horizontal and vertical axes represent respectively the two components of optical flow (u and v). The middle image shows the ground truth in that neighborhood and the right one shows the zoomed patch.

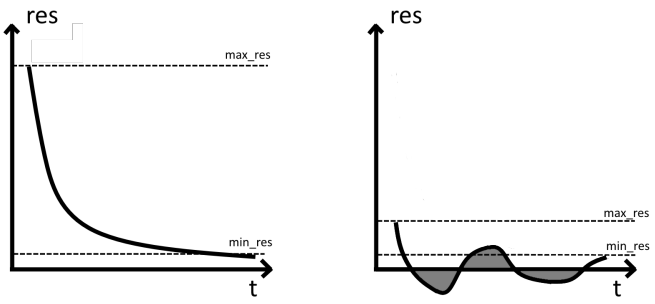
s_{var} to fit in the range [0, 1] in the following way :

$$w_{var}(\mathbf{x}) = \frac{s_{var}(\mathbf{x})}{max(s_{var})} \tag{2.45}$$

2.3.2 Temporal evolution of the residues

Due to the non-linear property of the signals under consideration, the true optical flow is estimated, in iterative methods, by accumulating the residual values of movement after each image linearizing with the optical flow estimated at the previous iteration. In the ideal case, when the optical flow is correctly estimated, the residual value gradually

decreases towards zero, as illustrated by figure 2.9(a). Thus, the shape of this curve is a good indicator of the convergence, and can consequently be an indicator of the quality of the result. Based on this observation, a reliability measure is proposed to compute locally the estimation stability. Two quantities, $w_{\Delta\epsilon}$ and w_{S_ϵ}, are evaluated.



(a) (b)

FIGURE 2.9: The expected residual value in KLT method as time flows (a) : the motion is approaching its true value in the ideal case. (b) : fluctuation of residues in real scenes when the optical flow reaches the near true motion.

First, the change of residual value from one iteration to another is used to show the way the estimated optical flow converges to the final value. From one iteration to the other, the residual value is expected to be decreasing, leading to a positive value :

$$w_{\Delta\epsilon} = max\left(0, \frac{\epsilon_k - \epsilon_{k+s}}{\epsilon_k}\right), \tag{2.46}$$

where ϵ_k and ϵ_{k+s} are the residual values at iterations k and $k+s$ with $s > 0$ the temporal step used to check for this quantity. This quantity is automatically normalized between [0,1]. It is null when residuals increase. Otherwise, it indicates the percentage of residuals decrease. The higher the decrease, the higher the weight $w_{\Delta\epsilon}$.

When the estimated optical flow reaches its stable state, the first quantity (equation 2.46) can not represent the stability since the residual value from one iteration to another varies in a small amplitude, as visible on figure 2.9b. Therefore, the second quantity considered is the sum of absolute residuals between iterations :

$$w_{S_\epsilon(x,y)} = \frac{\max_{(x,y)}(S_{k,k+s}(x,y)) - S_{k,k+s}(x,y)}{\max_{(x,y)} S_{k,k+s}(x,y)}, \tag{2.47}$$

where $S_{k,k+s}$ is a matrix, of the same dimensions as the input image, that collects for each point the accumulated sums of the absolute residuals from iteration k to iteration $k+s$. This measure corresponds to the total gray area in figure 2.9b. It is assumed that the fewer oscillations, the smaller this area and the higher the score are w_{S_ϵ}.

These two quantities computed from residuals are combined together to form the stability residual score w_{res} :

$$w_{res} = 0.5w_{\Delta\epsilon} + 0.5w_{S_\epsilon} \tag{2.48}$$

2.3.3 The mixture reliability score

The final reliability score is built by combining the three different measures defined above about the reliability of estimated optical flows : cornerness to ensure the gradient is high enough for estimation, local variance to evaluate the consistency of the estimations, and residual evolution for the stability of estimation.

This is made by using a minimum operator :

$$w_{mix} = min(w_{corner}, w_{var}, w_{res}) \tag{2.49}$$

Indeed, it is assumed that the optical flow is more reliable when the three quality requirements are fulfilled together. By using this reliability score, the optical values can be sorted by using the worst of their respective scores. The three criteria are evaluated in the experiments section (Section 2.6), both separately and in conjunction. They are compared with the reliability criterion based on the structure tensor described in equation 2.43. Beforehand, the next section explains the strategy used to take benefit of the reliability scores to improve the final optical flow estimation.

2.4 The propagation process

Starting from the first optical flow estimation and its reliability given by equation 2.49, the third stage of our algorithm aims to correct this optical flow map. It is made by maintaining the reliable estimations while correcting the less reliable ones. This correction is made by a propagation process which implies two information sources : the local color similarity where it is assumed that close pixels belonging to the same surface tend to have similar motions.

Figure 2.10 illustrates the key stages of the correction. In one of our previous work [MGB17], the optical flow is propagated in a global way where each pixel is examined to ensure that the good estimations correct the worst ones. The main drawback of this method is its great resources and time requirement.

To overcome this issue, a set of reliable *seeds* are first selected. They correspond to pixels for which the estimated optical flow is reliable, in the sense of the criterion given by equation 2.49. In order to ensure a good spatial distribution of the seeds through the image, a regular grid is built, with square cells of width W_c. In each cell, the pixel of maximum reliability score is detected and considered as a seed.

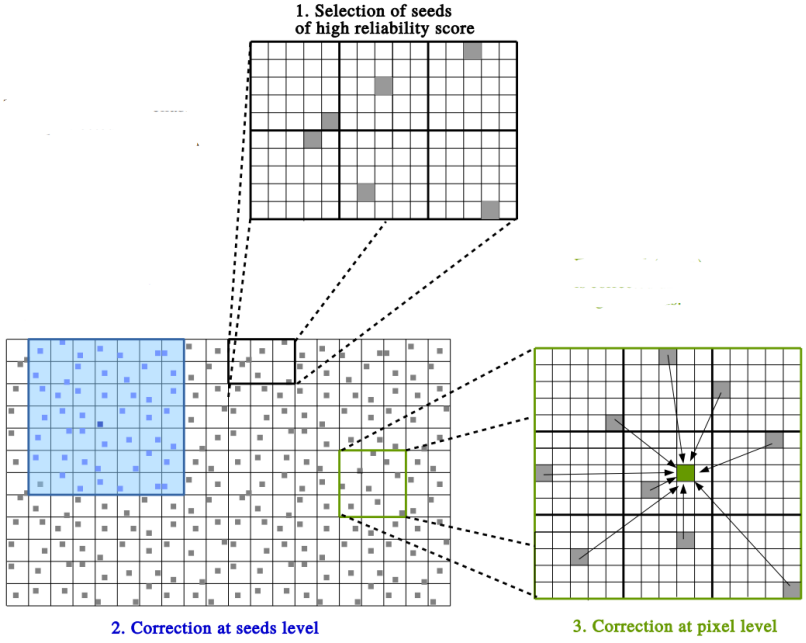


FIGURE 2.10: The three stages of the optical flow correction process. A regular grid of cells of width $\mathtt{W_c}$ is built (here $\mathtt{W_c} = 5$). (1) A seed is selected in each cell as the local maximum reliability score. (2) the optical flow is corrected in a sparse way at seeds level. Each seed (dark blue) is corrected by n close seeds (light blue) $n = 48$. (3) correction at pixel level : each pixel (green) is corrected under the influence of m (here $m = 9$) seeds (represented in gray).

A first correction of the optical flow values is achieved at the seed level (see Section 2.4.1) in order to consolidate their values. To finish, each motion value is corrected under the influence of its neighborhood seeds (Section 2.4.2). This new propagation technique requires very few iterations and hence reduces computation time significantly, as proven further in Section 2.6.

2.4.1 Sparse optical flow correction at seeds level

A first layer of correction is carried out at the seeds level. For each seed previously selected, a refinement is made by using n neighboring seeds around it in a window W_s, as illustrated by figure 2.10.

The propagation consists of correcting the optical flow values of the current seed by using the reliability score and the similarity level of its neighbor seeds. The new optical flow value of the current seed s, noted $\widehat{\mathbf{u}_e}(s)$, is estimated by using the seeds v located in its neighborhood W_s :

$$\widehat{\mathbf{u}_e}(s) = \frac{\sum_{v \in W_s, v \neq s} e_{simi}(v, s)\mathbf{u}_e(v)}{\sum_{v \in W_s, v \neq s} e_{simi}(v, s)} \tag{2.50}$$

Simultaneously, the reliability score of the seed is updated as follows :

$$\widehat{w}(s) = \frac{\sum_{v \in W_s, v \neq s} e_{simi}(v,s) w(v)}{\sum_{v \in W_s, v \neq s} e_{simi}(v,s)} \tag{2.51}$$

The similarity score plays an important role here. The higher this score is, the higher its impact on the correction is. Here, the similarity is based on color and spatial distances in the following way :

$$e_{simi}(v,s) = e^{-\frac{d_{color}(v,s)}{\sigma_c} - \frac{d_s(v,s)}{\sigma_s}}, \tag{2.52}$$

where d_{color} is the Euclidean RGB distance between point s and its neighbor v, d_s is their spatial Euclidean distance,σ_c and σ_s determine the degradation speed of color and spatial distance. This similarity measure provides an influence level of each point on the others based on their distance and color.

Then, the new optical flow is kept if its reliability is higher than the previous score at that point :

$$\text{If } \widehat{w}(s) \geq w(s) \text{ then } \begin{cases} \mathbf{u}_e(s) = \widehat{\mathbf{u}_e}(s) \\ w(s) = \widehat{w}(s) \end{cases} \tag{2.53}$$

2.4.2 Dense optical flow correction at pixel level

The two previous steps have provided reliable anchors which are then used to influence the correction of the dense optical flow in their local neighborhood. The correction procedure is similar to the previous seeds correction, except that : 1) it is made at each pixel except for the seeds ; 2) the process is made more locally i.e. less seeds are taken into account for correction.

Here, for each pixel p in the image, the m closest seeds are selected with $m < n$. Equations (2.50) to (2.53) hold except they are computed on p instead of s and the m neighbor seeds v are used instead of n.

The dual-layer of propagation adds advantageous effects to the final results. If the first propagation focuses on correction at coarse view, the second propagation focuses on refining the optical flow in a finer way.

The experiment section shows improvements in terms of precision and computation time in comparison to [MGB17].

2.5 Implementation

The numerical implementation of our method is given in table 2.1. The pyramidal images and the propagation of flows through the different levels are implemented in the similar way as in [WPZ+09]. We focus here on the main part of the proposed method.

TABLE 2.1: Implementation of the proposed algorithm

Input : Two successive color images I_1 and I_2
Output : Optical flow map $\mathbf{u}$ from I_1 to I_2
- Create L-levels pyramidal images I_1^l, I_2^l for l=1 ... L
- Initialize $k = 0$, $\mathbf{u}^l = \mathbf{u}_0 = 0$

For $l = L$ to 1
- Compute $\nabla I_1^l(\mathbf{x})$
- Compute w_{corner} from I_1^l using (2.42)
- Initialize $S_{k,k+s}$=0

For k = 1 to Max_Warps
- Interpolate $I_2^l(\mathbf{x} + \mathbf{u}^l)$
- Compute $\nabla I_2^l(\mathbf{x} + \mathbf{u}^l)$, I_t^l
- Estimate the residual motion $\mathbf{u}_{res}$
- $\mathbf{u}^l = \mathbf{u}^l + \mathbf{u}_{res}$
- Compute the sum of absolute residuals $S_{k,k+s} = S_{k,k+s} + |\mathbf{u}_{res}|$

If (k % s) = 1 :
- $\epsilon_k = \mathbf{u}_{res}$

If (k % s) = 0 :
- $\epsilon_{k+s} = \mathbf{u}_{res}$
- Compute w_{var} from (2.45) and w_{res} from (2.48)
- $w = \min(w_{corner}, w_{var}, w_{res})$
- Create seeds as mentioned in Section 2.4
- Correction at seeds level according to Section 2.4.1
- Correction at pixel level according to Section 2.4.2
- Apply the median filter to $\mathbf{u}^l$

Next
- Interpolate $\mathbf{u}^{l-1}$ from $\mathbf{u}^l$ if $l > 1$

Next

Regarding the residues criterion, a short time step s is considered for analysis. The reliability score is evaluated and the propagation is done at the same update speed of residual criteria. The Max_Warps should be divisible by s so that at the last warping iteration, the output of estimated optical flow u passes through the propagation.

2.6 Experiments

The performance of our method is evaluated on the Middlebury database [2] which is composed of eight sequences with ground-truth. There are three different image sizes : 1) Venus (420 × 380) ; 2) Dimetrodon, Hydrangea, RubberWhale (584 × 388) ; 3) Grove2, Grove3, Urban2, Urban3 (640 × 480).

The experiments are conducted by using KLT with pyramidal images (with 5 levels) and a warping scheme (12 iterations). For KLT, a window size of 5× 5 is considered.

The propagation follows the explanation in Section 2.4 where the estimated optical flow is passed through dual-layer of propagation after four consecutive image linearization iterations. The parameters for measuring the similarity of equation (2.52) are $\sigma_c = 25$ and $\sigma_d = 2$ and the searching number of neighbor points is $n = 50$ for the seeds correction and $m = 10$ for the dense refinement.

Our evaluation involves the error measures AAE (average angle error) and AEPE (average end-point error) recalled in [BSL+10].

This section is divided into two parts. The first one focuses on the evaluation and the comparison of the five reliability measures mentioned in Section 2.3. The second part evaluates the results of the refinement process and compares our approach to existing methods.

2.6.1 Study on the reliability criteria

Our first study deals with the evaluation of different reliability criteria evoked or proposed in this thesis. As mentioned in Section 2.3, besides the reliability criteria introduced in this work, the criterion defined from the structure tensor is also a prominent one. Therefore five criteria are evaluated : *i.e.* structure tensor (ST) of equation 2.43, cornerness of equation 2.42, residual of equation 2.48, variance of equation 2.44 and mixture of equation 2.49. In each case, the criterion is used solely to measure the error of the estimation The Average End Point Error (AEPE or EPE) is considered for the evaluation. In addition of the five reliability criteria, we also show the optimal measure of reliability proposed by "sparsification" in [BW06] and defined as :

$$c_{opt}(\mathbf{x}) = 1 - \frac{e(\mathbf{x})}{max(e(\mathbf{y})|\mathbf{y} \in N(\mathbf{x}))}, \tag{2.54}$$

where $e(\mathbf{x})$ is the AEPE at point $\mathbf{x}$ and $N(\mathbf{x})$ its neighbourhood. In our experiments, $N(\mathbf{x}) = 5$ which is the same size as the neighbourhood window used to estimate the optical flow by classical KLT method.

2. http ://vision.middlebury.edu/flow/data/

Finally, six criteria are compared. figure 2.11 shows the results obtained for 4 image pairs of the MiddleBury database at two stages of the process, (a) at first iteration i.e. at 5^{th} level of the pyramid (lower scale) and first warping and (b) after last iteration ($1^{r}st$ level, 12 iterations). The vertical axis represents the accumulated average end point error (EPE) while the horizontal axes represent the percentage of pixels accounted in the error measure (10 means 10% of the most reliable pixels of the image, 100 means all pixels). Of course, a criterion is considered as good when the average error is low. Moreover, its evolution should correspond to an increasing monotonous function of the % of pixels, i.e it should be lower when the more reliable points are considered and it should increase as soon as new points (less reliable) are progressively taken into account.

According to figure 2.11, it appears that only the curves based on variance (cyan) and mixture (red) are monotonously increasing. It can also be noticed that the criteria based on the structure tensor (blue) and cornerness (pink) generally provide high AEPE values. On the contrary, variance and mixture produce the lower errors in most cases. In figure 2.11(a) where the initial flow is null (there is no distortion in source images due to linearization), two criteria are competitive : the variance and the mixture. On overall, they basically represent well the precision of optical flow according to the reliability score. However, at the last iteration, as can be seen on figure 2.11(b), the mixture criterion appears to be more stable than the variance criterion.

The next part evaluates the impact of the whole refinement.

2.6.2 Refined optical flow

This section discusses about the results of the propagation method on the final average error of optical flow. As discussed above, five reliability criteria are introduced together with two seed selection strategy : sorting and grid. We search first the best combination between seed selection strategy and reliability measure then compare the results of our approach to the existing methods by using this combination.

2.6.2.1 Comparison of seeds selection strategies and reliability criteria

Besides the seeds selection strategy explained in Section 2.4, we explore an alternate strategy with a native *sorting strategy* where the reliability scores are sorted in a global way. Then the seeds are chosen from the highest to the lowest score with the neighbor constraint (no seed would be in the neighbor region N_{seed} of another). This choice leads to the random distribution of seeds through the image, where the distance between seeds is higher than $d_{seeds} = (N_{seed} - 1)/2$.

We first measure the average error and evaluate the ability of each criterion to be consistent with the precision computed from the ground-truth. The *sorting strategy* is

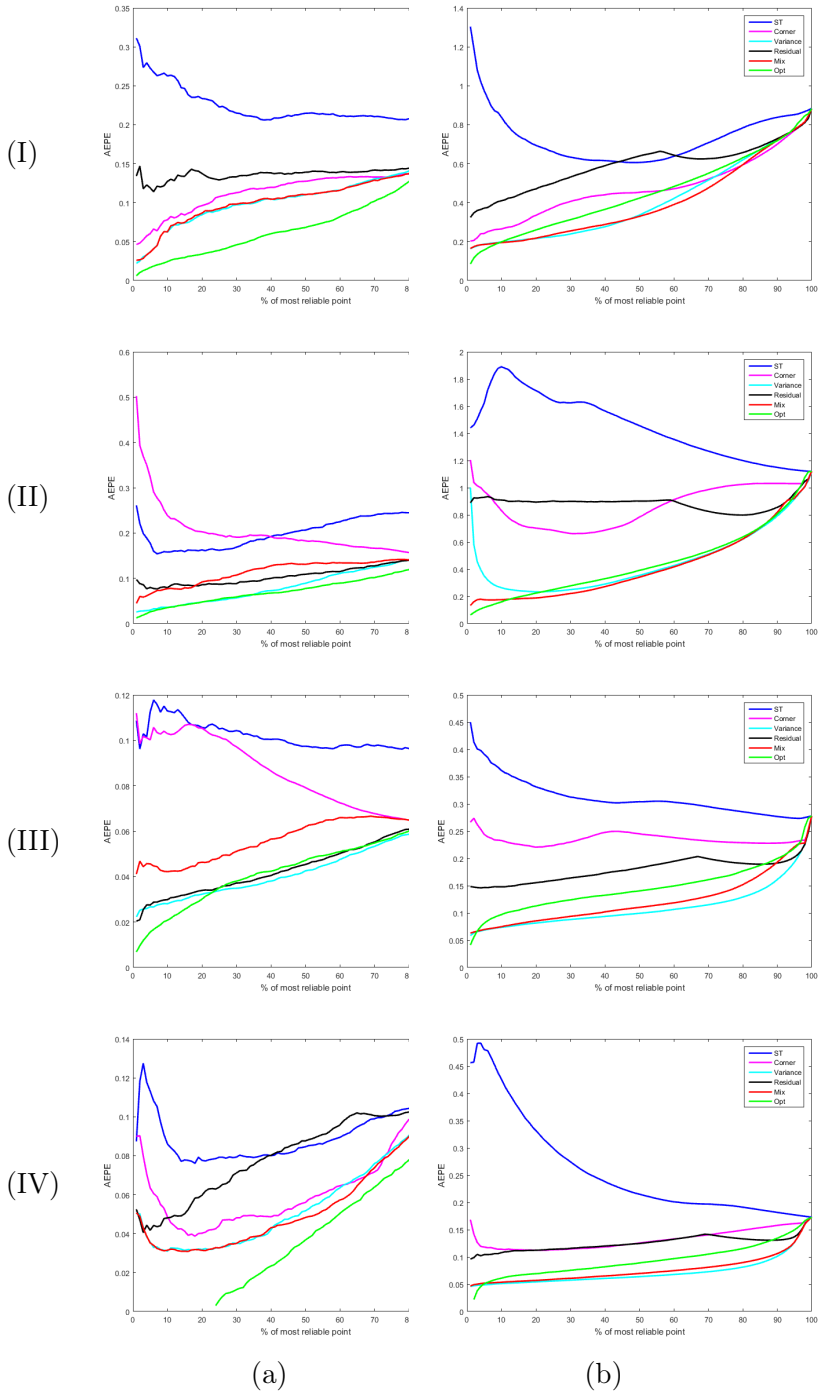


FIGURE 2.11: Reliability measure. Comparison of 4 sequences Venus (I),Urban3 (II), Grove2 (III) and Rubberwhale (IV) from the MiddleBury database at 5^{th} pyramidal level,1^{st} warping level (a) and at 1^{st} pyramidal level,12^{th} warping level (b)

considered first. The results of AAE and AEPE on each criterion are collected in table 2.5. Both residues and mixture criteria give the best results for 7 sequences and the cornerness is the best for 1 sequence. However, the average error on 8 sequences is globally lower when the mixture criterion is used.

Considering now the seeds selection using the grid strategy, the precision results collected in table 2.6 are globally better. In this case, the mixture criterion provides the best performances on real-life sequences : Venus, Dimetrodon, Hydrangea and RubberWhale. On synthesized sequences, the criteria "variance" and "variance+residual" give a better overall results than the mixture criterion. But the gap is not much so that the mixture criterion is still the best when counting the average error on 8 sequences.

Figure 2.12 illustrates the behaviour of the reliability measure of each criteria by applying our refinement method by propagation. Here also, the mixture and variance criteria show a coherent behaviour of the good reliability scores but the mixture criteria has the better final result. For example, table 2.2 shows the AEPE obtained on RubberWhale by using these criteria. Although for the first part of measure, the "variance" criterion seems to produce better results, the mixture criterion leads finally to a lower error when 100% of the pixels are taken into account.

	AEPE		
Criteria	10%	50%	100%
Var(before)	0.054	0.073	0.177
Var(after)	0.043	0.05	0.111
Mix(before)	0.057	0.077	0.176
Mix(after)	0.046	0.05	0.108

TABLE 2.2: Precision representation of two criteria before and after propagation at 10%, 50% and 100% on the sequence Rubberwhale according to the "sparsification" measure

According to the previous experiments, the "mixture" criterion is a useful indicator to evaluate the reliability of the optical flow estimates. Used in our propagation process, it finally produces the better precision.

Since the criterion is the combination of 3 independent criteria, it is interesting to evaluate the impact of each parameter in this mixture. Table 2.3 shows the percentage of contribution of each criterion to the "mixture" one. These results show that "variance" and "cornerness" measures have a predominant influence. However, the residual criterion, although having less influence, does improve the precision of the estimated optical flow, as shown by the results of table 2.6. Note that residuals are only available for KLT-based methods. This criterion can be removed, without significant loss of precision, when an other local optical flow technique is used instead of KLT. Therefore, the proposed algorithm can be used independently from any local optical flow technique, in a module that can be plugged in or out of the application whenever a refinement is required.

	Venus	Dimetrodon	Hydrangea	Rubber Whale	Grove2	Grove3	Urban2	Urban3
% var	99.6	28.9	37.8	36.8	51.3	98.1	98.5	95.5
% cornerness	0.2	71.1	61.9	63	48.6	0.9	0.5	2
% res	0.2	0	0.3	0.2	0.1	1	1	2.5

TABLE 2.3: The contribution of each criteria on "mixture" on 8 sequences of Middlebury dataset at 1st pyramidal level and 12th warp iteration.

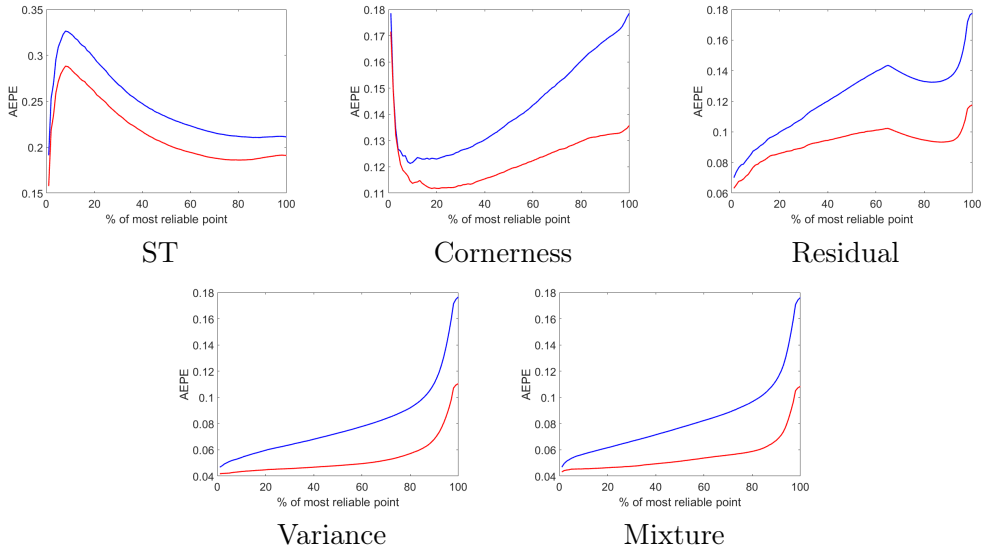
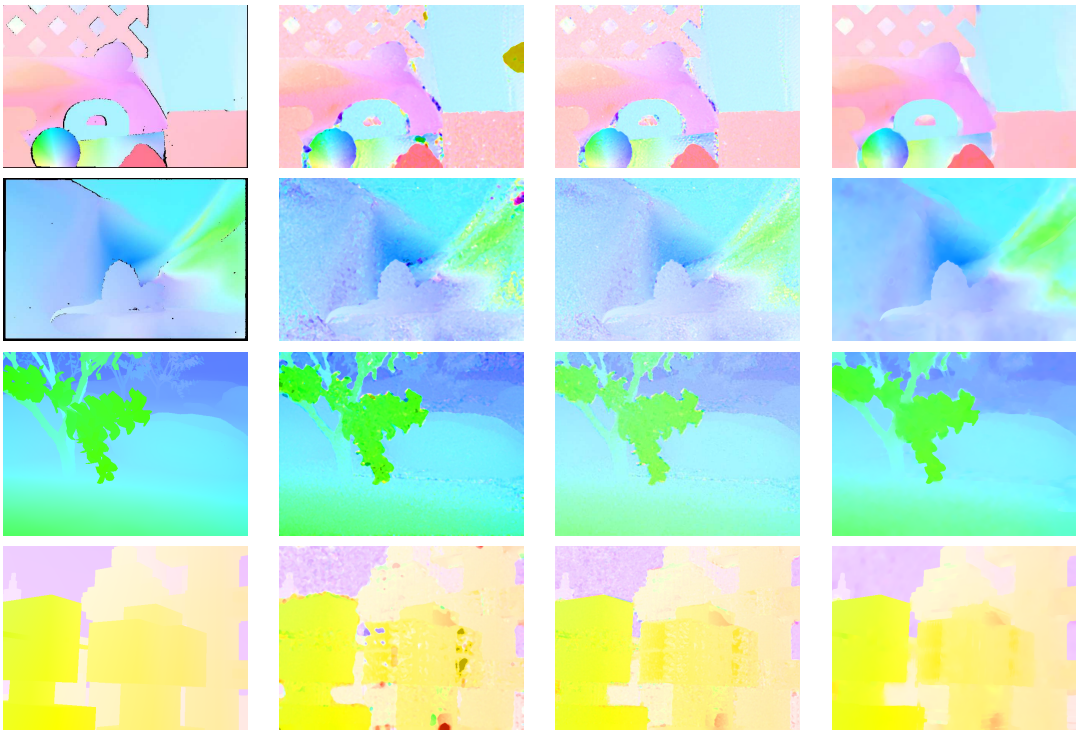


FIGURE 2.12: Comparison of the reliability measures for five criteria using the seeds selection based on grid strategy at 1st pyramidal level and 12th warp iteration (ST, corner, residues, variance and mixture) on Rubberwhale sequence. The blue curve and red curve represent the reliability measure before and after the propagation respectively.

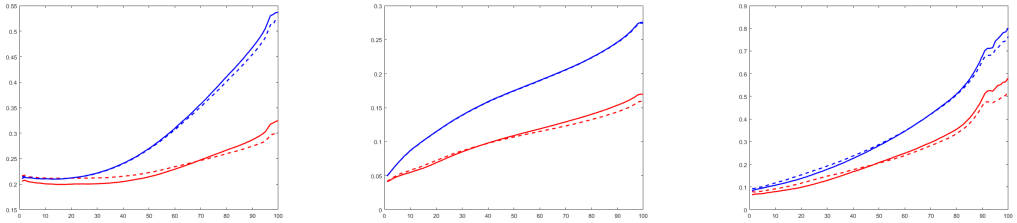
All these experiments have shown that the propagation process reduces the average error in all sequences under study. The local KLT estimation provides noisy flows, as visible on figure 2.13(b). After the refinement process, as described by figure 2.13 (d), the optical flow becomes smoother. According to the quantitative error measures, the mixture criterion provides competitive and consistent results. By comparing the results from table 2.5 and table 2.6, related to the two different seeds selection modes (grid selection and sorting), it can be seen that the grid strategy provides better results (AAE : 3.352, AEPE : 0.293) compared to sorting strategy (AAE : 3.593, AEPE : 0.308). figure 2.14 proves more this one by comparing the relationship between precision and reliability for the two selection strategies under mixture criterion, before and after refinement at the last iteration of algorithm. Each strategy is used independently for the whole estimation process from the first iteration at highest pyramidal level (5^{th} level, which corresponds to an image size divided by 16) to the last one at original image size. We observe the effect of propagation at the last iteration so that there is the difference of the before-propagation graph. The grid strategy always gives slightly better results with lower calculation time by passing the sorting step.

After all these experiments, for the best optical flow estimation by using our approach, we use the grid-based seeds selection strategy and the mixture reliability criterion to compare with existing methods in the next part.



(a) (b) (c) (d)

FIGURE 2.13: Flow color map according to Middlebury code : a) Ground truth, b) KLT, c) ours before propagation at the last iteration, d) ours after propagation on 4 sequences : RubberWhale, Dimetrodon, Grove2 and Urban3 from top to bottom respectively.



(a) (b) (c)

FIGURE 2.14: Comparison of reliability measure for two strategies : sorting (continuous line) and grid (dotted line) for 3 sequences at pyramidal level 1 and 12th warp : Venus(a), Dimetrodon(b), Urban3(c). The blue curve and red curve represent the reliability measure before and after the propagation respectively

2.6.2.2 Comparison with existing methods

This section intends to compare the performance of our method with existing methods on the training dataset : KLT [yB00], Horn and Schunck HnS [HS81], Black and Anandan BA [BA96] and our previous version of the Reliability Flow (RF) proposed in [MGB17].

Table 2.7 collects the AAE and AEPE for the 8 sequences. According to these results, our method surpasses all of the classical methods except for the sequence Urban2 for BA

method. Our improved version also gives better results than the previous RF with significant calculation improvement. Table 2.4 shows the computation times (Matlab, CPU i5-7300HQ) for the 3 groups of images : group 1 (420×380) : Venus, group 2 (584×388) : Dimetrodon, Hydrangea, RubberWhale and group 3 (640×480) : the rest. Each sequence in the same group takes the same amount of time to estimate the optical flow. Our method shows the good trade-off between the computation time and the precision of the estimation. It also shows a huge improvement compared to our previous proposal RF. In addition, if we compare to the BA method, ours has improvements on both precision and calculation time. Figure 2.15 shows the encoded flows on 4 sequences of Middlebury database. The proposed method produces more clear cut in the edges between different movements, especially on sequence RubberWhale (figure 2.15-first row) and Urban3 (figure 2.15-last row) even though the main estimator is local KLT. These results, ranked at 86/151 on Middlebury website, prove that our assumption about relationship between movement and color is applicable.

Group	1	2	3
RF :grid+mix	23.3	34.5	46.9
old RF	593	868	1184
KLT	13.3	18.1	24.66
HnS	17.7	23.4	35
BA	64.2	101	144.6

TABLE 2.4: Computation times (in seconds) of the methods under consideration.

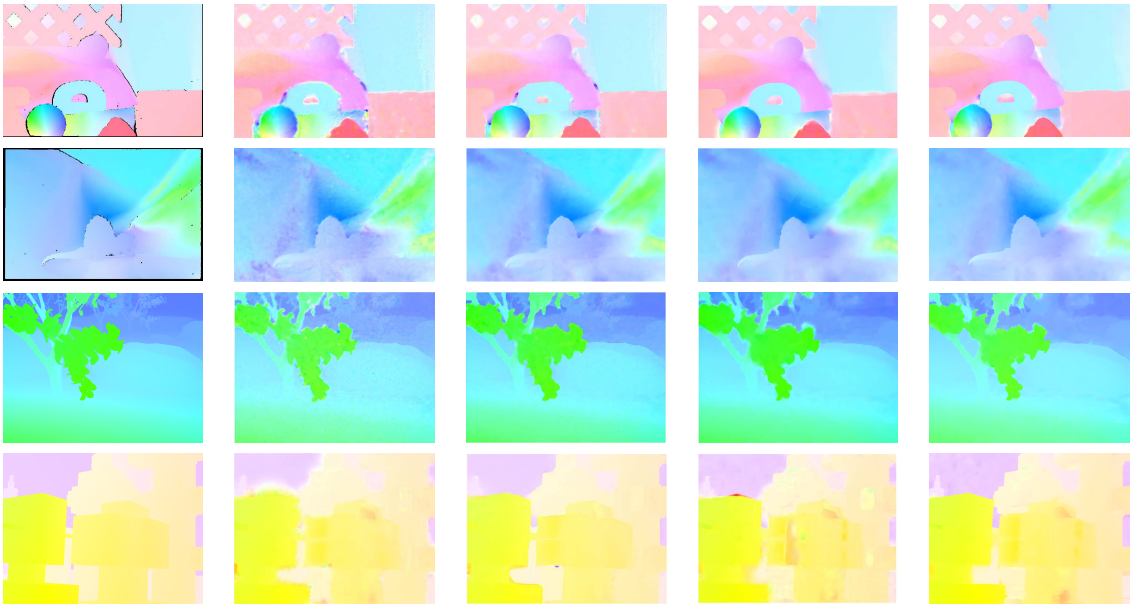


FIGURE 2.15: Comparison with existing methods (flow color map according to Middlebury code) : a) Ground truth b) HnS, c) BA, d) old RF and e) ours on 4 sequences : RubberWhale, Dimetrodon, Grove2 and Urban3 from top to bottom respectively.

We also extract partial result table of the submitted sequences on Middlebury dataset on table 2.8. The methods are sorted in ascending rank order. Note that the rank system

of Middlebury relies not only on the average error of all pixels but also on the error in the discontinuous or untextured regions, which are particularly prone to errors. It stays at a decent place for a local based method even though its performance can not surpass some modern more complex global methods [SSB10, BBPW04],

2.7 Conclusion

After analyzing the state-of-art related to motion estimation, we have proposed a new variant of the well-known KLT method that has the ability to evaluate and to take advantage of the quality of optical flow estimation at several stages. The main contributions of this chapter are :

— The definition of two criteria to evaluate the quality of the optical flow, considering first the case of the KLT method. One of the criteria, the local variance of optical flow, can be applied to any other local optical flow method ;
— A refinement of the optical flow by using a propagation process based on the reliability score. The precision of the optical flow is improved by assuming that objects show some local homogeneities of color or intensity, and that the local optical flow varies smoothly or is even homogeneous, allowing a *constancy propagation*.

In order to improve the estimations, we propose some perspectives to our method :

— The propagation could include additional constraints, such as scene structure information when available. The hypothesis of constancy propagation could be replaced by a different strategy that could be defined thanks to a given model of optical flow provided by structure information.
— The matching and the similarity criteria can integrate other features instead of or together with color. This could be object segmentation results or texture features.
— The propagation process could be accelerated, for example by paralleling the processing of each seed.

Using our proposed adaptive optical flow estimation, the next Chapter 3 presents a new 3D plane segmentation method called *uv*-velocity that will take advantage of the quality measure of the optical flow estimation.

Database	Venus		Dimetrodon		Hydrangea		RubberWhale		Groove2		Groove3		Urban2		Urban3		Average	
Method	AAE	AEPE	AAE	AEPE	AAE	AEPE	AAE	AEPE	AAE	AEPE	AAE	AEPE	AAE	AEPE	AAE	AEPE	AAE	AEPE
RF :sorted+ST	6.501	0.451	4.004	0.197	2.701	0.228	4.54	0.145	2.578	0.18	5.959	0.617	5.444	0.592	4.593	0.778	4.540	0.399
RF :sorted+corner	5.13	0.348	**3.392**	**0.169**	2.356	0.205	3.9	0.118	2.51	0.174	6.081	0.645	3.643	0.383	4.23	0.607	3.905	0.331
RF :sorted+res	5.2	0.354	3.699	0.182	**2.313**	**0.200**	3.643	0.111	2.157	0.153	**5.388**	**0.536**	**3.419**	**0.355**	**3.422**	**0.575**	3.655	0.308
RF :sorted+var	4.808	0.339	3.502	0.174	2.378	0.208	3.807	0.117	2.189	0.153	5.518	0.559	3.514	0.392	3.781	0.621	3.687	0.320
RF :sorted+mix	**4.515**	**0.324**	3.398	0.170	2.377	0.205	**3.619**	**0.110**	**2.085**	**0.146**	5.437	0.550	3.642	0.382	3.673	0.581	**3.593**	**0.308**

TABLE 2.5: Performance of the compared criteria under sorting strategy in the Middlebury database.

Database	Venus		Dimetrodon		Hydrangea		RubberWhale		Groove2		Groove3		Urban2		Urban3		Average	
Method	AAE	AEPE	AAE	AEPE	AAE	AEPE	AAE	AEPE	AAE	AEPE	AAE	AEPE	AAE	AEPE	AAE	AEPE	AAE	AEPE
RF :grid+ST	6.34	0.463	4.346	0.214	3.200	0.267	5.184	0.171	3.225	0.223	7.445	0.763	8.158	0.887	5.361	0.883	5.407	0.483
RF :grid+corner	6.245	0.372	3.111	0.158	2.766	0.234	4.341	0.135	3.374	0.228	7.358	0.784	3.924	0.437	4.342	0.584	4.432	0.3665
RF :grid+res	4.43	0.305	3.608	0.179	2.431	0.207	3.749	0.117	2.306	0.16	5.99	0.601	3.394	0.354	3.468	0.548	3.672	0.309
RF :grid+var	4.445	0.325	3.379	0.170	2.438	0.208	3.715	0.117	**2.125**	**0.149**	5.434	0.543	3.352	0.387	**2.823**	**0.485**	3.464	0.298
RF :grid+var+res	4.455	0.315	3.268	0.166	2.438	0.207	3.556	0.111	2.149	0.149	**5.380**	**0.541**	**3.144**	**0.368**	2.834	0.494	3.403	0.294
RF :grid+var+corner	4.156	0.301	3.103	0.156	2.392	0.203	3.518	0.109	2.251	0.155	5.461	0.549	3.157	0.384	2.874	0.493	3.364	0.293
RF :grid+mix	**4.100**	**0.298**	**3.103**	**0.156**	**2.392**	**0.203**	**3.491**	**0.108**	2.245	0.154	5.461	0.549	3.157	0.384	2.874	0.493	**3.352**	**0.293**

TABLE 2.6: Performance of the compared criteria under grid strategy in the Middlebury database.

Database	Venus		Dimetrodon		Hydrangea		RubberWhale		Groove2		Groove3		Urban2		Urban3		Average	
Method	AAE	AEPE	AAE	AEPE	AAE	AEPE	AAE	AEPE	AAE	AEPE	AAE	AEPE	AAE	AEPE	AAE	AEPE	AAE	AEPE
RF :grid+mix	4.100	0.298	**3.103**	**0.156**	**2.392**	**0.203**	**3.491**	**0.108**	**2.245**	**0.154**	**5.461**	**0.549**	3.157	0.384	**2.874**	0.493	**3.352**	**0.293**
old RF	**4.054**	**0.261**	3.26	0.165	2.422	0.221	3.558	0.114	2.514	0.170	5.940	0.624	3.919	0.518	4.429	0.739	3.762	0.351
KLT	9.643	0.584	4.936	0.241	3.148	0.289	5.824	0.179	3.433	0.252	7.551	0.81	6.121	0.705	7.284	1.056	5.992	0.5145
HnS	5.6	0.34	4.767	0.232	3.008	0.257	5.175	0.16	2.719	0.196	6.315	0.649	4.924	0.562	6.943	0.756	4.931	0.394
BA	4.095	0.255	4.199	0.205	2.665	0.231	4.384	0.132	2.274	0.159	5.711	0.584	**2.804**	**0.355**	3.528	**0.456**	3.708	0.297

TABLE 2.7: Performance of the compared methods in the Middlebury database.

Database	Army		Mequon		Schefflera		Wooden		Groove		Urban		Yosemite		Teddy		Average	
Method	AAE	AEPE	AAE	AEPE	AAE	AEPE	AAE	AEPE	AAE	AEPE	AAE	AEPE	AAE	AEPE	AAE	AEPE	AAE	AEPE
Layers++[SSB10]	**0.08**	**3.11**	**0.19**	**2.43**	**0.20**	**2.43**	**0.13**	**2.13**	**0.48**	**2.35**	**0.47**	3.81	**0.15**	2.74	**0.46**	**1.45**	**0.27**	**2.56**
Brox et al[BBPW04]	0.11	4.44	0.27	3.72	0.39	4.97	0.24	4.58	1.10	3.79	0.89	3.91	0.10	**2.22**	0.91	4.62	0.50	4.03
RF :grid+mix	0.13	4.77	0.60	8.72	0.65	8.49	0.25	4.55	0.76	2.92	0.57	**3.76**	0.16	2.98	0.86	4.59	0.50	5.10
LDOF[BM11]	0.12	4.60	0.32	4.67	0.43	5.63	0.45	5.80	1.01	3.52	1.10	4.84	0.12	2.46	0.94	4.85	0.56	4.55
FOLKI[BC05]	0.29	10.50	1.52	20.90	1.23	17.60	0.99	15.40	1.53	6.16	2.14	12.20	0.26	4.67	2.67	18.20	1.33	13.20

TABLE 2.8: Performance of our method with the others on submitted sequences.

Chapitre 3

From motion to structure : uv-velocity, an optical flow registration for planar segmentation

3.1 Introduction

The detection and characterization of planar surfaces in the 3D scene can provide useful homography information for example for perspective correction and panorama stitching [HZ03, Sze10]. In computer vision for robotics, it can be used to perform visual odometry [SVB+17, CPP12], which is the process of estimating camera pose *via* images, with less parameters than required for computing the essential matrix. Moreover, since our urban environments comprise many planar surfaces, profiting this information can boost the performance of state-of-art methods in visual odometry in terms of computation time. However, planar surfaces are not always available or easily detected from an image sequence captured using only one camera. In addition, detecting planar surfaces is generally an heavy process, which is somehow contradictory to the advantages above-mentioned.

Our study focuses on motion-based planar segmentation for monocular vision. Our idea is to detect the main planes of the scene, and to output a 2D segmentation map that corresponds to the 2D projection of the detected 3D planes. For that purpose, we introduce the concept of uv-velocity which is an accumulative voting space designed to detect planar surfaces efficiently and with a certain robustness thanks to its cumulative nature. This is the second block of our system described on figure 3.1. The proposed approach is a variation of the c-velocity method [BZ12] which is a generalization of the v-disparity that was proposed for stereo-vision [LAT02]. The method addresses the particular case of a unique camera mounted on a vehicle.

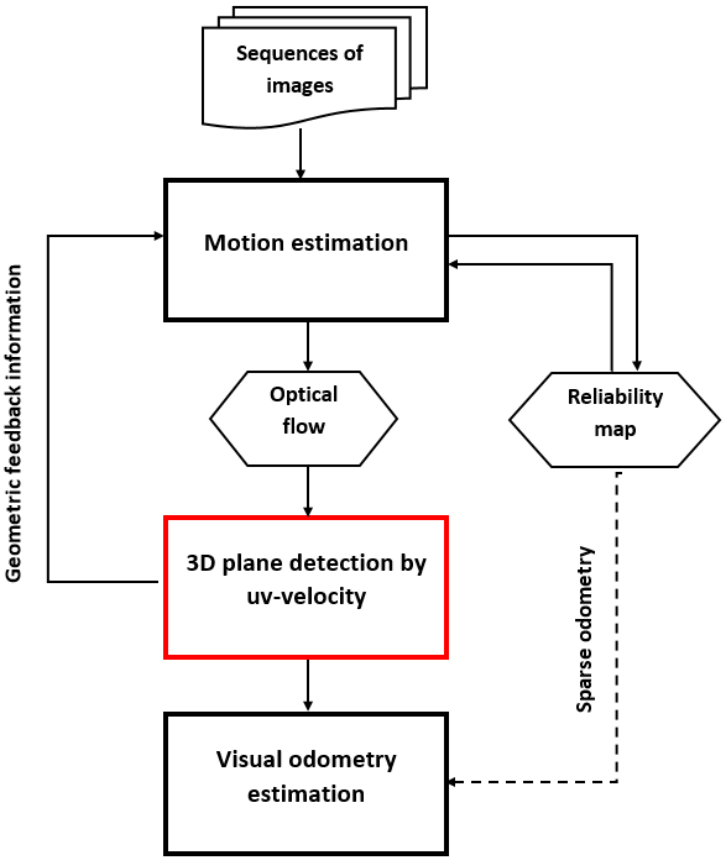


FIGURE 3.1: The developed block in this chapter, the *uv*-velocity and its application for plane segmentation

In this chapter, Section 3.2 evokes the main categories of methods used for planar detection and discusses their advantages and disadvantages. Then, the main assumptions about motion and plane models and their consequences on the design of the accumulative voting space is presented in Section 3.3. Section 3.4 recalls the principles of the initial method *c*-velocity and its main properties. Inspired by the *c*-velocity concept, Section 3.5 and 3.6 introduces the new voting strategy *uv*-velocity and addresses its advantages compared to *c*-velocity. Section 3.7 analyze in detail the voting spaces and discuss the strategy for parametric curve detection. Finally, Section 3.8 details the experiments conducted to compare the efficiency of the voting spaces on synthesized sequences as well as on real sequences. Our cumulative method is also compared to a classic parametric method where planar surfaces are found using the consistence of homography transformation [HZ03].

3.2 Segmentation of planes in a monocular dynamic system

Planar surface segmentation mostly revolves around the extraction of point clouds which are not always available, for instance when a simple embedded camera is used instead

of a 3D scanner. In that case, point clouds need to be estimated first through structure-from-motion techniques, where camera pose and 3D coordinates of keypoints on image have to be estimated [ÖVBS17]. The topic of planar region detection is addressed in the review [WCYL12], where methods are classified into tree main categories : stereo-based, monocular-based and multi-sensor fusion based method. In the monocular-based category, the authors distinguish three classes : projection invariance, homography based and optical flow based method.

In the first category, **projection invariance**, the objective is to define visual features that would share similar properties for each planar surface, and have distinctive values for two different planes. This can be extracted from points [SBSR92], segment and lines [GBB98] and allows simplifying the planar segmentation procedure. Then the planar segmentation consists in grouping similar invariants. Since these seminal works, a tremendous number of robust feature points like SIFT, SURF, FAST, BRIEF, ORB, etc. have been designed and more effort is now granted to homography-based and optical flow techniques.

Homography-based methods. These methods estimate a geometric transform between two views of the same scene. The methods of this category directly estimate the homography parameters of the planar surfaces, using points correspondences in a sequence of images. This is generally made in an iterative way, starting by the detection of the most dominant plane.

The points which comply the same transform are assumed to belong to the same planar surface. Three points located on the plane are enough to estimate the parameters of the transform. However, since there can be several planar surfaces of unknown location in the image, the question is how to select those points. This can be made by choosing points randomly, and by first detecting the most dominant plane. For noisy inputs, RANSAC techniques [FB81] (for *RANdom SAmple Consensus*) are often applied to provide a robust estimation. On the other hand, one can detect planar surfaces by the consistence of homography matrix between two images [Beb06] by again using RANSAC to eliminate the outliers between point-correspondences in N-views. As a consequence, the computation is long and the solution is not deterministic. The plane parameters are estimated by choosing three points randomly. The other points are added into the initial set according to their spatial distance to the current surface. The plane is confirmed when the set contains enough points. One can note also the work of [CD10], where the ground plane is detected using a Modified Expectation Maximization algorithm.

Instead of estimating globally the transform from a limited number of points and proceeding in a top down strategy, some approaches aggregate pixels locally according to they compliance to a geometric model. These bottom-up approaches start from a local estimation in the image and propagates the detection (growing techniques) or merge the resulting planar cells [AB08]. For example in [SMR06], the intensity image is segmented

into plane patches related to the projection of planar surfaces. Alternatively, starting from a first set of feature points extracted in a first frame, a Delaunay triangulation can be performed [Air]. On each segment, an affine homography can be estimated using consecutive frames. In [Jin14], a growing strategy is used. The seeds of the planes are first detected by scanning all the points with a fitting model. New points are added when a similarity criterion is met. The plane stops growing when of the points are scanned or when a stopping condition is met. Then, the technique passes to an other seed. The number of planes is determined *a priori* and can be higher than the actual number of planes (over-segmentation). The segmentation is achieved after a verifying and merging iterative process. The choice made on the fitting model and on the criteria is crucial to have the best results.

Optical flow based methods. These techniques exploit the relationship between motion and structure. The velocity is generally estimated in a dense way, at each possible position, contrary to the previous case where sparse correspondences were computed on the purpose of exhibiting planar structures. The motion map is analyzed through histograms or segmented according to a planarity criterion. Concerning this category of methods, we were more interested in the **voting strategies** which transform the inputs coordinates into parameters' voting space on which the points of a same plane reveal same parameters. These approaches can be related to the Hough Transform [Hou62, DH72], initially designed to identify simple parametric shapes, like circles or lines, using an accumulative voting space of shape parameters. This category of methods is less present in the literature compared to homography-based techniques. In [IF99], a randomized voting method is proposed to determine whether and object and its motion are planar or not, by assuming that a planar object is a polygon the vertices of which have been detected. In [SMR06], a progressive voting strategy is used to segment planar patches directly from the intensity image. To do so, the relation between the projections of one point onto a pair of images is linearized. Then, triplets of points progressively vote to define multiple planar seed regions. In [BELN11], a plane is considered as a parametric shape in 3-D coordinates of which coordinates of points $(X, Y, Z)^\top$ on the plane can be written :

$$\rho = Xcos(\phi)sin(\theta) + Ysin(\theta)sin(\phi) + Zcos(\theta), \tag{3.1}$$

where ρ is the distance of the plane from origin, $\phi \in [0, 2\pi]$ is the angle between the projected normal vector on XY-plane and OX, and $\theta \in [0, \pi]$ is the angle between normal vector and its projection on XY-plane. For each point $(X, Y, Z)^\top$, the technique iterates on all possible values of ϕ and θ to find the corresponding ρ and casts on the cumulative space. With three parameters, the voting space has three dimensions and is much more complicated to deal with, in comparison to Hough transforms for line detection which requires 2D voting spaces. Then, a search for prominent points in the voting space is performed to find the different planes. In order to simplify the analysis for indoor planes detection, Adan et al. [AH11] project the points onto the floor plane

in order to reduce the analysis to a 2D Hough transform. For stereovision, the planar segmentation can be simplified by using UV disparity [LAT02], where only two voting spaces are required to detect the main obstacles and the road in an urban environment. This method has been an inspiration for the design of the motion-based method c-velocity [BZ12] that is further studied in this thesis.

Discussion. Homography-based methods, also called parametric approaches, are generally faster than voting strategies because the estimation is direct. On the other hand, they are prone to false estimations of parameters and they are able to detect one plane at a time. However, by using RANSAC to estimate the plane parameters, the calculation time increases due to the try-and-error process of the technique. Moreover, its random nature makes it an un-deterministic technique. Most techniques require adapting parameters to data points in order to produce the best result and generally take a lot of time to compute, even there are researches trying to improve the speed [KEB91, XOK90, SWK07]. However, the only drawback of these techniques is that they rely on point clouds which are rarely available with standard vision systems.

The voting approaches are easy to implement and allow to find parametric shapes even when they are reasonnably noisy. However, their performance depends on the resolution of the voting spaces, which has to be defined to satisfy a trade-off between execution time and precision. Moreover, with the increasing number of parameters, voting approaches suffer from the complexity of multi-dimensional voting spaces, thus the process time is a drawback in these cases.

Hence, we propose an opportunistic technique based on a voting strategy, that can reveal the planar surfaces quickly under certain conditions of camera movement. The technique uses a dense matching (optical flow OF) between two successive frames and exploits the relation between OF and camera motion (*i.e.* ego-motion) to create a cumulative voting space where a parametric curve is emphasized when the plane exists in the scene. The planar segmentation is then simplified to a problem consisting in curve parameters fitting which is itself reduced to maxima finding in a voting space. The computation time should be reduced.

3.3 Main assumptions and model

First of all, let us consider the coordinates system of figure 3.2. $OXYZ$ is the real-world 3D coordinates system with an optical axis OZ. oxy is the image coordinates system which represents the basis system for each projection of a 3D point $P(X, Y, Z)$ onto image plane at $p(x, y)$. The relation between the coordinates of P and p can be proved

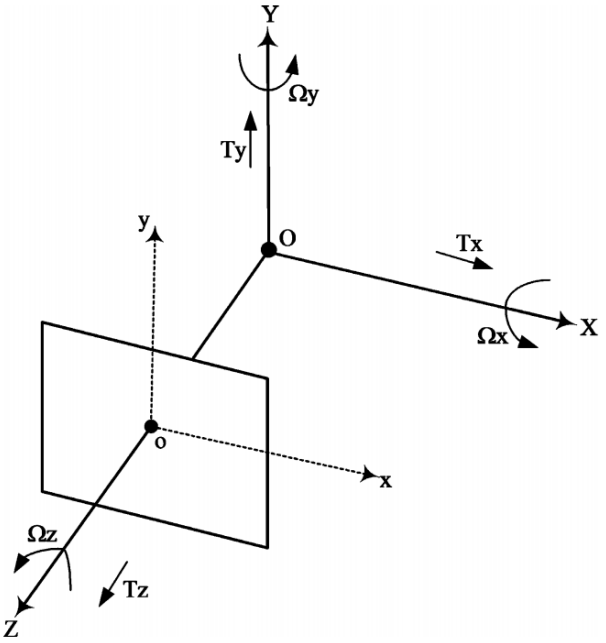


FIGURE 3.2: Coordinates of the system related to the monocular sensor and to real world, and notations used in the thesis.

simply by basic geometry :

$$\begin{aligned} x &= f\frac{X}{Z} \\ y &= f\frac{Y}{Z}, \end{aligned} \tag{3.2}$$

where f is the focal length of the sensor.

Higgin [LHP80] proves that in a rigid scene, when the pin-hole camera moves with translational movements $\mathbf{T} = [T_X, T_Y, T_Z]^\top$ and rotational movements $\Omega = [\Omega_X, \Omega_Y, \Omega_Z]^\top$, the two components of 2D motion (assimilated to optical flow) $[u, v]^\top$ of a point $[x, y]^\top$ on image plane can be expressed as :

$$\begin{aligned} u &= \frac{xy}{f}\Omega_X - \left(\frac{x^2}{f} + f\right)\Omega_Y + y\Omega_Z + \frac{xT_Z - fT_X}{Z} \\ v &= -\frac{xy}{f}\Omega_Y + \left(\frac{y^2}{f} + f\right)\Omega_X + x\Omega_Z + \frac{yT_Z - fT_Y}{Z} \end{aligned} \tag{3.3}$$

If a plane exists in the world coordinates with the normal vector $\mathbf{n} = [n_X, n_Y, n_Z]^\top$ and has distance d to the origin, its mathematical equation can be written using 3D plane equation and equation 3.2 :

$$|n_X X + n_Y Y + n_Z Z| = d \Longleftrightarrow \frac{|Z||n_X \frac{fX}{Z} + n_Y \frac{fY}{Z} + n_Z f|}{f} = d \tag{3.4}$$

Since all observable points are located in front of the image sensor, all Z values are positive $Z > 0$. Therefore, by using the relation (3.2), the equation (3.4) can be rewritten in the following way :

$$\frac{1}{fd}|n_X x + n_Y y + n_Z f| = \frac{1}{Z} \tag{3.5}$$

By replacing equation (3.5) into (3.3), we can model the relation between points of a plane and its optical flow :

$$\begin{aligned} u &= \frac{xy}{f}\Omega_X - \left(\frac{x^2}{f} + f\right)\Omega_Y + y\Omega_Z + \frac{(xT_Z - fT_X)|n_X x + n_Y y + n_Z f|}{fd} \\ v &= -\frac{xy}{f}\Omega_Y + \left(\frac{y^2}{f} + f\right)\Omega_X + x\Omega_Z + \frac{(yT_Z - fT_Y)|n_X x + n_Y y + n_Z f|}{fd} \end{aligned} \tag{3.6}$$

These equations were used in the study of [LHP80] to separate rotational movements from optical flow under certain assumptions. Horn et al. [HW88] use these equations as constraints to estimate optical flow in some particular cases such as pure translational or pure rotational movement of the camera. The cumulative method c-velocity, as well as the proposed uv-velocity, also exploit the special traits of these two equations to design specific voting spaces and achieve planar segmentation.

3.4 c-velocity

c-velocity [BZ12] has been inspired by Labayrade's work [LAT02] on V-disparity. Instead of using the stereovision, c-velocity is designed for temporal mono-vision. Instead of using the disparity, the optical flow is used to reveal the main structure of the scene. Using a calibrated and rectified stereovision rig, the extraction of planar surfaces is reduced to the detection of segments in the uv-disparity maps. Extracting planes in a monocular system, by using optical flow requires to formulate a few assumptions. As for uv-velocity, the use of c-velocity is limited, for the moment, to a few specific types of camera motion. The design of the c-velocity space and its analysis methods has been driven by two hypotheses :

First of all, the camera movement is translational. Indeed, c-velocity treats the pure translational movement of camera at first step, since the planes information lies totally on the translational component in the equation (3.6).

Secondly, the Manhattan world assumption is satisfied where the planes are parallel to the planes created by any two main axes. In the case of a mobile camera, either mounted on a vehicle, or on a mobile device, the scene consists of three categories of planar

surfaces : horizontal, lateral and frontal, relatively to the camera. This assumption holds particularly in urban scenes, when the camera is mounted on a car typically evolving on the horizontal road, which is most generally parallel to the buildings. The horizontal planes, for which the normal vector can be written as $\mathbf{n} = [0, 1, 0]^\top$, represent the road or the ground of the vehicle. It can also represent the ceiling of a room in indoor environments. The vertical planes, for which $\mathbf{n} = [1, 0, 0]^\top$, are related to the buildings likely to be present on each side of the road. It can also stands for walls in an indoor environment.

To finish, the frontal planes $\mathbf{n} = [0, 0, 1]^\top$ can represent some obstacles in front of the vehicle, or buildings in the distance of an urban scene. In indoor places, it can be related to obstacles (objects, pieces of furniture, etc.) or walls. With the two hypotheses above mentioned, the equation (3.6) can be simplified in three different ways depending on the type of plane under consideration. As camera moves from time t to $t + 1$, the term T_Z is viewed as an unknown constant in the equations (3.6). This is also true for the focal length f which depends on the camera and the distance d from the plane to the origin. Let x_{FOE} and y_{FOE} be the coordinates of the focus of expansion point FOE (depending only on the 3D translation) as expressed by the equation (3.7) :

$$\begin{aligned} x_{FOE} &= f\frac{T_X}{T_Z} \\ y_{FOE} &= f\frac{T_Y}{T_Z} \end{aligned} \tag{3.7}$$

All things considered, the expressions of the optical flow coordinates u, v as well as its norm w are significantly simplified, as shown in table 3.1.

In each of the three cases mentioned in table 3.1, an expression appears between the motion norm w and the variables x, y which can be written as : $w = Kc$ with a different constant K and a different expression c in each case :

$$\text{Horizontal planes} \quad c_H = |y|\sqrt{(x - x_{FOE})^2 + (y - y_{FOE})^2} \tag{3.8}$$

$$\text{Lateral planes} \quad c_L = |x|\sqrt{(x - x_{FOE})^2 + (y - y_{FOE})^2} \tag{3.9}$$

$$\text{Frontal planes} \quad c_F = \sqrt{(x - x_{FOE})^2 + (y - y_{FOE})^2} \tag{3.10}$$

In each case, the relationship between the motion norms and the c parameter is linear. Considering a two-dimensional voting space formed by the variables w and c, all pixels belonging to a same planar surface will vote for the same linear profile in this cumulative space. The slope K of each profile is determined by the distance d. Then the planes are

	Optical flow equations	Expression of motion norm w
Horizontal plane	$u = \frac{T_Z(x - x_{FOE})\lvert y\rvert}{fd}$ $v = \frac{T_Z(y - y_{FOE})\lvert y\rvert}{fd}$	$w = \sqrt{u^2 + v^2} = \frac{T_Z}{fd}\lvert y\rvert\sqrt{(x - x_{FOE})^2 + (y - y_{FOE})^2}$ $w = \frac{T_Z}{fd}c_H$
Lateral plane	$u = \frac{T_Z(x - x_{FOE})\lvert x\rvert}{fd}$ $v = \frac{T_Z(y - y_{FOE})\lvert x\rvert}{fd}$	$w = \sqrt{u^2 + v^2} = \frac{T_Z}{fd}\lvert x\rvert\sqrt{(x - x_{FOE})^2 + (y - y_{FOE})^2}$ $w = \frac{T_Z}{fd}c_L$
Frontal plane	$u = \frac{T_Z(x - x_{FOE})}{d}$ $v = \frac{T_Z(y - y_{FOE})}{d}$	$w = \sqrt{u^2 + v^2} = \frac{T_Z}{d}\sqrt{(x - x_{FOE})^2 + (y - y_{FOE})^2}$ $w = \frac{T_Z}{d}c_F$

TABLE 3.1: Development of optical flow norm based on 3 predefined plane models.

detecting by extracting the different signatures exhibited by the voting space. The next section focuses on the analysis of the c-velocity.

3.4.1 Voting space of c-velocity

For each of the three plane categories, a different voting space is needed since the constant term K and the c-value are expressed differently, as explained by equations (3.8) to (3.10). In order to better understand the relationships between the image and c-velocity space, let us have a glance to the iso-motion contour maps of figure 3.3, for the three categories of planes and for two different FOE. These lines reveal the distribution of points x, y where c-value and w are constant, therefore which all vote to a same single point in the voting space.

The analysis of the iso-motion contour maps allows us to visualize the interference produced by the contributions of other planes on the current voting space. As can be seen on figure 3.3, the distribution of iso-motion contours on three planes are totally different from each other, which means that the intervention of outliers is limited. For example, in the horizontal plane voting space, the iso-motion map consists of horizontal oriented curves parallel to each other. Each curve casts only one point (c,w) in that voting space if these points on that curve belong to horizontal plan. However the vote will be scattered on many w-values alongside these iso-motion curves if these points belong to lateral or frontal plane according to their proper iso-motion contour. Thus, it makes their votes insignificant at the end on this voting space. As a consequence, these outliers votes can be removed easily by a simple thresholding.

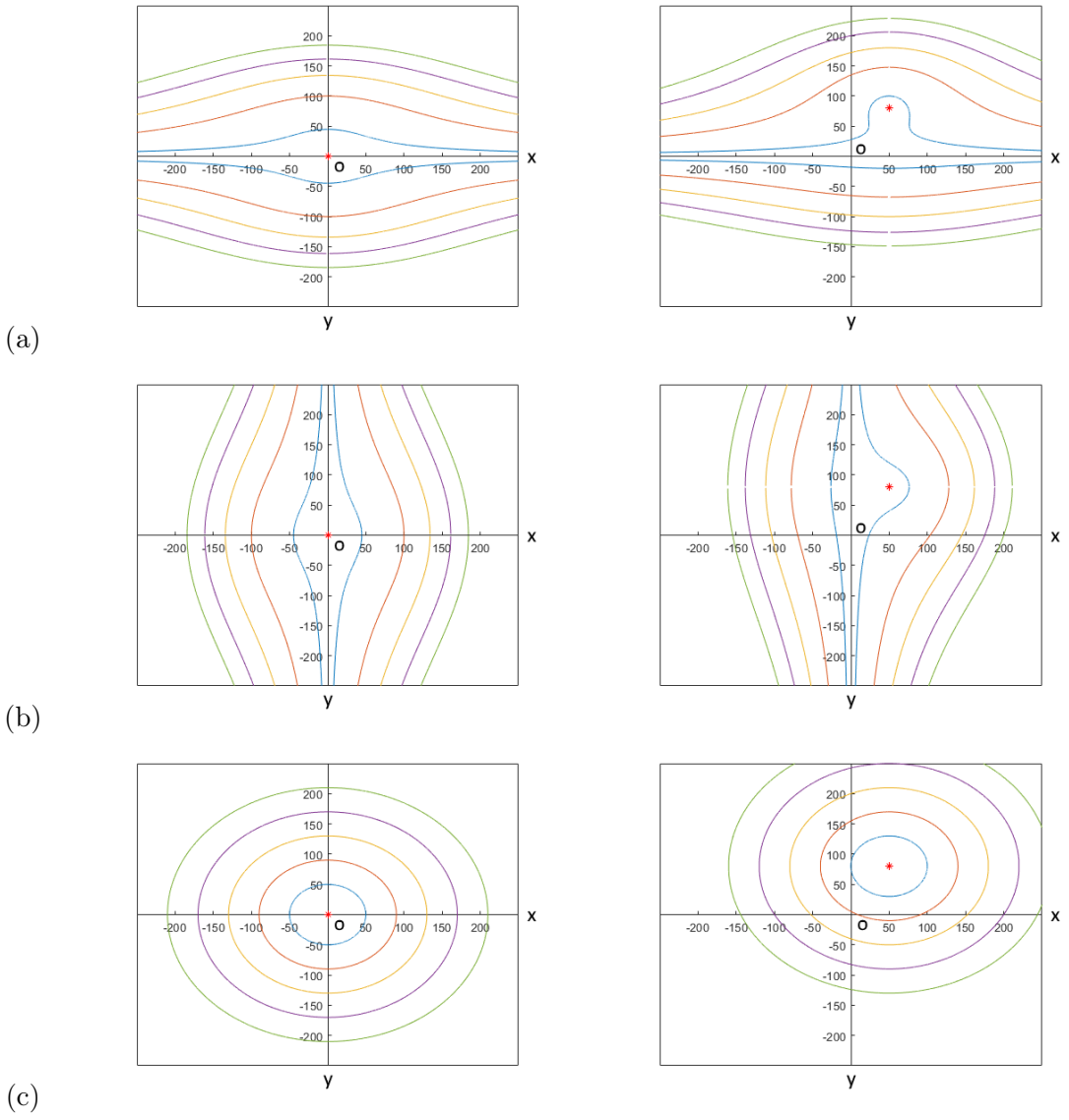


FIGURE 3.3: Examples of iso-motion contours (lines of constant velocity norm w) in the three different voting spaces : horizontal (a), lateral (b) and frontal (c). The horizontal and vertical axes represent the ox and oy on image (pixel). The left column shows the iso-motion contours for $x_{FOE} = y_{FOE} = 0$ while the right column shows the iso-motion contours for $x_{FOE} = 50, y_{FOE} = 80$. The red dot represents the FOE point. The line of same color draws the position of points having the same w (hence c), the value c increases as it goes further away the FOE point (blue→ orange→yellow →purpose→green).

It can be seen on figure 3.3 (a), for horizontal plane ($|Y| = d$, in mathematical terms), the plane with equation $\{Y = 0 \Rightarrow y = 0, \forall Y\}$ constitutes a degenerate case, for which all points project onto a straight line in the image. It is then impossible to detect this element. The same analysis can be done for lateral planes for $X = 0$. These configurations correspond to cases which are not realistic in the case of mobile robotics, since it would correspond to an horizontal plane (respectively a vertical one) that intersects with the optical axis and is perpendicular to the image plane. In other words, it would correspond to levitating planes.

With the interpretation made from the iso-motion contour maps, the voting space is defined as an accumulative matrix of two dimensions (c and w). In our study, c is the ordinate and w is the abscissa. The voting spaces for three planes are created following the steps described in Table 3.2.

INPUT : Optical flow (u, v)
OUTPUT : 3 voting spaces$(horizontal_vs, lateral_vs, frontal_vs)$
• Compute c_H, c_L, c_F according to Table 3.1 for each pixel
• Compute w from u and v
• Create 3 empty matrices $horizontal\_vs, lateral\_vs, frontal\_vs$
• For each point $p(x, y)$ in image :
Increment one value at $horizontal\_vs(c_H(x, y), w(x, y))$
Increment one value at $lateral\_vs(c_L(x, y), w(x, y))$
Increment one value at $frontal\_vs(c_F(x, y), w(x, y))$

TABLE 3.2: Procedure used to create the c-velocity voting spaces.

Theoretically, following the steps described in Table 3.2 leads to the voting spaces where linear profiles are revealed when planar surfaces are present in the scene.

It is important to address the problem of the discretization of each axis of the voting space, which is also a well-known problem in the Hough transform. The range of values that the variable c of horizontal and lateral planes can take is very wide and depends both on the image dimensions and on the FOE location.

For example, if we have an image of 320×240 and x_{FOE}=$y_{FOE} = 0$, which is the case where maximum of c is minimized, the c_H-value can reach 32000. To remedy this problem, instead of using the linear property $w = Kc$, a quadratic one is adapted by replacing c-value by $\sqrt{c}$ for **horizontal and lateral** planes.

$$w = K \times c = K \times (\sqrt{c})^2 = K \times c_s^2 \tag{3.11}$$

With the new c_s-value, the horizontal and lateral voting spaces are much compact while maintaining one parameter K to be estimated. As an example, figure 3.4 shows the 3 voting space under simulation. The detailed explanation on the creation of the simulated

planes is given further in Section 3.8.1. Each type of plane is drawn once : horizontal (red), lateral (green) and frontal (yellow) in the image. The frontal plane still keeps the linear profile and the quadratic relation appears clearly on each horizontal and lateral voting space corresponding to each type of plane.

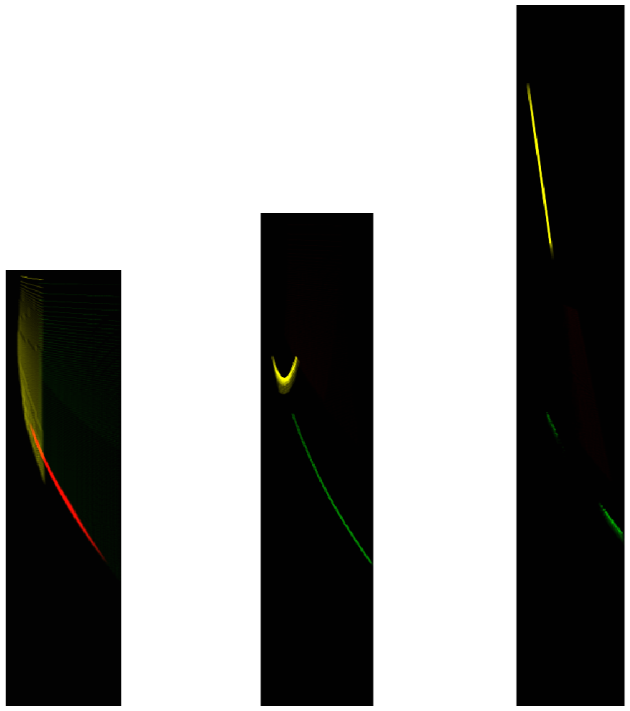


FIGURE 3.4: The horizontal, lateral and frontal voting spaces created from the optical flow computed from the simulation of pure translation movement, the votes of horizontal, lateral and frontal are represented in red, green and yellow respectively.

After creating the voting spaces, the one-parameter-quadratic and one-parameter-linear profiles are detected using Hough transform. It has been proved in the work of Qiong Nie [NBM12] that the creation of the c-velocity voting spaces is not required. Instead, the Hough space of parameters is formed directly from w and c and the estimation is done in a mutual way to improve the precision. This improves considerably the computation speed and effectiveness of the c-velocity.

The next Section introduces our contribution on the velocity-based voting spaces, by explaining the new uv-velocity strategy. The case of a pure translational motion of the camera is first considered. Then the approach is further extended to rotations.

3.5 uv-velocity on pure translational movement

As explained previously, the c-velocity can detect three kinds of planes under pure translational movement of the camera. However, the combination of u and v, through the

norm w, makes it difficult to develop further. Indeed, if the camera motion comprises a rotation component Ω_X, Ω_Y or Ω_Z, the expression of the norm w becomes extremely complicated. Hence, it becomes impossible to deduce a relationship between w and c. Equation (3.12) shows the expression of w derived from equation (3.6) when rotational movement Ω_Y is involved :

$$w = \sqrt{\left(\left(\frac{x^2}{f} + f\right)\Omega_Y + \frac{(xT_Z - fT_X)\alpha}{fd}\right)^2 + \left(-\frac{xy}{f}\Omega_Y + \frac{(yT_Z - fT_Y)\alpha}{fd}\right)^2}, \tag{3.12}$$

where $\alpha = |n_X x + n_Y y + n_Z f|$. Hence, we propose a new method called uv-velocity to detect planar surfaces which is able to expand the application domain of the original c-velocity technique. We investigate the capacity of uv-velocity in different conditions to prove its progressive trait. In order to alleviate the weaknesses of the c-velocity, we propose to study the possibility of a different voting space, which would use directly the components of the motion vectors u, v instead of their norm w. The first question that has to be addressed is the number and the shape of such voting spaces that would allow to detect the three types of planar surfaces : horizontal, lateral and frontal. First of all, we focus on the same assumptions as the c-velocity, *i.e.* a translational motion of the camera in a Manhattan world. The previous notations are kept. Table 3.3 collects the

	Optical flow equations	Observation
Horizontal plane	$u = \frac{T_Z(x - x_{FOE})\lvert y\rvert}{fd}$ $v = \frac{T_Z(y - y_{FOE})\lvert y\rvert}{fd}$	$u = \frac{T_Z(yx - yx_{FOE})}{fd} = \frac{T_Z yx - yfT_X}{fd} = F(x,y)$ $v = \frac{T_Z(y^2 - yfy_{FOE})}{fd} = \frac{T_Z y^2 - yfT_Y}{fd} = F(y)$
Lateral plane	$u = \frac{T_Z(x - x_{FOE})\lvert x\rvert}{fd}$ $v = \frac{T_Z(y - y_{FOE})\lvert x\rvert}{fd}$	$u = \frac{T_Z(x^2 - xx_{FOE})}{fd} = \frac{T_Z x^2 - xfT_X}{fd} = F(x)$ $v = \frac{T_Z(xy - xy_{FOE})}{fd} = \frac{T_Z xy - xfT_Y}{fd} = F(x,y)$
Frontal plane	$u = \frac{T_Z(x - x_{FOE})}{d}$ $v = \frac{T_Z(y - y_{FOE})}{d}$	$u = \frac{T_Z x - fT_X}{fd} = F(x)$ $v = \frac{T_Z y - fT_Y}{fd} = F(y)$

TABLE 3.3: Sum-up of the relationships between uv and xy, for the three categories of planar surfaces.

three relationships between the optical flow components (u, v) and the pixel coordinates (x, y). Note the sign of $|y|$ is neglected, we will further discuss this point. A parametric relationship is observed between (u, v) and (x, y) elements. For sake of clarity, we

introduce the following coefficients, that are constant for a given planar surface :

$$a_Z = \frac{T_Z}{fd}; \qquad a_Y = \frac{T_Y}{fd}; \qquad b_X = \frac{T_X}{d}; \qquad b_Y = \frac{T_Y}{d}.$$

The simplified relationships between the motion components (u, v) and the spatial coordinates (x, y) are collected in table 3.4.

	Optical flow equations
Horizontal plane	$u = a_Z xy - b_X y$ $v = a_Z y^2 - a_Y y$
Lateral plane	$u = a_Z x^2 - b_X x$ $v = a_Z y^2 - b_Y y$
Frontal plane	$u = a_Z x - b_X$ $v = a_Z y - b_Y$

TABLE 3.4: Relationships between (u, v) and (x, y) for specific plane models and camera motion

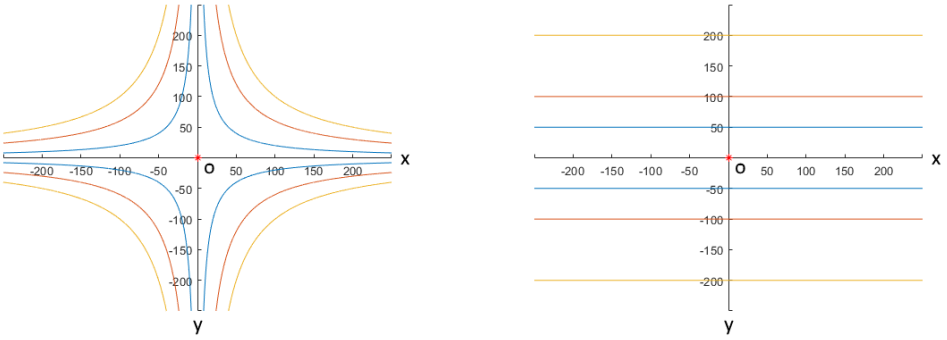
According to table 3.4, six different voting spaces can be designed. In fact, only two of them are needed. Regarding the case of horizontal planes, the v-element is a quadratic function of y, while the u component depends on xy, which is more difficult to handle in a voting space. For lateral planes, u is a quadratic function of x and for frontal planes, both relationships are linear. Therefore, according to these observations, only two bidimensional voting spaces $\mathbf{U}(u, x)$ and $\mathbf{V}(v, y)$ are needed to detect the signatures of the planar elements.

Like c-velocity, we also observe iso-motion contours of these two voting spaces in order to investigate the inter-model influence on each voting space.

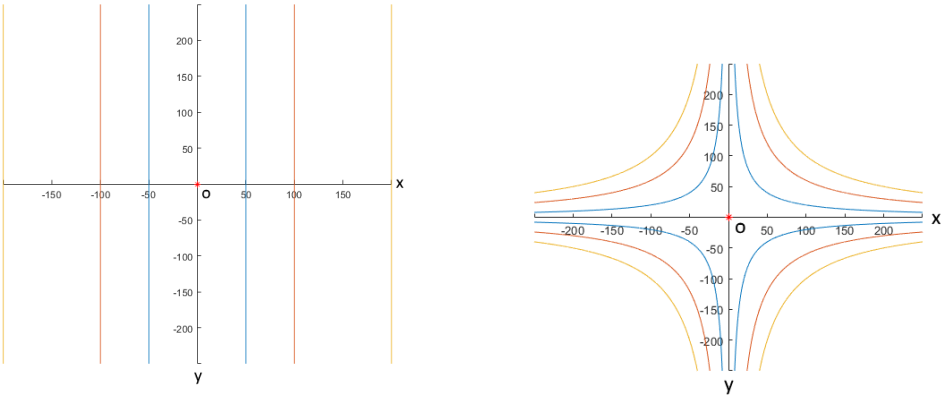
Figure 3.5 shows the iso-motion lines for each category of planes on two voting spaces $\mathbf{U}$ and $\mathbf{V}$. The iso-motion contours in this case draw the distribution of points having the same u or v elements on image plane.

First, let us focus on the horizontal plane in figure 3.5 (first row). The iso-v contours (right column) form straight lines parallel to the ox axis, so y is constant. Thus, these pixels create a consensus for the point (v, y). Considering all (v, y) values of a given horizontal plane, a prominent parabola $v = F(y)$ of significant votes appears in $\mathbf{V}$. Let us consider an image that also contains lateral planes, which corresponds to the iso-v lines of figure 3.5 (second row on the right). For a given y, the votes are scattered on various v-values, which can cause small interference for the detection of horizontal planes. However, the voting strategy proposed for parabola detection (described in section 3.7.1) allows to tackle the problem.

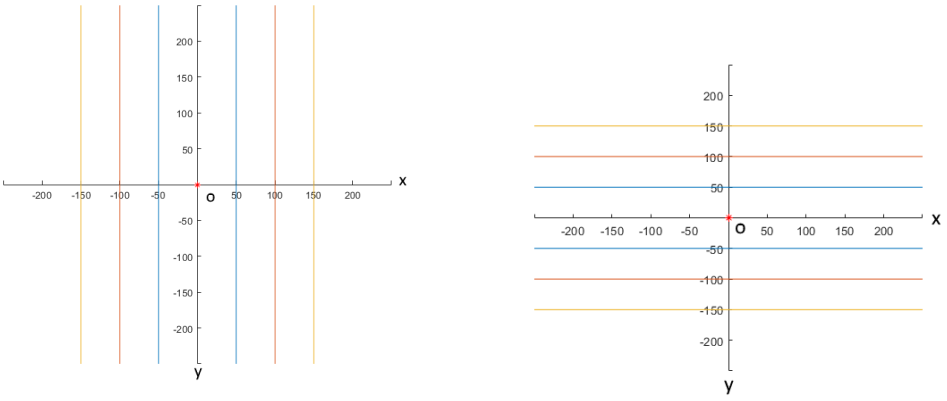
This explanation holds for lateral planes by inverting y and x and replacing v by u, and $\mathbf{V}$ by $\mathbf{U}$. The lateral plane creates a parabola $u = F(x)$ in $\mathbf{U}$. To finish, the iso-motion contours for frontal planes show linear relations on both $\mathbf{U}$ and $\mathbf{V}$.



Horizontal



Lateral



Frontal

FIGURE 3.5: The iso-motion contours drawn in the image space (y in abscissa and x in ordinate), for 3 types of planes and $x_{FOE} = y_{FOE} = 0$ (represented as a red dot). First column : lines of similar $|u|$ in voting space $\mathbf{U}$. Second column : lines of similar $|v|$ in voting space $\mathbf{V}$. The value ($|u|$ or $|v|$ increases as it goes further away from FOE point (blue→orange→yellow).

As a consequence, with the uv-velocity strategy, only two voting spaces are needed for detecting three types of planes. The creation steps of these voting spaces are detailed in table 3.5.

INPUT : Optical flow map (u, v)
OUTPUT : 2 voting spaces($\mathbf{U}(u, x)$ and $\mathbf{V}(v, y)$)
• Create 2 empty matrices $\mathbf{U}$ and $\mathbf{V}$
• For each point p of coordinates (x, y) in image :
Increment one value at $\mathbf{U}(u(x, y), x)$
Increment one value at $\mathbf{V}(v(x, y), y)$

TABLE 3.5: Procedure used to create the $\mathbf{U}$ and $\mathbf{V}$ voting spaces.

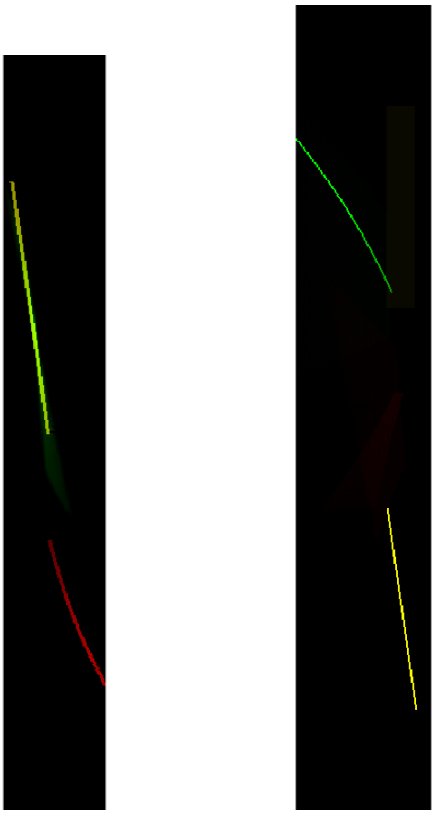
Considering a simulated image composed by the 2D motion of synthetic 3D moving planes like the one in the c-velocity part, figure 3.6 illustrates the contribution of each plane on the two voting spaces. These results corroborate the theoretical analysis. Indeed, a parabola curve appears for horizontal and lateral planes on $\mathbf{V}$ and $\mathbf{U}$ respectively, while the frontal plane produces a straight line on both voting spaces. In addition, the interventions of outliers models (lateral plane (yellow) on $\mathbf{V}$ voting space or horizontal plane (red) on $\mathbf{U}$) is minimized, which consolidates the above analysis of the iso-motion contours.

Discussion on v-disparity and uv-velocity. Let us show that the v-disparity [LAT02] is a special case of the uv-velocity. Note that the u and v notations do not refer to the same parameters : u and v are the pixel coordinates in Labayrade's work whereas they correspond to motion components in ours. According to the system depicted in [LAT02], which is a rectified stereoscopic system, the disparity is the horizontal (lateral) translation of each point from the left view to the right view. It corresponds to a translational motion where $T_X = T_Z = 0$ and $T_Y \neq 0$. Therefore, considering our notations, where u, v correspond to the motion, the component u is nul at each pixel. The horizontal motion components $v \neq 0$ correspond to the disparity.

Consequently, the relationships of table 3.3 are reduced to the expressions collected in table 3.6.

We retrieve the linear relationship between the spatial coordinates y and the disparity v, that is $v = Ky$ for the horizontal plane (the road) and the constant v for frontal plane which are already analyzed in v-disparity. Here, we see another linear relation $v = Kx$ for lateral planes which leads to another potential voting space $\mathbf{V}$ to detect these kinds of planes with disparity.

After this remark on the uv-disparity, the next section studies the extension of the uv-velocity, when more complex motions are considered.



(a) (b)

FIGURE 3.6: The **V**(a) and **U**(b) voting space created from the optical flow computed from the simulation of pure translation motion of 3D planes, the votes of horizontal, lateral and frontal are represented in red, green and yellow respectively.The horizontal axe represents the $v(u)$-values and the vertical axe represents the $y(x)$-values

	Observation
Horizontal plane	$u = 0$ $v = \frac{-yT_Y}{d} = F(y)$
Lateral plane	$u = 0$ $v = \frac{-xT_Y}{d} = F(x)$
Frontal plane	$u = 0$ $v = \frac{-T_Y}{d} = \mathbf{const}$

TABLE 3.6: Relationships between uv and xy in the special cas of the uv-disparity.

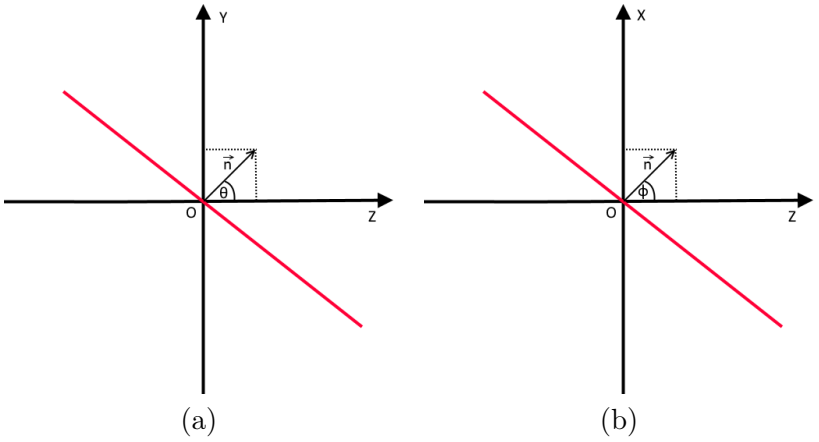
3.6 *uv*-velocity : extended planes and additional rotational movements

As we can see from the previous section, *uv*-velocity allows to characterize the optical flow of planes into parametric curves under the same assumptions and hypotheses than the original *c*-velocity, in particular the translational motion model of the camera. The expected advantage of the *uv*-velocity is an easier implementation and a faster execution, with similar results in terms of precision.

This section explores extensive cases of the *uv*-velocity where rotation is taken into account, therefore extending the notion of the three planes categories considered previously.

3.6.1 Extended planes

The previous explanation of the *uv*-velocity considers three main categories of planes (horizontal, lateral and frontal) by assuming a translation motion of the camera along the optical axis. Considering now the possibility of rotations, it is necessary to use another definition of this classification. We now consider two main categories : extended horizontal (ex-horizontal) and extended lateral (ex-lateral) planes.



(a) (b)

FIGURE 3.7: The new two types of planes : ex-horizontal (a) and ex-lateral (b). The red line represents the plane with normal vector **n**.

An ex-horizontal plane, described on figure 3.7 (a) is perpendicular to the (ZOY) plane (its normal **n** is therefore parallel to (ZOY)). If the angle between the normal vector **n** and (OZ) is θ, the normal vector of the plane is expressed as $\mathbf{n} = [0, sin(\theta), cos(\theta)]^\top$. The ex-horizontal planes represent the tilted road or obstacles in front of the camera. The horizontal and frontal planes are two special cases of ex-horizontal planes where the angle are $\theta = \frac{\pi}{2}$ and $\theta = 0$ respectively.

In a same way, the ex-lateral plane depicted by figure 3.7 (b) has a normal vector ($\mathbf{n} = [sin(\phi), 0, cos(\phi)]^\top$) parallel to (XOZ) plane with the angle ϕ between the normal vector

and (OZ). The ex-lateral planes allow us to model a wider range of buildings on the two sides of the road as well as different shapes of obstacles. Similarly to ex-horizontal planes, the lateral and frontal planes are special cases of ex-lateral planes where the angles are $\phi = \frac{\pi}{2}$ and $\phi = 0$ respectively. For example, when a vehicle moves forward behind another vehicle (a frontal plane) and then turns left or right, the frontal vehicle becomes a ex-lateral plane.

3.6.2 *uv*-velocity with additional rotational movement

Looking back to equation (3.6) together with the new definition of the extended planes, this section studies the ability of *uv*-velocity to exhibit the extended planes introduced previously. Let us consider that the motion of the camera is composed of a translation and either a rotation components Ω_Y (around axis OY) or Ω_X (around axis OX).

• Additional rotation Ω_Y :

Motion	$(T_X, T_Y, T_Z) \neq (0,0,0)$ $(\Omega_X, \Omega_Z) = (0,0), \Omega_Y \neq 0$
Equation	$u = -\left(\frac{x^2}{f} + f\right)\Omega_Y + \frac{xT_Z - fT_X}{Z}$ $v = -\frac{xy}{f}\Omega_Y + \frac{yT_Z - fT_Y}{Z}$
Plan ex-horizontal	$u = -\left(\frac{x^2}{f} + f\right)\Omega_Y + \frac{xT_Z - fT_X}{fd} \times \lvert ysin(\theta) + fcos(\theta)\rvert$ =F(x,y) $v = -\frac{xy}{f}\Omega_Y + \frac{yT_Z - fT_Y}{fd} \times \lvert ysin(\theta) + fcos(\theta)\rvert$ =F(x,y)
Plan ex-lateral	$u = -\left(\frac{x^2}{f} + f\right)\Omega_Y + \frac{xT_Z - fT_X}{fd} \times \lvert xsin(\phi) + fcos(\phi)\rvert$ =**F(x)** $v = -\frac{xy}{f}\Omega_Y + \frac{yT_Z - fT_Y}{fd} \times \lvert xsin(\phi) + fcos(\phi)\rvert$ =F(x,y)

TABLE 3.7: Overview of equations on each type of plane when $\Omega_Y \neq 0$.

Rotating around OY axis is the most common rotation for mobile robots or vehicles as it represents the turning left or right. The table 3.7 recalls the assumptions made on the camera motion model and provides the resulting OF equations, first for the general case, then for both categories of extended planes.

According to these equations, only the ex-lateral planes family can be recovered by analyzing the $\mathbf{U}(u,x)$ voting space since a quadratic relationship exists between u and

x, as demonstrated by equation (3.13) :

$$\begin{aligned} u &= \left(\frac{T_Z sin(\phi)}{fd} - \frac{\Omega_Y}{f}\right) x^2 + \left(\frac{-T_X sin(\phi) + T_Z cos(\phi)}{d}\right) x + f\Omega_Y - \frac{fT_X cos(\phi)}{d}, \forall x >= -fcotan(\phi) \\ u &= \left(-\frac{T_Z sin(\phi)}{fd} - \frac{\Omega_Y}{f}\right) x^2 + \left(\frac{T_X sin(\phi) - T_Z cos(\phi)}{d}\right) x + f\Omega_Y + \frac{fT_X cos(\phi)}{d}, \forall x < -fcotan(\phi) \end{aligned} \tag{3.13}$$

As this relation has two forms according to the position of x and the threshold $fcotan(\phi)$, an ex-lateral plane will be represented by up to two parabolas separated by

$$x_{thresh} = -fcotan(\phi)$$

in the voting space.

• Additional rotation Ω_X :

Motion	$(T_X, T_Y, T_Z) \neq (0,0,0)$ $(\Omega_Y, \Omega_Z) = (0,0), \Omega_X \neq 0$
Equation	$u = \frac{xy}{f}\Omega_X + \frac{xT_Z - fT_X}{Z}$ $v = \left(\frac{y^2}{f} + f\right)\Omega_X + \frac{yT_Z - fT_Y}{Z}$
Plane ex-horizontal	$u = \frac{xy}{f}\Omega_X + \frac{xT_Z - fT_X}{fd} \times \lvert ysin(\theta) + fcos(\theta)\rvert$ =F(x,y) $v = \left(\frac{y^2}{f} + f\right)\Omega_X + \frac{yT_Z - fT_Y}{fd} \times \lvert ysin(\theta) + fcos(\theta)\rvert$ =**F(y)**
Plane ex-lateral	$u = \frac{xy}{f}\Omega_X + \frac{xT_Z - fT_X}{fd} \times \lvert xsin(\phi) + fcos(\phi)\rvert$ =F(x,y) $v = \left(\frac{y^2}{f} + f\right)\Omega_X + \frac{yT_Z - fT_Y}{fd} \times \lvert xsin(\phi) + fcos(\phi)\rvert$ =F(x,y)

TABLE 3.8: Overview of equations on each type of plane when $\Omega_X \neq 0$.

The rotation Ω_X around the axis OX (the tilt) occurs less frequently than Ω_Y rotations on vehicles. It can occur on speed bumps for example. This kind of motion would occur more often on Unmanned Aerial Vehicles (UAV) or with everyday life mobile equipments like smartphones and tablets, where the camera movement is free. In this case the *uv*-velocity analysis can bring useful information about the scene.

As we can see in table 3.8, only the ex-horizontal planes can be exhibited as parabola in the $\mathbf{V}(v, y)$ voting space :

$$v = \left(\frac{T_Z sin(\theta)}{fd} + \frac{\Omega_X}{f}\right) x^2 + \left(\frac{-T_Y sin(\phi) + T_Z cos(\theta)}{d}\right) x + f\Omega_X - \frac{fT_Y cos(\theta)}{d}, \forall y >= -fcotan(\theta)$$
$$v = \left(-\frac{T_Z sin(\theta)}{fd} + \frac{\Omega_X}{f}\right) x^2 + \left(\frac{T_Y sin(\theta) - T_Z cos(\theta)}{d}\right) x + f\Omega_X + \frac{fT_Y cos(\theta)}{d}, \forall y < -fcotan(\theta) \tag{3.14}$$

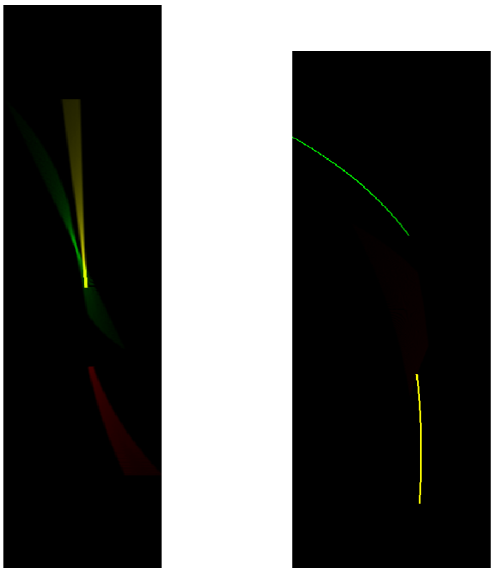


FIGURE 3.8: The **V** and **U** voting spaces created from the optical flow computed from the simulation with additional $\Omega_Y = \pi/20$ rotation. The ex-lateral plane, the frontal plane and horizontal plane are drawn in green, yellow and red respectively. The votes' color of these planes match their planes'.The horizontal axe represents the $v(u)$-values and the vertical axe represents the $y(x)$-values

Thus, when the camera motion involves rotations, the use of uv-velocity is limited to a few categories of planes. The pan motion, *i.e.* rotation along the Y axis, allows the detection of ex-lateral planes, while the tilt rotation allows the detection of ex-horizontal planes.

In order to validate these results, figure 3.8 shows the resulting uv-disparity voting spaces for Ω_Y rotation, with the color convention mentioned in the description. The voting space $\mathbf{V}(v, y)$ exhibits scattered curves associated to the three surfaces. According to the table 3.7 , this voting space is difficult to interpret, since the parameters of the expressions depend also on the x values which are not involved in this space. As a consequence, the intensity of these votes is low. The voting space $\mathbf{U}(u, x)$ displays the parabolic signatures of two ex-lateral planes : the lateral surfaces (in green) and the frontal surface (in yellow), which confirms the theory stated previously.

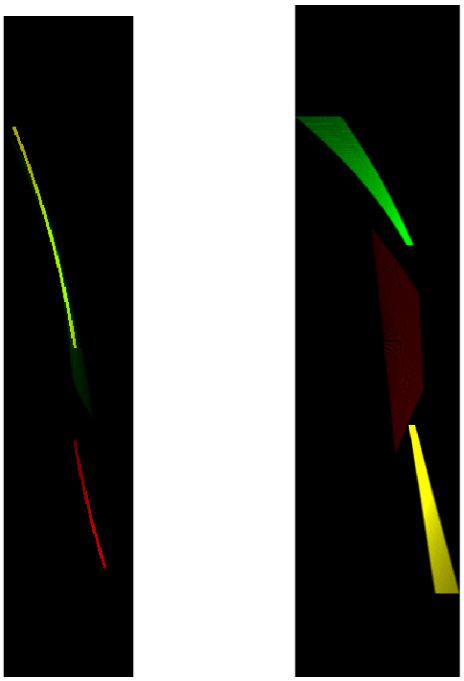


FIGURE 3.9: The **V** and **U** voting spaces created from the optical flow computed from the simulation with additional $\Omega_X = \pi/20$ rotation. The ex-lateral plane, the frontal plane and horizontal plane are drawn in green, yellow and red respectively. The votes' color of these planes match their planes'.The horizontal axe represents the $v(u)$-values and the vertical axe represents the $y(x)$-values

In a similar way, Figure 3.9 displays the uv-disparity voting spaces when a rotation Ω_X occurs. Here, $\mathbf{U}(u, x)$ can not be exploited. $\mathbf{V}(v, y)$ exhibits the signatures of two ex-horizontal planes : the horizontal (in red) and lateral planes (in green).

Thus, the simulation experiments confirm the parabolic relationships between $u - x$ for ex-lateral planes and $v - y$ for ex-horizontal planes.

As a conclusion, the $\mathbf{V}(v, y)$ and $\mathbf{U}(u, x)$ voting spaces bring the possibility to detect the extended planes with the additional rotational camera motion.

Sections 3.5 and 3.6 have studied the mathematical expressions for uv-velocity which can be able to address a wider range of applications in comparison to c-velocity, while maintaining the simplicity by using the direct transforms $u = F(x)$ and $v = F(y)$ to form the voting spaces (see table 3.9).

Table 3.9 compares the c-velocity and uv-velocity in terms of input data and dimensions, with H and W the image dimensions. The voting space dimension formula of c-velocity is based on horizontal voting space with $x_{FOE} = y_{FOE} = 0$. The parameters w_{max}, v_{max}, v_{min}, u_{max}, u_{min} are the maximum and minimum values of w, v, u.

	c-velocity	uv-velocity
Input	c_H, c_L, c_F and $w = \sqrt{u^2 + v^2}$	u and v
Voting space dimension	$\sqrt{\frac{H}{2}\sqrt{(H/2)^2 + (W/2)^2}} \times w_{max}$	$H \times \lvert v_{max} - v_{min} \rvert$ $W \times \lvert u_{max} - u_{min} \rvert$

TABLE 3.9: Comparison between voting space of c-velocity and uv-velocity .

Although the dimensions of the voting spaces are competitive between c-velocity and uv-velocity, uv-velocity has the advantages of avoiding the complicated expression of c-value and w-value before forming the voting spaces. In addition, only two voting spaces are needed to achieve the task of c-voting space. The performances of these two methods are compared further in Section 3.8 through simulations and experiments.

The next part gives practical details about the analysis of the voting spaces, *i.e.* the detection of the parametric curves in the voting spaces followed by the image segmentation into planar regions.

3.7 Analysis of the voting spaces

It has been shown that there are parametric curves revealed by the $\mathbf{V}(v, y)$, $\mathbf{U}(u, x)$ voting spaces when planes are present in the image. This section analyzes the behaviour of these curves on those voting spaces, including the c-velocity voting spaces, starting with the case of a pure translational motion model in paragraph 3.7.1. The case of a rotation is studied in a second step in paragraph 3.7.2.

3.7.1 Pure translational movement with three main planes

c-velocity voting space. c-velocity uses three voting spaces, one per category of planes : horizontal, lateral and frontal. According to Section 3.4.1, the horizontal and lateral voting spaces exhibit the parabola $w = K \times c^2$, while the frontal voting space reveals a linear relationship $w = K \times c$. Since each voting space is used to estimate one parameter, it is simple to find K following the steps described in table 3.10.

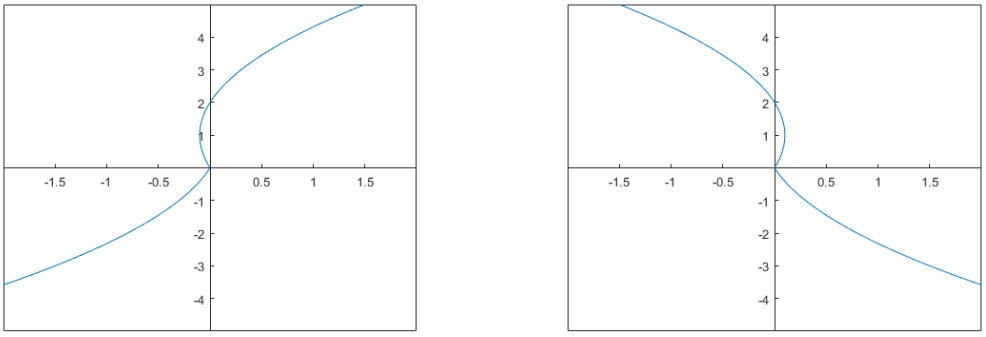
The threshold τ_I (I for inliers) is used to eliminate the low votes (outliers). The τ_N parameter is used to detect multi-parabola if they ever exist in voting space. The threshold is determined by experiences. In our case $\tau_N = H_{vs}/4$ where H_{vs} is the height of each voting space.

INPUT : Voting space $\mathbf{C}(c, w)$
OUTPUT : K
Create table TAB for stocking parameter $K = [0, \Delta, 2\Delta...N]$
For each point $p(i, j) > \tau_I$ in voting space $\mathbf{C}(c, w)$
    Compute $K_{ij} = \frac{i}{j^2}$
    Find the position of K_{ij} in TAB : $pos = \frac{K_{ij}}{\Delta}$
    Increase $TAB(pos)$ by one unit
End
Search in TAB voted position $peak$ where $TAB(peak) > \tau_N$
Return the K at $TAB(peak)$

TABLE 3.10: Parabola parameter estimation for c-velocity.

The straight lines related to frontal planes are estimated by using the steps of table 3.10 but with the linear relationship between i and j instead of the quadratic one.

***uv*-velocity voting space.** As we analyze the relation between $u - x$ and $v - y$ in table 3.3, the real form on function $v = F(y)$ (or $u = F(x)$) is not a natural parabola, as shown by figure 3.10, because of the absolute values $|y|$ (or $|x|$).



$T_Z > 0$ $T_Z < 0$

FIGURE 3.10: The two cases of "parabola" defined by a function $v = F(y)$. Horizontal and vertical axes represent the v-value and y respectively.

The parabola crosses the axis $v = 0$ at $y = 0$ and $y = y_{FOE}$ and changes sign as it crosses the limit $y = 0$. We show that this situation never occurs to the parabola to be detected for 3 categories of planar surfaces. For example, in horizontal plane $|Y| = d$, the projection on image plane on oy-axis is $y = f\frac{Y}{Z}$. So the depth of the plane determines the vertical expansion of the plane $y = F(Z)$. Supposing that $d \neq 0$ to avoid degenerated case where $|Y| = 0 \Rightarrow y = 0, \forall Z$, the horizontal plane appears as a straight line in image, and we have :

$$\lim_{Z\to\infty} y = f\frac{Y}{Z} = 0 \tag{3.15}$$

So the projection of a horizontal plane never crosses the ox, hence the sign can be safely removed. This observation is important for the segmentation procedure afterwards.

On each voting space, a parabola and a straight line are exposed because of the availability of three planes in image. As the number of voting spaces of uv-velocity decreases, the complexity of curve profile increases.

The parabola on these voting spaces does not follow the relation $y = Ax^2$ like c-velocity but has the form $y = Ax^2 + Bx$ (see table 3.3). We can use the Hough Transform to estimate directly two parameters A and B. The complexity is double compared to c-velocity in this case. However if x_{FOE} and y_{FOE} are known, the number of parameters can be reduced to only one for both parabola and line detection. Luckily, the FOE point, where all OF converge or diverge, is easy to be estimated by a voting scheme since there are only translational movements. Consequently, we consider the cases where FOE is not at infinity ($T_Z \neq \infty$).

$\star$ *Detection of the parabola curves (for horizontal and lateral planar regions)*

The parabola curve appears only for horizontal and lateral planes. Because the interpretation is almost the same in both cases, we consider only the equations related to horizontal planes for explanation. The original quadratic equation of horizontal plane in **V** voting space is :

$$v = \frac{T_Z}{fd}y|y| - \frac{T_Y}{d}|y| = \frac{T_Z}{fd}(y|y| - |y|y_{FOE}) \tag{3.16}$$

By removing the absolute values, this equation can be written as :

$$\begin{aligned} v &= K(y^2 - yy_{FOE}), \forall\, sign(yT_Z) \geq 0 \\ v &= -K(y^2 - yy_{FOE}), \forall\, sign(yT_Z) < 0, \end{aligned} \tag{3.17}$$

where $K = \frac{|T_Z|}{fd} > 0$. Then we find the relation :

$$v = sign(yT_Z)Kv', K > 0 \tag{3.18}$$

For a determined optical flow, the v' is the constant parabola which is drawn from the equation $v' = y^2 - yy_{FOE}$. The value v of optical flow is scaled proportionally to v' by the constant K. So instead of estimating directly two parameters A and B, the parabola can be found by estimating only $K > 0$. Note that the parameter K depends only on the proportional relationship between v and v', which makes the estimation easier.

The same interpretation for parabola can be deduced in an analogous way for lateral planes. **V** is replaced by **U**, leading to the following equations (3.19) and (3.20).

$$\begin{aligned} u &= K(x^2 - xx_{FOE}) = |K|u', \forall\, sign(xT_Z) \geq 0 \\ u &= -K(x^2 - xx_{FOE}), \forall\, sign(xT_Z) < 0 \end{aligned} \tag{3.19}$$

$$u = sign(xT_Z)Ku' \tag{3.20}$$

The constant which links u and the reference u' is found through different x-values. The steps of finding parameter K of these parabolas are detailed in table 3.11.

INPUT : Voting space vs, y_{FOE} or x_{FOE}
OUTPUT : K
Create table TAB for stocking parameter K with step Δ with N elements
Pre-compute $v'(j) = j^2 - jy_{FOE}$ or $u'(j) = j^2 - jx_{FOE}$, $\forall j$
For each point $p(i,j) > \tau_I$ in voting space vs
Compute $K_{ij} = \lvert\frac{v(p(i,j))}{v'(i)}\rvert$ or $K_{ij} = \lvert\frac{u(p(i,j))}{u'(i)}\rvert$
Find the position of K_{ij} in TAB : $pos = \frac{K_{ij}}{\Delta}$
Increase $TAB(pos)$ by one unit
End
Search in TAB voted positions $peak$ where $TAB(peak) > \tau_N$
Return the K at $TAB(peak)$

TABLE 3.11: Parabola parameter estimation for uv-velocity.

The role of each parameter is the same as the one in table 3.10, except that $v(p(i,j)), u(p(i,j))$ are the v, u-values at the position j in the voting space. Since K_{ij} is found by dividing $\frac{v(p(i,j))}{v'(i)}$, we avoid points where $v' \approx 0$. Moreover, the OF of magnitude under 0.5 is not relevant because of the weak precision of OF estimation and the models are not distinctive on these points (for example, on **V** voting space, each curve goes through y_{FOE} by value $v \approx 0$).

⋆ *Detection of the lines (for frontal planar regions)*

The detection of the straight lines corresponding to frontal planes in voting spaces **V** or **U** is more simple than the detection of the parabola. The expressions to be determined are given by equations (3.21) and (3.22) :

$$v = \frac{T_Z}{fd}(y - y_{FOE}) = Ky' \tag{3.21}$$

$$u = \frac{T_Z}{fd}(x - x_{FOE}) = Kx' \tag{3.22}$$

The straight lines always go through point $(0, y_{FOE})$ in $\mathbf{V}$ or $(0, x_{FOE})$ in $\mathbf{U}$ so we can shift the origin to that point. Consequently, the number of parameters is reduced to only one parameter K. The estimation is described in table 3.12.

INPUT : Voting space vs, y_{FOE} or x_{FOE}
OUTPUT : K
Create table TAB for stocking parameter K with step Δ with N elements
For each point $p(i,j) > \tau_I$ in voting space vs
Compute $K_{ij} = \lvert\frac{v(p(i,j))}{j-y_{FOE}}\rvert$ or $K_{ij} = \lvert\frac{u(p(i,j))}{i-x_{FOE}}\rvert$
Find the position of K_{ij} in TAB : $pos = \frac{K_{ij}}{\Delta}$
Increase $TAB(pos)$ by one unit
End
Search in TAB voted positions $peak$ where $TAB(peak) > \tau_N$
Return the K at $TAB(peak)$

TABLE 3.12: Line parameter estimation for uv-velocity.

Pixel Segmentation. Knowing the profile of each curve, a re-projection process is needed to segment the image pixels into planes according to the pair $w - c$ for c-velocity or $u - x$ and $v - y$ for uv-velocity. A fast and simple strategy consists in computing the reference value of each profile w_{ref}, u_{ref} or v_{ref} then compare the actual w,u or v with the reference to see if it fits in a limit δ. Then we can decide whether a pixel belongs to a profile and aggregate the pixel to the correspond plane. The summary of the reference values are shown in table 3.13.

	c-velocity	uv-velocity
Horizontal plane	$w_{ref} = Kc_H^2$	$v_{ref} = sign(yT_Z)Kv'$
Lateral plane	$w_{ref} = Kc_L^2$	$u_{ref} = sign(xT_Z)Ku'$
Frontal plane	$w_{ref} = Kc_f$	$v_{ref} = Ky'$ or $u_{ref} = Kx'$

TABLE 3.13: The used reference value to segment pixel into planar surface

As we can see in table 3.13, the norm of OF w is calculated straightforward thanks to the estimated K and the c-value. But the u_{ref} and v_{ref} are a bit more complex to compute since the formula involves the sign of T_Z, which is unknown.

However, in the pure translation movement case, this factor can be estimated by observing the sign of u and v of OF in the 4 quadrants separated by $x = x_{FOE}$ and $y = y_{FOE}$. Figure 3.11 shows the typical OF shapes in the image when the camera moves forward ($T_Z > 0$). The dominant sign of u and v are detailed in table 3.14. A voting scheme is used to determine the dominant sign in each quadrant. One or all 4 quadrants can be analyzed to ensure the sign of T_Z.

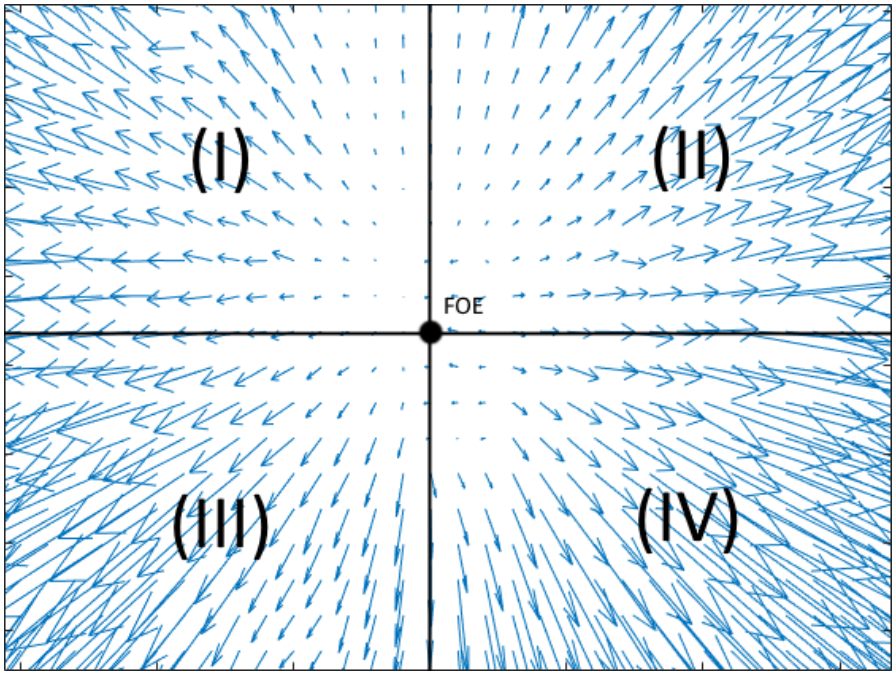


FIGURE 3.11: The typical distribution of OF on 4 quadrants divided by FOE in case $T_Z > 0$.

Quadrant	sign(u)	sign(v)
I	+	+
II	-	+
III	+	-
IV	-	-

(a)

Quadrant	sign(u)	sign(v)
I	-	-
II	+	-
III	-	+
IV	+	+

(b)

TABLE 3.14: Dominant sign of and on each quadrant in two case : $T_Z > 0$ (a) and $T_Z < 0$ (b).

After the detection of the main curves in the voting spaces using Hough transform, a reference model is determined for each of them. The classification of the pixels consists in assigning them to the closest model. However, the outliers points (which do not belong to a planar surface) can also be assigned to one of the models when their optical flow coincidentally satisfies the comparison condition. This effect can be seen on the voting spaces of figure 3.4 and figure 3.6 where the main curve interferes with votes emanating from other models. In order to partially remedy this problem, knowing that horizontal or lateral planes projections on image never cross the middle line of the image (*oy* or *ox*

axes), we can analyze the voting space and segment the image on each half of image to avoid the interference of other models for horizontal and lateral planes.

In practice, that is in an urban environment, most planes are horizontal and lateral. The frontal planes are smaller (in terms of area in image) so their votes are not as numerous as the others. Thus, a hierarchical strategy is chosen, by first estimating the horizontal and lateral planes. The voting space dedicated for frontal planes is formed in a second stage, using the remaining pixels.

3.7.2 With additional rotational movement and ex-planes

As studied in Section 3.6, when rotation is considered, the voting spaces **U** and **V** has the capacity to reveal the ex-planes.

The equations (3.13) and (3.14) show that each plane can produce two different portions of parabola, depending on the pixel coordinates and the orientation of the surface. However, in most practical cases, a plane leads to only one parabola in the voting space. To prove that statement, we analyze the case of ex-lateral planes, by focusing on the **U** voting space.

The two parabolas of this kind of planes are separated by the threshold $x_{thresh} = -fcotan(\theta)$ from the expression $|xsin(\theta)+fcos(\theta)|$ which can be expressed in two cases :

$$\begin{aligned} xsin(\theta) + fcos(\theta), \forall x \geq -fcotan(\theta) \\ -xsin(\theta) - fcos(\theta), \forall x < -fcotan(\theta) \end{aligned} \tag{3.23}$$

If θ is small, the impact of $xsin(\theta)$ becomes negligible, we can safely remove the absolute sign, that is the case of frontal planes. In the case θ has average value, considering a camera with focus f, pixel size s_p, image size $W \times H$. The maximum value x that the image plane can capture at $Z = f$ (on image plane) is $x_{max} = s_p W/2$. The limit angle θ for which the x_{thresh} still lies inside the image is expressed by the following equation (3.24) :

$$\begin{aligned} -x_{max} < x_{thresh} < x_{max} \\ x_{max} > fcotan(\theta) > -x_{max} \\ \frac{x_{max}}{f} > cotan(\theta) > -\frac{x_{max}}{f} \\ |\theta| > |arccotan(\frac{x_{max}}{f})| \end{aligned} \tag{3.24}$$

Let θ_{lim} be the angle limit defined by $\theta_{lim} = arccotan(\frac{x_{max}}{f})$. A plane's angle $|\theta| < |\theta_{lim}|$ will lead to a threshold point x_{thresh} located outside the image vision field (out of zone created by $-x_{max}Ox_{max}$). Hence, it produces only one parabola in the voting space.

Those planes with $|\theta| > |\theta_{lim}|$ never reach the x_{thresh}. Indeed, considering the plane shown on figure 3.12 where $\theta > 0 \Rightarrow cot(\theta) > 0 \Rightarrow x_{thresh} < 0$. That means the x_{thresh} is always staying on the opposite side of the ray OA. The sign of x_{thresh} is inverted if $\theta < 0$, but the explication is still the same. Calling θ_0 the angle between Ox_{thresh} and OZ, A the projection of optical center on the plane and T the intersection between the ex-lateral plane and OZ, then at $Z = f$:

$$x_{thresh} = -f cotan(\theta) = -f\frac{OA}{AT} \tag{3.25}$$

We also have, according to the figure 3.12

$$tan(\theta_0) = \frac{|x_{thresh}|}{f} = \frac{OA}{AT} = cotan(\theta) \tag{3.26}$$

However, we also have $cotan(\theta) = tan(\frac{\pi}{2} - \theta)$. So the equation (3.26) becomes :

$$tan(\theta_0) = tan(\frac{\pi}{2} - \theta) \Rightarrow \theta_0 = \frac{\pi}{2} - \theta \tag{3.27}$$

Hence the angle between the Ox_{thresh} and OA is $\theta + \theta_0 = \frac{\pi}{2}$. This leads to the ray Ox_{thresh} parallel to the plane. Hence, the plane never crosses the x_{thresh} on image plane. When the $\theta \to \pi/2$, the x_{thresh} reaches the middle of image but at the same time the plane becomes pure lateral where its projection on image does not cross $x = 0$ as proved before. So in any case, there is only one parabola for an ex-lateral plane on voting space.

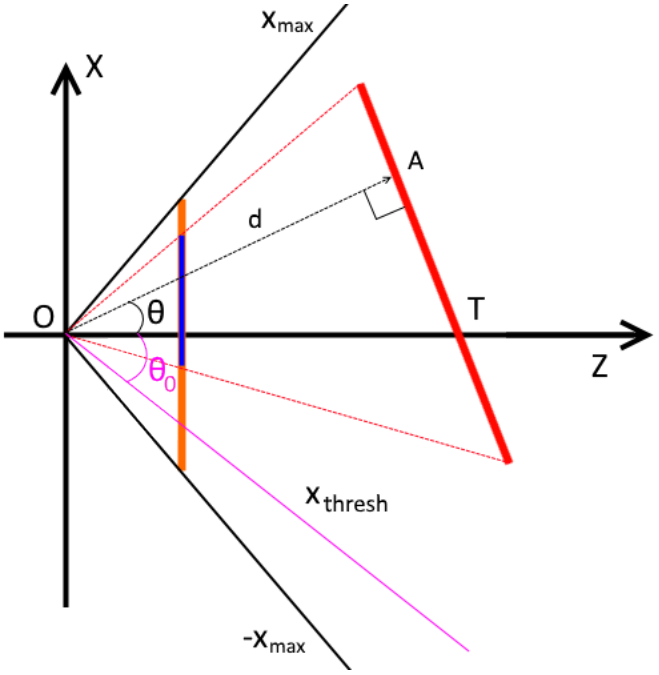


FIGURE 3.12: Illustration of ex-lateral planes red line and its projection (blue line) on image plane (orange line) .

For ex-horizontal planes, the interpretation is similar. We replace OX by OY and the angle is ϕ instead of θ. Therefore, for realistic situations, the voting space exhibits only

one parabola for each type of plane.

The parabolas follow the generic equation $y = Ax^2 + Bx + C$, where three parameters have to be estimated. In that case, a RANSAC technique is chosen instead of the Hough transform in order to reduce the calculation time and memory resources. Three points are randomly chosen in the voting space to compute the unknown A, B and C of the estimated parabola :

$$\begin{aligned} A &= \frac{y_3(x_2 - x_1) + y_2(x_1 - x_3) + y_1(x_3 - x_2)}{D}, \\ B &= \frac{y_3^2(x_1 - x_2) + y_2^2(x_3 - x_1) + y_1^2(x_2 - x_3)}{D}, \\ C &= \frac{y_2 y_3(y_2 - y_3)x_1 + y_3 y_1(y_3 - y_1)x_j + y_1 y_2(y_1 - y_2)x_3}{D}, \end{aligned} \tag{3.28}$$

where $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ are the coordinates of the 3 points and $D = (y_1 - y_2)(y_1 - y_3)(y_2 - y_3)$. The estimated parameters define the reference parabola. Then, points of the voting space are associated to this shape and considered as inliers when they are close enough to the curve (threshold τ_d), considering an Euclidean distance. Then, when the number N of inliers is high enough, the estimated parameters A, B and C are kept and the process stops. The details about this analysis are given in table 3.15, where max_iter is the maximum number of iterations for RANSAC. In the case of multiple parabola on the image, we remove the inliers of the previous estimation on the threshed voting space then proceed to the estimation again if the number of remaining pixels is higher than the 30% of voting space's height.

INPUT : Voting space vs, y_{FOE} or x_{FOE}
OUTPUT : A, B, C
Threshold the voting space vs with τ_I
For i =1 :max_iter
Choose randomly 3 points where $vs(i, j) = 1$ and extract their coordinates
Compute the A, B, C following Equ.3.28
Compute the distance d_{ij} of each point to the new estimated parabola
Find and count the N points for which $d_{ij} < \tau_d$
The parameter A, B, C are chosen and break the loop when $N > \tau_N$
End

TABLE 3.15: Procedure used for RANSAC estimation of the three parameters A, B, C of the parabola.

Once the parabola is detected and characterized, the relationship between motion (u, v) and spatial location (x, y) is fully determined. It corresponds to a set of reference motion values v_{ref} or u_{ref}. Each motion value (u, v) in the image is compared to the reference velocity and aggregated to the corresponding planar surface when it is similar enough

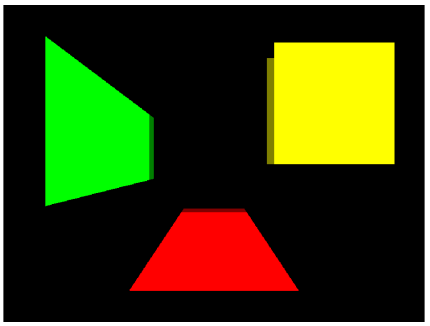


FIGURE 3.13: The 3 simulated planes with a camera movement $T = [0, 0, 0.5]^\top$: horizontal (red), lateral(green) and frontal (yellow). The brighter colors show the projection of the planes after movement.

(considering a tolerance of δ). The procedure is similar to the one explained in the "Pixel Segmentation" part above.

This section has covered the characteristics of voting spaces of c-velocity and uv-velocity and depicted steps to estimate the parameters of curves from these voting spaces. The next section shows the performance of the segmentation between these two voting spaces and compares the cumulative method to a direct homography-based method.

3.8 Experiments

The experiments are conducted following two scenario, first on synthesized images then on real sequences to compare the uv-velocity with c-velocity. The expected results are : a lighter computation, a lower amount of memory required while preserving the quality of the planar segmentaion. Then, the accumulative voting method uv-velocity is compared to a parametric method where planes are segmented by estimating the homography transformation [HZ03].

3.8.1 Simulation

For the first simulation experiment, we have created a point cloud for each of the three planes : horizontal, lateral and frontal. Then, these point clouds are projected to the image plane to form the first-view synthesized image. The coordinates of points are changed according to the pure translational movement of the camera and are projected again onto the image plane to form the second-view synthesized image. While projecting on image plane, the optical flow for each point is computed at the same time. The resulting OF map is used to create the voting space. Figure 3.13 shows the first simulation where the camera motion is a pure translational movement $T = [0, 0, 0.5]^\top$. We compare the results between c-velocity and uv-velocity in terms of segmentation quality, number of operators and memory use.

3.8.1.1 *c*-velocity

The three binarized voting spaces used in the *c*-velocity method are represented in figure 3.14. A parabola appears clearly on each voting space, confirming the presence of each category of planar surfaces, as discussed previously in Section 3.4.1. The frontal voting space uses the original c_F-value, that is without the root square, since its range is not too large compared to the other two *c*-values. Hence, we see a straight line on voting space figure 3.14(c) .

The parabola and lines are detected by analyzing each voting space (see figure 3.14). By using the segmentation strategy described in Section 3.7, we obtain the result shown on figure 3.16(a).

3.8.1.2 uv-velocity

Simulation for a translation of the camera. Under the same simulation as in the previous part, the two voting spaces **U** and **V** of *uv*-velocity are shown in figure 3.15 (a) and 3.15 (c).

The following parameters are used :

U	$\tau_I = 0.1H$	$\tau_N = 0.3W$	$\Delta = 0.5e-4$
V	$\tau_I = 0.1W$	$\tau_N = 0.3H$	$\Delta = 0.5e-4$

The thresholds are chosen by experiences in order to have a good visualization of the curves on voting spaces. The detected parabolas are shown in figure 3.15(b) and figure 3.15(d). A parabola and a straight line are found on each voting space. By using the same strategy used for *c*-velocity, the resulting segmentation is displayed on figure 3.16(b).

Thus, according to this simulation, *c*-velocity and *uv*-velocity have efficiently transposed the problem of plane segmentation to a problem of model fitting in a 2D space. Finally, the segmentation results are qualitatively similar, as illustrated in figure 3.16. Indeed, by looking at the table 3.16 comparing the accuracy of segmentation between two approaches, the performances are almost identical. The accuracy of segmentation is computed by the equation 3.29 :

$$Accuracy = \frac{TP + TN}{N} \tag{3.29}$$

where TP (true positive) is the number of pixels correctly labeled in a plane region, TN (true negative) is the number of pixels correctly attributed in a non-plane region and N is the total number of available pixels.

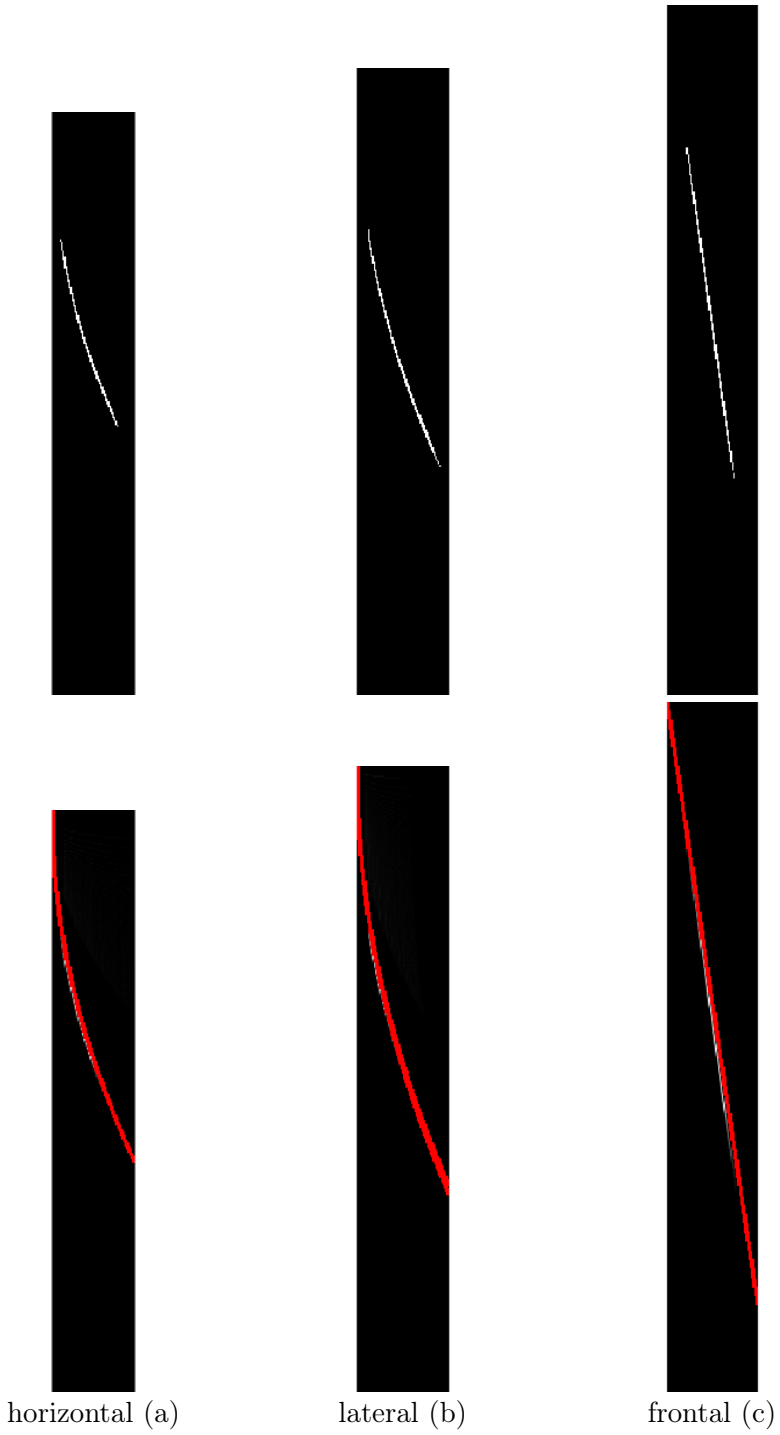
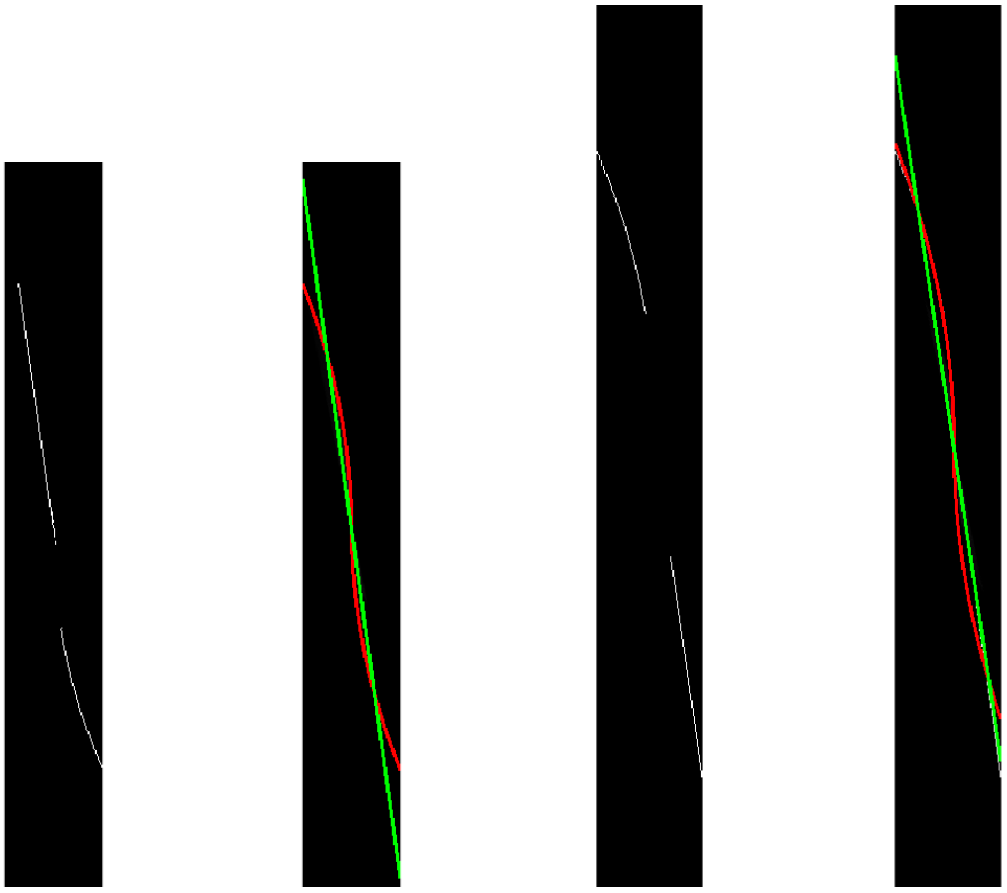


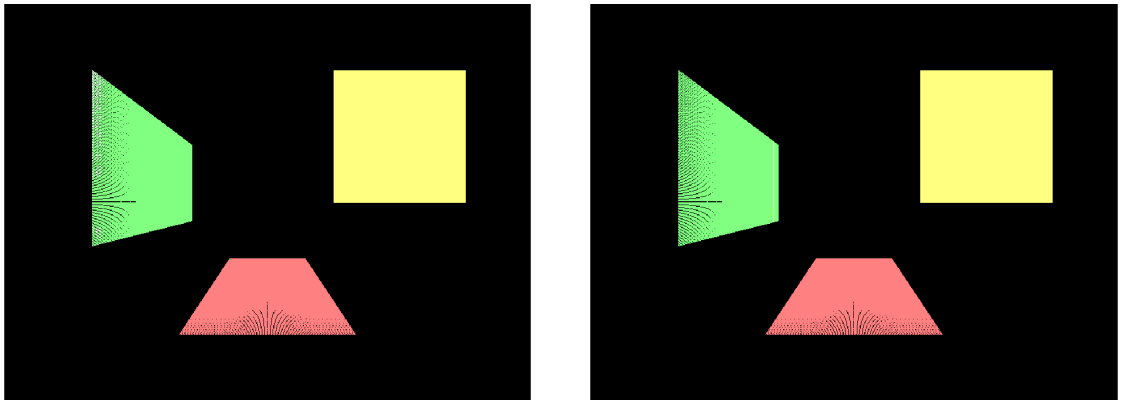
FIGURE 3.14: The three binarized voting spaces for a pure translation movement $T = [0, 0, 0.5]\top$ (top row) and the detected curve on these voting spaces (bottom row). The horizontal axe represents the c-values, the vertical axe represents the w-values

Table 3.17 shows a more detailed comparison, regarding the total amount of memory bytes required, the number of operations, and the execution times. Obviously, uv-velocity needs only two voting spaces and is much more simple an rapid to compute.



(a) (b) (c) (d)

FIGURE 3.15: The voting spaces (**V** (a), **U**(c)) with the detected parabola (red) and line (green) on each voting space (**V**(b) and **U** (d)) for the experiment illustrated by figure 3.13. The horizontal axe represents the u or v values, the vertical axes represents the H or W value for **V** or **U** respectively



(a) (b)

FIGURE 3.16: The 3 planes detected by (a) c-velocity and (b) uv-velocity : horizontal (red), lateral(green) and frontal (yellow)

Consequently, considering a translational motion of the camera, the uv-velocity can replace c-velocity and is able to detect the three basic planes : horizontal, lateral and frontal.

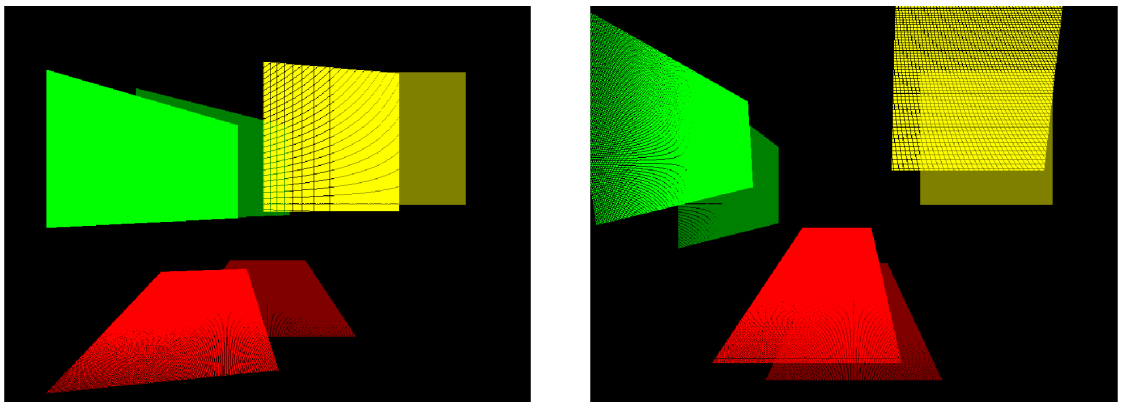
Plane	*uv*-velocity	*c*-velocity
Horizontal	1	1
Lateral	0.99	0.99
Frontal	0.99	1

TABLE 3.16: Accuracy comparison for each plane segmentation model between *c*-velocity and *uv*-velocity

	c-velocity	uv-velocity
Memory usage	$2.008M$ float-unit	$125.4K$ float-unit
Nb.operator	$2.0958M$	$730.8K$
Execution time	12 seconds	6 seconds

TABLE 3.17: The detail of memory usage, number of operators and execution time for *c*-velocity and *uv*-velocity for the simulation experiments for an image size 800×600.

Simulation for rotations of the camera. The next simulation considers a camera movement composed of a translation $\mathbf{T}$ with a rotation component, either around OX (Ω_X) or around OY (Ω_Y) axis. The simulated planes are shown on figure 3.17(a) and (b) respectively for $\Omega_Y \neq 0$ and $\Omega_X \neq 0$.



(a) (b)

FIGURE 3.17: The 3 simulated planes with the camera movement $T = [-0.5, 0.1, 0.5]^\top$ and $\Omega_Y = -\pi/20$(a), $\Omega_X = -\pi/20$ (b) : ex-horizontal (red), ex-lateral(green) and frontal (yellow). The brighter color show the projection of the planes after movement.

The two corresponding voting spaces $\mathbf{U}$ and $\mathbf{V}$ after thresholding are shown in figure 3.18(a) and (c).

The parabolas are detected by using RANSAC, as detailed by table 3.15 with $\tau_I = 0.1W$ for $\mathbf{V}$ or $\tau_I = 0.1H$ for $\mathbf{U}$. The $max\_iter = 100$, $\tau_d = 4$ and $\tau_N = 0.3N_T$ where N_T is

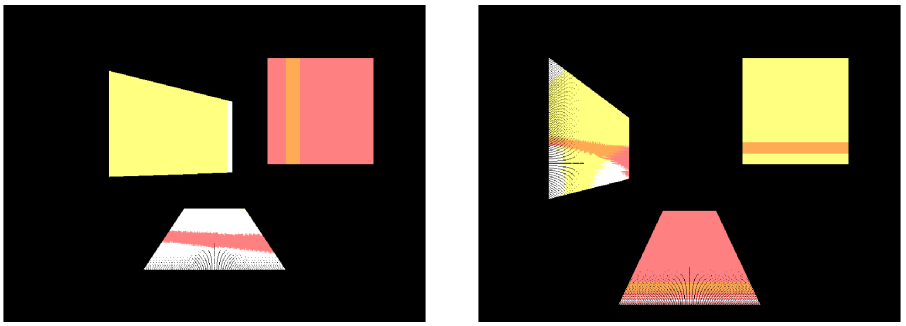
(a) (b) (c) (d)

FIGURE 3.18: Simulation results for a motion of the camera composed of a translation $T = [-0.5, 0.1, 0.5]^\top$ and a rotation $\Omega_Y = -\pi/20$ (see figure 3.17(a)). (a) and (c) : voting spaces **V** and **U**. (b) and (d) : detected parabolas (in red) in **V** and **U**. The horizontal axis represents the u or v values, the vertical axis represents the H or W value for **V** or **U** respectively

the total number of voters. The parameters are chosen according to the experiences in the experiments.

It can be seen on figure 3.18 (b) and (d) that display the binary voting space **U** and its segmentation, that the parabolas are well detected. This is due to the fact that the number of votes is high enough and well distributed on the parabolic shape. Unfortunately, on the voting space **V**, no parabola is found since the votes are weak and scattered across the voting space for all types of planes, as shown by the equations of table 3.7. The final segmentation results are shown in figure 3.19(a). In this case, it is impossible to classify planes that cross the principal horizontal/vertical lines, so the resulting segmentation is inevitably incorrect as well. The accuracy decreases to **0.91** for segmentation of ex-lateral plane (yellow) and **0.85** for segmentation of frontal plane (red).

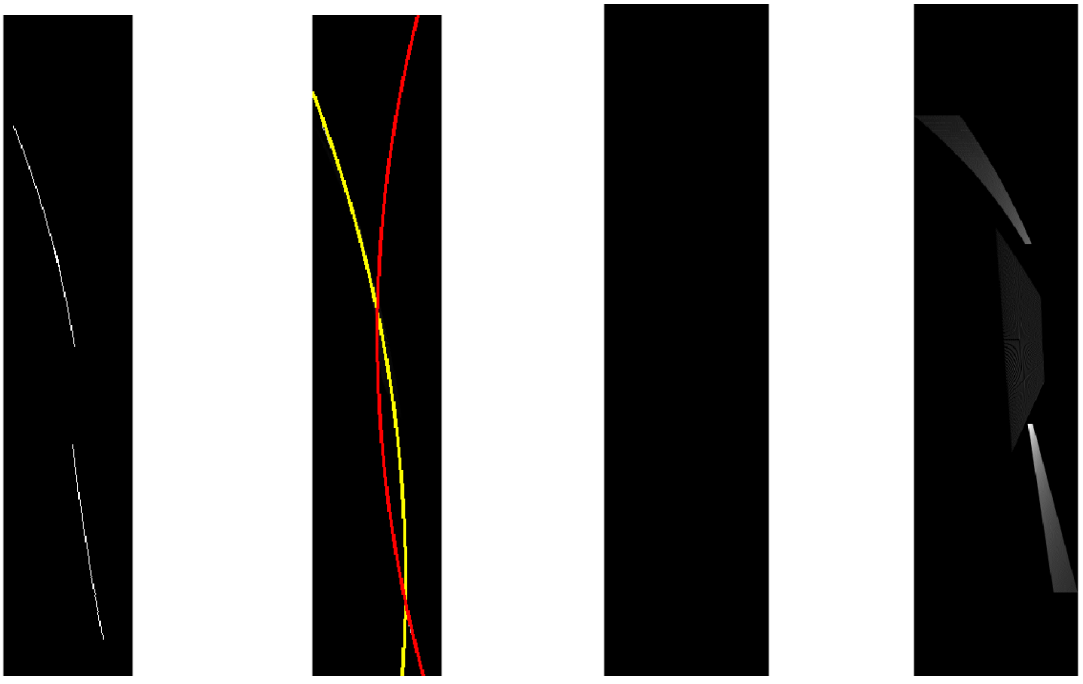
In the case of a rotational motion Ω_X visible on figure 3.17 (b), an analogous observation can be made on the two voting spaces **V** and **U** displayed on figure 3.20(a) and (c). In this case, the parabolas are found on **V** voting space. The segmentation is shown in figure 3.19(b). It undergoes the same defects as when Ω_Y rotation is considered. Although the two ex-horizontal planes are roughly extracted, the interference of other planes on voting space still causes a false segmentation. In this case, the accuracy drops to **0.81** for segmentation of ex-horizontal plane (red) and **0.75** for frontal plane segmentation (yellow).



(a) (b)

FIGURE 3.19: Segmentation of the planes corresponding to two parabolas detected in voting space $\mathbf{U}$(a) with addition Ω_Y and $\mathbf{V}$ with addition Ω_X

It is inevitable to avoid a false segmentation based only on optical flow models, especially for the frontal plane model, because pixels can belong to many planes. Beside the principal plane, a pixel always belongs to a global plane that groups all pixels who share the same Z. To improve the quality of the segmentation, we can envision to combine classical plane segmentation methods based on other feature similarity together with uv-velocity to refine the segmentation.



(a) (b) (c) (d)

FIGURE 3.20: The voting spaces ($\mathbf{V}$ (a), $\mathbf{U}$(c)) and found parabola (red) on each voting space ($\mathbf{V}$(b) and $\mathbf{U}$ (d)) for the experiment at figure 3.17(b).The horizontal axe represents the u or v values, the vertical axes represents the H or W value for $\mathbf{V}$ or $\mathbf{U}$ respectively

3.8.2 Real Sequences

To compare c-velocity with uv-velocity in real sequences, we use the KITTI dataset [MHG18]. Two types of OF are available for these sequences : the ground truth coming from KITTI dataset and the estimation OF produced by the method proposed in [SRB10].

3.8.2.1 Experiments for a translational ego-motion.

We first select the sequences for which the ego-motion of the camera is a pure translation. The OF ground truth values are only available sparsely on the road and on the obstacles. Hence, the threshold used to binarize the voting space is chosen as $min(0.02H, 0.02W)$ (H and W are image dimension) to adapt to the limit number of votes on the voting space. It is reduced five times compared to the previous experiments. Three sequences are chosen in the dataset to illustrate the behavior of c-velocity and uv-velocity. Figure 3.21, displays the ground truth and the estimated OF for one sequence. The ground truth of segmentation is created manually and shown in figure 3.22. Since the ground truth of optical flow is sparse and focuses only on the road region, we adapt the ground truth of segmentation to work only on the road (figure 3.22(a)) and compare the performance of the segmentation only for horizontal plane using (the lateral plane are not available). However, with the dense estimated optical flow, the lateral planes can also be detected, so we also compare the accuracy of lateral plane segmentation (figure 3.22(b)).

(a) (b)

FIGURE 3.21: The ground truth (a) and the estimated OF (b) of the sequence.

For the *true* optical flow (the optical flow provided by the dataset), the corresponding c-velocity and uv-velocity voting spaces are shown in figures 3.23, 3.24 and 3.25 respectively. As horizontal and lateral planes never cross the central axis of the image, we divide these two voting spaces into two halves in order to reduce the interference produced by other models. In fact, these two planes are considered first. Then the remaining pixels are used to detect the frontal planes.

(a) (b)

FIGURE 3.22: The ground truth of segmentation (red for horizontal plane, green for lateral plane) for sparse ground truth optical flow (a) and dense optical flow (b).

(a) (b) (c)

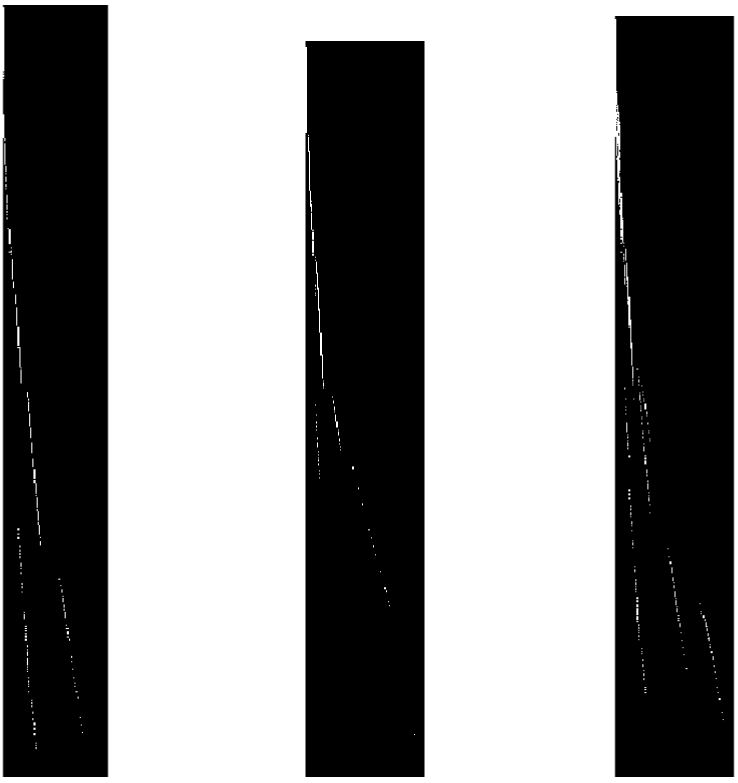
FIGURE 3.23: Horizontal voting spaces corresponding to OF of the half-top (a) and the half-bottom (b) of image in figure 3.21(a). The found parabola on half-bottom voting space is drawn in red (c). The horizontal axis represents the w-value and the vertical axis represents the c-values

The profiles of the curves are characterized as explained in Sections 3.8.1.1 and 3.8.1.2. The segmentation results are shown in Figure 3.26.

While the true OF allows to produce sharp curves in the voting spaces, the estimated OF produces thicker curves. This is visible on figure 3.27.

Despite the imprecision of the estimated optical flow, the binarized voting space of c-velocity (see figures 3.28 and 3.29) and uv-velocity (see figure 3.30) can still reveal the parabolic curves thanks to the consensus votes. However, the consequence of that imprecision is that the curve detection is more difficult and the planar segmentation is less precise than with the true OF (see figure 3.31). Table 3.18 recaps the accuracy of segmentation of uv-velocity and c-velocity approaches for this sequence. Both approaches have nearly the same accuracy

At this stage, from the simulations to the experiments on real sequences, from precise optical flow to the estimated one, the segmentation results are quite similar for both c-velocity and uv-velocity. However, for uv-velocity, the voting space is more simple to build. This is visible in table 3.19 which compares the two methods in terms of memory



(a) (b) (c)

FIGURE 3.24: The lateral voting space for half-left (a) and half-right (b) and frontal voting space (c) for OF in figure 3.21(a). No parabola is found on these voting spaces. The horizontal axis represents the w-value and the vertical axis represents the c-values

Plane	uv-velocity	c-velocity
Horizontal (sparse OF)	0.95	0.89
Horizontal (dense OF)	0.90	0.89
Lateral (dense OF)	0.93	0.93

TABLE 3.18: Accuracy on each plane segmentation between c-velocity and uv-velocity for the studied sequence

usage, number of operators, and execution times. Consequently, uv-velocity can totally replace c-velocity.

3.8.2.2 Experiments for translation and rotation

In this section, we choose other sequences from the KITTI dataset where a rotational movement exists. There are mostly rotations around the OY axis since they correspond to images acquired by cameras mounted on a vehicle. The uv-velocity is able to segment the horizontal plane by applying the steps already described for the pure translational movement and ex-lateral plane models by using RANSAC to estimate a 3-parameters parabola on voting space $\mathbf{U}$ (see figure 3.32).

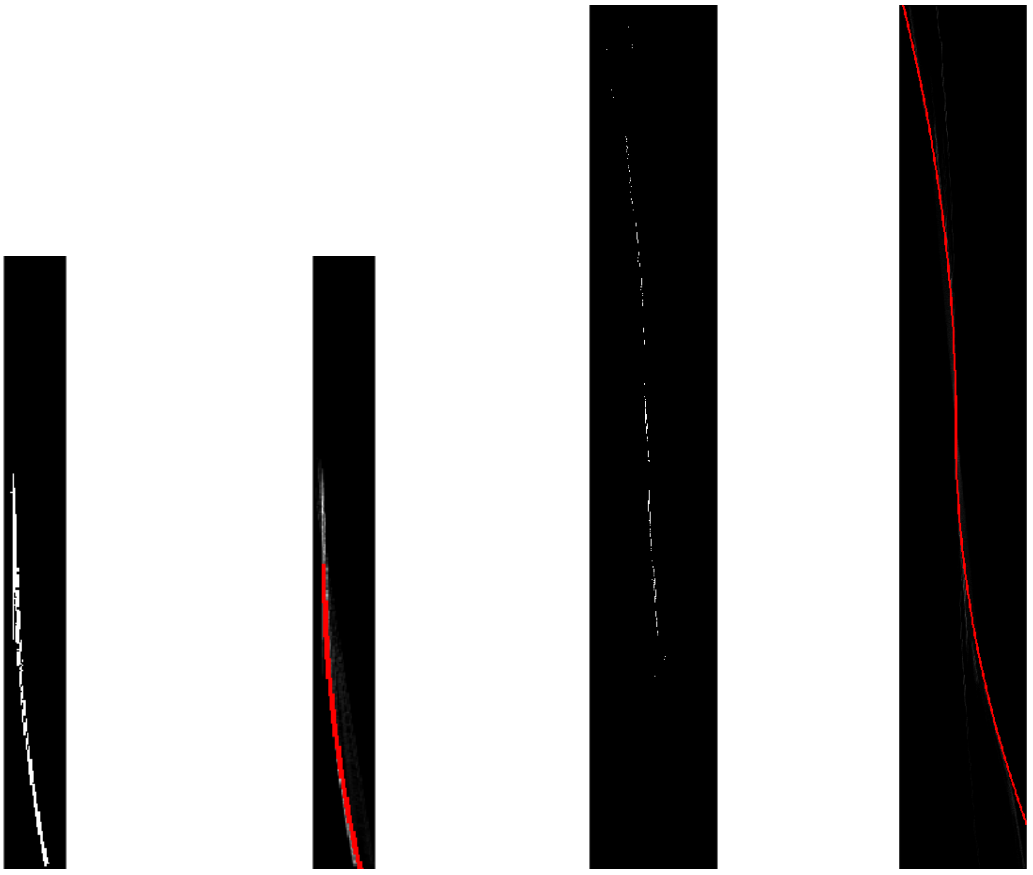


FIGURE 3.25: The **U**(a) and **V**(c) voting space for OF in figure 3.21(a). The detected parabola on **U**and **V**are drawn in red (b and d respectively).The horizontal axis represents the u or v values, the vertical axis represents the H or W value for **V** or **U** respectively.

	c-velocity	uv-velocity
Memory usage	$1.998M$ float-units	$131.9K$ float-units
Nb.operator	$2.1338M$	$729.4K$
Execution time	15 seconds	9 seconds

TABLE 3.19: The detail of average memory usage, operator and execution time of c-velocity and uv-velocity for the real sequence experiments with image size 1242×375.

Beside the same features as c-velocity, uv-velocity can adapt to additional rotation movements around OY axis (Ω_Y) or OX axis (Ω_X) to segment planar surfaces in the scene. Based on table 3.20 which displays the accuracy of the segmentation on two studied sequences, the precision of extended planes stays only at adequate level especially for the ex-lateral plane.

FIGURE 3.26: Segmented image according to the found parabola for c-velocity (top) and uv-velocity (bottom) on the ground truth OF. The horizontal plane is drawn in red

Plane	(1)	(2)
Ex-horizontal	0.91	0.90
Ex-lateral	0.84	0.85

TABLE 3.20: Accuracy on each extended plane segmentation using uv-velocity with additional rotational movement Ω_Y

3.8.3 Parametric versus voting method

The last experiments of the chapter aim to compare an homography-based method and our voting optical flow-based method uv-velocity. By matching points between two frames, the parametric method can be used to detect planes by estimating a consistent homography matrix ($\mathbf{H}$) from a set of points among all matching points. $\mathbf{H}$ can be found by using the DLT method (Direct Linear Transform) introduced in [HZ03] together with a RANSAC technique to remove outliers. The method will be detailed in the next chapter. We use six sequences, with the ground truth of OF, to find the horizontal plane. The masks of available optical flow are shown in figure 3.33(a). The hand-made ground truth of segmentation is adapted to these available points and shown in 3.33(b).

Figure 3.34 shows segmentation of the horizontal plane obtained with the parametric method (Figure 3.34(a)) and with uv-velocity (Figure 3.34 (b)). In terms of quality, both methods have decent results. We can remark however that the voting method gives a more dense segmentation map. Moreover, in the last two rows (sequence 5 and 6) of figure 3.34, even if there is a rotational motion component in the 3D motion, the road segmentation is still good even if the algorithm was designed for pure translational

(a) (b)

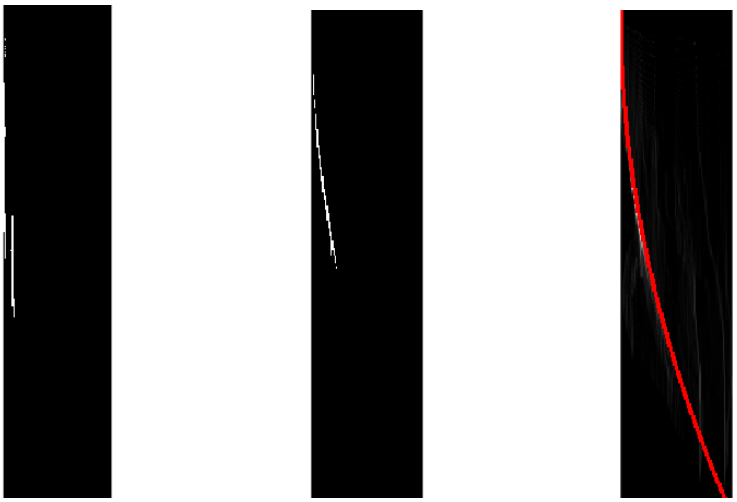
FIGURE 3.27: Comparison between original voting spaces created from ground truth of optical flow (a) and estimated optical flow (b). Top line shows c-velocity voting space for horizontal plane. Bottom line displays the $\mathbf{V}$ voting space

motion. Table 3.21 captures the accuracy of the segmentation between two approaches. Clearly, the segmentation by uv-velocity performs better than the classical homography approach.

Sequence	uv-velocity	classical homography
(1)	0.96	0.91
(2)	0.95	0.90
(3)	0.93	0.77
(4)	0.95	0.86
(5)	0.88	0.66
(6)	0.78	0.77

TABLE 3.21: The horizontal plane segmentation's accuracy of two approaches based on the hand-made segmentation ground-truth

In terms of calculation time, the voting method takes only 10 seconds while the parametric one uses averagely 120 seconds even the ground truth OF is only available in



(a) (b) (c)

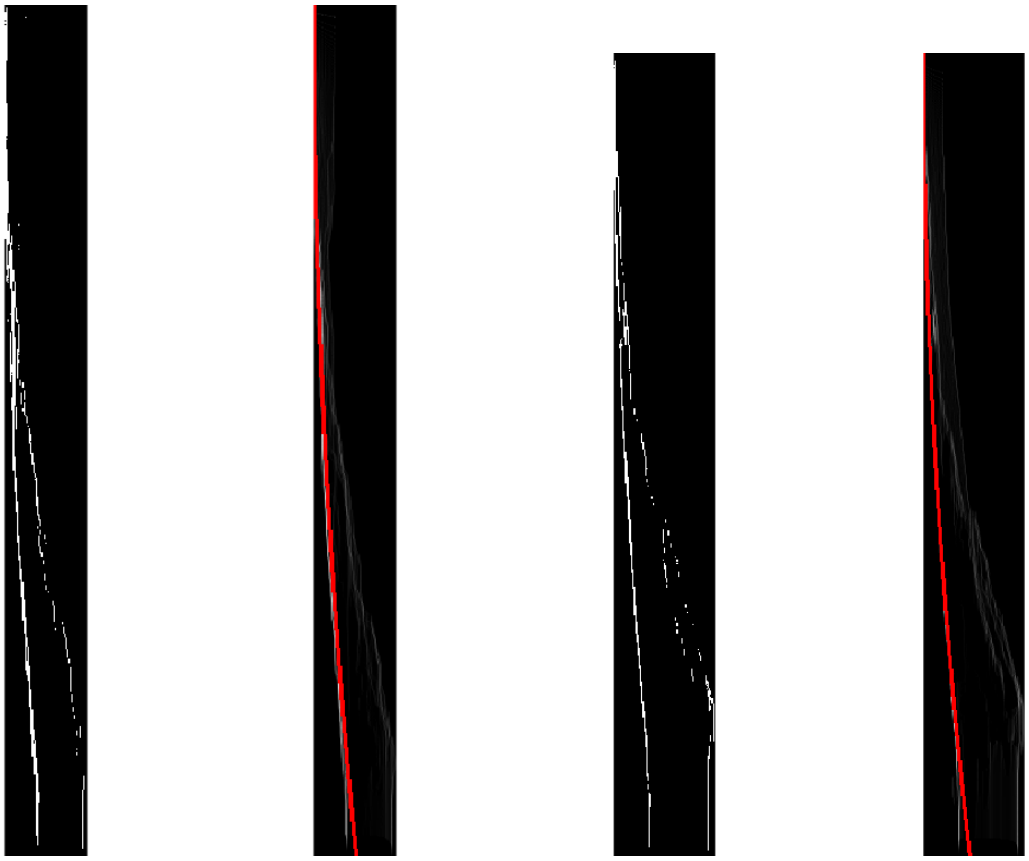
FIGURE 3.28: The c-velocity horizontal voting space for half-top (a) and half-bottom (b) for OF in figure 3.21 and the found parabola on half-bottom voting space is drawn in red (c).The horizontal axis represent the w-value and the vertical axis represents the c-values

approximately 30 % of the image. The difference comes from the size of data to be processed. In uv-velocity, each pixel (for which a motion has been estimated) votes once, which represents a large amount of data, but eventually only a few significant votes are used for model fitting. In the parametric method, a good homography matrix has to be estimated at each iteration of RANSAC, and for all pixels of the image.

3.9 Conclusion

This chapter has addressed the problem of monocular optical-flow based plane segmentation. It started by recalling the principles of the c-velocity method proposed in [BZ12] and has proposed the following contributions :

- A new method called uv-velocity, which can be seen as a generalization of the v-disparity method.
- An improvement of the performance in terms of computation time and reduction of memory resources. Instead of three voting spaces needed by c-velocity, only two are required for uv-velocity.
- An extension of the main paradigm. Indeed, the original approach was limited to a translational ego-motion of the camera. Even if this hypothesis is realistic when the camera is mounted on a vehicle which moves forward, we have extended the applicative spectrum of the method by considering a rotation component in the motion model.
- A comparison with a classic plane segmentation based on homography estimation and RANSAC.



(a) (b) (c) (d)

FIGURE 3.29: The c-velocity lateral voting space for half-left (a) and half-right (c) and frontal voting space (c) for OF in figure 3.21. Two most prominent parabolas are found on these two voting spaces (b) and (d). The horizontal axis represent the w-value and the vertical axis represents the c-values

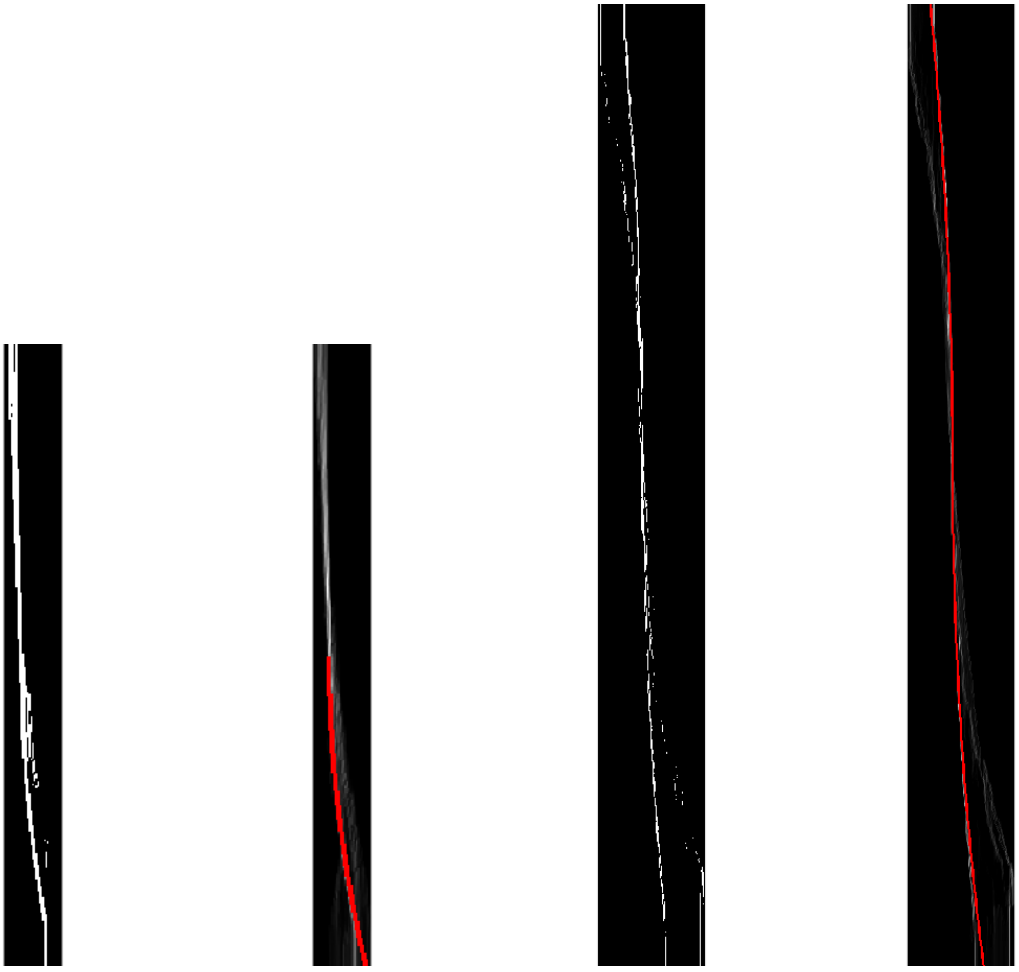
Note that the quality of the initial motion map (optical flow) plays an important role on the precision of the curves parameters and on the quality of the segmentation. Our contribution related to optical flow estimation presented in chapter 2 will undoubtedly be useful to improve plane segmentation using uv-velocity. This idea will be developed in chapter 5.

The work presented in this chapter has many perspectives, among them let us quote :

— The optical flow method proposed in chapter 2 provides a relevance map, that can be used as weights in the voting space.

— Plane segmentation when rotations are involved could be improved by considering gradually more general models depending on the targeted application.

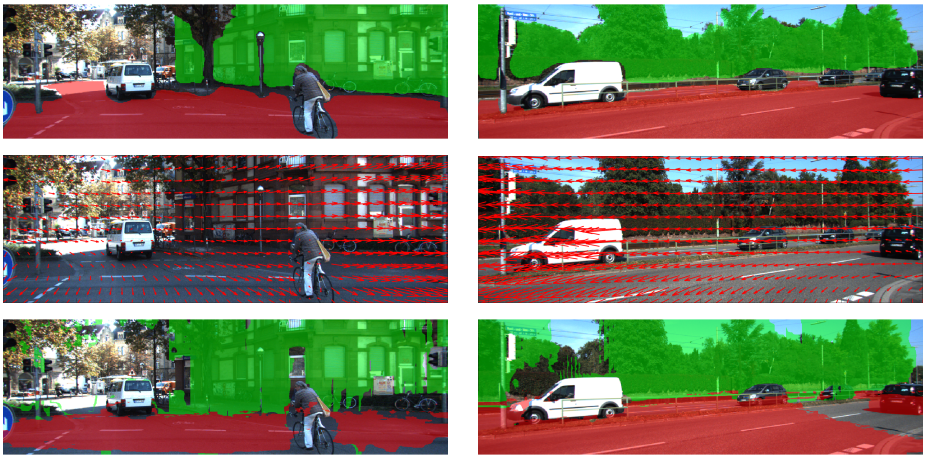
This chapter proposed to exploit optical flow to segment a scene into planar surfaces using only two successive images from one moving camera. The next chapter will focus on the whole sequence and will propose strategies to get benefit of the planar segmentation to improve motion estimation over all the frames of the sequence.



(a) (b) (c) (d)

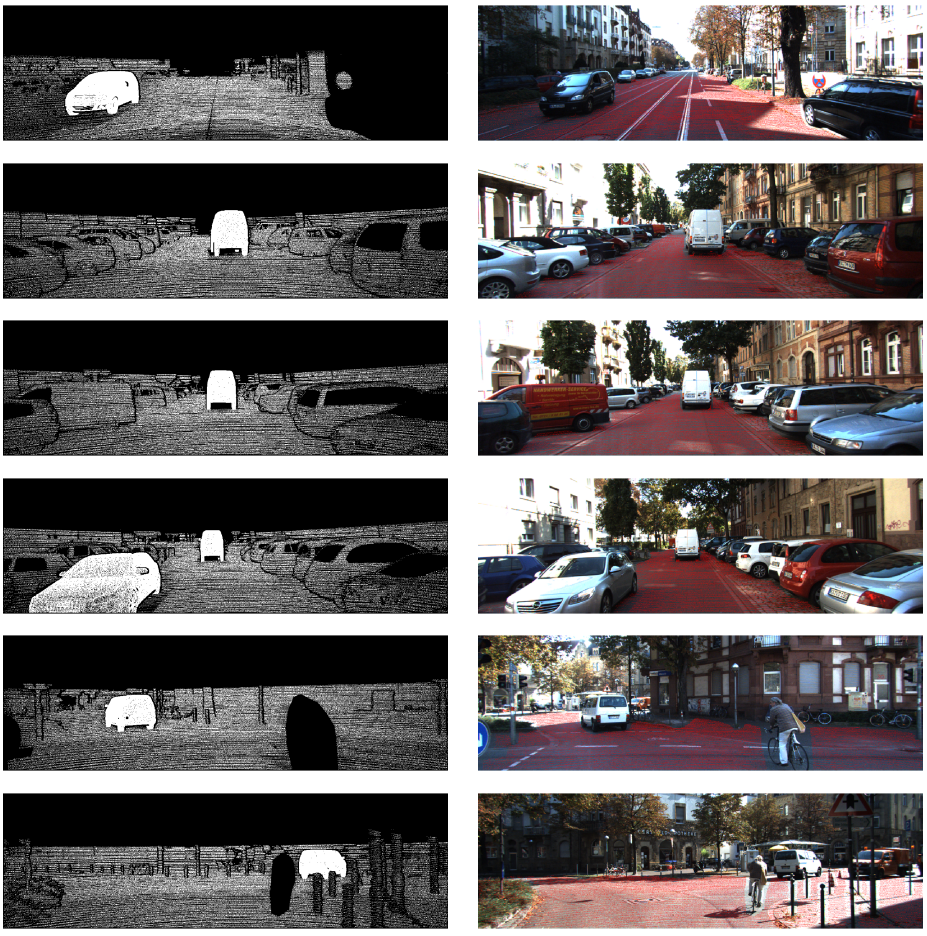
FIGURE 3.30: The **U**(a) and **V**(c) voting space for OF in figure 3.21. The found parabola on **V**is drawn in red (b). Two parabola are found on **U**(d) two side of voting space. No line is found on two voting spaces. The horizontal axis represents the u or v values, the vertical axis represents the H or W value for **V** or **U** respectively

FIGURE 3.31: Segmented image according to the detected parabola for c-velocity (top) and uv-velocity (bottom) on the estimated OF. The horizontal plane is drawn in red, the lateral plane is coloured in green.



(a) (b)

FIGURE 3.32: The ground truth segmentation of horizontal plane(red) and extended plane (green) of two sequences (called 1 and 2) on the first row. The OF (second row) and corresponding plane segmentation (third row) based on uv-velocity. The ex-horizontal and ex-lateral planes are colored in red, and green respectively



(a) (b)

FIGURE 3.33: The mask of available ground truth optical flow (a) and the ground truth of segmentation for horizontal plane (b) for 6 sequences indexed (1),(2),(3),(4),(5),(6) respectively from top to bottom

(a) (b)

FIGURE 3.34: The segmentation results from a parametric method (a) and our voting method (b).

Chapitre 4

From structure to motion

The previous chapter has proposed a method that exploits optical flow to exhibit the main planar structures of a scene. The relationship between optical flow and geometric information is simplified through uv-velocity and the planar segmentation is reduced to the detection of curves in the voting spaces. We have shown also in the previous chapter that the quality of the estimated optical flow has a direct impact on curve sharpness in the voting space. Inversely, the idea that comes naturally to mind is to exploit the structure detection provided by the uv-velocity to improve the quality of the optical flow. These round trips between optical flow estimation and planar structures detection are easily conceivable when one considers an entire sequence and not only two successive images. The process is made iteratively. It starts from the first pair of successive images, estimates optical flow, detects and characterizes the planar regions by using the uv-velocity. The arising information about the structure could be exploited as a prior for the estimation of the optical flow in the subsequent frame. Starting from these observations, this chapter presents all the necessary bricks for the implementation of a complete planar structures system of detection based on optical flow. In addition, as the concept of structure is intimately linked to that of the 3D motion, we also propose to present a path of integration of our approach in a complete visual odometry system and highlight the main added values compared to classical approaches that are based on a global homography estimation. Here, our idea is to estimate one homography per image plane, therefore in a more local way. This should filters out the outliers from the estimation and improve finally the global precision. This chapter studies the final block of our opportunistic system as the application emerged from the scene analyzes (figure 4.1)

This chapter is divided into two sections. Section 4.1 presents our work on iterative optical flow refinement using prior knowledge on the geometric structure. Then, section 4.2 deals with a complete visual odometry system that exploits geometry and plane detection.

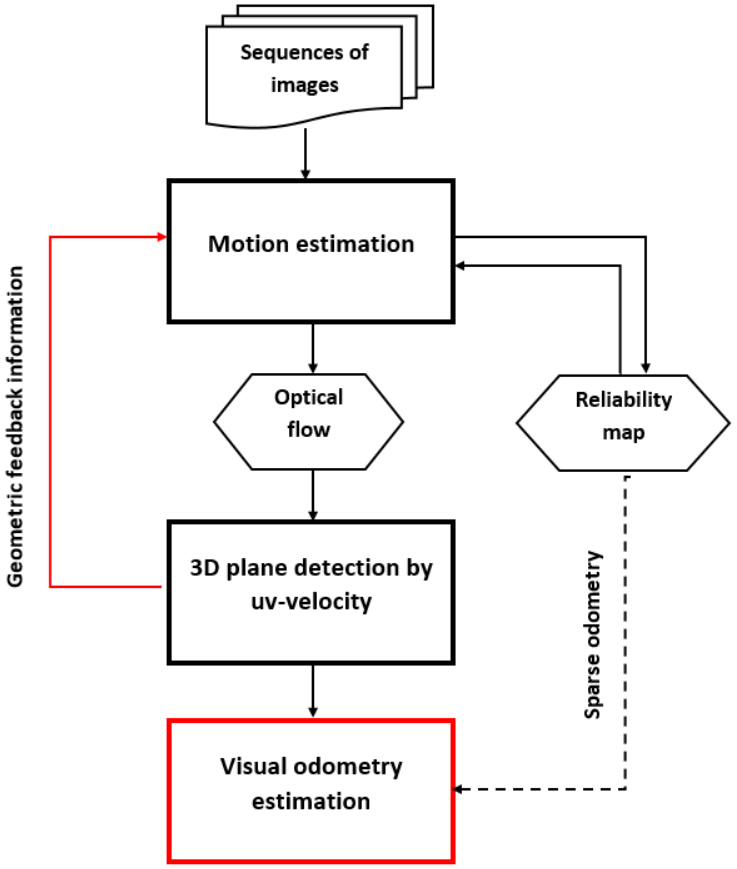


FIGURE 4.1: Developed block in this section, the feedback geometric information to improve optical flow and odometry estimation from the planar segmentation

4.1 Optical flow estimation using geometric information

Adding geometric information to optical flow estimation is studied in the late 20^{th} in the work of Horn and Berthold [HW88]. Instead of estimating optical flow as a translational vector with two unknown components $[u, v]^\top$, the authors attempt to directly estimate the six components of the camera motion $\mathbf{M} = [T_X, T_Y, T_Z, \Omega_X, \Omega_Y, \Omega_Z]$ like in equation (3.3). Then, the vector $\mathbf{M}$ is estimated by minimizing the luminosity constraint mentioned in equation (2.1). The estimation is possible when a few assumptions are made in order to reduce the number of unknown variables to be estimated, for example : the depth or the rotation is known or the movement is a pure rotation. A global parametric motion model, such as linear [Far00] or affine [SSB10] can be estimated using the whole image in order to get the dominant motion of the camera. However, this assumption does not hold when the scene contains different motions. In [SSB10], the author defines a global cost function to be minimized for simultaneous image segmentation and optical flow estimation, using a variety of affine models. More precisely, two levels of optimization have to be solved, one for segmentation and one for optical flow estimation, are interleaved, which requires heavy processing.

In a different way, *uv*-velocity allows to get the equation of the main planes. If we can determine the plane equation for each pixel, then a global energy function can be formulated to find the best setting for segmentation and deduce the optical flow. Unfortunately, *uv*-velocity can not assign a model to pixels which do not belong to a plane region. Therefore, the segmentation and the optical flow estimation have to be done separately. The experiments of Section 3.8 have shown that *uv*-velocity can segment planar surfaces quite well when optical flow is estimated correctly. However, optical flow estimates are prone to errors. Consequently, the resulting segmentation map can be imprecise, as illustrated by figure 4.2. As it can be seen, a part of the road is included in the lateral planes category. On the road, which does not contain enough texture, the segmentation is incomplete.

FIGURE 4.2: Example of segmentation result coming from the analysis of the *uv*-velocity computed from the proposed optical flow estimation.

Since the analysis of *uv*-velocity and the extraction of the parametric curves are based on a consensus search, the estimation of the parameters can be correct even if the optical flow map is incomplete or partly erroneous. However, after assigning a model to each pixel, the resulting segmentation will remain as incomplete or erroneous as the initial motion map. We face a sort of chicken and egg problem where we want to use the segmentation map to improve the precision of the next optical flow while a correct optical flow is required to get a precise segmentation.

Consequently, in order to separate the two problems, we first assume that the plane segmentation is known with a good precision, and that the optical flow models are well estimated for each of the regions. We propose a deviation of our optical flow estimation approach integrating the plane motion models in the estimation.

4.1.1 Optical flow estimation with planar constraint

Most optical flow estimation methods assume that the initial optical flow is equal to zero and estimate the optical flow iteratively by minimizing an energy function. In our case, it is considered that a partial planar segmentation is available together with the motion model of each region.

Our method estimates optical flow using the classical local KLT method as a first estimation, followed by a propagation process that refines this rough estimation. The energy

function of KLT of section 2.2.2 is recalled as :

$$\underset{\mathbf{u}(\mathbf{x})}{\operatorname{argmin}} E = \sum_{\mathbf{k}\in N(\mathbf{x})} \left[I_t(\mathbf{k}) + \nabla I^\top(\mathbf{k})\mathbf{u}(\mathbf{x})\right]^2 \tag{4.1}$$

Suppose that M planes have been detected and characterized. Each of them has its own optical flow model noted $\widehat{\mathbf{u}}_m, m \in [1, 2, .., M]$ determined by uv-velocity. These planes are well identified in the image. Instead of using a label $m \in [1, 2, ..., M]$, we introduce another notation that will be easier to include in the optimization function : the variable $s_m(\mathbf{x}), m \in [1, 2, .., M]$ represents the ownership of pixels $\mathbf{x}$ to the m^{th} plane. A pixel can belong to only one plane, therefore the sum of the s_m at this point is maximum 1. All pixels are not included in a planar region, so the sum can be null for this point. This leads to the following condition :

$$\sum_{m\in M} s_m(\mathbf{x}) = 1 \; or \; \sum_{m\in M} s_m(\mathbf{x}) = 0 \tag{4.2}$$

We then modify the energy function (4.1) to take into account geometric information in the estimation :

$$\underset{\mathbf{u}(\mathbf{x})}{\operatorname{argmin}} E = \sum_{\mathbf{k}\in N(\mathbf{x})} \left[I_t(\mathbf{k}) + \nabla I^\top(\mathbf{k})\mathbf{u}(\mathbf{x})\right]^2 + \lambda \sum_{m\in M} s_m(\mathbf{x})[\mathbf{u}(\mathbf{x}) - \widehat{\mathbf{u}}_m(\mathbf{x})]^{\mathbf{2}} \tag{4.3}$$

λ is a weighting coefficient that gives more or less importance to the geometric motion model with respect to brightness change constraint. The minimization of this cost function should improve the optical flow estimation in pixels that are assigned to the correct motion model. In some cases however, false segmentation should occur because motion estimation is constrained to follow a false plane model.

After adding this new constraint, the rest of the algorithm remains unchanged (see Chapter 2).

Numerical solution. The cost function (4.3) is minimized in a slightly different way as the one analyzed in Chapter 2. The solution is found by linearization and partial derivation. The motion vector $\mathbf{u}(\mathbf{x})$ is expressed as the sum of the previous estimate $\mathbf{u}_0(\mathbf{x})$ and a step $\delta\mathbf{u}(\mathbf{x})$ such that $\mathbf{u}(\mathbf{x}) = \mathbf{u}_0(\mathbf{x}) + \delta\mathbf{u}(\mathbf{x})$. The final optical flow values, that is the solution of the minimization process, is reached after several iterations.

$$E(\delta\mathbf{u}(\mathbf{x})) = \sum_{\mathbf{k}\in N(\mathbf{x})} \left[I_t(\mathbf{k}+\mathbf{u}_0(\mathbf{x}))+\nabla I^\top(\mathbf{k}+\mathbf{u}_0(\mathbf{x}))\delta\mathbf{u}(\mathbf{x})\right]^2+\lambda \sum_{m\in M} s_m(\mathbf{x})\left[\mathbf{u}_0(\mathbf{x})+\delta\mathbf{u}(\mathbf{x})-\widehat{\mathbf{u}}_m(\mathbf{x})\right]^2 \tag{4.4}$$

The $\mathbf{u}_0(\mathbf{x}) = \begin{bmatrix} u_0 & v_0 \end{bmatrix}^\top$ is the optical flow estimated at previous iteration (or the initialized value at first iteration) the residual value $\delta\mathbf{u}$ is the variable to be computed in this

case. Calling $\mathbf{k}_0 = \mathbf{k} + \mathbf{u}_0(\mathbf{x})$, the expression (4.4) is simplified in the following way :

$$E(\delta\mathbf{u}(\mathbf{x})) = \sum_{\mathbf{k}\in N(\mathbf{x})} \big[I_t(\mathbf{k}_0) + \nabla I^\top(\mathbf{k}_0)\delta\mathbf{u}(\mathbf{x})\big]^2 + \lambda \sum_{m\in M} s_m(\mathbf{x})\big[\mathbf{u}_0(\mathbf{x}) + \delta\mathbf{u}(\mathbf{x}) - \widehat{\mathbf{u}}_m(\mathbf{x})\big]^2 \tag{4.5}$$

Then, we differentiate the expression (4.5) according to $\delta\mathbf{u}(\mathbf{x})$:

$$\sum_{\mathbf{k}\in N(\mathbf{x})} 2\big[\nabla I^\top(\mathbf{k}_0)\delta\mathbf{u}(\mathbf{x})) + I_t(\mathbf{k}_0)\big]\nabla I(\mathbf{k}_0) + \lambda \sum_{m\in M} 2s_m(\mathbf{x})\big[\mathbf{u}_0(\mathbf{x}) + \delta\mathbf{u}(\mathbf{x}) - \widehat{\mathbf{u}}_m(\mathbf{x})\big] \tag{4.6}$$

Since the solution of our cost function is the motion value that sets (4.6) to zero, the following equation has to be solved :

$$\sum_{\mathbf{k}\in N(\mathbf{x})} \big[\nabla I(\mathbf{k}_0)\nabla I^\top(\mathbf{k}_0)\delta\mathbf{u}(\mathbf{x})) + I_t(\mathbf{k}_0)\nabla I(\mathbf{k}_0)\big] + \lambda \sum_{m\in M} s_m(\mathbf{x})\big[\mathbf{u}_0(\mathbf{x}) + \delta\mathbf{u}(\mathbf{x}) - \widehat{\mathbf{u}}_m(\mathbf{x})\big] = 0 \tag{4.7}$$

which can be written differently as :

$$\begin{aligned} &\sum_{\mathbf{k}\in N(\mathbf{x})} \big[\nabla I(\mathbf{k}_0)\nabla I^\top(\mathbf{k}_0)\delta\mathbf{u}(\mathbf{x}))\big] + \lambda \sum_{m\in M} s_m(\mathbf{x})\delta\mathbf{u}(\mathbf{x}) = \\ &- \sum_{\mathbf{k}\in N(\mathbf{x})} \big[I_t(\mathbf{k}_0)\nabla I(\mathbf{k}_0)\big] - \lambda \sum_{m\in M} s_m(\mathbf{x})\big[\mathbf{u}_0(\mathbf{x}) - \widehat{\mathbf{u}}_m(\mathbf{x})\big] \end{aligned} \tag{4.8}$$

The equation (4.8) can be expressed as a linear relation :

$$\mathbf{A}\delta\mathbf{u}(\mathbf{x}) = \mathbf{B}, \tag{4.9}$$

where $\mathbf{A} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} b_1 \\ b2 \end{bmatrix}$. Then the $\delta\mathbf{u}(\mathbf{x})$ is found by dividing $\mathbf{B}$ by $\mathbf{A}$.

Re-calling $\nabla I = \begin{bmatrix} I_x \\ I_y \end{bmatrix}$ and $\widehat{\mathbf{u}}_m = \begin{bmatrix} \widehat{u}_m \\ \widehat{v}_m \end{bmatrix}$, the coefficients of $\mathbf{A}$ and $\mathbf{B}$ can be computed as :

$$
\begin{aligned}
a_1 &= \sum_{\mathbf{k}\in N(\mathbf{x})} \mathbf{I}_x^2(\mathbf{k}_0) + \lambda \sum_{m\in M} s_m(\mathbf{x})\widehat{u}_m(\mathbf{x}) \\
a_2 = a_3 &= \sum_{\mathbf{x}\in N(\mathbf{x})} \mathbf{I}_x(\mathbf{k}_0)\mathbf{I}_y(\mathbf{k}_0) \\
a_4 &= \sum_{\mathbf{x}\in N(\mathbf{x})} \mathbf{I}_y^2(\mathbf{k}_0) + \lambda \sum_{m\in M} s_m\widehat{v}_m(\mathbf{x}) \\
b1 &= -\sum_{\mathbf{x}\in N(\mathbf{x})} \mathbf{I}_t(\mathbf{k}_0)\mathbf{I}_x(\mathbf{k}_0) - \lambda \sum_{m\in M} s_m(\mathbf{x})\big[u_0(\mathbf{x}) - \widehat{u}_m(\mathbf{x})\big] \\
b2 &= -\sum_{\mathbf{x}\in N(\mathbf{x})} \mathbf{I}_t(\mathbf{k}_0)\mathbf{I}_y(\mathbf{k}_0) - \lambda \sum_{m\in M} s_m(\mathbf{x})\big[v_0(\mathbf{x}) - \widehat{v}_m(\mathbf{x})\big]
\end{aligned}
\tag{4.10}
$$

Now that the cost function has been defined, as well as its numerical solving, it can be applied to a concrete scenario.

4.1.2 Experiments

In these experiments, we use again some selected sequences from the KITTI dataset for optical flow evaluation in which the camera motion is a translation. We address the problem of estimating the optical flow on the road, which is known to be a difficult problem because of the lack of texture. Our objective is to improve the optical flow estimation on the road region by using our new proposed energy function.

We first acquire the road segmentation from the ground truth made manually from the experiments on segmentation in chapter 3. Figure 4.3 shows the ground truth of horizontal plane on three studied sequences.

Then, the motion model of the road can be estimated by the parameter K estimated in the $\mathbf{V}$ voting space created by optical flow (in the case of pure translation) :

$$
\begin{aligned}
u_1 &= K(x - x_{FOE})|y| \\
v_1 &= K(y - y_{FOE})|y|
\end{aligned}
\tag{4.11}
$$

This optical flow can come from the ground truth or from our estimation method. To stick with the purpose of this experiment, we use the model provided by the estimated optical flow explained in chapter 2. Let us denote this method RF (for Reliable Flow). Thanks to the consistency to noise offered by uv-velocity, the quality of the model remains intact despite of the imprecision of the optical flow. The table 4.1 compares the parameter K obtained from two types of optical flow and proves that the parameter found from estimated optical flow does not vary much from one found by the *true* optical flow.

FIGURE 4.3: The mask of the road segmentation on 3 studied sequences (enumerated I, II and III from top to bottom)

Sequence	K_{GT}	K
(I)	$6.084 * 10^{-4}$	$6.099 * 10^{-4}$
(II)	$6.9138 * 10^{-4}$	$6.9319 * 10^{-4}$
(III)	$8.2758 * 10^{-4}$	$8.011 * 10^{-4}$

TABLE 4.1: Parameter K determined using a ground truth optical flow (K_{GT}) and using an estimated optical flow (K) for 3 sequences I, II and III selected from the KITTI database

Next, with the segmentation of the road, the motion model of the road is then included in the cost function with the help of the mask s_1 corresponding to the ground truth of segmentation (see figure 4.3). In this context, there is only one region to be improved $M = 1$.

After that, to start the refinement of the optical flow estimation, we initialize the optical flow ($\mathbf{u}_0 = [u_0, v_0]^\top$) with the optical flow estimated in the previous frame. A pyramidal structure is used for the estimation, as in chapter 2. However, instead of using five pyramidal levels, only the two first levels are considered because the estimation should be improved by using the knowledge on the previous motion model of the road. With $\lambda = 10$, the refined optical flow (denoted Model-RF) is estimated and shown together with the original optical flow RF in figure 4.5.

Finally, in order to evaluate the method, we will compare the estimated optical flow with the ground truth optical flow provided by the KITTI dataset. Since the ground truth is sparse, we only compute the error on the available pixels, represented in white in figure 4.4.

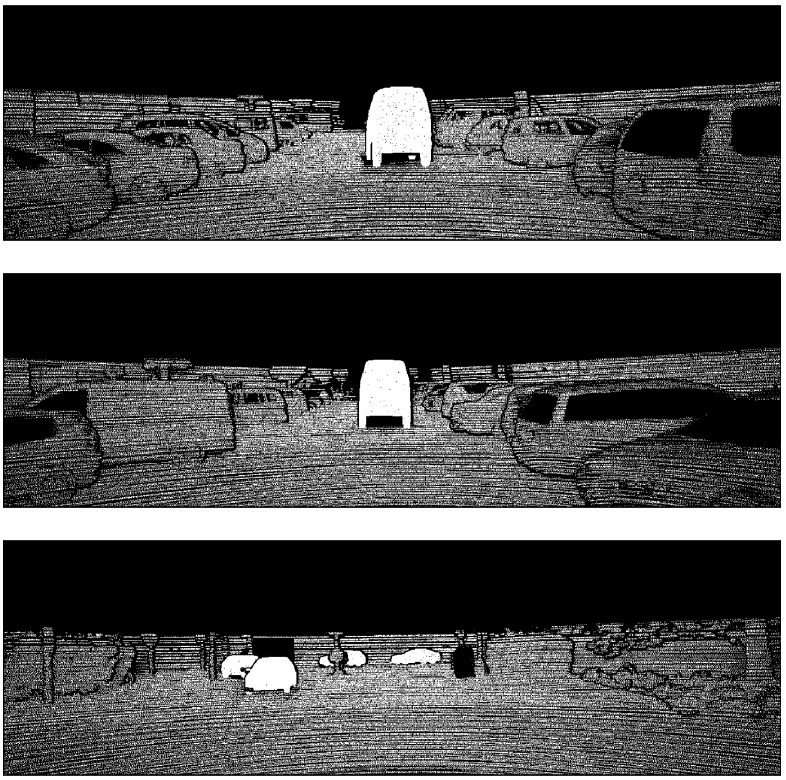


FIGURE 4.4: The mask of available ground truth optical flow on 3 example sequences I,II and III from top to bottom

According to table 4.2 which compares the error of two optical flow approaches, the precision of the optical flow is increased when integrating the road model on the estimation. While the angular error does not change much, the end point error shows a noticeable improvement.

Sequence	I		II		III	
	AAE	AEPE	AAE	AEPE	AAE	AEPE
Classical RF	4.92	4.22	5.65	6.31	19.83	11.33
Model RF	4.85	3.40	5.62	5.57	4.68	3.13

TABLE 4.2: Comparison the error between classical RF and new model RF

Figure 4.6 compares the segmentation maps obtained with each optical flow method, that is the initial RF method (without any prior on the structure) and the Model-RF, which includes the motion model as a constraint. It can be seen that the results on all the three selected sequences are visually more satisfactory on the road region.

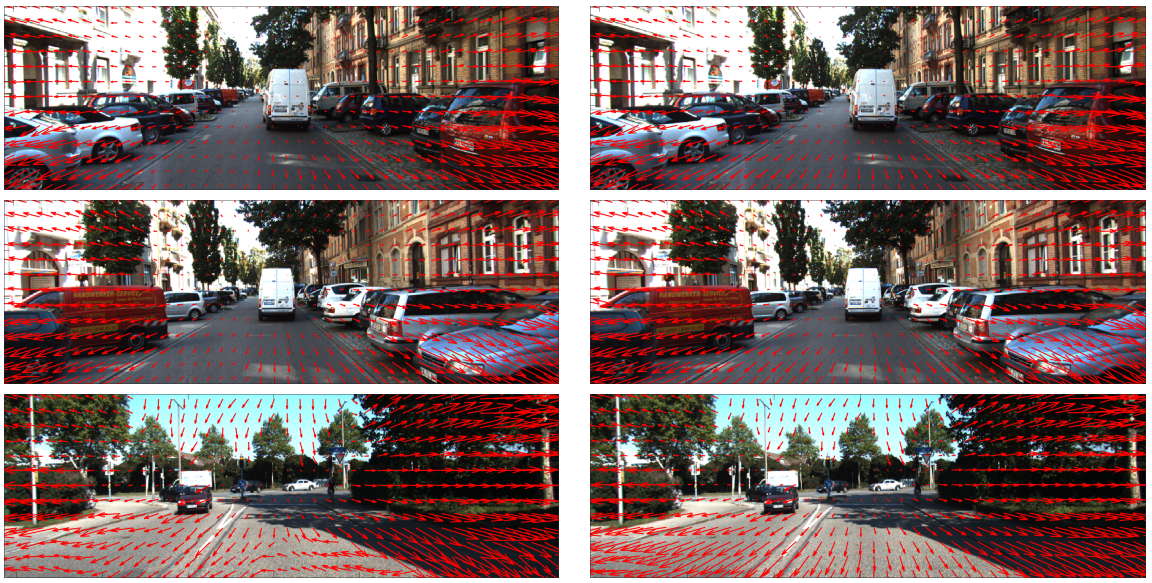


FIGURE 4.5: Illustration of optical flow of the classical RF method(a) and Model RF method(b) for 3 selected image sequences of the KITTI data base

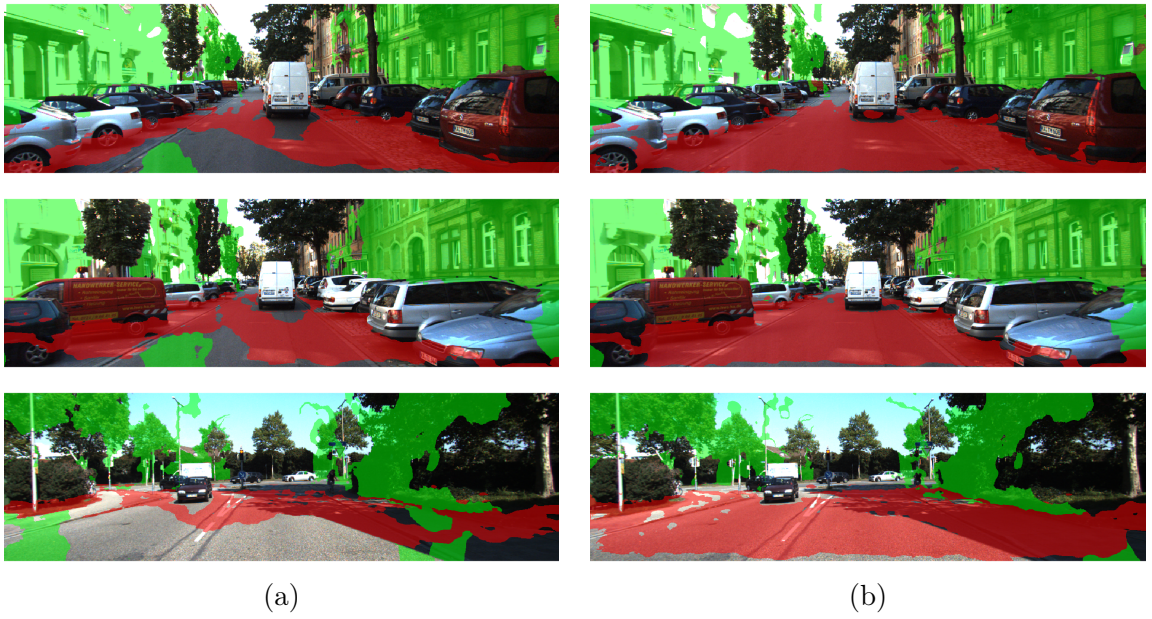


FIGURE 4.6: The plane segmentation from uv-velocity created by RF optical flow(a) and Model RF optical flow(b)

As demonstrated by table 4.3, the accuracy of the segmentation is improved by including the motion model into the optical flow cost function.

With the new energy function, both optical flow and plane segmentation are improved. Here also, these results illustrate the tight relation between the precision of the optical flow and the quality of plane segmentation.

Sequence	RF	Model RF
(I)	0.86	0.93
(II)	0.88	0.93
(III)	0.72	0.91

TABLE 4.3: The accuracy of the road segmentation between two optical flow

4.1.3 Discussions

This experiment has focused on the estimation of the optical flow on the road region, which is reputed to be challenging because of the lack of texture. According to the experiments made in this section, the proposed cost function has produced more precise optical flow values, by exploiting the motion model of the planar surface provided by uv-velocity. The other planes of the scene, like lateral buildings, can be found more easily because they generally contain enough texture or edges.

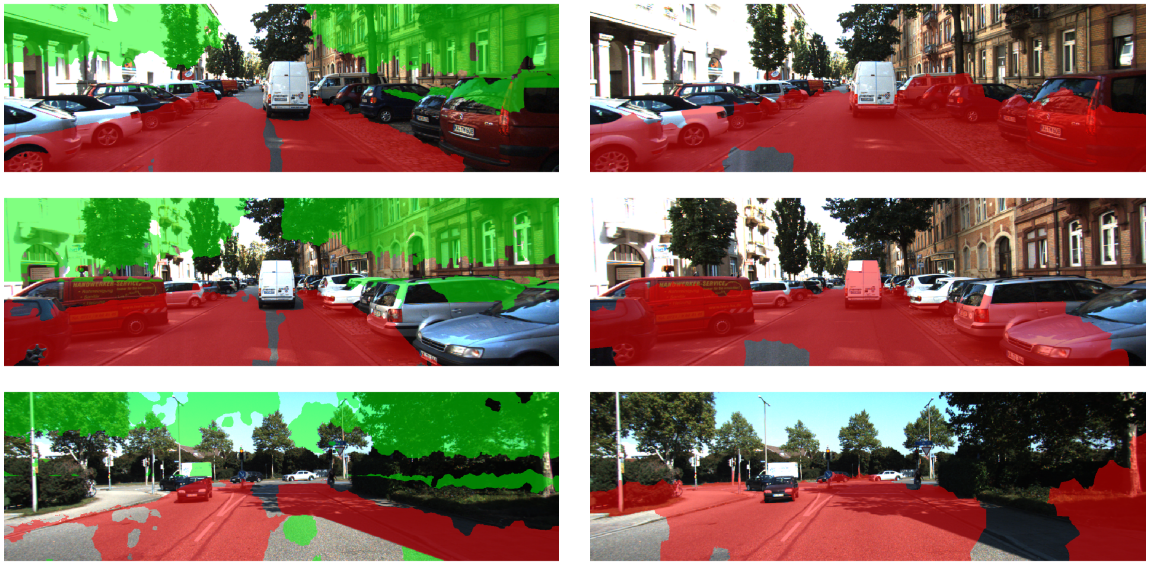
The optical flow method proposed in chapter 2 assumes that the motion values are constant locally in a limited neighborhood, for points of the same object. As a perspective work, the two main steps of the method, that is the auto-evaluation of the reliability and the propagation process could be modified by using a quadratic or linear relationship between a pixel and its neighbor. This relationship could be parameterized depending on the type of planar surface.

In this experiment, the problem of optical flow estimation has been addressed by using the *true* segmentation for initialization (the map s_m). Such information is not available naturally from a pair of images. At this stage of our work, the pre-segmentation provided by the analysis of uv-velocity (see figure 4.6(a)) is not precise enough to correct the optical flow. The segmentation should be improved by using additional information like appearance. An attempt is proposed in figure 4.7, where we try to recover the good segmentation from the output of uv-velocity by using the probabilistic Naive Bayes model and Graph Cut algorithm. Unfortunately, this is not conclusive enough, as explained hereafter.

• The pre-segmentation provided by the uv-velocity can be exploited as seeds to build a statistical model (through a Naive Bayesian model). The results are displayed on figure 4.7(a). Here, we use the orientation of optical flow to model the planes. Each pixel has then two probabilities : belonging to the road and not belonging to the road. The decision is done by finding the dominant probability. The red region is decided as the road, the blue region is decided as "non-road" and the other pixels are not decisive. It can be seen that pixels which are falsely labelled can interfere negatively in the minimization of the cost function, by adding a incorrect constraint. Therefore, the segmentation results are not improved, there is over-segmentation and missing pixels.

• An another approach to refine the segmentation is to use the graph cut [BK01], where the road is considered as the foreground and the rest as background. The result seems more stable than the Bayes probabilistic model (figure 4.7(b)). The red region is considered as road and the rest is "non-road". However, there are still some important segmentation defects, for example holes on the road. The graph cut tends to over-segment the cars on the two sides because of the imprecision of the seminal segmentation provided by *uv*-velocity. There are some improvements to do in order to better associate these regions to the background. Thus, the segmentation refinement after *uv*-velocity is still an open issue.

The next section explores a second application of the *uv*-velocity, where the segmentation into planar regions is used as an input for a visual odometry method.



(a) (b)

FIGURE 4.7: The segmentation refinement from the pre-segmentation figure 4.6(b) by using the Naïve Bayes (a) and Graph Cut (b)

4.2 Visual odometry using homography estimation from a fast segmentation

Visual odometry is the task of determining the position and orientation of a camera by exploiting the apparent changes and motion through a sequence of images. The images can be captured from single or stereoscopic sensors. There are two main methodologies to tackle the problem, namely feature-based methods and direct methods.

Feature-based methods consist in extracting keypoints in each image and pairing them between successive frames. The resulting set of correspondences is then used to estimate

a global 3D transform and find the geometric 3D properties. In [ÖVBS17], the author resumes a large number of methodologies to recover 3D information with feature matching between frames. The results are maintained good with a forward estimation and a backward optimization called "bundle adjustment". Normally, to be able to estimate the absolute translation motion or real size of environment, we must know either the depth of pixels or the true movement of the camera. Otherwise, the recovery can only estimate the size and the translation movement up to a scale factor. Noticeably, in [NNB04], the author mentions that the exact visual odometry can be recovered with mono-vision rigs which is very interesting for our approach but it is not mentioned how the scale factor is obtained. These approaches have the advantage to be fast to compute and they allow to easily remove the outliers of the estimation. However, the reconstructed scene is sparse and fragmented in keypoints of the scene.

Direct methods model directly the displacement of points in image by expressing camera motion. The movement is estimated and camera odometry is recovered by minimizing a photometric error. This methodology requires to know the depth for each point [FPS14] or some special hypotheses like pure rotational/translation movements [HW88, IA00]. The calculation is heavier than for feature-based methods but the 3D reconstruction is made in a dense way. Some other research works combine other sensors like IMU [UESC16] to improve the precision of the estimation.

Considering that the calibration matrix is known, we propose in this section a new technique to estimate odometry from the output of uv-velocity : a rough plane segmentation.

The motion of a planar surface can be seen as a homography transform which embeds the information about the odometry. Taking the advantage of having the segmented planes on the scene, we compute one step further in our opportunistic system the homography matrix to yield the odometry information of camera. We experiment in this section the capacity to recover the camera orientation and translation direction from any plane segmentation. In section 4.2.1, we first recall some basic notations used in two-view geometry. Section 4.2.2 details our proposed developments to integrate the segmented regions into a complete visual odometry estimation by homography matrix. Then, experiments are conducted (only forward-estimation) in section 4.2.3 to assess the effectiveness of this approach in two aspects :

• Comparing the homography matrix estimation time with and without uv-velocity support

• Comparing the performance of our approaches versus the traditional feature based method where the essential matrix is formed by feature matching and decomposed to yield the odometry.

4.2.1 Overview of protective geometry for single and two-views

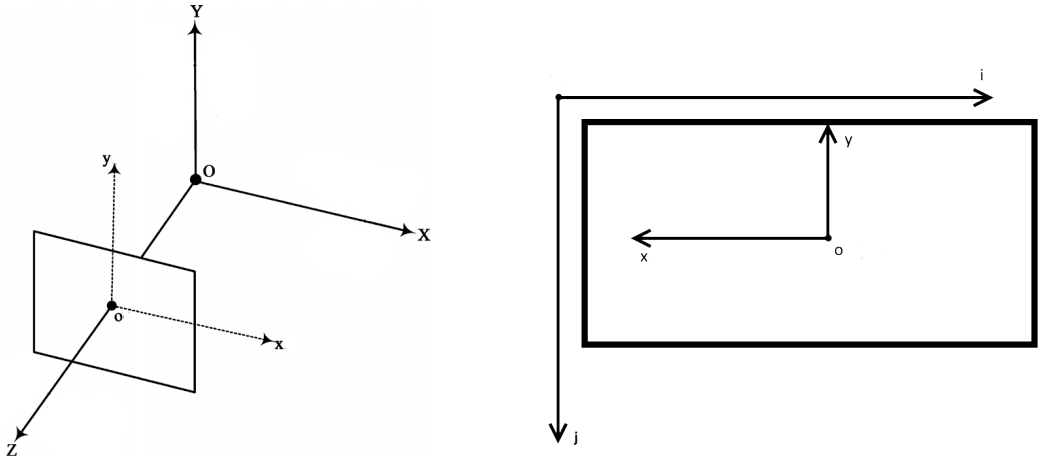


FIGURE 4.8: Coordinates of the system

Single view geometry : We consider the coordinates system of Fig.4.8 (a), where **OXYZ** is the world coordinate system, *oxy* is the image coordinates system and i, j are the image pixel coordinates. A point $P(X, Y, Z)$ in the real scene will have its pixel coordinates on image plane $p(i, j)$ with camera having a focal length f :

$$\mathbf{i} = \begin{bmatrix} i \\ j \\ 1 \end{bmatrix} = \frac{1}{f} \begin{bmatrix} -fp_x & 0 & i_c \\ 0 & -fp_y & j_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ f \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} -fp_x & 0 & i_c \\ 0 & -fp_y & j_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \tag{4.12}$$

where p_x, p_y are the pixel dimensions on X and Y directions and i_c, j_c are the pixel coordinates of image center o. From the equation (4.12), the camera calibration $\mathbf{C}$ is defined as the matrix which allows to project the 3D points in camera coordinates into homogeneous image pixel coordinates. It has 3 degrees of freedom (p_x, p_y and f) :

$$\begin{bmatrix} i \\ j \\ 1 \end{bmatrix} = \frac{1}{Z}\mathbf{C} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \propto \mathbf{C} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{4.13}$$

Two-view geometry : This part studies the relation between pixel coordinates of a pair of identical cameras. The coordinates systems are $O_1X_1Y_1Z_1$ and $O_2X_2Y_2Z_2$ like in figure 4.9. Supposing that the world coordinates system is the coordinate system of the first camera, such that $O_1X_1Y_1Z_1$ is equivalent to $OXYZ$. A point $\mathbf{P} = [X, Y, Z]^\top$ in world coordinates has coordinates in second view ${}^2\mathbf{P} = [X_2, Y_2, Z_2]^\top$. Calling ${}^2\mathbf{R}_1$ is the rotational matrix from the first coordinates to the second ones and ${}^1\mathbf{T}_2$ is the coordinates of second view on the first camera coordinates (world coordinate), then we

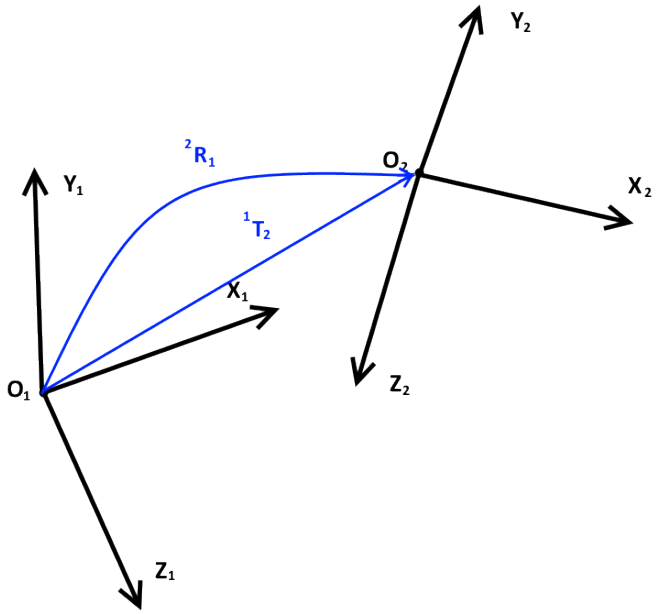


FIGURE 4.9: Illustration of two camera coordinates systems.

have the relation :

$$ {}^2\mathbf{P} = {}^2\mathbf{R}_1(\mathbf{P} - {}^1\mathbf{T}_2) = {}^2\mathbf{R}_1\mathbf{P} - {}^2\mathbf{R}_1{}^1\mathbf{T}_2 = \begin{bmatrix} {}^2\mathbf{R}_1 & {}^2\mathbf{T}_1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \tag{4.14} $$

where ${}^2\mathbf{T}_1$ is the coordinates of world (1st camera) projected on the 2nd camera coordinate system. The 3×4 matrix $\mathbf{Q}_{12} = \begin{bmatrix} {}^2\mathbf{R}_1 & {}^2\mathbf{T}_1 \end{bmatrix}$ is called the extrinsic matrix of camera. It reflects the external parameter of camera : orientation and position on to the world coordinates. It has 6 degrees of freedom (3 for rotation and 3 for translation). This matrix is used to transform homogeneous coordinates from one view to another view.

By combining the two equations (4.13) and (4.14), we have a complete projection matrix ($\mathbf{Pr}$) from the world point to camera pixel coordinate by $\mathbf{Pr}_{12} = \mathbf{CQ}_{12}$:

$$ \begin{bmatrix} {}^2i \\ {}^2j \\ 1 \end{bmatrix} \propto \mathbf{C}\ {}^2\mathbf{P} = \mathbf{CQ}_{12} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{Pr}_{12} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{4.15} $$

Using all these basic notations of multi-view projection, the next section aims to present the approach that we carried out to find odometry after planar segmentation by *uv*-velocity.

4.2.2 Integration of the segmentation map into odometry estimation

To estimate the odometry, our approach is composed of three main consecutive steps starting from two successive images of the input sequence. The first step consists in estimating the optical flow and segmenting the image into planar regions, as explained in Chapter 3. Even if the quality of the segmentation is not perfect, we will see that it allows to reduce the searching area required for classical homography estimation.

One homography matrix of one plane is enough to estimate the odometry. Different regions yield different plane's vector normal but the rotation matrix and direction of translation movement stay the same. So, in the first step, any region can be used for the next steps. Indeed, since the uv-velocity extracts the main planar surfaces of the scene through a voting strategy, it is assumed that each of the detected region contains enough points for homography estimation.

The next two steps consist in finding and decomposing the homography matrix.

4.2.2.1 Homography estimation

If a group of points belong to a planar surface, assuming that plane is aligned to the world plane XOY (Z=0 for all pixels), we can deduce the relation between the homogeneous pixel coordinates between two views :

$$ {}^2\mathbf{i} = \begin{bmatrix} {}^2i \\ {}^2j \\ 1 \end{bmatrix} \propto \mathbf{C}\mathbf{Q}_{02} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} \propto \mathbf{C}\mathbf{H}_{02}\mathbf{H}_{01}^{-1}\mathbf{C}^{-1} \begin{bmatrix} {}^1i \\ {}^1j \\ 1 \end{bmatrix} = \mathbf{C}\mathbf{H}_{12}\mathbf{C}^{-1} \begin{bmatrix} {}^1i \\ {}^1j \\ 1 \end{bmatrix} = \mathbf{G}_{12}\ {}^1\mathbf{i} \quad (4.16) $$

where $\mathbf{H}_{ab}$ and $\mathbf{G}_{ab}$ is the homography matrix and projective homography matrix respectively, both having size of 3×3, from a^{th}-view to b^{th}-view and the index $a, b \in \{0, 1, 2\}$. The 0^{th} index represents the world coordinates.

For each matching pair $\{{}^1\mathbf{i}, {}^2\mathbf{i}\}$, only the matrix $\mathbf{G}$ can be estimated and two equations can be deducted :

$$ \begin{aligned} -g_{11}{}^1i - g_{12}{}^1j - g_{13} +^2 i(g_{31}{}^1i + g_{32}{}^1j + g_{33}) = 0 \\ -g_{21}{}^1i - g_{22}{}^1j - g_{23} +^2 j(g_{31}{}^1i + g_{32}{}^1j + g_{33}) = 0 \end{aligned} \quad (4.17) $$

Since $\mathbf{G}$ is a matrix with 8 degrees of freedom (the values g_{ij} are up-to-scale), a 4-points matching provides enough equations to solve for $\mathbf{G}$:

$$\mathbf{A}_k\mathbf{g} = \begin{bmatrix} -{}^1i_1 & -{}^1j_1 & -1 & 0 & 0 & 0 & {}^2i_1{}^1i_1 & {}^2i_1{}^1j_1 & {}^2i_1 \\ 0 & 0 & 0 & -{}^1i_1 & -{}^1j_1 & -1 & {}^2j_1{}^1i_1 & {}^2j_1{}^1j_1 & {}^2j_1 \\ -{}^1i_2 & -{}^1j_2 & -1 & 0 & 0 & 0 & {}^2i_2{}^1i_2 & {}^2i_2{}^1j_2 & {}^2i_2 \\ 0 & 0 & 0 & -{}^1i_2 & -{}^1j_2 & -1 & {}^2j_2{}^1i_2 & {}^2j_2{}^1j_2 & {}^2j_2 \\ & & & & \vdots & & & & \\ -{}^1i_i & -{}^1j_i & -1 & 0 & 0 & 0 & {}^2i_i{}^1i_i & {}^2i_i{}^1j_i & {}^2i_i \\ 0 & 0 & 0 & -{}^1i_i & -{}^1j_i & -1 & {}^2j_i{}^1i_i & {}^2j_i{}^1j_i & {}^2j_i \end{bmatrix} \begin{bmatrix} g_{11} \\ g_{12} \\ g_{13} \\ g_{21} \\ g_{22} \\ g_{23} \\ g_{31} \\ g_{32} \\ g_{33} \end{bmatrix} = 0 \quad (4.18)$$

$\mathbf{A}_4$ is a 8×9, if the matching is correct and any 3 points are not colinear, $\mathbf{A}_4$ will have rank 8 and the 1D null space of A is the unique solution $\mathbf{g}$.

But in practice, choosing 4 coplanar points with precise matching on other frame is trivial. Furthermore, the solution $\mathbf{g}$ is sensible to the matching quality. Thus using only 4-points does not give robust result.

To leverage this problem, more matching points are considered. In matrix $\mathbf{A}_k$ with $k > 4$, if all of the matching are correct, then $\mathbf{A}$ still has rank 8 and the solution is unique. Otherwise, rank of $\mathbf{A}$ will be higher than 8, then $\mathbf{h}$ is the vector that minimize the term 4.19 subjected to $||\mathbf{g}|| = 1$

$$\mathbf{g}^\top \mathbf{A}_k^\top \mathbf{A}_k \mathbf{g} \quad (4.19)$$

Another way to estimate the homography matrix by using multiple points consist in minimizing the geometric distance (4.20) measuring the distance between the mapped points by homography and the matched ones :

$$\sum_k d({}^2\mathbf{i}_k, \mathbf{G}\ {}^1\mathbf{i}_k), \quad (4.20)$$

where $d(\mathbf{p}_1, \mathbf{p}_2)$ is the distance between two vectors $\mathbf{p}_1$ and $\mathbf{p}_2$. The two approaches translated by equations (4.19) and (4.20) are discussed in [HZ03]. Both solutions are prone to errors when there are outliers in the set of matching points, *i.e.* points that do not belong to the investigated plane.

To deal with the outliers, we use RANSAC to estimate matrix $\mathbf{G}$ with more precision. Calling S the set of N matching points on the scene. S_k is the k^{th} pair $\{{}^1\mathbf{i}_k, {}^2\mathbf{i}_k\}$. The RANSAC algorithm to find the matrix $\mathbf{H}$ is shown in table 4.4.

The τ_N,τ_d are parameters of RANSAC representing the required minimum number of inliers and maximum distance d_i of a pair S_i using the estimated $\mathbf{G}$.

INPUT : S,τ_N,τ_d
OUTPUT : G
Initialize Sin as an empty set, $num\_inliers = 0$
For n=1 :$max\_iteration$
  Choose 4 points randomly in which any 3 of them are not collinear and form A_4
  Find the $\mathbf{g}$ that satisfies $A_4\mathbf{g} = 0$ then form G from $\mathbf{g}$
  For each pair S_k is S :
    Compute ${}^2\mathbf{i'}_k = \mathbf{G}\ {}^1\mathbf{i}_k$
    Compute ${}^1\mathbf{i'}_k = \mathbf{G}^{-1}\ {}^2\mathbf{i}_k$
    Measure distance $d_k = d({}^2\mathbf{i'}_k, {}^2\mathbf{i}_k) + d({}^1\mathbf{i'}_k, {}^1\mathbf{i}_k)$
   If $d_k \leq \tau_d$
     Add S_k into Sin
     Increase $num\_inliers$ by 1
   End
  End
  If $num\_inliers \geq \tau_N$
    Re-estimate $\mathbf{g}$ with new set Sin by minimizing the term 4.19
    Break the loop
  End
End
Return $\mathbf{G}$ from $\mathbf{g}$

TABLE 4.4: RANSAC algorithm to estimate G

The flexibility of RANSAC allows us to use various error measures to improve the results. The first estimation uses the 4-points technique to quickly estimate $\mathbf{g}$. The validation step checks the geometric correspondence between points and the estimated model and eliminates the points that do not fit the model (the outliers). Then, when the number of inliers is high enough, all inliers are use to estimate $\mathbf{G}$ again. From the planes extracted previously, the RANSAC tries to find the best projective homography matrix. Theoretically, the algorithm does not take much time to compute since the segmentation produced by *uv*-velocity provides a set of points that are already located on a same planar surface. With a calibrated camera, *i.e.* when $\mathbf{C}$ is known, the Euclidean homography matrix $\mathbf{H}_{12}$ that projects a point from camera coordinate $OX_1Y_1Z_1$ to the second camera coordinate system $OX_2Y_2Z_2$ can be found using the following relation :

$$\mathbf{H}_{12} \propto \mathbf{C}^{-1}\mathbf{G}_{12}\mathbf{C} \tag{4.21}$$

4.2.2.2 Homography Decomposition

The homography matrix $\mathbf{H}$ can be decomposed to reveal the relative translation movements and orientation between two views as it contains information about the rotation $\mathbf{R}$, $\mathbf{t}$ and plane normal vector $\mathbf{n}$ ($\mathbf{t} = \frac{\mathbf{T}}{d}$, d is the distance of the plane to origin) :

$$\mathbf{H} = \mathbf{R} + \mathbf{t}\mathbf{n}^\top \tag{4.22}$$

If $\mathbf{i}_2 = \mathbf{C}\mathbf{H}_{12}\mathbf{C}^{-1}\mathbf{i}_1$, then :

$$\mathbf{H}_{12} = {}^2\mathbf{R}_1 + {}^2\mathbf{t}_1{}^1\mathbf{n}^\top \tag{4.23}$$

The classical approach used to decompose the homography matrix [FL88, ZH96] gives 8 solutions (4 pairs of distinguished ones and in each pair, the solution is opposite to each other). The number of solutions is then reduced to 2 (called **A** and **B**) under the development of Malis [MV07] where these two conditions are satisfied :

— The two views have to be on the same side of the plane.

— Camera is calibrated and the observed points always lie in front of the camera.

For the visual odometry estimation of a mobile robot, these conditions are always true. From the homography matrix H_{12} found in the previous step, by following the method described in [MV07]), we found the four possible sets of solutions after decomposing the homography matrix :

$$\begin{array}{l} R_A, \mathbf{t}_A, \mathbf{n}_A \\ R_A, -\mathbf{t}_A, -\mathbf{n}_A \\ R_B, \mathbf{t}_B, \mathbf{n}_B \\ R_B, -\mathbf{t}_B, -\mathbf{n}_B \end{array}$$

The decomposition of homography matrix always yields many solutions. To choose the good one, additional information about the movement or the plane is needed.

Following the steps presented in this section gives the camera odometry. The next section shows the experiments and the performance of estimated odometry based on *uv*-velocity.

4.2.3 Experiments

We know that from an homography matrix, the camera odometry can be estimated up to a scale factor for translation (direction of translation). In this section, we compare the performance of the homography estimation between our fast segmentation using *uv*-velocity and a traditional homography method. We suppose here that the matching has already been achieved, whatever its nature (dense or sparse). Then, we conduct a second set of experiments on the KITTI sequences dedicated to visual odometry to compare

the precision of the estimation between our approach and the feature based method based on decomposing essential matrix. These experiments show that beside the 3D planar segmentation, additional information like visual odometry can be deduced from uv-velocity.

4.2.4 Simulation

The experiment is conducted in a simulated framework. The calibration matrix $\mathbf{C}$ is known and is unique for the two views ($f = 0.04$, $p_x = p_y = 1e-5$, $i_o = j_o = 400.5$) :

$$\mathbf{C} = \begin{bmatrix} -400 & 0 & 400.5 \\ 0 & -400 & 400.5 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.24}$$

The first view (camera 1) is considered as the world coordinates :

$$\mathbf{Q}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.25}$$

The second view has a rotation $\Omega_Y = -0.15rad$ compared to the first view and its coordinates on first view are ${}^1\mathbf{T}_2 = [-0.4, 0.2, 0.1]^\top$. Hence, the projective matrix is :

$$\mathbf{Q}_2 = [{}^2\mathbf{R}_1, -{}^2\mathbf{R}_1 {}^1\mathbf{T}_2] = \begin{bmatrix} 0.988 & 0 & 0.149 & 0.38 \\ 0 & 1 & 0 & -0.2 \\ -0.149 & 0 & 0.988 & -0.159 \end{bmatrix} \tag{4.26}$$

Since the first view is the world coordinate system, the orientation of the second camera is expressed as :

$${}^1\mathbf{R}_2 = \begin{bmatrix} 0.988 & 0 & -0.149 \\ 0 & 1 & 0 \\ 0.149 & 0 & 0.988 \end{bmatrix} = \begin{bmatrix} \mathbf{r}_x & \mathbf{r}_y & \mathbf{r}_z \end{bmatrix} \tag{4.27}$$

Calling the estimated orientation and coordinates of the second camera respected to the first one respectively ${}^1\hat{\mathbf{R}}_2$ and ${}^1\hat{\mathbf{T}}_2$, the quality of estimation is measured using two indicators :

• The difference between the estimated normalized coordinates ${}^1\hat{\mathbf{t}}_2$, of the second camera with respect to the first one, and the true one ${}^1\mathbf{t}_2$:

$$ {}^1\mathbf{T}_2 = \begin{bmatrix} -0.4 \\ 0.2 \\ 0.1 \end{bmatrix} \propto \begin{bmatrix} -0.8729 \\ 0.4364 \\ 0.2182 \end{bmatrix} = {}^1\mathbf{t}_2 \tag{4.28} $$

Then the distance is measured as :

$$ d_t({}^1\mathbf{t}_2, {}^1\hat{\mathbf{t}}_2) = ||{}^1\mathbf{t}_2 - {}^1\hat{\mathbf{t}}_2||_2 \tag{4.29} $$

• The precision of the estimation of the 3 main axes compared to the true one in equation (4.27). Each column of ${}^1\hat{R}_2$ ($\begin{bmatrix} \hat{\mathbf{r}}_x & \hat{\mathbf{r}}_y & \hat{\mathbf{r}}_z \end{bmatrix}$) represents the coordinates of three 2nd camera's principal axes O_2X, O_2Y, O_2Z respectively with respect to the first camera :

$$ d_R({}^1\mathbf{R}_2, {}^1\hat{\mathbf{R}}_2) = ||\mathbf{r}_x - \hat{\mathbf{r}}_x||_2 + ||\mathbf{r}_y - \hat{\mathbf{r}}_y||_2 + ||\mathbf{r}_z - \hat{\mathbf{r}}_z||_2 \tag{4.30} $$

The cloud points of 3 planes are first created on the world coordinates. Then, we project each 3D points into image planes to create the first view and the second view by using $\mathbf{CQ_1}$ and $\mathbf{CQ_2}$. While projecting these points, we also keep track of the matching between points. Hence, the matching is totally exact in this experiments.

The first step aims to estimate the homography matrix $\mathbf{H}$. On the one hand, *uv*-velocity can be used to quickly find a plane in the scene by following the steps described in Section 4.2.2 with the RANSAC method detailed in table.4.4. On the other hand, we can apply a direct RANSAC method on all the pixels.

The $\mathbf{G}$ matrix is found within a plane on image. In both cases, we use $\tau_d = 10, \tau_N = 0.3N, max\_iteration = 100$ (N is the number of matched points).

Once the projective homography matrix $\mathbf{G}$ is found, the Euclidean homography matrix $\mathbf{H}$ is computed by using equation (4.21). The matrix $\mathbf{H}$ is then decomposed to reveal the ${}^1\hat{\mathbf{R}}_2$ and ${}^1\hat{\mathbf{t}}_2$. The table 4.5 gives a comparison between the two approaches.

	Homography with *uv*-velocity	Traditional homography
Execution time	6 seconds	13 seconds
d_R	**0.053**	**0.062**
d_t	**0.0664**	**0.0583**

TABLE 4.5: Performance comparison between two approaches in estimating camera odometry : Homography using *uv*-velocity and traditional homography.

It can be noted that the precision of the homography matrix between the two approaches is not too far from each other. However, with the planar information from *uv*-velocity, the execution time reduces significantly from 13 to 6 seconds. These experiments prove

that, under the same matching inputs, the plane segmentation with *uv*-velocity helps reducing a lot of calculation. This is due to the use of a selected region considered as a plane on the image instead of searching on the whole image to find such a region. Next experiments measure the precision of our approach using real image sequences.

4.2.5 Experiments on real sequence

In the experiments presented in this section, we use the KITTI dataset for odometry estimation. The data set contains 10 independent sequences with available ground truth, each data set has numerous instances of vehicles. Each instance has two images captured from a stereo-camera system. The camera calibration matrix is also provided in the document (see equation 4.31). As in the experiment above, homography estimation with *uv*-velocity helps improving the computation speed compared to the classic homography method. At the end, these two approaches give results that are close to each other. So in this part, we compare the precision of the visual odometry by decomposing the homography matrix supported by *uv*-velocity and by decomposing the essential matrix on 3 sequences of KITTI dataset that are shown in figure 4.10.

$$\mathbf{C} = \begin{bmatrix} 718.8560 & 0 & 607.1928 \\ 0 & 718.8560 & 185.2157 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.31}$$

Since we can not retrieve the true translation movement with only one camera, we focus on the precision of the translation direction and the orientation of the camera. On each sequence, we use 11 frames to compute 10 estimations of $\hat{\mathbf{t}}$ and $\hat{\mathbf{R}}$. The first frame of each sequence is called *zero*-frame which is considered as the origin of the trajectory. The i^{th} estimation uses the $(i-1)^{th}$-frame and the i^{st} frame to estimate relative odometry between these two frames. Since the ground truth gives us the absolute pose of the camera, *i.e* the relative pose between the i^{th} frame and *zero*-frame : ${}^0\mathbf{R}_{GT_i}$ and ${}^0\mathbf{T}_{GT_i}$, for each i^{th} estimation (${}^{i-1}\hat{\mathbf{R}}_i$ and ${}^{i-1}\hat{t}_i$), the i^{th} error measure is computed as :

$$\begin{aligned} d^i_{\mathbf{R}} &= d_R({}^0\mathbf{R}_{GT_i}, {}^0\hat{\mathbf{R}}_{i-1} * {}^{i-1}\hat{\mathbf{R}}_i) = d_R({}^0\mathbf{R}_{GT_i}, {}^0\hat{\mathbf{R}}_i) \\ d^i_t &= d_t({}^0\hat{t}_{i-1} + {}^{i-1}\hat{\mathbf{t}}_i, {}^0\mathbf{t}_{GT_{i-1}} + {}^{i-1}\mathbf{t}_{GT_i}) = d_t({}^0\hat{\mathbf{t}}_i, {}^0\mathbf{t}_{GT_i}), \end{aligned} \tag{4.32}$$

where ${}^{i-1}\mathbf{t}_{GT_i} = \frac{{}^0\mathbf{T}_{GT_i} - {}^0\mathbf{T}_{GT_{i-1}}}{||{}^0\mathbf{T}_{GT_i} - {}^0\mathbf{T}_{GT_{i-1}}||}$

Our approach is using the homography matrix computed from the points segmented by *uv*-velocity and considered as planar surfaces. The optical flow is computed by our method to feed the *uv*-velocity. Then from voting spaces of *uv*-velocity, we quickly found one plane on the scene (horizontal plane a priori) to estimate the homography matrix.

FIGURE 4.10: The three examined sequences called I,II and III respectively from top to bottom

The compared approach uses the essential matrix computed by a sparse matching of feature points on image. We use SURF features to match keypoints between two frames and use RANSAC to estimate the fundamental matrix (equation 4.33). The estimation and decomposition of the essential matrix is achieved by using the Machine Vision Toolbox [Cor07].

$$^{2}i^{\top}\mathbf{F}^{1}i = 0 \tag{4.33}$$

The essential matrix is then found :

$$\mathbf{E} = \mathbf{C}^{\top}\mathbf{F}\mathbf{C} \tag{4.34}$$

Decomposing the essential matrix allows to extract the rotational matrix and the translation direction, like the decomposition of the homography matrix. Supposing the direction of the translation movement is known, we can determine the correct solution among the results given by the decomposition. This experiment focuses on the precision of the visual odometry. Table 4.6 recaps the results concerning the average error measures of both camera orientation and translation direction. The errors produced by uv-velocity are displayed together with the errors produced by the method based on the essential matrix, considering the estimations made on 10 consecutive frames. Figure 4.11 also shows the evolution of these errors.

The results of table 4.6 and figure 4.11 show that the homography decomposition yields

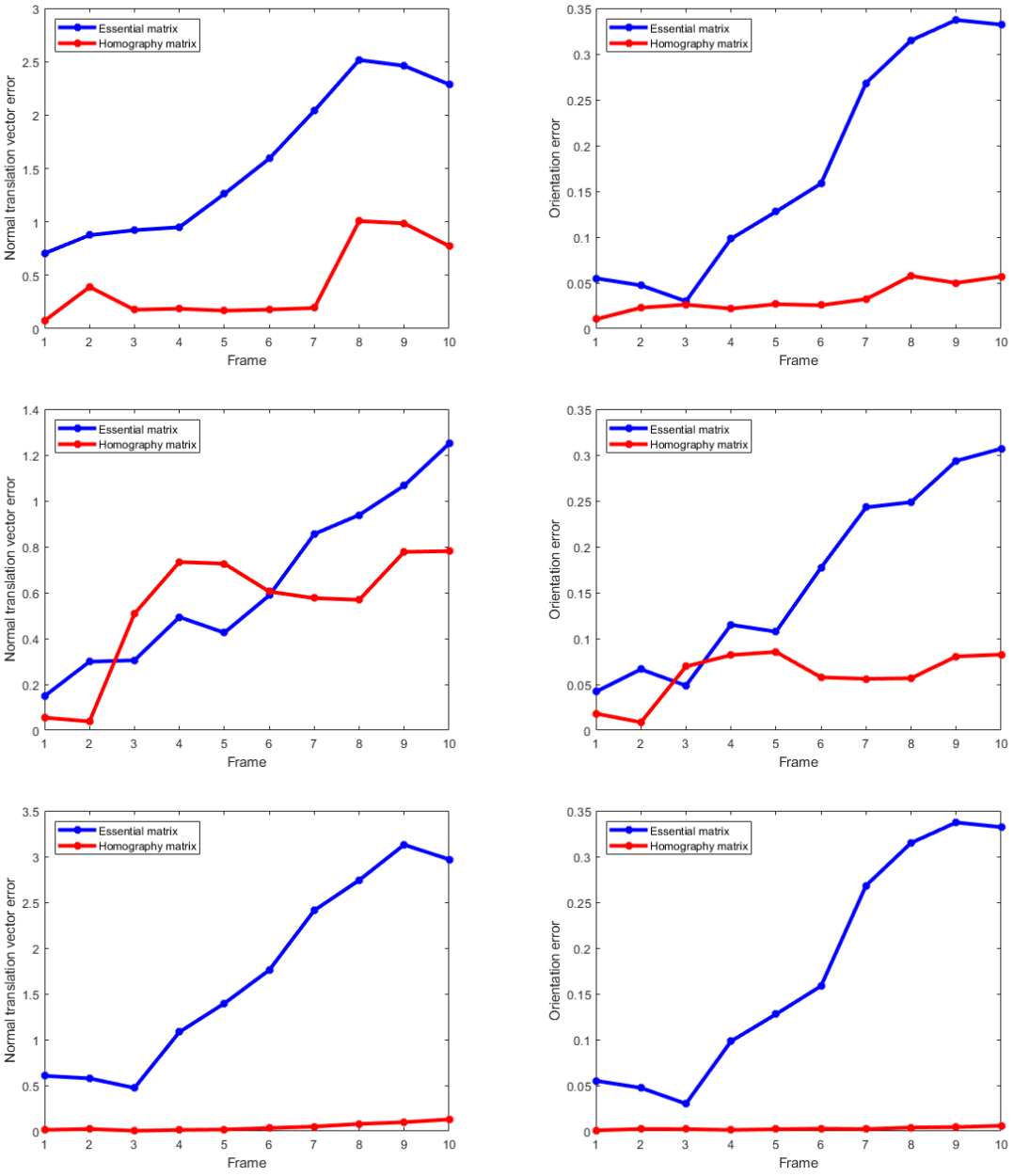


FIGURE 4.11: Evolution of the two measures on 10 frames consecutive : (a) translation direction and (b) orientation of camera on 3 sequences I,II,III from top to bottom respectively

Approach	Average error	(I)	(II)	(III)
Homography matrix	$\bar{d}_R$	0.033	0.060	0.003
	$\bar{d}_t$	0.414	0.537	0.046
Essential matrix	$\bar{d}_R$	0.177	0.165	0.177
	$\bar{d}_t$	1.561	0.637	1.714

TABLE 4.6: Comparison between the camera orientation and translation movement between the mentioned two approaches

better precision than using the essential matrix estimated from matching feature points although the execution time is not an advantage because of the dense optical flow estimation. However, for the defence of *uv*-velocity, it produces not only odometry information but also provides information about the structure of the scene through the planar segmentation.

4.3 Conclusion

In this chapter, we propose two usages of *uv*-velocity and its resulting plane segmentation :

— An adaptive model to integrate geometric information to improve optical flow estimation

— A complete usage of the whole opportunistic system : visual odometry together with scene reconstruction

The geometric feedback information is useful to refine the optical flow on non-texture surfaces. The proposed method has a lot of potential extensions. Instead of using a binary decision for the planar ownership of each pixel, we could build an ownership probability map, so that each pixel could belong to several planar structures with a given probability. This could solve the problem of false segmentation. In that case, an additional strategy is needed to update these probabilities alongside with the minimization of the new energy function.

For the visual odometry estimation, the homography matrix is computed fast using the planar regions provided by *uv*-velocity. However, the overall calculation time of the whole processing chain has to be reduced. A more important effort should be made on the optical flow block.

Integrating the odometry and 3D planes from uv-velocity to create a complete SLAM application is also one of the potential use of the proposed approach.

Chapitre 5

Variations of the proposed system

5.1 Introduction

This chapter presents some variations of the proposed processing chain that goes from motion estimation to visual odometry. It aims to show the flexibility of the proposed framework, regarding especially the optical flow block. As the core input of our system, this block also requires improvement in terms of precision and calculation time. We proposed in Chapter 4 a technique to increase the precision of the optical flow estimation by integrating the geometric information in the energy function. In this chapter, we explore the capacity of the optical flow method to adapt to different situations : first when a color camera is not available and gray information has to be exploited, second when optical flow is estimated sparsely, on a restricted number of points. Once again, it is a question of instantiating the opportunism of the approach through some examples showing how the global system adapts to the availability or not of some information. The extension of our optical flow to gray images should illustrate its versatility to any kind of input data, such as depth or texture information for example. Estimating the optical flow in a sparse way should lead to a faster computation while maintaining the precision. It has to be verified whether or not the uv-velocity can provide an accurate planar segmentation with a restricted number of motion values, even if this voting strategy has been initially designed for dense motion map. We study in this chapter the last branches of our opportunistic system : the variation of optical flow approach and its effect on the sub-blocks (see figure 5.1).

Therefore, the speed should be increased at the cost of a lower estimation density. The experiments conducted in this chapter address this question by evaluating uv-velocity, as well as its applications for planar segmentation and visual odometry estimation. Section 5.2 evaluates our optical flow estimation in its *degraded state*, that is for gray images instead of color, and by considering a sparse estimation instead of a dense one.

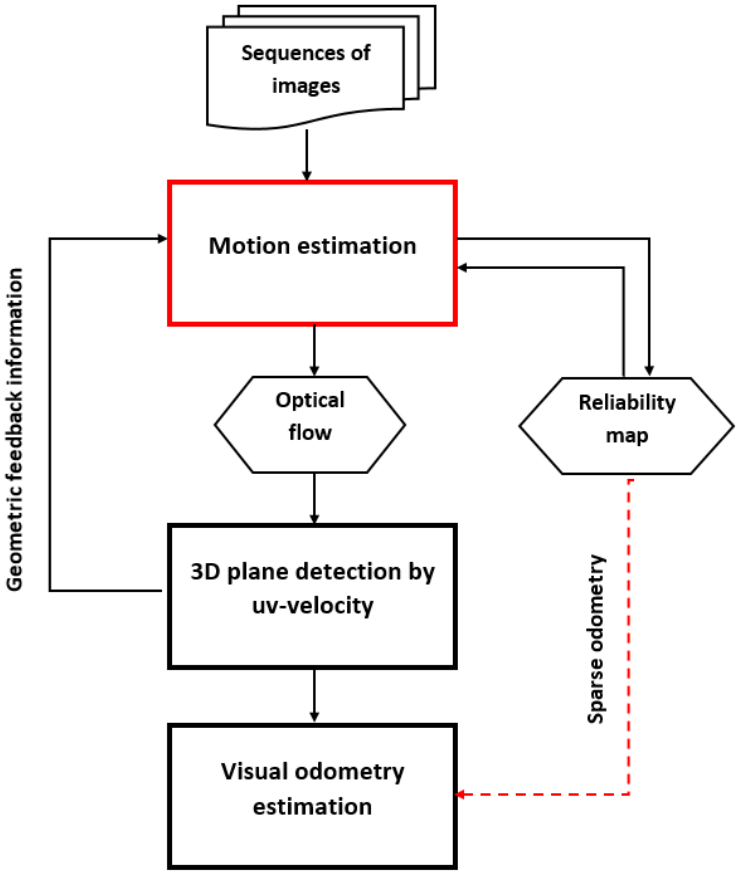


FIGURE 5.1: The developed block in this chapter, the *uv*-velocity and its application for plane segmentation

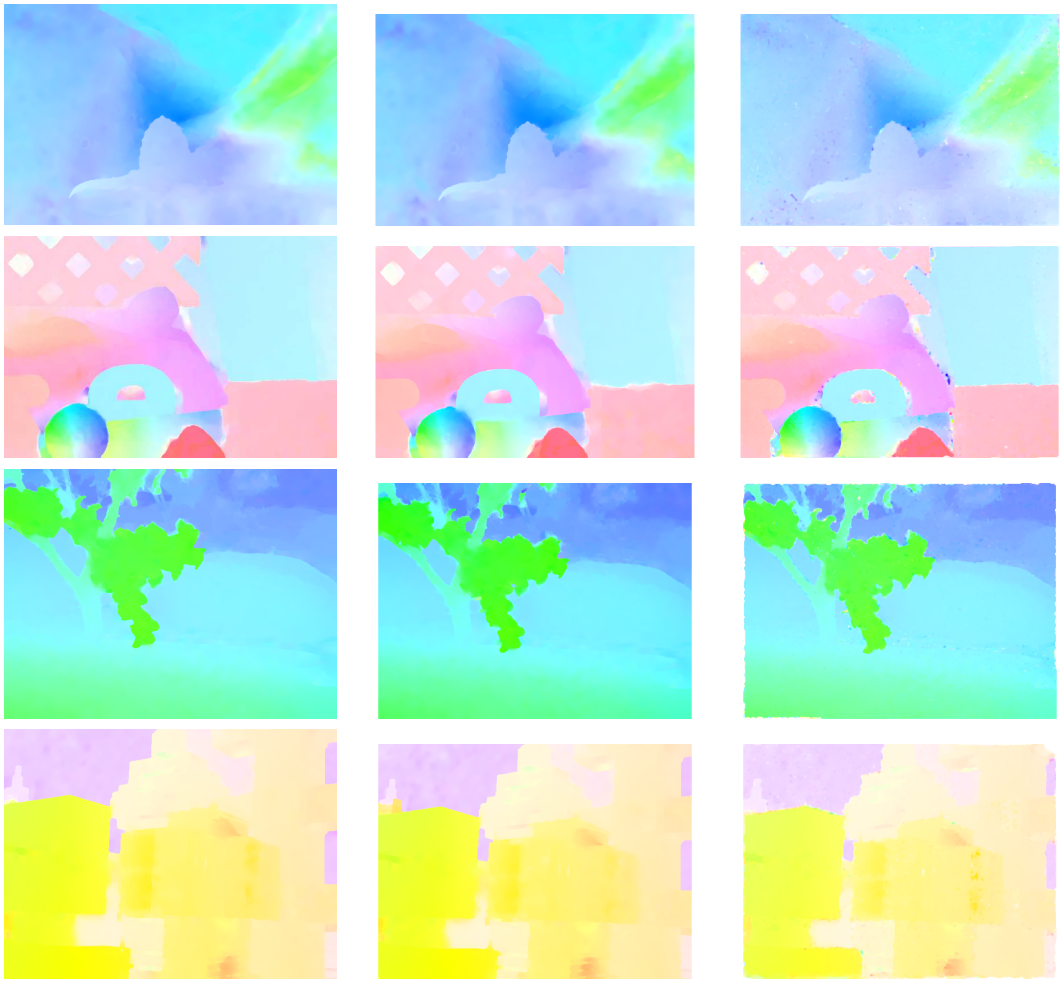
5.2 Optical flow

Chapter 2 has proposed a dense matching between two frames which exploits the color information to locally refine the motion through a propagation process. Let us now explore two variations of the method : first by considering to monochrome image ; second by stopping the propagation in order to output a sparse motion map.

Gray image. It was previously assumed in our method, denoted *Reliability Flow* (RF), that objects in the image are distinguished by their color. The refinement is made iteratively, where the motion of each color pixel is influenced by motion of its neighbors according to a color similarity measure. Even the first rough estimation of the motion by the KLT technique uses all three channels. In the case of gray images where the color information is not available, the adaptation of the algorithm is straightforward. The number of channels is reduced to one in equation (2.52), the color distance d_{color} is changed from Euclidean distance between two 3×1 vectors to an absolute difference of

intensities. The rest of the algorithm remains the same concerning the computation of the optical flow reliability, and the refinement under the local uniform gray constraint.

This variation of our method is evaluated on the Middlebury dataset, with the same images as in chapter 2, and using the same error measures, that is AEPE and AAE. The results are collected in Table 5.1. Since the AEPE correlates with AAE, only the AAE values are displayed. As expected, without the color information the global performance slightly decreases. Visually, there are slight differences between the optical flow estimated by color and by intensity, as can be seen respectively on figures 5.2(a) and 5.2 (b). This result shows that without color information, the intensity can be used without degrading too much the quality of the estimation.



(a) (b) (c)

FIGURE 5.2: Color encoded optical flow of 4 sequences : Dimetrodon, Rubberwhale,Grove2 and Urban3 (from top to bottom) of RF-color (a), RF-gray(b) and RF-sparse(c)

Sparse Estimation. In some applications that use optical flow as an input, the estimation does not need to be 100% dense. The *uv*-velocity requires enough data related

to planar surfaces to be able to form the corresponding parametric curve since it is based on a voting process. However, the experiments of chapter 3 have shown that even with the sparse ground truth optical flow of the KITTI database, the segmentation results are still good. In these experiments, the estimation of the OF is limited to the seed points found in the first step of the propagation. The first KLT estimation and the evaluation of the reliability are still done in all pixels but the propagation proceeds only the seeds and their neighbors in a patch of size 5×5. The same patches are used for the KLT estimation at next iteration. This estimation is called *RF sparse*. The error measures are shown in table 5.1. The *RF (sparse, seed)* measures the error only on the seeds which are the valid points for the estimation, the *RF (sparse, all pixels)* indicates the error measures for all of the pixels. The average error of these seeds is even better than the original method not only because it takes into account less points than the classical RF but because they allow a good estimation. The result on the RF (sparse, all pixels) shows a rapid increase of the error when the other pixels are taken into account. Visually on figure 5.2(c), the global optical flow looks worse than the original RF (figure 5.2(a)) which is coherent with the error measure of table 5.1.

Table 5.2, where the notations of Chapter 2 are kept, shows the calculation time of the sparse estimation using propagation on 3 image groups with different sizes : group 1 :420×380,group 2 :584×388, group 3 : 640×480. It can be observed a reduction of 30% compared to the classical RF. These experiments show that the proposed dense optical flow method can be adapted to perform a sparse estimation, which allows a significant reduction of the calculation time while preserving the performance on the valid pixels.

Database	Venus	Dimetrodon	Hydrangea	Rubberwhale	Grove2	Grove3	Urban2	Urban3	Average
RF (color)	4.100	3.103	2.392	3.491	2.245	5.461	3.157	2.874	3.352
RF (gray)	4.19	3.45	2.52	3.96	2.59	5.89	3.41	3.36	3.68
RF (sparse,seed)	**3.97**	**3.02**	**2.03**	**2.75**	**1.73**	**4.08**	**2.78**	**2.73**	**2.89**
RF (sparse, all pixel)	5.42	3.43	2.48	4.00	3.28	6.67	4.36	4.41	4.26

TABLE 5.1: Comparison of AAE between classical RF, RF-gray and RF-sparse

Group	1	2	3
RF dense	23.3	34.5	46.9
RF sparse	17	23.3	33

TABLE 5.2: Computation times (in seconds) between the dense and sparse RF

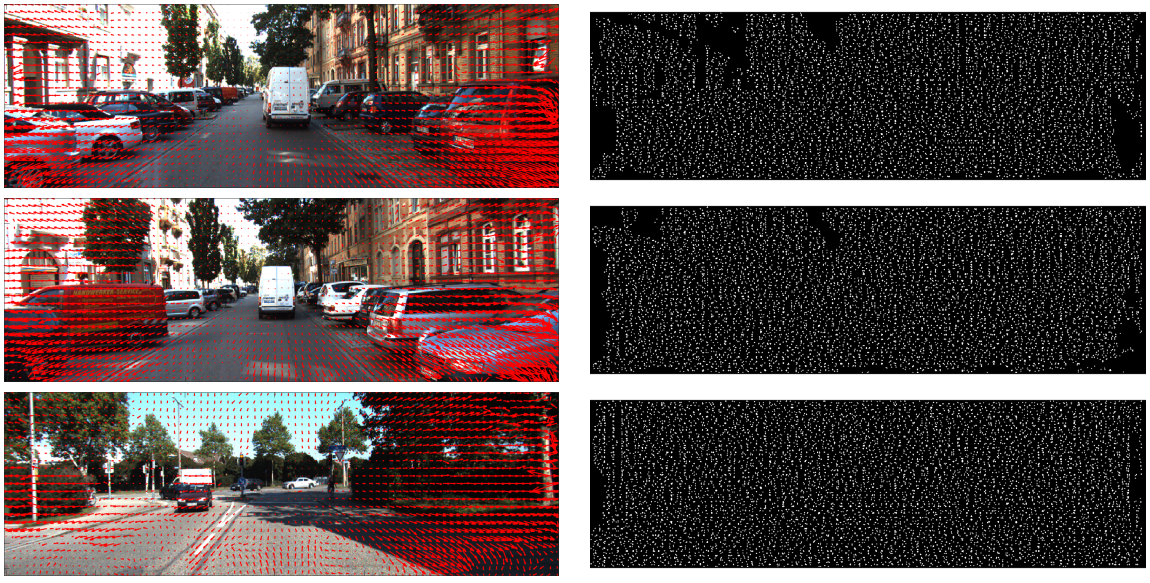
The sparse optical flow is then used to create the voting space and to estimate the visual odometry via homography estimation. The next section studies the stability of these applications under this estimation.

5.3 uv-velocity

We use the 3 sequences presented in the Chapter 4.1. The sparse optical flow is shown in figure 5.3 together with the seeds maps. We first measure the parameters of the road curve on the voting space $\mathbf{V}(v, y)$ which is characterized by the parameter K as follows :

$$\begin{aligned} u_1 &= K(x - x_{FOE})|y| \\ v_1 &= K(y - y_{FOE})|y| \end{aligned} \tag{5.1}$$

Table 5.3 collects the parameters estimated from the sparse optical flow, as well as from dense optical flow and ground truth optical flow. Although having fewer points available to vote, the parameter is still correctly estimated.



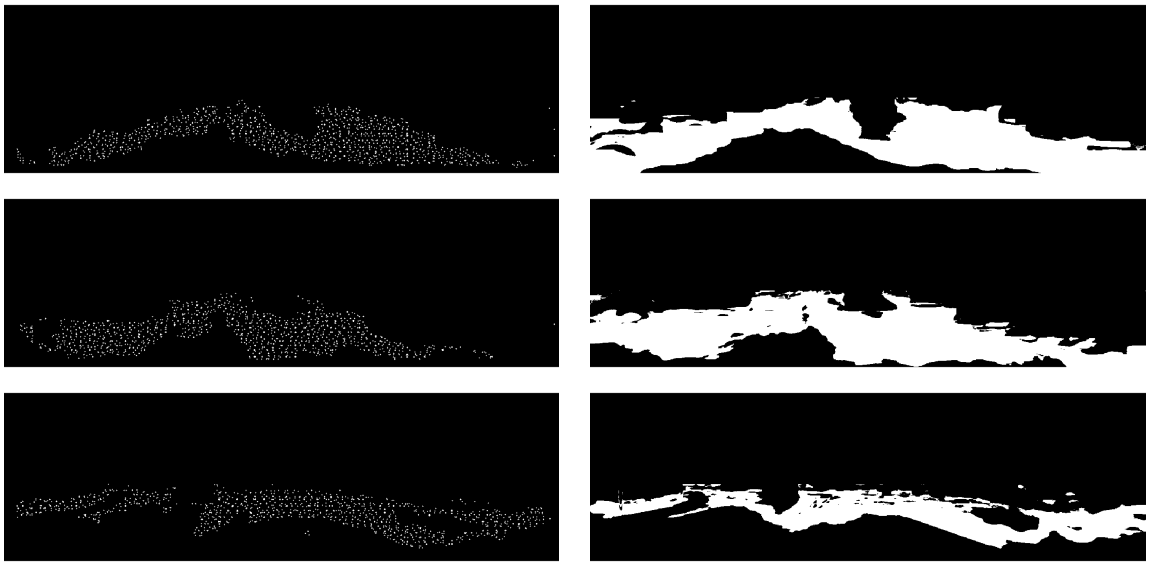
(a) (b)

FIGURE 5.3: The sparse estimated optical flow (a) and the corresponding mask of seeds (b) of 3 examples sequences I,II and III

Sequence	K_{GT}	K_{dense}	K_{sparse}
(I)	6.084×10^{-4}	6.099×10^{-4}	6.01×10^{-4}
(II)	6.9138×10^{-4}	6.9319×10^{-4}	7.0×10^{-4}
(III)	8.2758×10^{-4}	8.011×10^{-4}	8.21×10^{-4}

TABLE 5.3: Parameter K found by ground truth optical flow(K_{GT}), dense (K_{dense} and sparse K_{sparse} optical flow

The segmentation from sparse optical flow produces almost the same results as the dense estimation displayed on figure 5.4(b), except that the points are scattered. Figure 5.4 (a) shows the segmented plane of 3 examined sequences on a background colored in black for an easier observation. In each case, segmentation errors occur.



(a) (b)

FIGURE 5.4: Comparison of road plane segmentation based on the sparse (a) and dense (b) optical flow of 3 examples sequences I,II and III (from top to bottom)

While maintaining the precision of the curve parameter on *uv*-velocity voting space, the sparse optical flow allows a faster calculation time and can be used to estimate the parameters of the planes or their homographies. The next part evaluates the stability of the visual odometry using only a limited number of points, that is the points that are detected and considered as co-planar by *uv*-velocity.

5.4 Visual odometry

The experiments of the previous chapter on the estimation of the visual odometry are carried on by using the sparse optical flow and its plane identification from *uv*-velocity. The three sequences of the Section 4.2.3 are considered here again together with their 11 frames. The table 5.4 shows the average errors of odometry estimation for the dense optical flow and for the sparse optical flow. Figure 5.5 shows the evolution of the error measures for each optical flow.

Overall, the odometry estimation holds its quality when using the sparse estimation, as shown by the slight difference of precision between the dense optical flow and the sparse one. However, under the sparse estimation, the calculation time is reduced significantly, from an average of 50 seconds to 2 seconds. This result once again proves that our seeds selected by the proposed reliability criteria are relevant to serve as indicators for the precision of the optical flow estimation.

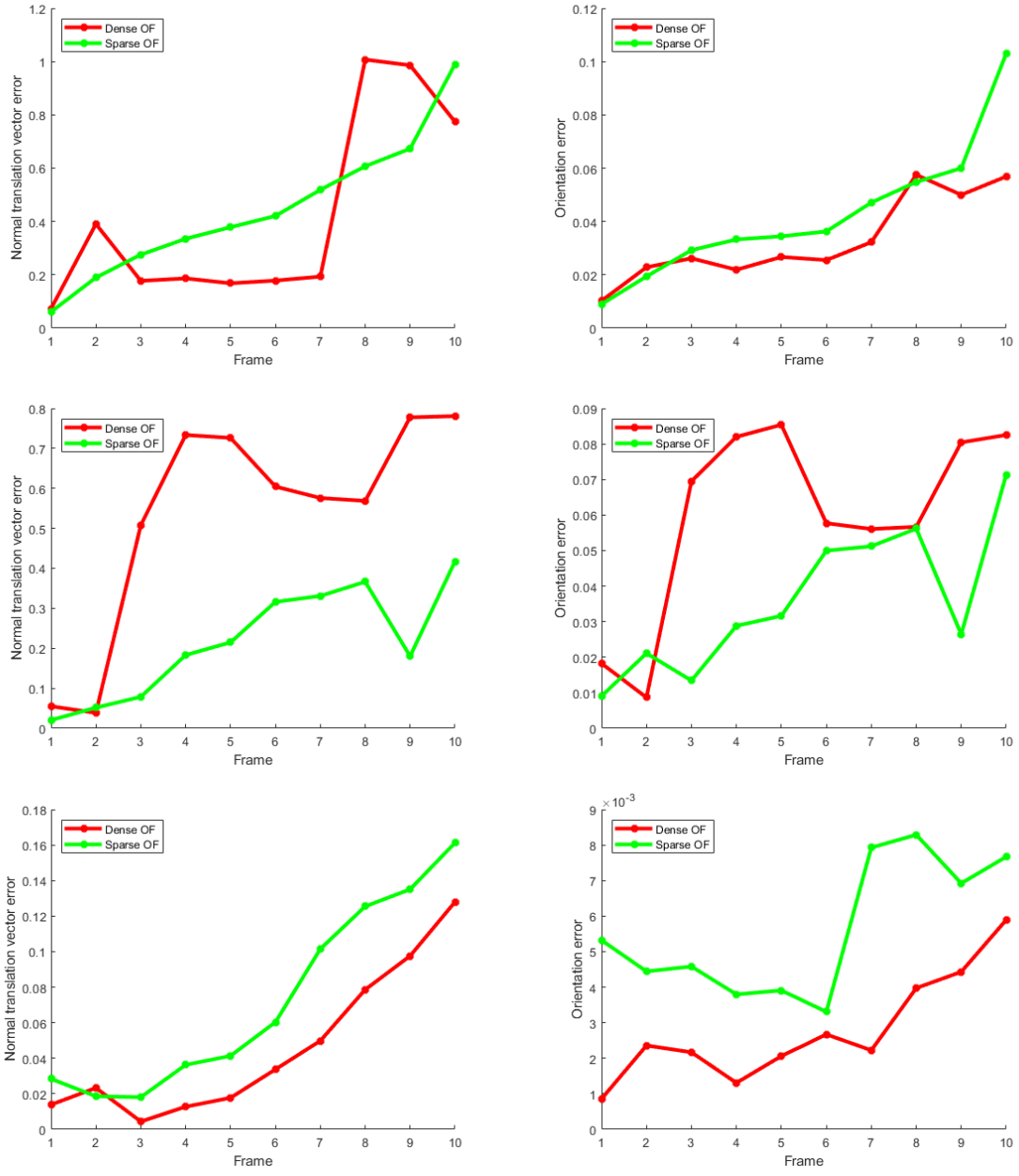


FIGURE 5.5: Evolution of the two measures : (a) translation direction and (b) orientation of camera on 3 sequences I,II,III from top to bottom respectively

Approach	Average Error	(I)	(II)	(III)
Dense optical flow	$\bar{d}_R$	0.033	0.060	0.003
	$\bar{d}_t$	0.414	0.537	0.046
Sparse optical flow	$\bar{d}_R$	0.043	0.036	0.006
	$\bar{d}_t$	0.445	0.216	0.073

TABLE 5.4: Comparison between the camera orientation and translation movement between the homography formed using the dense optical flow and the sparse one

5.5 Conclusion

This chapter has explored the flexibility of our optical flow estimation, when modifying the nature of the input data (from color to gray) and the density level of the motion map. The choice of the points for sparse estimation, based on the proposed reliability criteria, has proved to be effective for each stage of the proposed processing chain : uv-velocity, segmentation and odometry estimation. The density of the estimation is just enough to keep the performance of uv-velocity while speeding up the whole process. With this huge improvement in calculation time for optical flow estimation, the opportunistic system could be achieved in real-time in the future.

Chapitre 6

Conclusion

This thesis has proposed some contributions towards a vision-based opportunistic system for mobile robot which relies on the only information provided by a single camera, without any additional sensor. The objective is to take the maximum profit from the monoscopic sensor, by exploiting color and motion to obtain information on both the structure of the scene and the ego-motion.

This system, that we have started to design, should first evaluate the relevance of the available input attributes, which could be intensity, color or texture. From these cues, the motion would be estimated by using the most relevant or reliable ones, in order to adapt to different contexts and different scenes. It would offer the possibility to produce a dense or a sparse motion map, depending on the targeted application and depending on the computing and memory resources. From this motion map, the most relevant values are used to exhibit the main planar regions through a voting strategy, which offers a good immunity to noise. Then, the estimated structure can be used to improve the next motion estimation. It can also provide information about the ego-motion. Ideally, the system should also be able to evaluate the quality of each intermediate result, in order to be able to call a complementary method when needed.

At this stage, there are still work to do to reach the initial expectations, in particular regarding the adaptability to different contexts, or the self-evaluation. However, we have tried to define a whole optical flow-based processing chain, which comprises a level of self-evaluation (in the optical flow method) and includes different levels of feedback : from motion to structure, and from structure to motion.

6.1 Contributions

Optical flow refinement using iterative propagation under color, proximity and flow reliability constraints

In Chapter 2, optical flow methods have been studied deeply because this is the key input for our system. Beside the numerous research about optical flow estimation, we propose a method which is able to self-evaluate the quality of its estimation. This analysis is made on the basis of the classical KLT method, but the philosophy could be extended to other techniques.

After a first estimation, a reliability map is built by using three criteria : the cornerness, the motion local uniformity and the temporal evolution of the residues. The motion map is then refined iteratively in a hierarchical way, and the reliability map is modified alongside. First of all, the most relevant motion values are used as the seeds of the propagation process. A first step consists in a sparse correction at seeds level. Then the optical flow is corrected densely at pixel level. This propagation process is guided by the color similarity in a neighborhood weighted by the reliability scores. Progressively, through the iterations, the overall reliability of the motion map is increased. The experiments have shown a correlation between the evolution of the estimation errors and the reliability scores. Eventually, according to the experiments performed on the Middlebury dataset, the refined optical flow improves the quality of the estimation.

The Chapter 5 has illustrated the versatility of the optical flow method. First, it can be applied to monochrome images. Then, the propagation process aiming to produce a dense map, can be stopped at different levels to form a sparser map. Since the optical flow is estimated in a hierarchical way and starts with the most reliable values, it can be decided to compute a sparse reliable map or to estimate a dense but less reliable map.

From motion to structure : uv-velocity, an optical flow registration for planar segmentation

Next, from the motion map, a new optical flow registration called uv-velocity is developed to detect rapidly three kinds of planar surfaces present in the scene : lateral, frontal and horizontal. To begin with, we have considered the assumptions of a Manhattan world, in which the robot moves forward along the Z axis. Like its predecessor, the so-called c-velocity, the method transforms the input motion map into an accumulated voting space where the planes are exhibited as 2D parametric curves, either straight lines or parabolas. The planar segmentation is therefore reduced to a curve fitting problem.

The new method is more straightforward than the c-velocity, in the way that the two voting spaces are directly built using two relationships : u as a function of x and v as a function of y. Consequently, it does not require any approximation as in c-velocity, for which the c axis has to be sampled. Once the curves have been detected, a motion model

is estimated directly from the shapes parameters. The experiments on KITTI optical flow dataset have shown similar performances for both techniques, with a much lower resource demand for *uv*-velocity.

In contrast to its predecessor, the new *uv*-velocity can be expanded to a wider range of situations. The ego-motion model can now include rotations. In the same way, it is able to detect planar surfaces that are oriented with respect to the camera coordinate system.

Chapter 5 has successively tested the *uv*-velocity with a sparser motion map. While voting strategies generally implies a dense input, it is shown that a sparse but reliable flow can also be used in such cumulative methods. However, note that the emerging segmentation is less precise, but is computed faster.

From structure to motion

Once estimated, the arising structure can help improving the motion estimation in the next frames of the sequence, not only the motion map but also the ego-motion. Beside a planar segmentation, *uv*-velocity provides a simplified motion model for each plane. These motion models can be injected as a constraint in the cost function to be minimized for optical flow estimation. Therefore the optical flow estimation is guided toward the geometric constraint alongside with the classical brightness and smoothness constraints. The experiments conducted on the KITTI dataset have shown that such a strategy can indeed improve the precision of the optical flow as well as the output segmentation. The plane segmentation is also used to estimate the homography matrix which contains the odometry information of the camera. By focusing the estimation on one of the planar regions of the image, the homography matrix is found quickly with RANSAC algorithm. Experiments have shown that the estimated odometry is also more precise than using a direct feature-based method. From the fast plane detection, the odometry estimation completes the processing chain of our opportunistic system to reveal rapidly the main structure of the scene and to provide the navigation system with the supplementary odometry information.

The system is designed for urban scenes or indoor scene where the context can be simplified in planar surfaces. As mentioned previously, the geometry analysis is funded on several assumptions concerning the scene and the ego-motion, even if the *uv*-velocity has alleviated theses restrictions. Our objective, fully and consciously accepted, was to push monovision possibilities as far as possible. Most robot systems integrate several sensors, therefore the methods proposed in this thesis could certainly benefit from these additional sources of information. It could be used as a alternate or a complementary approach to a stereovision strategy. It can be used as a rough estimation that could be refined when need by a more complete multi-sensor framework.

6.2 Perspectives

Optical flow estimation

The motion map is the most important block of the system, because the performance of all downstream algorithms depends directly on the quality of the optical flow. It has an impact on the sharpness of the curves to be extracted in uv-velocity and therefore on the precision of the motion models, as well as the precision of the segmentation.

The refinement of the optical flow is based on local constraints like color and motion local uniformity, that allows to provide a dense map of good quality. However, by analyzing the model of optical flow on planar surfaces, these constraints are not suitable anymore because of the linear or quadratic relation between optical flows of neighboring pixels. Thus an adaptive propagation strategy should also include information about the local geometric structure. In Chapter 4, the structure was included in the minimization process but it would be interesting to modify the refinement process in the same way.

Beside the intensity information used to estimate the optical flow, a feature-based matching could be integrated into the energy function for better interpreting the 3D motion.

Initially, we also wanted to use other low-level features such as texture information, in addition to color or intensity. This could be achieved by using Local Binary Patterns or Gabor filters for example. Our optical flow estimation method should be easily extended to this feature because any local constraint can be used in the refinement process, the color similarity being replaced by a texture similarity. It would have been interesting to give more or less importance to each feature depending on their relevance (for example the amount of gradient in the patch under consideration).

In order to take benefit of the planar segmentation to improve the estimation of the optical flow, each pixel is assigned to one plane, with a binary ownership. However, when the segmentation is imprecise, the binary assignation can be detrimental for the quality of the estimation. This binary ownership should be replaced by probabilities. Therefore, the cost function should include a new term that would represent the ownership of a pixel to each region, and therefore to each motion model.

Planar segmentation and characterization by uv-velocity

The analysis of the voting spaces is crucial for the detection of the planar surfaces. It could be improved in different ways. For example, the resolution of the ordinate axes, that is u or v resolution could be improved by integrating sub-pixel precision. In this work, we use scale factor of 1, which means each pixel on horizontal axis represents one unity of optical flow component. A more fine scale factor would give a more clear voting space. The choice is also as important as the windows size of local optical flow estimation method. Eventually, for a specific application, the scale can be optimized knowing the average amplitude of optical flow.

Since our strategy was to use a binary threshold and a simple Hough transform to analyze the voting spaces, there is room for improvement. The threshold has an important impact on the size of planes to be segmented. The method shows no difficulty to reveal the large planes of the scene. However, it is more complicated to segment the smaller planes due to the noise. It would be interesting to use a more advanced technique. Probably, deep learning methods could be a response, but it would require a large amount of uv-velocity results data. Also, a multiscale analysis could be tested. Otherwise, depending on the targeted application, obstacle detection or scene reconstruction for example, a secondary object recognition method could be used to boost the performance of the system.

Odometry estimation

Since the odometry can only be recovered at a scaled factor, it needs an additional information to be able to recover the true displacement of translation motion, which can come from the stereo-vision, the depth camera or IMU. In particular cases, the real translation movement can be estimated without the supplementary peripheries, using real time SLAM by mono-vision. SLAM (Simultaneous Localization and Mapping) consists in building and updating the map and identifying the location of the mobile robot in the map. The state of the art has been developed for 25 years with many techniques (filters, optimization techniques) and sensors (camera, IMU, GPS, encoder...), as explained in the review [BAYG17]. With our opportunistic system, under pure translation movement of the mobile robot, the segmentation and exact visual odometry can be retrieved when the four following conditions are satisfied :

— the quality of optical flow is adequate,
— the world consists of planes, which is true for human world
— the camera mounted on the robot does not change its height d during operation which is also true for mobile rolling robots
— the camera is calibrated, and in particular f is known

When camera observes the scene as the robot moves, the uv-velocity finds the planar surfaces appearing in the image. The ground plane is always available in this context. With the configuration of the camera, the ground is always an horizontal plane. Under pure translation movement of the robot we have the following relationships (already mentioned in table 3.3) :

$$\begin{aligned} u =& K(x - x_{FOE})|y| \\ v =& K(y - y_{FOE})|y|, \end{aligned} \tag{6.1}$$

where $K = \frac{T_Z}{fd}$ is the estimated parameter. Knowing f and d, we can compute T_Z, *i.e.* the true displacement in the Z-direction. For the ex-horizontal plane, the expression between

optical flow and camera movements is given as :

$$\begin{aligned} u &= \frac{1}{fd_{exh}}\big(yxT_Zsin(\theta) + f(xT_Zcos(\theta) - yT_Ysin(\theta)) - f^2T_Ycos(\theta)\big) \\ v &= \frac{1}{fd_{exh}}\big(y^2T_Zsin(\theta) + fy(T_Zcos(\theta) - T_Ysin(\theta)) - f^2T_Ycos(\theta)\big) \end{aligned} \tag{6.2}$$

By estimating the three parameters A,B,C of parabola $v = Ay^2 + By + C$ on voting space $\mathbf{V}(v, y)$, we have the following equations system :

$$\begin{cases} \frac{T_Zsin(\theta)}{fd_{exh}} = A \\ \frac{T_Zcos(\theta) - T_Ysin(\theta)}{d_{exh}} = B \\ \frac{fT_Ycos(\theta)}{d_{exh}} = C \end{cases} \tag{6.3}$$

Since f and d are known, we basically can solve this equation systems with 3 variables : T_Y, d_{exh} and θ.

The same analysis can be done for ex-lateral planes. From the quadratic relation between u and x on $\mathbf{U}(u, x)$ voting space :

$$u = \frac{1}{fd_{exl}}\big(x^2T_Zsin(\phi) + fx(T_Zcos(\phi) - T_Xsin(\phi)) - f^2T_Xcos(\phi)\big) \tag{6.4}$$

we also have the equations system :

$$\begin{cases} \frac{T_zsin(\theta)}{fd_{exl}} = A \\ \frac{T_zcos(\theta) - T_zsin(\theta)}{d_{exl}} = B \\ \frac{fT_xcos(\theta)}{d_{exl}} = C \end{cases} \tag{6.5}$$

The three variables T_X, d_{exl} and ϕ can be computed from these equations. With all translation components n T_X, T_Y, T_Z, the distance of plane from the origin and the angle of ex-planes, it is possible to reconstruct the scene with the real scale. Obviously, there are still more insight works to realize this idea.

Bibliographie

[AB08] A. Amintabar and B. Boufama. Homography-based plane identification and matching. In *2008 15th IEEE International Conference on Image Processing*, pages 297–300, Oct 2008.

[AH11] A. Adan and D. Huber. 3d reconstruction of interior wall surfaces under occlusion and clutter. In *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 275–281, May 2011.

[AHPB13] O. Mac Aodha, A. Humayun, M. Pollefeys, and G. J. Brostow. Learning a confidence measure for optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(5) :1107–1120, May 2013.

[Air] Kelson Rômulo Teixeira Aires. Plane detection from monocular image sequences. In *IASTED*.

[ASO10] T. Azuma, S. Sugimoto, and M. Okutomi. Egomotion estimation using planar and non-planar constraints. In *2010 IEEE Intelligent Vehicles Symposium*, pages 855–862, June 2010.

[AWS] Luis Alvarez, Joachim Weickert, and Javier Sánchez. Reliable Estimation of Dense Optical Flow Fields with Large Displacements. *International Journal of Computer Vision*, 39(1) :41–56.

[BA91] M. J. Black and P. Anandan. Robust dynamic motion estimation over time. In *, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91*, pages 296–302, June 1991.

[BA96] Michael J. Black and P. Anandan. The Robust Estimation of Multiple Motions : Parametric and Piecewise-Smooth Flow Fields. *Computer Vision and Image Understanding*, 63(1) :75–104, January 1996.

[BAYG17] Guillaume Bresson, Zayed Alsayed, Li Yu, and Sébastien Glaser. Simultaneous Localization And Mapping : A Survey of Current Trends in Autonomous Driving. *IEEE Transactions on Intelligent Vehicles*, XX :1, 2017.

[BBC+14] N. Bernini, M. Bertozzi, L. Castangia, M. Patander, and M. Sabbatelli. Real-time obstacle detection using stereo vision for autonomous ground vehicles : A survey. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 873–878, Oct 2014.

[BBH03] M. Z. Brown, D. Burschka, and G. D. Hager. Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8) :993–1008, Aug 2003.

[BBM09] T. Brox, C. Bregler, and J. Malik. Large displacement optical flow. pages 41–48. IEEE, June 2009.

[BBPW04] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In Tomás Pajdla and Jiří Matas, editors, *Computer Vision - ECCV 2004*, pages 25–36, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[BC05] G. Le Besnerais and F. Champagnat. Dense optical flow by iterative local window registration. In *IEEE International Conference on Image Processing 2005*, volume 1, pages I–137, Sept 2005.

[BDS15] Xiuxiu Bai, Xiaoshe Dong, and Yuanqi Su. Edge Propagation KD-Trees : Computing Approximate Nearest Neighbor Fields. *IEEE Signal Processing Letters*, 22(12) :2209–2213, December 2015.

[Beb06] Richard ; Parvin Bahram ; Koracin Darko ; Remagnino Paolo ; Nefian Ara ; Meenakshisundaram Gopi ; Pascucci Valerio ; Zara Jiri ; Molineros Jose ; Theisel Holger ; Malzbender Tom Bebis, George ; Boyle. *Planar Surface Detection in Image Pairs Using Homographic Constraints*, volume 10.1007/11919476. 2006.

[BELN11] Dorit Borrmann, Jan Elseberg, Kai Lingemann, and Andreas Nüchter. The 3d hough transform for plane detection in point clouds : A review and a new accumulator design. *3D Research*, 2(2) :3, Nov 2011.

[BGW91] J. Bigun, G. H. Granlund, and J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8) :775–790, August 1991.

[BHS98] Alireza Bab-Hadiashar and David Suter. Robust Optic Flow Computation. *International Journal of Computer Vision*, 29(1) :59–77, August 1998.

[BK01] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. In Mário

Figueiredo, Josiane Zerubia, and Anil K. Jain, editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 359–374, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[BM11] T Brox and J Malik. Large Displacement Optical Flow : Descriptor Matching in Variational Motion Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3) :500–513, March 2011.

[Bou00] Jean-yves Bouguet. Pyramidal implementation of the Lucas Kanade feature tracker. *Intel Corporation, Microprocessor Research Labs*, 2000.

[BRFK14] Frederic Besse, Carsten Rother, Andrew Fitzgibbon, and Jan Kautz. PMBP : PatchMatch Belief Propagation for Correspondence Field Estimation. *International Journal of Computer Vision*, 110(1) :2–13, October 2014.

[BSL+10] Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. A Database and Evaluation Methodology for Optical Flow. *International Journal of Computer Vision*, 92(1) :1–31, November 2010.

[BTS17] Christian Bailer, Bertram Taetz, and Didier Stricker. Optical Flow Fields : Dense Correspondence Fields for Highly Accurate Large Displacement Optical Flow Estimation. *arXiv :1703.02563 [cs]*, March 2017. arXiv : 1703.02563.

[BW02] Thomas Brox and Joachim Weickert. Nonlinear Matrix Diffusion for Optic Flow Estimation. In Luc Van Gool, editor, *Pattern Recognition*, number 2449 in Lecture Notes in Computer Science, pages 446–453. Springer Berlin Heidelberg, September 2002. DOI : 10.1007/3-540-45783-6_54.

[BW06] A. Bruhn and J. Weickert. *A Confidence Measure for Variational Optic flow Methods*, pages 283–298. Springer Netherlands, Dordrecht, 2006.

[BWSB12] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *European Conf. on Computer Vision (ECCV)*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, October 2012.

[BZ12] Samia Bouchafa and Bertrand Zavidovique. c-Velocity : A Flow-Cumulating Uncalibrated Approach for 3d Plane Detection. *International Journal of Computer Vision*, 97(2) :148–166, April 2012.

[CAS08] S Chhaniyara, K Althoefer, and L Seneviratne. Visual odometry technique using circular marker identification for motion parameter estimation. In L Marques, AD Almeida, and MO Tokhi, editors, *Advances in mobile robotics*, number 17, pages 1069 – 1076. World Scientific Pub Co Inc, 2008.

[CD10] D. Conrad and G. N. DeSouza. Homography-based ground plane detection for mobile robot navigation using a modified em algorithm. In *2010 IEEE International Conference on Robotics and Automation*, pages 910–915, May 2010.

[CJL+13] Zhuoyuan Chen, Hailin Jin, Zhe Lin, Scott Cohen, and Ying Wu. Large Displacement Optical Flow from Nearest Neighbor Fields. pages 2443–2450. IEEE, June 2013.

[CM87] E. De Castro and C. Morandi. Registration of translated and rotated images using finite fourier transforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5) :700–703, Sept 1987.

[Cor07] P. Corke. Matlab toolboxes : robotics and vision for students and teachers. *IEEE Robotics Automation Magazine*, 14(4) :16–17, Dec 2007.

[CPP12] Diego Cheda, Daniel Ponsa, and Antonio Manuel López Peña. Monocular egomotion estimation based on image matching. In *ICPRAM*, 2012.

[CS05] Chih-Hao Chen and Kai-Tai Song. Complete coverage motion control of a cleaning robot using infrared sensors. In *IEEE International Conference on Mechatronics, 2005. ICM '05.*, pages 543–548, July 2005.

[CYT11] Ce Liu, Jenny Yuen, and Antonio Torralba. SIFT Flow : Dense Correspondence across Scenes and Its Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5) :978–994, May 2011.

[DH72] Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1) :11–15, January 1972.

[DM00] Douglas DeCarlo and Dimitris Metaxas. Optical flow constraints on deformable models with applications to face tracking. *International Journal of Computer Vision*, 38(2) :99–127, Jul 2000.

[DMC15] Nameirakpam Dhanachandra, Khumanthem Manglem, and Yambem Jina Chanu. Image segmentation using k -means clustering algorithm and subtractive clustering algorithm. *Procedia Computer Science*, 54 :764 – 771, 2015. Eleventh International Conference on Communication Networks, ICCN 2015, August 21-23, 2015, Bangalore, India Eleventh International Conference on Data Mining and Warehousing, ICDMW 2015, August 21-23, 2015, Bangalore, India Eleventh International Conference on Image and Signal Processing, ICISP 2015, August 21-23, 2015, Bangalore, India.

[DRMS07] Andrew J. Davison, Ian D. Reid, Nicholas Molton, and Olivier Stasse. Monoslam : Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29 :1052–1067, 2007.

[DT05] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, June 2005.

[Far00] G. Farneback. Fast and accurate motion estimation using orientation tensors and parametric motion models. volume 1, pages 135–139. IEEE Comput. Soc, 2000.

[Far03] Gunnar Farnebäck. *Two-Frame Motion Estimation Based on Polynomial Expansion*, pages 363–370. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[FB81] Martin A. Fischler and Robert C. Bolles. Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6) :381–395, June 1981.

[FBYJ00] David J. Fleet, Michael J. Black, Yaser Yacoob, and Allan D. Jepson. Design and Use of Linear Models for Image Motion Analysis. *International Journal of Computer Vision*, 36(3) :171–193, February 2000.

[FL88] Olivier Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. Technical Report RR-0856, INRIA, June 1988.

[For07] Per-Erik Forssen. Maximally Stable Colour Regions for Recognition and Matching. pages 1–8. IEEE, June 2007.

[FPS14] C. Forster, M. Pizzoli, and D. Scaramuzza. Svo : Fast semi-direct monocular visual odometry. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15–22, May 2014.

[FZB02] H. Foroosh, J. B. Zerubia, and M. Berthod. Extension of phase correlation to subpixel registration. *IEEE Transactions on Image Processing*, 11(3) :188–200, Mar 2002.

[GBB98] Patrick Gros, Olivier Bournez, and Edmond Boyer. Using local planar geometric invariants to match and model images of line segments. *Computer Vision and Image Understanding*, 69(2) :135–155, February 1998.

[GG16] Fatma Güney and Andreas Geiger. Deep discrete flow. In *Asian Conference on Computer Vision (ACCV)*, 2016.

[GM17] M. Garrigues and A. Manzanera. Fast semi dense epipolar flow estimation. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 427–435, March 2017.

[GSC79] Georges Giralt, Ralph Sobek, and Raja Chatila. A multi-level planning and navigation system for a mobile robot : A first approach to hilare. In *Proceedings of the 6th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'79, pages 335–337, San Francisco, CA, USA, 1979. Morgan Kaufmann Publishers Inc.

[GT96] Andrea Giachetti and Vincent Torre. Refinement of Optical Flow Estimation and Detection of Motion Edges. In *In ECCV96*, pages 151–160, 1996.

[Hou62] Paul VC Hough. Method and means for recognizing complex patterns, December 18 1962. US Patent 3,069,654.

[HS81] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3) :185–203, August 1981.

[HW88] Berthold K. P. Horn and E. J. Weldon. Direct methods for recovering motion. *International Journal of Computer Vision*, 2(1) :51–76, Jun 1988.

[HZ03] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003.

[IA00] M. Irani and P. Anandan. About direct methods. In Bill Triggs, Andrew Zisserman, and Richard Szeliski, editors, *Vision Algorithms : Theory and Practice*, pages 267–277, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[IF99] Atsushi Imiya and Iris Fermin. Voting method for planarity and motion detection. *Image and Vision Computing*, 17(12) :867 – 879, 1999.

[JHG99] B. Jähne, H. Haussecker, and P. Geissler. *Handbook of Computer Vision and Applications : Signal processing and pattern recognition*. Number Vol. 2 in Handbook of Computer Vision and Applications. Academic Press, 1999.

[Jin14] Tammam ; Cheng Fei Jin, Zhi ; Tillo. Planar surfaces detection on depth map using patch based approach. 2014.

[KEB91] N. Kiryati, Y. Eldar, and A.M. Bruckstein. A probabilistic hough transform. *Pattern Recognition*, 24(4) :303 – 316, 1991.

[KJ12] Kaiming He and Jian Sun. Computing nearest-neighbor fields via Propagation-Assisted KD-Trees. pages 111–118. IEEE, June 2012.

[KLL13] Tae Hyun Kim, Hee Seok Lee, and Kyoung Mu Lee. Optical Flow via Locally Adaptive Fusion of Complementary Data Costs. pages 3344–3351. IEEE, December 2013.

[KMG08] Claudia Kondermann, Rudolf Mester, and Christoph Garbe. *A Statistical Confidence Measure for Optical Flows*, pages 290–301. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[KTDVG16] Till Kroeger, Radu Timofte, Dengxin Dai, and Luc Van Gool. *Fast Optical Flow Using Dense Inverse Search*, pages 471–488. Springer International Publishing, Cham, 2016.

[LAT02] R. Labayrade, D. Aubert, and J. P. Tarel. Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation. In *IEEE Intelligent Vehicle Symposium, 2002*, volume 2, pages 646–651 vol.2, June 2002.

[LCR03] Haiying Liu, R. Chellappa, and A. Rosenfeld. Accurate dense optical flow estimation using adaptive structure tensors and a parametric model. *IEEE Transactions on Image Processing*, 12(10) :1170–1180, October 2003.

[LHP80] Hugh Christopher Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Proceedings of the Royal Society of London. Series B, Biological sciences*, 208 1173 :385–97, 1980.

[LK81] Bruce D. Lucas and Takeo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. pages 674–679, 1981.

[LKM04] F. Lauze, P. Kornprobst, and E. Mémin. A coarse to fine multiscale approach for linear least squares optical flow estimation. 2004.

[LL97] Jianhua Lu and M. L. Liou. A simple and efficient search algorithm for block-matching motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(2) :429–433, Apr 1997.

[LNHS05] Kai Lingemann, Andreas Nüchter, Joachim Hertzberg, and Hartmut Surmann. High-speed laser localization for mobile robots. *Robotics and Autonomous Systems*, 51(4) :275 – 296, 2005.

[LSD15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[LYMD13] J. Lu, H. Yang, D. Min, and M. N. Do. Patch match filter : Efficient edge-aware filtering meets randomized search for fast correspondence field estimation. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1854–1861, June 2013.

[MGB17] Tan Khoa Mai, Michèle Gouiffès, and Samia Bouchafa. Optical flow refinement using reliable flow propagation. In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer*

Graphics Theory and Applications - Volume 6 : VISAPP, (VISIGRAPP 2017), pages 451–458, 2017.

[MHG15] Moritz Menze, Christian Heipke, and Andreas Geiger. Discrete optimization for optical flow. In *German Conference on Pattern Recognition (GCPR)*, volume 9358, pages 16–28. Springer International Publishing, 2015.

[MHG18] Moritz Menze, Christian Heipke, and Andreas Geiger. Object scene flow. *ISPRS Journal of Photogrammetry and Remote Sensing (JPRS)*, 2018.

[MN01] M. Middendorf and H. H. Nagel. Estimation and interpretation of discontinuities in optical flow fields. In *Eighth IEEE International Conference on Computer Vision, 2001. ICCV 2001. Proceedings*, volume 1, pages 178–183 vol.1, 2001.

[MV07] Ezio Malis and Manuel Vargas. Deeper understanding of the homography decomposition for vision-based control. Research Report RR-6303, INRIA, 2007.

[NBCL06] Amaury Negre, Christophe Braillon, Jim Crowley, and Christian Laugier. Real-time Time-To-Collision from variation of Intrinsic Scale. In *Proc. of the Int. Symp. on Experimental Robotics*, Rio de Janeiro (BR), France, July 2006. voir basilic : http ://emotion.inrialpes.fr/bibemotion/2006/NBCL06/ address : Rio de Janeiro (BR).

[NBM12] Q. Nie, S. Bouchafa, and A. Merigot. Voting spaces cooperation for 3d plane detection from monocular image sequences. In *2012 3rd International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 135–140, Oct 2012.

[NG98] H.-H. Nagel and A. Gehrke. Spatiotemporally adaptive estimation and segmentation of OF-fields. In Hans Burkhardt and Bernd Neumann, editors, *Computer Vision — ECCV'98*, number 1407 in Lecture Notes in Computer Science, pages 86–102. Springer Berlin Heidelberg, June 1998. DOI : 10.1007/BFb0054735.

[Nis04] D. Nister. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6) :756–770, June 2004.

[NM02] Yao Nie and Kai-Kuang Ma. Adaptive rood pattern search for fast block-matching motion estimation. *IEEE Transactions on Image Processing*, 11(12) :1442–1449, Dec 2002.

[NNB04] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I, June 2004.

[NRB14] Nirmal Jith O. U., S. Avinash Ramakanth, and R. Venkatesh Babu. Optical flow based on Approximate Nearest Neighbor Field. pages 1–5. IEEE, January 2014.

[OG99] J. Oliensis and Y. Genc. New algorithms for two-frame structure from motion. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 737–744 vol.2, Sept 1999.

[OKK11] K. Okuyama, T. Kawasaki, and V. Kroumov. Localization and position correction for mobile robot using artificial visual landmarks. In *The 2011 International Conference on Advanced Mechatronic Systems*, pages 414–418, Aug 2011.

[ÖVBS17] Onur Özyesil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey on structure from motion. *CoRR*, abs/1701.08493, 2017.

[PLBC16] Aurélien Plyer, Guy Le Besnerais, and Frédéric Champagnat. Massively parallel lucas kanade optical flow for real-time video processing applications. *Journal of Real-Time Image Processing*, 11(4) :713–730, Apr 2016.

[PYLP11] Yanwei Pang, Yuan Yuan, Xuelong Li, and Jing Pan. Efficient hog human detection. *Signal Processing*, 91(4) :773 – 781, 2011.

[SBSR92] D. Sinclair, A. Blake, S. Smith, and C. Rothwell. Planar region detection and motion recovery. In David Hogg and Roger Boyle, editors, *BMVC92*, pages 59–68, London, 1992. Springer London.

[SCS+10] R. Szeliski, B. Curless, S. M. Seitz, N. Snavely, Y. Furukawa, and S. Agarwal. Reconstructing rome. *Computer*, 43 :40–47, 06 2010.

[SMB04] M. Singh, M. Mandal, and A. Basu. Robust KLT tracking with Gaussian and Laplacian of Gaussian weighting functions. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 4, pages 661–664 Vol.4, August 2004.

[SMR06] G. Silveira, E. Malis, and P. Rives. Real-time robust detection of planar regions in a pair of images. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Beijing, China, October 2006.

[SOCM01] H. S. Stone, M. T. Orchard, Ee-Chien Chang, and S. A. Martucci. A fast direct fourier-based algorithm for subpixel registration of images. *IEEE Transactions on Geoscience and Remote Sensing*, 39(10) :2235–2243, Oct 2001.

[SRB10] Deqing Sun, Stefan Roth, and Michael J. Black. Secrets of optical flow estimation and their principles. pages 2432–2439. IEEE, June 2010.

[SRB14] Deqing Sun, Stefan Roth, and Michael J. Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision (IJCV)*, 106(2) :115–137, 2014.

[SSB10] Deqing Sun, Erik B. Sudderth, and Michael J. Black. Layered image motion with explicit occlusions, temporal consistency, and depth ordering. In *NIPS*, 2010.

[ST94] Jianbo Shi and Tomasi. Good features to track. In *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, June 1994.

[SVB13] M. Stoll, S. Volz, and A. Bruhn. Joint trilateral filtering for multiframe optical flow. In *2013 IEEE International Conference on Image Processing*, pages 3845–3849, September 2013.

[SVB+17] Olivier Saurer, Pascal Vasseur, Rémi Boutteau, Cédric Demonceaux, Marc Pollefeys, and Friedrich Fraundorfer. Homography Based Egomotion Estimation with a Common Direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(2) :327–341, February 2017.

[SWK07] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient ransac for point-cloud shape detection. *Computer Graphics Forum*, 26(2) :214–226, June 2007.

[Sze10] Richard Szeliski. *Computer Vision : Algorithms and Applications*. Springer-Verlag, Berlin, Heidelberg, 1st edition, 2010.

[TNNL02] JD Tardós, J Neira, PM Newman, and J J Leonard. Robust mapping and localization in indoor environments using sonar data. 21(4) :311–330, 2002.

[TZ99] P. H. S. Torr and A. Zisserman. Feature Based Methods for Structure and Motion Estimation. In *Vision Algorithms : Theory and Practice*, pages 278–294. Springer, Berlin, Heidelberg, September 1999.

[UESC16] Vladyslav C. Usenko, Jakob Engel, Jörg Stückler, and Daniel Cremers. Direct visual-inertial odometry with stereo cameras. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1885–1892, 2016.

[VBW08] Levi Valgaerts, Andrés Bruhn, and Joachim Weickert. A Variational Model for the Joint Recovery of the Fundamental Matrix and the Optical Flow. In *Pattern Recognition*, pages 314–324. Springer, Berlin, Heidelberg, June 2008.

[VRS14] Christoph Vogel, Stefan Roth, and Konrad Schindler. *View-Consistent 3D Scene Flow Estimation over Multiple Frames*, pages 263–278. Springer International Publishing, Cham, 2014.

[VSR13] C. Vogel, K. Schindler, and S. Roth. Piecewise rigid scene flow. In *2013 IEEE International Conference on Computer Vision*, pages 1377–1384, Dec 2013.

[VSR15] Christoph Vogel, Konrad Schindler, and Stefan Roth. 3d scene flow estimation with a piecewise rigid scene model. *International Journal of Computer Vision*, 115(1) :1–28, Oct 2015.

[WB15] Jonas Wulff and Michael J. Black. Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[WC00] O. Wijk and H.I. Christensen. Localization and navigation of a mobile robot using natural point landmarks extracted from sonar data. *Robotics and Autonomous Systems*, 31(1) :31 – 42, 2000.

[WCPB09] Andreas Wedel, Daniel Cremers, Thomas Pock, and Horst Bischof. Structure- and motion-adaptive regularization for high accuracy optic flow. pages 1663–1668. IEEE, September 2009.

[WCYL12] Z. L. Wang, B. G. Cai, F. Z. Yi, and M. Li. Reviews on planar region detection for visual navigation of mobile robot under unknown environment. In Gary Lee, editor, *Advances in Automation and Robotics, Vol. 2*, pages 593–601, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[WDEH95] P. Weckesser, R. Dillmann, M. Elbs, and S. Hampel. Multiple sensor processing for high-precision navigation and environmental modeling with a mobile robot. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, volume 1, pages 453–458 vol.1, Aug 1995.

[Wei97] Y. Weiss. Smoothness in layers : Motion segmentation using nonparametric mixture estimation. In *, 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997. Proceedings*, pages 520–526, June 1997.

[WPZ+09] Andreas Wedel, Thomas Pock, Christopher Zach, Horst Bischof, and Daniel Cremers. An improved algorithm for tv-l1 optical flow. In Daniel Cremers, Bodo Rosenhahn, Alan L. Yuille, and Frank R. Schmidt, editors, *Statistical and Geometrical Approaches to Visual Motion Analysis*, pages 23–45, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[WRHS13] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. DeepFlow : Large displacement optical flow with deep matching. In *IEEE Intenational Conference on Computer Vision (ICCV)*, Sydney, Australia, December 2013.

[WSLB17] Jonas Wulff, Laura Sevilla-Lara, and Michael J. Black. Optical flow in mostly rigid scenes. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[XJM10] Li Xu, Jiaya Jia, and Yasuyuki Matsushita. Motion detail preserving optical flow estimation. pages 1293–1300. IEEE, June 2010.

[XOK90] Lei Xu, Erkki Oja, and Pekka Kultanen. A new curve detection method : Randomized hough transform (rht). *Pattern Recognition Letters*, 11(5) :331 – 338, 1990.

[yB00] Jean yves Bouguet. Pyramidal implementation of the lucas kanade feature tracker. *Intel Corporation, Microprocessor Research Labs*, 2000.

[YD97] Y. Yacoob and L. S. Davis. Temporal multi-scale models for flow and acceleration. In *, 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997. Proceedings*, pages 921–927, June 1997.

[YDB12] Yan Niu, A. Dick, and M. Brooks. Locally Oriented Optical Flow Computation. *IEEE Transactions on Image Processing*, 21(4) :1573–1586, April 2012.

[YgWGl12] Z. Yong-guo, C. Wei, and L. Guang-liang. The navigation of mobile robot based on stereo vision. In *2012 Fifth International Conference on Intelligent Computation Technology and Automation*, pages 670–673, Jan 2012.

[YL15] Jiaolong Yang and Hongdong Li. Dense, accurate optical flow estimation with piecewise parametric model. pages 1019–1027. IEEE, June 2015.

[ZBSL17] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017.

[ZH96] Zhongfei Zhang and Allen R. Hanson. 3d reconstruction based on homography mapping. In *In ARPA Image Understanding Workshop*, pages 0249–6399, 1996.

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Vers un système de vision artificielle opportuniste pour l'analyse de scènes complexes à partir de caméras embarquées

Mots clés : flot optique, détection de surfaces planaires, odométrie visuelle

Résumé : L'objectif de ces travaux de thèse consiste à proposer un système de vision pour l'analyse de scènes dynamiques dit « opportuniste » En ce sens, il est orienté vers une tâche applicative précise et profite des connaissances à priori fournies par l'application ainsi que des indices disponibles provenant de la scène perçue (couleur, texture, géométrie) selon leur pertinence.

Cette thèse propose de pousser jusqu'au bout la vision monoculaire basée sur l'analyse d'images issues d'une caméra embarquée sur un véhicule mobile. Dans un premier temps est proposée une nouvelle méthode d'estimation du flot optique, information directement estimable à partir d'une séquence d'images. L'approche se base sur la génération d'une carte de fiabilité pour raffiner le flot optique à travers un processus itératif profitant d'informations disponibles telles que la couleur. La carte de mouvement ainsi obtenue est ensuite exploitée pour une détection rapide des plans 3D principaux. Pour cela, une approche cumulative, appelée *uv*-velocité, exploitant les propriétés géométriques du champ des vecteurs de mouvement, est développée. Elle permet de détecter les surfaces planaires en partant d'hypothèses concernant la nature de l'ego-mouvement. Contrairement à l'approche déjà existante *c*-velocité, la méthode proposée permet une stratégie de vote plus progressive qui prend en compte plus de modèles d'ego-mouvement et plus de modèles de surfaces planaires

Le modèle de mouvement de chaque surface détectée est réintégré à la méthode d'estimation du flot optique qui devient une méthode d'optimisation sous contrainte de validité du modèle planaire afin d'améliorer la précision de l'estimation du flot optique. Par ailleurs, nous montrons dans cette thèse comment un processus d'odométrie visuelle peut tirer profit de la méthode de détection de surfaces planaires.

L'approche d'estimation du flot optique est évaluée en termes de précision et de temps d'exécution sur la base de données Middlebury. En ce qui concerne la *uv*-velocité, la validation est faite aussi bien sur des flots simulés que sur des images de la base de données de KITTI.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France

Title : Towards an opportunistic artificial vision system for analysing complex scenes from onboard cameras

Keywords : optical flow, plane segmentation, visual odometry

Abstract: The thesis intends to develop the bricks of an opportunistic vision system for dynamic scene analysis, an opportunistic system that would be guided by the applicative task, that would benefit from any knowledge and prioris made available by the application, and take profit of all available cues (color, texture, geometry) depending on their quality and relevance. The context of color monocular vision is considered, with a camera embedded on a mobile platform. A dense optical flow technique is first proposed. After a rough estimation, a reliability map is computed and is used for refining the motion map, through an iterative propagation process constrained by local information, starting by the color cues.
This motion map is then analyzed for rough and fast plane segmentation. A cumulative approach called *uv*-velocity has been developed. It allows the fast exhibition of prominent planar surfaces under certain assumption related the ego-motion. Contrary to its predecessor, the so-called *c*-velocity, it allows a more progressive voting strategy, it avoids using sampling, it is not limited to translations of the camera and can detect a wider range of surfaces..
The motion models related to each surface can then be re-injected as a constraint in the estimation of the next optical flow. The raw and fast planar segmentation produced by *uv*-velocity can be used to fasten the estimation visual odometry.
The results of optical flow estimation remain acceptable in terms of precision and execution time (tested on Middleburry dataset) which can be the input for creating the voting space to detect the planes on image. After the simulations and real experiments on KITTI dataset, *uv*-velocity shows its potential to be the polyvalent image registration on plane detection and opportunistic alert for the system.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France