On Modeling Ego-Motion Uncertainty for Moving Object Detection from a Mobile Platform

Dingfu Zhou, Vincent Fremont, Benjamin Quost, Bihao Wang

On Modeling Ego-Motion Uncertainty for Moving Object Detection from a Mobile Platform

Dingfu Zhou[1,2], Vincent Frémont[1,2] , Benjamin Quost[1,2] and Bihao Wang[1,2]

***Abstract*—In this paper, we propose an effective approach for moving object detection based on modeling the ego-motion uncertainty and using a graph-cut based motion segmentation. First, the relative camera pose is estimated by minimizing the sum of reprojection errors and its covariance matrix is calculated using a first-order errors propagation method. Next, a motion likelihood for each pixel is obtained by propagating the uncertainty of the ego-motion to the Residual Image Motion Flow (RIMF). Finally, the motion likelihood and the depth gradient are used in a graph-cut based approach as region and boundary terms respectively, in order to obtain the moving objects segmentation. Experimental results on real-world data show that our approach can detect dynamic objects which move on the epipolar plane or that are partially occluded in complex urban traffic scenes.**

I. Introduction and Related Work

Vision-based driver assistance system (DAS) is a complex and challenging task in urban traffic scenarios. In particular, moving object detection from dynamic scene analysis is essential for obstacle avoidance and path planning, and has numerous applications in autonomous and semi-autonomous driving. Indeed, being able to detect dynamic obstacles (vehicles, cyclists, or pedestrians) and to estimate their positions and motion tendency can increase the safety in both autonomous and semi-autonomous driving. Moving objects information also helps to improve the precision of Vision-based Simultaneous Localization and Mapping (VSLAM) and Structure-from-Motion (SfM) approaches which mainly rely on static environment assumptions [1].

Several vision-based motion detection approaches have been proposed over the last decade. Using one camera, approaches like background subtraction [2], adaptive background model [3] or optical flow measurement [4] can be used when the camera is static. More details about motion detection from a static camera can be found in [5]. The problem becomes much more complex when the camera and the surrounding objects move simultaneously. Indeed, the camera motion induces location changes of all the image pixels. Therefore, geometrical constraints are essential to distinguish between static and moving parts of the image. Two-view geometrical constraints (known as epipolar constraints) can be used to detect moving pixels. However it cannot detect objects moving on the epipolar plane (degenerate case). Other constraints, such as flow vector bound constraints [6] and multi-frame epipolar constraints [7], have been used to detect the objects moving on the epipolar plane. Using two cameras, a dense or sparse disparity map can be calculated to reconstruct 3D information of the environment [8]. By combining disparity information with feature tracking or optical flow computation, the 3D scene flow can be reconstructed and used to detect moving objects [9], [10].

Following [4], we propose to detect moving using the Residual Image Motion Flow (RIMF), which quantifies the difference between the measured optical flow and global image motion flow (pixel changes caused by camera motion). The global image motion flow between two consecutive frames can be determined by a function of the current scene depth and the relative camera pose. Regions with significant RIMF are detected as potential moving objects. In order to avoid a large number of false positives or misdetections, the noise in the RIMF estimation process should be considered. In [10] and [11], the uncertainties of the real optical flow and 3D scene flow have been modeled respectively to detect the moving objects. However, they just roughly modeled the uncertainty of the ego-motion information obtained from other sensors (GPS/IMU).

Unlike these methods, our approach is only based on two consecutive stereo images: no other sensor information is required. Furthermore we detail how the ego-motion uncertainty may be taken into account so as to improve the RIMF computation. Therefore, in this paper we propose a moving object detection approach based on [12] with two main contributions. First, a first-order error propagation framework is used to take into account the ego-motion imprecision which result from the uncertainty in feature extraction and matching. The uncertainty propagation strategy is also applied in the RIMF computation to build a motion likelihood for each possible image pixel (when its disparity value is available). This information quantifies the likelihood for a pixel to be moving or not. Then, a segmentation of the moving objects is performed using graph-cuts based motion segmentation on the motion likelihood and depth information, which helps to reduce the noise in the local optical flow estimation process.

Our paper is organized as follows: first, we present the overview of our approach in Section II. Section III introduces the key steps of our moving object detection algorithm, including ego-motion estimation and uncertainty computation, motion likelihood calculation and graph-cuts based motion segmentation. Simulation experiments to test the uncertainty estimation and real experiments results for moving object detection are presented in Section IV. Finally, the paper ends with conclusions and future work.

II. System Overview

Fig.1 outlines the main steps of our moving object detection system based on two consecutive stereo image pairs. The dense disparity map at frame $t-1$ and the optical flow (dense or sparse) between frame $t-1$ and t are estimated

The authors are with [1]*Université de Technologie de Compiègne (UTC)*, [2]*CNRS Heudiasyc UMR* 7253*, France.* E-mail: {dingfu.zhou, vincent.fremont, benjamin.quost, bihao.wang}@hds.utc.fr

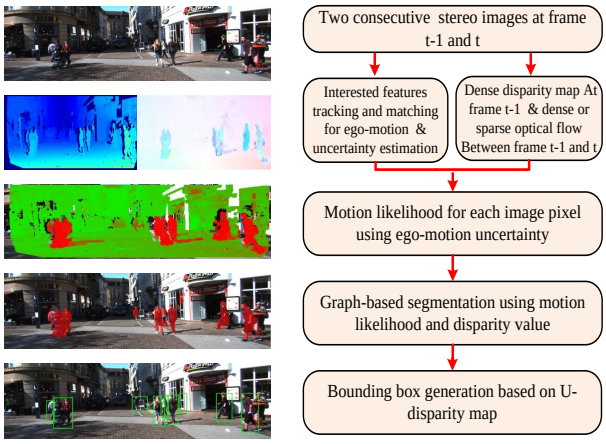


Figure 1: **Moving objects segmentation framework overview.**

respectively. At the same time, the feature points tracked and matched between two consecutive stereo image pairs are used to estimate the camera relative pose and its uncertainty. Then, the global motion flow (image changes caused by the camera ego-motion) is calculated for each image pixel using the camera motion and the depth information obtained from the disparity map. The RIMF, which is used to measure the difference between the measured optical flow and the global image motion flow, is used to distinguish between moving and non-moving pixels. In order to handle the noise involved in the RIMF calculation, the covariance matrix of RIMF is used to estimate the motion likelihood for each image pixel. It is computed using a first-order strategy to propagate the uncertainty from both disparity and ego-motion estimation procedures to the RIMF estimation. Using the motion likelihood and disparity values, a graph-cut based motion segmentation approach is then applied to segment the image into regions of moving and non-moving pixels. Finally, the bounding boxes of the moving objects are obtained by combining region-growing and U-disparity map information.

III. Moving Object Detection Framework

Two successive stereo frames are considered in the motion detection procedure. We assume that the stereo rig undergone an unconstrained motion ($\mathbf{R}$, $\mathbf{T}$) between the two successive frames, where $\mathbf{R}$ and $\mathbf{T}$ are respectively the rotational and translational components of the motion. The left image at time $t-1$ is considered as the reference image. A 3D point P at frames $t-1$ and t is noted as $P_{t-1}(X_{t-1}, Y_{t-1}, Z_{t-1})$ and $P(X_t, Y_t, Z_t)$ respectively. The selected 2D points (x^L_{t-1}, y^L_{t-1}), (x^R_{t-1}, y^R_{t-1}), (x^L_t, y^L_t), (x^R_t, y^R_t) are matched and tracked [13] (in the left and right images at time $t-1$ and t (a bucketing technique is used to ensure that features points are well spread in the whole image regions). Assuming the origin of 3D coordinate system is coincident with the left camera center, the 3D world points can be obtained as follow:

$$(X_{t-1}, Y_{t-1}, Z_{t-1})^T = \frac{b}{d}(x^L_{t-1} - u_0, y^L_{t-1} - v_0, f)^T \quad (1)$$

where $d = x^L_{t-1} - x^R_{t-1}$ is the disparity value for the scanline $y = y^L_{t-1}$. The variables f, b and (u_0, v_0) are the camera intrinsic parameters known as the focal length, the baseline and the principal point coordinates.

A. Ego-motion Estimation and Uncertainty Computation

1) Ego-motion Estimation : Given the points matching in four images for two consecutive frames, the relative pose of the camera can be estimated by minimizing the sum of the reprojection errors using non-linear minimization approaches. First, the feature points from the previous frame are reconstructed in 3D via triangulation and using the camera intrinsic parameters. Then these 3D points are re-projected into current image frames using the camera motion as below:

$$\hat{\mathbf{x}}^i_t = \mathbf{f}(\mathbf{\Theta}, \mathbf{x}^i_{t-1}) = \begin{bmatrix} Pr^L(K\,[\mathbf{R}|\mathbf{T}]\,P^i_{t-1}) \\ Pr^R(K\,[\mathbf{R}|\mathbf{T}]\,P^i_{t-1}) \end{bmatrix} \quad (2)$$

where $\hat{\mathbf{x}}_{t,i} = (\hat{x}^L_{t,i}, \hat{y}^L_{t,i}, \hat{x}^R_{t,i}, \hat{y}^R_{t,i})^T$ and $\mathbf{x}_{t-1,i} = (x^L_{t-1,i}, y^L_{t-1,i}, x^R_{t-1,i}, y^R_{t-1,i})^T$ are the predicted and measured image points at image t and $t-1$ respectively. The vector $\mathbf{\Theta} = (r_x, r_y, r_z, T_z, T_y, T_z)^T$ represents the six degrees of freedom of the relative pose. Let Pr^L and Pr^R be the image projections of the 3D world points into the left and right image (non-homogeneous coordinates). Let P^i_{t-1} be the 3D point in the previous frame, which is calculated using Eq. (1). The parameter vector $\mathbf{\Theta}$ can be estimated using the minimization of the following cost function which is built using the geometric distance of the predicted and measured image points in time t as:

$$F(\mathbf{\Theta}, \mathbf{x}) = \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2_{\Sigma_{\mathbf{x}_t}} = \|\mathbf{x}_t - \mathbf{f}(\mathbf{\Theta}, \mathbf{x}_{t-1})\|^2_{\Sigma_{\mathbf{x}_t}} \quad (3)$$

where $\|\ .\ \|^2_\Sigma$ denotes the squared Mahalanobis distance according to the covariance matrix Σ. A Gaussian-Newton iterative optimization method is used to solve the optimization problem presented in Eq. (3).

2) Error Propagation: For vision systems, robust motion estimation should not only provide an estimate of the camera motion, but also an estimate of the uncertainty associated with this solution. Let the i_{th} loop matched features be $(x^L_{t-1,i}, y^L_{t-1,i}), (x^R_{t-1,i}, y^R_{t-1,i}), (x^L_{t,i}, y^L_{t,i}), (x^R_{t,i}, y^R_{t,i}), i = 1, \cdots, N$ in stereo image pairs in $t-1$ and t frames. By stacking these features in vectors, new vectors can be defined as: $\mathbf{x} \in \mathbb{R}^{8N}$ represents all the features and $\mathbf{x}_t \in \mathbb{R}^{4N}$, $\mathbf{x}_{t-1} \in \mathbb{R}^{4N}$ represent the features at time t and $t-1$ respectively. To be robust against outliers (mismatched features or features on moving objects), a RANSAC strategy is applied to estimate the relative pose between the two successive frames. Assuming that all the inliers used for the final minimization of Eq. (3) are good matched features with only additive Gaussian noise, the associated probability density function has the following form:

$$\mathbf{x} \sim \boldsymbol{N}\left(\begin{bmatrix} \mu_{\mathbf{x}_{t-1}} \\ \mu_{\mathbf{x}_t} \end{bmatrix} , \begin{bmatrix} \Sigma_{\mathbf{x}_{t-1}} & 0 \\ 0 & \Sigma_{\mathbf{x}_t} \end{bmatrix} \right) \quad (4)$$

where, μ_t and Σ_t (respectively, μ_{t-1} and Σ_{t-1}) are the mean and the covariance of the image features at time t (resp. $t-1$). Then the parameters accuracy only depends on the precision of the detected feature locations in the image plane.

In [14] and [15], the covariance matrix of the estimated parameters which considers the uncertainty of $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$ respectively, can be obtained with the following model:

$$\Sigma_\Theta = (\frac{\partial g}{\partial \mathbf{\Theta}})^{-1}(\frac{\partial g}{\partial \mathbf{x}})^T \Sigma_{\mathbf{x}} \frac{\partial g}{\partial \mathbf{x}}(\frac{\partial g}{\partial \mathbf{\Theta}}^{-1})^T \tag{5}$$

where, $g(\mathbf{x}, \mathbf{\Theta}) = \frac{\partial F(\mathbf{X},\mathbf{\Theta})}{\partial \mathbf{\Theta}}$, is the partial derivative of $F(\mathbf{\Theta}, \mathbf{x})$ w.r.t each component of Θ. The matrix $\Sigma_{\mathbf{x}}$, which has been defined in Eq. (4), is the covariance matrix of the measured features at time $t-1$ and t.

In order to get the partial derivatives, $\frac{\partial g}{\partial \Theta}$ and $\frac{\partial g}{\partial \mathbf{x}}$ of $g(\mathbf{\Theta}, \mathbf{x})$ .w.r.t each components of Θ and $\mathbf{x}$, and since $\frac{\partial g}{\partial \Theta}$ and $\frac{\partial g}{\partial \mathbf{X}}$ are parts of the Hessian of $F(\mathbf{\Theta}, \mathbf{x})$, we have:

$$\frac{\partial g}{\partial \mathbf{\Theta}} = \frac{\partial^2 \mathbf{F}}{\partial \mathbf{\Theta}^2} = 2(\frac{\partial \mathbf{f}}{\partial \mathbf{\Theta}})^T \Sigma_{\mathbf{x}_t}^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{\Theta}} - 2\frac{\partial^2 \mathbf{f}}{\partial \mathbf{\Theta}^2} \Sigma_{\mathbf{x}_t}^{-1}(\mathbf{x}_t - \mathbf{f}(\mathbf{\Theta}, \mathbf{x}_{t-1})) \tag{6}$$

$$\frac{\partial g}{\partial \mathbf{x}} = \frac{\partial^2 \mathbf{F}}{\partial \mathbf{\Theta} \partial \mathbf{x}} = \begin{bmatrix} \frac{\partial^2 \mathbf{F}}{\partial \mathbf{\Theta} \partial \mathbf{x}_c} \\ \frac{\partial^2 \mathbf{F}}{\partial \mathbf{\Theta} \partial \mathbf{x}_{t-1}} \end{bmatrix} = \begin{bmatrix} -2\Sigma_{\mathbf{x}_t}^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{\Theta}} \\ \frac{\partial^2 \mathbf{F}}{\partial \mathbf{\Theta} \partial \mathbf{x}_{t-1}} \end{bmatrix} \tag{7}$$

$$\frac{\partial^2 \mathbf{F}}{\partial \mathbf{\Theta} \partial \mathbf{x}_{t-1}} = 2(\frac{\partial \mathbf{f}}{\partial \mathbf{x}_{t-1}})^T \Sigma_{\mathbf{x}_t}^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{\Theta}} - 2\frac{\partial^2 \mathbf{f}}{\partial \mathbf{\Theta} \partial \mathbf{x}_{t-1}} \Sigma_{\mathbf{x}_t}^{-1}(\mathbf{x}_t - \mathbf{f}(\mathbf{\Theta}, \mathbf{x}_{t-1})) \tag{8}$$

Equation. (6) and (8) are both a sum of two terms, a first part with only first-order derivatives and a second part which is the product of second-order derivatives and the residual part $\mathbf{x}_t - \mathbf{f}(\mathbf{\Theta}, \mathbf{x}_{t-1})$. If the final solution of Eq. (3) has a zero residual or very small residual, one can remove the second-order derivatives parts from Eq. (6)-(8) to get a first-order estimate as below:

$$\Sigma_{\mathbf{\Theta}} = (\frac{\partial \mathbf{f}}{\partial \mathbf{\Theta}}^T \Sigma_{\mathbf{X}_c}^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{\Theta}})^{-1} + A^T \Sigma_{\mathbf{X}_p} A \tag{9}$$

where,

$$A = \frac{1}{2}\left(\frac{\partial \mathbf{f}}{\partial \mathbf{\Theta}}^T \Sigma_{\mathbf{X}_c}^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{\Theta}}\right)^{-1} (\frac{\partial \mathbf{f}}{\partial \mathbf{\Theta}})^T \Sigma_{\mathbf{X_c}}^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{X}_p}$$

Note that using only the first right-hand term in Eq. (9) is called as partially method (only the current frame feature noise has been considered) which has been described in [16].

However, due to the noise of the measured features in the previous and the current frames, the residuals of the final solution of Eq. (3) may not be zero. So a second-order error propagation model which considers residuals will be more suitable. A second-order error propagation result can be obtained by substituting Eq. (6,8) into Eq. (5).

B. Residual Image Motion Flow

Given a pixel position $\mathbf{x}_{t-1}$, it is possible to compute the image position of pixel $\hat{\mathbf{x}}_t$ at time t using the following equation [16]:

$$\hat{\mathbf{x}}_t = \mathbf{K}\mathbf{R}\mathbf{K}^{-1}\mathbf{x}_{t-1} + \frac{\mathbf{K}\mathbf{t}}{z} \tag{10}$$

where $\mathbf{x}_{t-1}$ and $\hat{\mathbf{x}}_t$ are the normalized homogeneous image pixels at time $t-1$ and t respectively. The matrix $\mathbf{K}$ encapsulates the camera intrinsic parameters, z is the depth of the image pixel $\mathbf{x}_{t-1}$. Normalizing Eq. (10) and substituting $z = \frac{bf}{d}$ into Eq. (10), the following equation can be obtained:

$$\begin{cases} x_t = \frac{r_{00}fb(x_{t-1}-u_0)+r_{01}fb(y_{t-1}-v_0)+r_{02}f^2b+dfT_x}{r_{20}b(x_{t-1}-u_0)+r_{21}b(y_{t-1}-v_0)+r_{22}fb+dT_z} + u_0 \\ y_t = \frac{r_{10}fb(x_{t-1}-u_0)+r_{11}fb(y_{t-1}-v_0)+r_{12}f^2b+dftT_y}{r_{20}b(x_{t-1}-u_0)+r_{21}b(y_{t-1}-v_0)+r_{22}fb+dT_z} + v_0 \end{cases} \tag{11}$$

where $r_{ij}, i, j = 1, 2, 3$ are the rotation matrix coefficient in line-column format. For image pixels from 3D static points, the global image motion flow caused by the camera motion only can be expressed as:

$$\begin{pmatrix} u_x \\ v_y \end{pmatrix} = \begin{pmatrix} x_t - x_{t-1} \\ y_t - y_{t-1} \end{pmatrix} \tag{12}$$

However, for moving objects, the image motion is caused by both their motion and the camera displacement. Assuming that the real optical flow estimated from the image between time t and $t-1$ is (u'_x, v'_y), it is possible to define the RIMF as:

$$\overrightarrow{p_{(m)}} = (u_x - u'_x, v_y - v'_y)'_{(m)} \tag{13}$$

C. Motion Likelihood Estimation

Once the RIMF has been computed using Eq. (13), it can be used to separate the image pixels into moving or non-moving parts. Comparing the absolute RIMF difference $|\overrightarrow{p_{(m)}}|$ to a fixed threshold does not lead to a satisfying result to differentiate moving pixels from static ones. Points with different 3D world locations have different image motions. Also, the estimation uncertainty, e.g. camera motion and pixel depth, have different influences on the image points. Ignoring these uncertainties could lead to a large number of false positives. In our case, the uncertainty in the RIMF is propagated from the image pixels noise to the final estimation using a first order Gaussian approximation.

As in Eq. (13), the RIMF is a function of the camera motion $\mathbf{\Theta}$, the pixel location (x_{t-1}, y_{t-1}) in the previous frame, the depth of its corresponding 3D point d and the measured optical flow (u'_x, v'_y). The uncertainty of the measured optical flow will not be considered in this work because it only affects the detection result locally. However, a linear approximation of the RIMF covariance can be calculated as:

$$\Sigma_{RIMF} = \mathbf{J}\mathbf{C}\mathbf{J}^T$$

where $\mathbf{J}$ represents the Jacobian matrix with respect to the camera motion $\mathbf{\Theta}$, the pixel position (x_{t-1}, y_{t-1}) in the previous frame and the disparity value d in previous frame, and $\mathbf{C}$ is the covariance matrix of all the input variables:

$$\mathbf{J} = \begin{pmatrix} \frac{\partial u_x}{\partial r_x} & \frac{\partial u_x}{\partial r_y} & \frac{\partial u_x}{\partial r_z} & \frac{\partial u_x}{\partial t_x} & \frac{\partial u_x}{\partial t_y} & \frac{\partial u_x}{\partial t_z} & \frac{\partial u_x}{\partial x} & \frac{\partial u_x}{\partial y} & \frac{\partial u_x}{\partial d} \\ \frac{\partial v_x}{\partial r_x} & \frac{\partial v_x}{\partial r_y} & \frac{\partial v_x}{\partial r_z} & \frac{\partial v_x}{\partial t_x} & \frac{\partial v_x}{\partial t_y} & \frac{\partial v_x}{\partial t_z} & \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} & \frac{\partial v_x}{\partial d} \end{pmatrix}$$

,

$$\mathbf{C} = \begin{pmatrix} \Sigma_\Theta & \mathbf{0} \\ \mathbf{0} & \Sigma_1 \end{pmatrix}$$

with Σ_Θ the covariance matrix associated to the camera motion estimated in Sec.III-A and $\Sigma_1 = diag\left(\sigma_x^2, \sigma_y^2, \sigma_d^2\right)$. The variances σ_x and σ_y describe the image noise coming from features detection. As in [10], the disparity uncertainty can be considered as an approximate standard Gaussian Distribution and the variance can be approximated by a linear function,

$$\sigma_d(x,y) = \sigma_0 + \gamma U_d(x,y) \tag{14}$$

where σ_0 and the γ are two constants parameters and $U_d(x,y)$ is the uncertainty on the disparity value at position (x,y). Here, the matching cost is used as a confidence measure of the disparity value (further details can be found in [17]).

Assuming a stationary world and Gaussian error propagation, static pixels can be expected to follow a Gaussian distribution with zero mean and covariance matrix Σ_{RIMF}. The Mahalanobis distance associated to the RIMF can be calculated as follows:

$$\mu_{p_{(m)}} = \sqrt{p_m^T \Sigma_{RIMF}^{-1} p_m} \tag{15}$$

Since $\mu^2_{p_{(m)}}$ is χ^2 distributed, the motion likelihood $\xi_{motion}(x)$ of the RIMF of each pixel can be computed according to its $\mu_{p_{(m)}}$ value.

D. Graph-based Segmentation

Graph-cut (GC) is an energy minimization framework widely used in image segmentation. Further details may be found in [18] [19]. Usually independent moving objects are layered in the depth direction in real traffic scenes and may thus easily be distinguished from the neighboring background. Here, GC is applied to depth images in order to refine the results obtained from likelihood motion estimation. In particular, we detail how to build our cost function for segmentation, which is composed of two terms related to local and contextual segmentation:

$$E(L) = E_r(L) + \lambda E_b(L)$$

where $L = \{l_1, l_2, \cdots, l_p\}$ is a binary vector, $l_i \in \{0,1\}$ is the label of the pixel (1 if moving, and 0 if static). Here, E_r and E_b are called the regional and boundary terms, respectively. The former term $E_r(L)$, ensures that pixels with a high motion likelihood are to be detected as moving; the latter, that adjacent pixels with similar depths should share the same label. The parameter λ is used as a trade-off between both terms.

1) Region Term: The motion likelihood of each pixel can be used directly to build the region term E_r of the energy function.

$$E_r = -\sum_{\Omega}\{L(x)\xi_{motion}(x) + (1 - L(x))\xi_{static}(x)\}$$

where Ω represents all the image field and ξ_{static} is a fixed prior likelihood to describe a point to be static. We set $\xi_{static} = 0.5$ in our experiments.

2) Boundary Term : Usually, the depth maps can be used for object segmentation [20] because the depth of objects is significantly different from the background. Therefore, we propose to use the depth $de(x) = \frac{bf}{D(x)}$ (D is the disparity value) to measure the similarity between two pixels x_i and x_j as follows:

$$B(x_i, x_j) = exp(-\sqrt{2}(|de(x_i) - de(x_j)|))$$

Then, the boundary term can be expressed as below:

$$E_b = \sum_{\Omega} \sum_{\hat{x} \in N_4(x)} B(x_i, x_i)|L(\hat{x}) - L(x)|$$

where $N_4(x)$ is the 4-neighborhood of a pixel x. In order to improve the segmentation efficiency, down-sampling is used in the motion likelihood estimation and moving objects segmentation steps, retaining one pixel out of four in both dimensions of the image.

E. Bounding Box Generation and Verification

After the segmentation step detailed above, bounding boxes should be generated for every moving object before performing tracking or recognition. Note that errors may come from partial detection (e.g., legs or arms of pedestrians) or redundancies (such as shadows). Object verification and region growing may be used to remove redundancies and to integrate parts detection using the dense disparity map. The U- and V- disparity maps [21] are two variants of the disparity map that are widely used for road and obstacle detection. In the U-disparity map, an upright object will form a horizontal line because of the same disparity value. This information may be used to obtain the width of the bounding box. Then, region growing can be applied to get the height of the bounding box from the disparity map. According to [22], the real world height of the objects could be estimated as below:

$$h_i = h_c + \frac{(y_i - y_0) z \cos\theta}{f} \tag{16}$$

Here, h_i and h_c are the height of object i and camera respectively in real world coordinate; θ is the camera tilt angle and f is the camera focal length; z is the depth of the object to the camera; y_0 and y_i are the horizon position and top of the objects in image coordinate (the origin of coordinates is assumed at left bottom). Assuming that moving objects are not higher than 2.5 meters, some obvious false positives may be filtered. For this purpose, the horizontal position is first computed using the V-disparity map. Then, the actual height of the object h_i is calculated using Eq. (16). Finally we retain only the objects which height is between 0.5m and 2.5m.

IV. Experiments

A. Motion Uncertainty Estimation

1) Monte Carlo Experiments : In our simulation experiments, both intrinsic and extrinsic parameters of the stereo rig are assumed to be known. The relative pose of the stereo cameras between two successive frames is fixed before generating the image features. We generate 3D points from a uniform distribution. Then they are projected into the four images using the appropriate projection matrices. We use bucketing techniques in order to ensure that features points are well spread in all the image regions.

A Monte-Carlo-like experiment is used to obtain an estimate of the covariance matrix as ground truth. At each time, the measured features are generated using Eq. (4) are used as inputs in Eq. (3) to obtain the optimal parameters in $\mathbf{\Theta}$. Covariance matrices of the Monte-Carlo method can be calculated from N independently estimated $\mathbf{\Theta}$. We set $N = 500$ in our experiments.

Simulation experiments have been conducted to compare the performance of the different approaches. Fig. 2 clearly

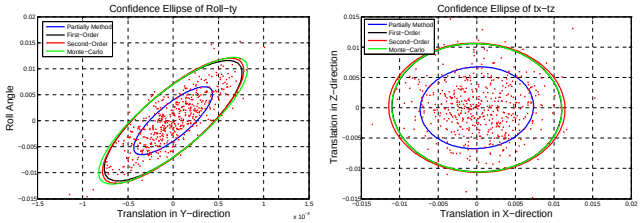


Figure 2: **Motion Uncertainty estimates**. Covariances between (a) t_y and r_x and (b) t_x and t_y, estimated using Monte-Carlo (green), first- and second-order techniques (black and red) using Eq. (5) respectively, and partially method (blue) using in [16].

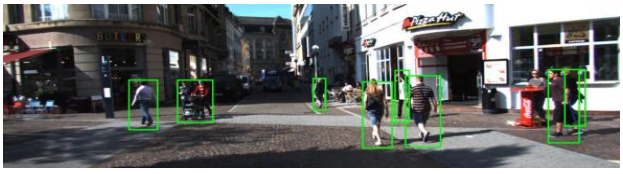
shows that the first and second-order methods proposed in Sec.III-A perform better than the classical partially method technique, the latter being slightly superior to the former; both perform almost as well as the Monte-Carlo approximation. For the sake of computational efficiency, we used first-order method in all our experiments.

B. Moving Object Detection

(a) Motion likelihood image

(b) Segmentation result using a fixed threshold

(c) Graph-based segmentation result using depth



(d) Bounding boxes generation and verification

Figure 3: **Moving Object Detection Flow .** (a) motion likelihood (red: moving, green: static). (b) Segmentation using a fixed threshold $\xi_{motion} = 0.75$. (c) Graph-based segmentation using depth information, We choose $\lambda = 1$ and $\xi_{static} = 0.5$ for all our experiments. (d) Final result after bounding box generation and verification.

In order to validate our moving object detection approach, we used the video sequences from the KITTI dataset [23] with a resolution of 1240×370 pixels and 10 frames per second. The video sequences are acquired from a SVS installed on the roof of a vehicle. Details about the setup can be found in [23]. Five video sequences acquired in different road situations were used to test our moving object detection technique. In the inner city sequence, the host vehicle was driven at a low speed (15 km/h) because of the complex road conditions. In the suburban road, the speed went up to 50 km/h. First, the stereo disparity map [24] and optical flow (dense [25] or sparse [13]) are computed before the moving object detection step. At the same time, the relative pose of the camera between two consecutive frames and its covariance are estimated as mentioned in III-A, and we assumed that the covariance of the features in Eq. (4) is 0.5 pixel. To compute the variance of disparity in Eq .(14), we set $\sigma_0 = 0.25$ and $\gamma = 0.075$ empirically.

Figure 4: **Detection results on a campus sequence.**

Fig.3 shows the key steps of our moving object detection approach. Fig.3(a) is the motion likelihood image, the stationary and moving parts are respectively displayed in green and red. Fig.3(b) and 3(c) show two detections results based on a fixed threshold and graph-based method. From the results, we can see that our graph-cut-based approach performs better to detect moving objects than when using a fixed threshold. Note that the verification in the bounding box generation step allowed to avoid a false positive detection. Fig.4 shows the detection results in a campus sequence. During this sequence, the camera turned from left to right at a high speed. Our algorithm proved to be efficient in this context too, allowing to detect the cyclists behind the trees to the left. We also tested our algorithm on a suburban road; the detection results are displayed in Fig.5. In this sequence, both the camera and the object vehicle move at urban speed (about 50km/h). In this sequence, we used sparse instead of dense optical flow because of the high changes in the images between two successive frames. The opposite driving vehicles were detected at a range of 40m, which remains sufficient for an appropriate reaction of the driver. The white car moving in front of the camera was also properly detected even if it moves in the same direction that

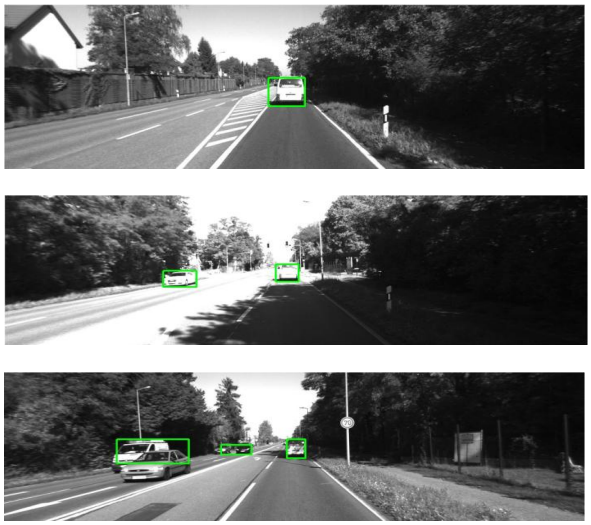


Figure 5: **Detection results on a suburban road.**

True moving (True positive)	False moving (False positive)	False static (False negative)	True static (True negative)
221	55	34	n/a

Table I: Performance of our moving object detection and segmentation approach

the ego vehicle. Fig.6 displays the results obtained on three inner city sequences. In crowded streets, the host vehicle moved slowly, which makes detecting moving objects easier, even when they move on the epipolar plane. Note that the algorithm also detected partially occluded objects because of the use of a dense approach. Despite the effectiveness of our algorithm, some false negative and false positive detections happen in the real image sequences. In Fig. 4, a cyclist (red elliptical box) has not been detected because the related 3D information cannot be reconstructed in this frame (the cyclist can not be seen in right camera). False positives also appeared, such as in Fig. 6 (a) with red bounding box, due to reflections on windows in the scene.

Table I describes our detection results in one inner city sequence with 153 frames (Fig. 6 (b)). The ground truth of the object's locations have been included in the dataset for this sequence. Here, only the moving objects whose distance is less than 30m are considered. The detection result is only based on two adjacent frames. True moving represents the number of detected moving object bounding boxes in the whole sequence. False moving means static objects that have been detected as moving and false static are the moving objects that are not detected. The true static objects are not taken into account because our algorithm focuses on moving objects only. In this sequence, our algorithm obtains a precision of 79.5% along with a recall of 86.7%.

All the experiments have been realized on a standard laptop (Intel Core i7) with Matlab R2013a processing environment. When the dense optical flow is used, the total average computational time is about 30 seconds for each frame. The optical flow calculation step takes about 15 seconds. Around 4.5 seconds is spent on the motion likelihood computation, 5 seconds on the graph-cut based segmentation and 5 seconds on the bounding boxes generation. Computing ego-motion and estimating the uncertainty only takes about 0.2 seconds. Although our Matlab implementation is not real-time, it compares favorably with respect to [26] (7 minutes per frame) and further accelerations could be achieved by C/C++ implementation with parallel/GPU computing.

V. Conclusions and Future Work

In this paper, a novel approach has been proposed to detect moving objects from two consecutive stereo frames by modeling the ego-motion uncertainty and using a graph-cut based motion segmentation. The ego-motion uncertainty estimated through a first-order error propagation model is used to obtain the motion likelihood for each image pixel. Pixels with a high motion likelihood and a similar depth are detected as a moving object based on a graph-cut motion segmentation approach. Additionally, a fast recognition of moving objects becomes possible based on our segmentation results. Detection results in several different real video sequences show that our proposed algorithm is highly robust with respect to global (camera motion) and local (optical flow) noise. The ego-motion error has been considered using its covariance matrix and uncertainties in optical flow can be eliminated by graph-cut segmentation procedure. Furthermore, our approach works with all image pixels and arbitrarily moving objects (including partially occluded) can be detected.

Future work will be firstly to consider a robust multiple objects tracking, using for example a PHD Filter, to obtain a stable detection results by reducing false positive detections. Further, fusing detection results coming from other sensors (lidar or radar) will also be tested to improved the detection results. Furthermore, categories information like pedestrian, car or others about the moving objects can be used on each bounding box to focus on moving pedestrians only.

References

[1] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localization and mapping (SLAM): part II. *Robotics & Automation Magazine, IEEE*, 13:108–117, 2006.

[2] Yaser Sheikh, Omar Javed, and Takeo Kanade. Background subtraction for freely moving cameras. In *ICCV*, pages 1219–1225, 2009.

[3] Alberto Faro, Daniela Giordano, and Concetto Spampinato. Adaptive background modeling integrated with luminosity sensors and occlusion processing for reliable vehicle detection. *Intelligent Transportation Systems, IEEE Transactions on*, 12(4):1398–1412, 2011.

[4] Ashit Talukder and Larry Matthies. Real-time detection of moving objects from moving vehicles using dense stereo and optical flow. In *IROS*, volume 4, pages 3718–3725, 2004.

[5] Jacinto C. Nascimento and Jorge S. Marques. Performance evaluation of object detection algorithms for video surveillance. *Multimedia, IEEE Transactions on*, 8:761–774, 2006.

[6] Abhijit Kundu, K. Madhava Krishna, and Jayanthi Sivaswamy. Moving object detection by multi-view geometric techniques from a single camera mounted robot. In *IROS*, pages 4306–4312, 2009.

[7] Soumyabrata Dey, Vladimir Reilly, Imran Saleemi, and Mubarak Shah. Detection of independently moving objects in non-planar scenes via multi-frame monocular epipolar constraint. In *ECCV*, pages 860–873. 2012.

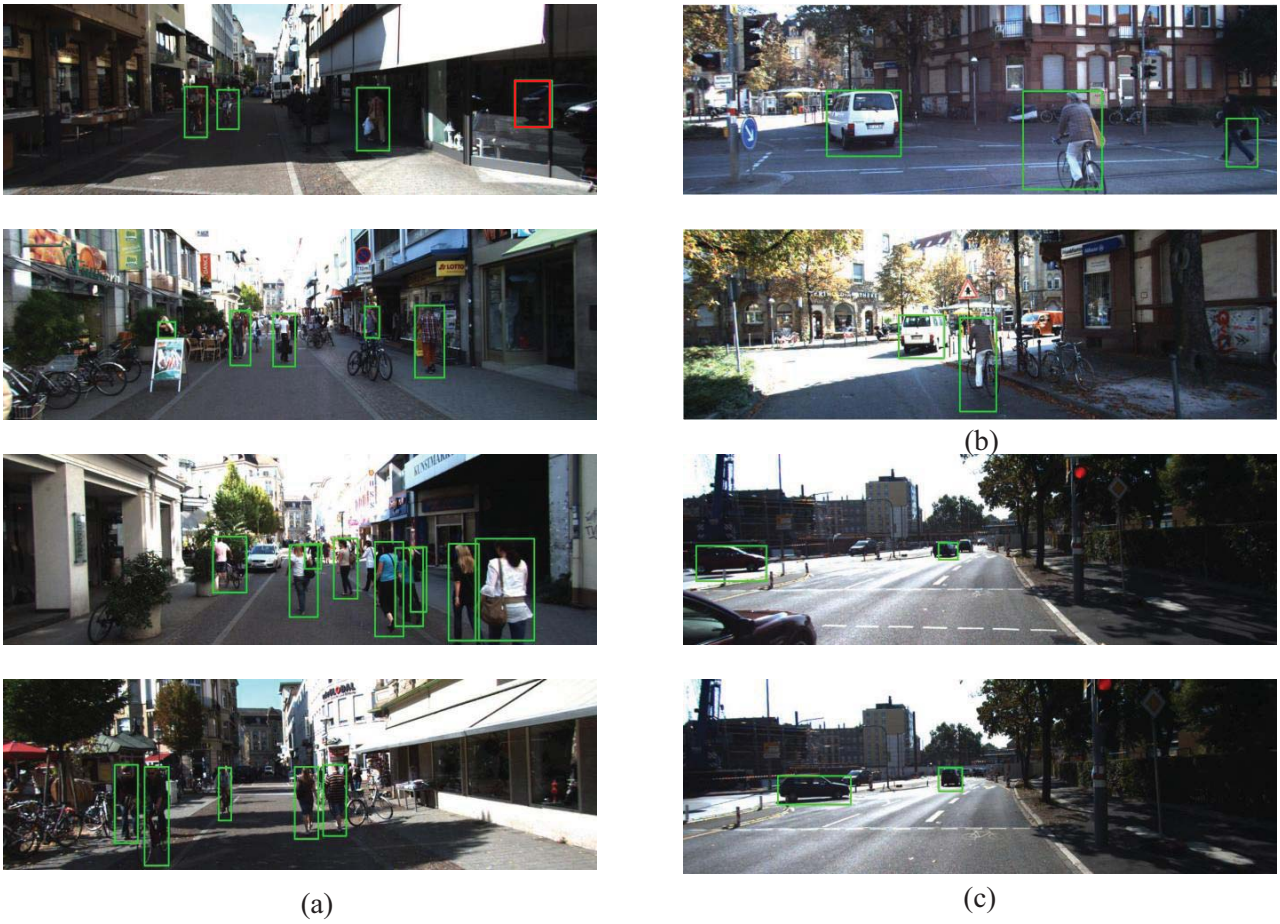


Figure 6: **Detection results on three different inner city sequences.** (a) crowded street (b) relatively simple environment (c) intersection

[8] Heiko Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *CVPR*, volume 2, pages 807–814, 2005.

[9] Philip Lenz, Julius Ziegler, Andreas Geiger, and Martin Roser. Sparse scene flow segmentation for moving object detection in urban environments. In *IV*, pages 926–932, 2011.

[10] Andreas Wedel. *Stereo scene flow for 3D motion analysis*. Springerverlag London Limited, 2011.

[11] Victor Romero-Cano and Juan I. Nieto. Stereo-based motion detection and tracking from a moving platform. In *IV*, pages 499–504, 2013.

[12] Dingfu Zhou, Vincent Fremont, Benjamin Quost, et al. Moving objects detection and credal boosting based recognition in urban environments. *CIS & RAM*, pages 24–29, 2013.

[13] Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *IV*, pages 963–968, 2011.

[14] John C. Clarke. Modelling uncertainty: A primer. *University of Oxford. Dept. Engineering science, Tech. Rep*, 2161:98, 1998.

[15] Robert M Haralick. Propagating covariance in computer vision. *International journal of pattern recognition and artificial intelligence*, 10(05):561–572, 1996.

[16] Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[17] Xiaoyan Hu and Philippos Mordohai. A quantitative evaluation of confidence measures for stereo vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2121–2133, 2012.

[18] Yuri Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in ND images. In *ICCV*, volume 1, pages 105–112, 2001.

[19] Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient ND image segmentation. *IJCV*, 70:109–131, 2006.

[20] Antonio HernÃ¡ndez-Vela, Nadezhda Zlateva, Alexander Marinov, Miguel Reyes, Petia Radeva, Dimo Dimov, and Sergio Escalera. Graph cuts optimization for multi-limb human segmentation in depth maps. In *CVPR*, pages 726–732, 2012.

[21] Zhencheng Hu and Keiichi Uchimura. UV-disparity: an efficient algorithm for stereovision based scene analysis. In *IV*, pages 48–54, 2005.

[22] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Putting objects in perspective. *IJCV*, 80:3–15, 2008.

[23] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, pages 3354–3361, 2012.

[24] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient large-scale stereo matching. In *ACCV*, pages 25–38. 2011.

[25] Ce Liu. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Massachusetts Institute of Technology, 2009.

[26] Rahul Kumar Namdev, Abhijit Kundu, K. Madhava Krishna, and C. V. Jawahar. Motion segmentation of multiple objects from a freely moving monocular camera. In *ICRA*, pages 4092–4099, 2012.