



Moving object detection and segmentation in urban environments from a moving platform

Dingfu Zhou, Vincent Frémont, Benjamin Quost, Yuchao Dai, Hongdong Li

► To cite this version:

Dingfu Zhou, Vincent Frémont, Benjamin Quost, Yuchao Dai, Hongdong Li. Moving object detection and segmentation in urban environments from a moving platform. Image and Vision Computing, 2017, 68, pp.76-87. 10.1016/j.imavis.2017.07.006 . hal-01609038

HAL Id: hal-01609038

<https://hal.science/hal-01609038>

Submitted on 3 Oct 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Moving Object Detection and Segmentation in Urban Environments from a Moving Platform

Dingfu Zhou^{a,b}, Vincent Frémont^a, Benjamin Quost^a, Yuchao Dai^b,
Hongdong Li^{b,c}

^a*Sorbonne Universités, Université de Technologie de Compiègne, CNRS, UMR 7253, Heudiasyc-CS 60 319, 60 203 Compiègne Cedex, France*

^b*Research School of Engineering, Australian National University*

^c*ARC Centre of Excellence for Robotic Vision*

Abstract

This paper proposes an effective approach to detect and segment moving objects from two time-consecutive stereo frames, which leverages the uncertainties in camera motion estimation and in disparity computation. First, the relative camera motion and its uncertainty are computed by tracking and matching sparse features in four images. Then, the motion likelihood at each pixel is estimated by taking into account the ego-motion uncertainty and disparity in computation procedure. Finally, the motion likelihood, color and depth cues are combined in the graph-cut framework for moving object segmentation. The efficiency of the proposed method is evaluated on the KITTI benchmarking datasets, and our experiments show that the proposed approach is robust against both global (camera motion) and local (optical flow) noise. Moreover, the approach is dense as it applies to all pixels in an image, and even partially occluded moving objects can be detected successfully. Without dedicated tracking strategy, our approach achieves high recall and comparable precision on the KITTI benchmarking sequences.

Keywords: Moving Object Detection, Ego-Motion Uncertainty, Motion Segmentation

1. Introduction

Over the past decades, many researchers from different fields such as robotics, automotive engineering and signal processing have been devoting themselves to the development of intelligent vehicle systems. Making the

vehicles to automatically perceive and understand their 3D environment is a challenging and important task. Due to the improvement of the sensor technologies, processing techniques and researchers' contributions, several Advanced Driver Assistance Systems (ADASs) have been developed for various purposes such as forward collision warning systems, parking assist systems, blind spot detection systems and adaptive cruise control systems. Currently, some popular Vision-based Simultaneous Localization and Mapping (VSLAM) and Structure-from-Motion (SfM) [1] systems have also been applied in ADASs or autonomous vehicle, such as the recent popular ORB-SLAM [2].

However, most of these systems assume a static environment, and thus have faced some difficulties in inner urban areas where dynamic objects are frequently encountered. Usually, moving objects are considered as outliers and RANSAC strategy is applied to get rid of them efficiently. However, this strategy will fail when the moving objects are the dominant part of the image. Thus, efficiently and effectively detecting moving objects turns out to be a crucial issue for the accuracy of such systems.

In this article, we focus on the specific problem of moving object detection. We propose a detection and segmentation system based on two time-consecutive stereo images. The key idea is to detect the moving pixels by compensating the image changes caused by the global camera motion. The uncertainty of the camera motion is also considered to obtain reliable detection results. Furthermore, color and depth information is also employed to remove some false detection.

1.1. Related Works

Moving object detection has been investigated for many years. Background subtraction is a commonly used approach for tackling this problem in videos obtained from a static camera: then, regions of interest can easily be detected [3]. Adaptive Gaussian Mixture Models are well known for modelling the background by recursively updating the Gaussian parameters and simultaneously setting the appropriate number of components for each pixel [4]. However, background subtraction cannot be applied to handle the problem when the camera also moves. Due to the camera motion, both the camera and objects motions are coupled in the apparent 2D motion field. The epipolar constraint is classically used for motion detection between two

views [5]. However, it fails in a degenerate case ¹. Other constraints, such as the flow vector bound constraint [6, 7] have been used together with epipolar constraint to detect the degenerate motion.

An alternative to detecting moving objects using the fundamental matrix is 2D planar homography [8, 9]. Homography is used as a global image motion model that makes it possible to compensate the camera motion between two consecutive frames. Pixels which are consistent with the homography matrix are recognized as the static planar background, while these inconsistent ones may belong to moving objects or to static 3D structure with large depth variance (parallax pixels). In order to remove the parallax pixels, additional geometric constraints [8] or clustering strategies [9] may be used.

Compared to monocular vision, stereo vision system (SVS) provides depth or disparity information using images provided by the left and right cameras. Dense or sparse depth/disparity maps computed by global [10] or semi-global [11] matching approaches can be used to build 3D information on the environment. Theoretically, by obtaining the 3D information, any kind of motion can be detected, even the case of degenerate motion mentioned above. In [12], 3D point clouds are reconstructed from linear stereo vision systems first and then objects are detected based on a spectral clustering technique from the 3D points. Common used methods for Moving Object Detection (MOD) in stereo rig can be divided into sparse feature based [13, 14] and dense scene flow-based approaches [15, 16, 17].

Sparse feature-based approaches fail when few features are detected on the moving objects. Then, dense flow-based methods can be used instead. In [15], a prediction of the optical flow between two consecutive frames is calculated based on a function of the current scene depth and ego-motion. From the difference between the predicted and measured flow fields, large non-zero regions are classified as potential moving objects. Although this motion detection scheme provides dense results, the system may be prone to producing a large number of misdetections due to the noise involved in the perception task. Other improved approaches have been developed [18] and [16] to limit misdetections, by considering the uncertainties on the 3D scene flow [18] or on the 2D real optical flow [16]. However, such approaches roughly model the uncertainty of the ego-motion obtained from other sensors

¹The 3D point moves along the epipolar plane formed by the two camera centers and the point itself, whereas its 2D projections move along the epipolar lines.

(GPS or IMU). In fact, the camera ego-motion has a global influence on the predicted optical flow; therefore, its uncertainty should be well considered to improve detection accuracy.

1.2. Structure and Contributions of This Paper

In this paper, we aim at detecting the foreground moving object from two consecutive stereo image pairs. An object is considered as moving if its location in the absolute world coordinate frame changes between two consecutive frames. This article is an extension of previously-published conference papers [19, 20] with a new review of the relevant state-of-the-art, new theoretical developments and extended experimental results. In [19], we proposed a framework to detect moving objects using the Residual Image Motion Flow (RIMF). In order to improve detection performance, we refined this framework in [20] by considering the uncertainty during the RIMF computation procedure. Both of these works have never been published together; and the aim of this paper is to show the global scope of vision-based perception systems that have been proposed for moving objects detection in intelligent vehicles applications.

The main additional contributions of the present paper are the following: first, we propose a dense moving object detection and segmentation system by using two consecutive stereo frames, which effectiveness is demonstrated on the public KITTI dataset. Unlike in [21] where moving objects are detected by tracking based on sparse features, we compute the dense optical flow in every image pixel. This makes it possible to detect small and partly-occluded moving objects. Next, compared to [20], we add color together with depth information into the graph-cuts framework to improve object segmentation. Then, the 3D density map is used to generate bounding boxes: this strategy proves to be effective to avoid redundancy detection, such as shadow. Finally, we evaluate our proposed moving object segmentation algorithm at the pixel level on the KITTI dataset. At the same time, we also test it at the bounding box level on different real traffic sequences with ground truth. Its effectiveness is demonstrated with respect to related works [15].

This paper is organized as follows: First, Section 2 gives an overview of our proposed moving object detection system. Next, moving pixel detection and motion segmentation are introduced in details in Sections 3 and 4 respectively. Then, we evaluate our proposed system on different KITTI image sequences. The experimental results and analysis are presented in Section 5. Finally, the paper ends with a short conclusion and future works.

2. Problem Setup

As mentioned above, using the epipolar geometry from a monocular camera does not make it possible to detect moving objects when their motion is degenerate. In order to overcome this issue, a stereo system may be employed. The binocular images are recorded by a vehicle-mounted stereo system. We assume that the system is calibrated; thus, a simple rectification [22] can be used to align the left and right images. We denote b as the calibrated baseline for the stereo head. Additionally, the left and right rectified images have identical focal length f and principal point coordinates as $p_0 = (u_0, v_0)^T$.

Given two time-consecutive stereo images from time $t-1$ and t , as shown in Fig. 1, the origin of the world system is assumed to be coincident with the left camera's local coordinate system at time $t-1$. The Z -axis coincides with the left camera optical axis and points forwards, the X -axis points to the right and the Y -axis points downwards. All the coordinate systems are right handed. The main difficulty of moving object detection is caused by the

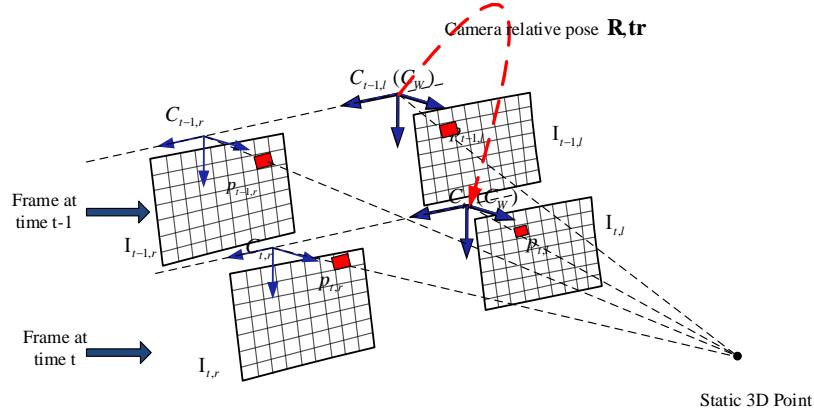


Figure 1: Coordinate frames of the considered stereo-vision system

existence of multiple relative motions – the vehicles' motion and the other objects' independent motions. The position of a pixel extracted from a static background point is $\mathbf{p}_{t-1} = (u_{t-1}, v_{t-1}, 1)^T$ in the previous frame $t-1$, and its image position $\mathbf{p}_t = (u_t, v_t, 1)^T$ in frame t can be predicted by [5, Chapter 9, page 250]:

$$\mathbf{p}_t = \mathbf{K}\mathbf{R}\mathbf{K}^{-1}\mathbf{p}_{t-1} + \frac{\mathbf{K}\mathbf{tr}}{Z_{t-1}}, \quad (1)$$

142 where \mathbf{K} is the camera’s intrinsic parameter matrix, \mathbf{R} , \mathbf{tr} are the relative
 143 camera rotation and translation (the pose), and Z_{t-1} stands for the depth of
 144 the 3D point X in frame at $t - 1$.

145 In order to detect the moving objects in the image, a straightforward
 146 idea is to compensate the camera motion by Eq. (1) first. Then the residual
 147 image, calculated as the difference between the current and previous ones
 148 compensated in motion, highlights both the pixels belonging to moving ob-
 149 jects and the pixels related to motion error estimation. For the sake of clarity,
 150 we first define three different flow-based expressions:

- 151 • the *Global Image Motion Flow* (GIMF) represents the predicted image
 152 changes caused by the camera motion only, that can be calculated using
 153 Eq. (1).
- 154 • The *Measured Optical Flow* (MOF) represents the real dense optical
 155 flow estimated using image processing techniques [23].
- 156 • The *Residual Image Motion Flow* (RIMF) is used to measure the dif-
 157 ference between MOF and GIMF.

158 The RIMF can be used to distinguish between pixels related to moving
 159 and non-moving objects. In order to calculate the RIMF, the MOF and
 160 GIMF should be computed first. Remark that computing the latter requires
 161 both information on the camera motion (ego-motion) and on the depth value
 162 of the pixels. This paper does not address the issues of computing the dense
 163 optical flow [23] and disparity map [24]: we simply use the results from state-
 164 of-the-art methods. More precisely, we exploit the approach proposed in [25]
 165 in order to compute the dense optical flow and dense disparity map. We
 166 then use them directly as inputs of our system. The whole system can be
 167 summarized by the following three steps:

- 168 1. **Moving pixel detection.** In this step, the moving pixels are detected
 169 by compensating the image changes caused by camera motion. In or-
 170 der to improve the detection results, the camera motion uncertainty is
 171 considered.
- 172 2. **Moving object segmentation.** After the moving pixel detection, a
 173 graph-cut based algorithm is used to remove false detections by con-
 174 sidering both the color and disparity information.
- 175 3. **Bounding box generation.** Finally, the bounding boxes are gener-
 176 ated for each moving object by using the UV-disparity map analysis.

For a better understanding, a flowchart of the proposed system is given in Fig. (2), in which the three main steps are highlighted by specific bounding boxes.

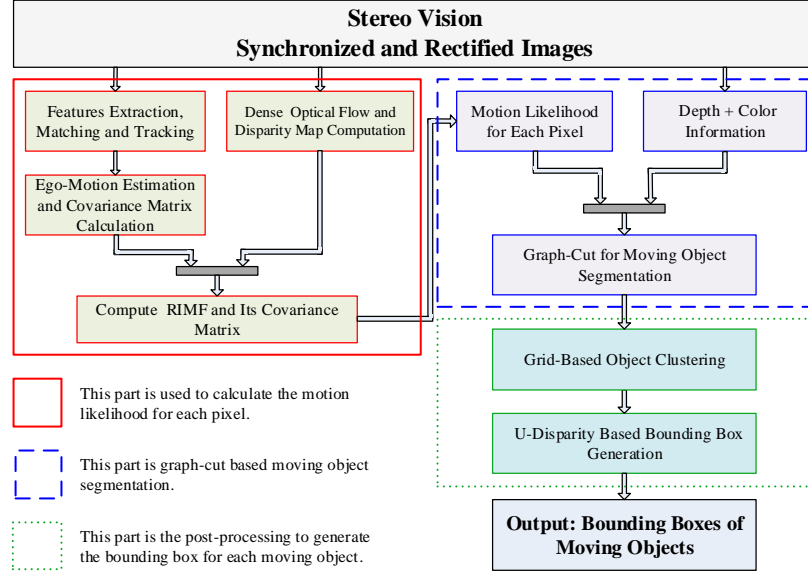


Figure 2: Framework of the moving object detection and segmentation system.

3. Moving Pixel Detection

As described in Fig. 1, four images are considered: two at time $t - 1$ and two at time t . The left image $\mathbf{I}_{t-1,l}$ in the previous frame is considered as the reference image. The right image in the previous frame, and the left and right images in the current frame are represented as $\mathbf{I}_{t-1,r}$, $\mathbf{I}_{t,l}$ and $\mathbf{I}_{t,r}$, respectively. Similarly, we define $(u_{t-1,l}, v_{t-1,l})$, $(u_{t-1,r}, v_{t-1,r})$, $(u_{t,l}, v_{t,l})$ and $(u_{t,r}, v_{t,r})$ as corresponding image points in the previous and current stereo frames.

3.1. Ego-Motion Estimation and Uncertainty Computation

Given a set of corresponding points 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

193 via triangulation and using the camera intrinsic parameters. Then these
 194 3D points are re-projected onto the current image frames using the camera
 195 motion as below:

$$\hat{\mathbf{x}}_t^i = \mathbf{f}(\Theta, \mathbf{x}_{t-1}^i) = \begin{bmatrix} Pr^l(\mathbf{K}, \Theta, \mathbf{x}_{t-1}^i) \\ Pr^r(\mathbf{K}, \Theta, \mathbf{x}_{t-1}^i) \end{bmatrix}, \quad (2)$$

196 where $\hat{\mathbf{x}}_t^i = (\hat{u}_{t,l}^i, \hat{v}_{t,l}^i, \hat{u}_{t,r}^i, \hat{v}_{t,r}^i)^T$ are the predicted image points in the current
 197 frame and $\mathbf{x}_{t-1}^i = (u_{t-1,l}^i, v_{t-1,l}^i, u_{t-1,r}^i, v_{t-1,r}^i)^T$ are the detected image points
 198 in the previous frames. The vector $\Theta = (r_x, r_y, r_z, tr_z, tr_y, tr_x)^T$ represents
 199 the six degrees of freedom of the relative pose. Let Pr^l and Pr^r be the
 200 image projections of the 3D world points into the left and right images (non-
 201 homogeneous coordinates).

202 In general, the optimal camera motion vector $\hat{\Theta}$ can be obtained by min-
 203 imizing the weighted squared error of measurements and predictions

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmin}} F(\Theta, \mathbf{x}) = \underset{\Theta}{\operatorname{argmin}} \sum_{i=1}^N \|\mathbf{x}_t^i - \mathbf{f}(\Theta, \mathbf{x}_{t-1}^i)\|_{\Sigma}^2, \forall i = 1 \cdots N. \quad (3)$$

204 where $\mathbf{x}_t^i = (u_{t,l}^i, v_{t,l}^i, u_{t,r}^i, v_{t,r}^i)^T$ are the matched points in the current frame
 205 by using tracking and matching strategies [26] and where $\|\cdot\|_{\Sigma}^2$ stands for
 206 the squared Mahalanobis distance according to the covariance matrix Σ .

207 Although the optimal motion vector $\hat{\Theta}$ can be obtained by minimizing
 208 Eq. (3), its accuracy also depends on the precision of the matched and
 209 tracked features' positions in the images. Let $\mathbf{x} = [\mathbf{x}_{t-1}, \mathbf{x}_t] \in \mathbb{R}^{8N}$ represent
 210 all points and $\mathbf{x}_{t-1} \in \mathbb{R}^{4N}, \mathbf{x}_t \in \mathbb{R}^{4N}$ stand for the points at times $t-1$
 211 and t , respectively. We assume that all points considered in the optimization
 212 procedure are well-matched pixel features with only additive Gaussian noise:

$$\mathbf{x} \sim \mathbf{N}(\mu, \Sigma), \quad (4)$$

213 where $\mu = (\mu_{\mathbf{x}_{t-1}}, \mu_{\mathbf{x}_t})^T$ and $\Sigma = \operatorname{diag}(\Sigma_{\mathbf{x}_{t-1}}, \Sigma_{\mathbf{x}_t})$ are the mean and the
 214 covariance of the features. The Gauss-Newton optimization of Eq.(3) can
 215 converge rapidly if the starting point is close to the optimal point. A real
 216 vision-based system requires both a robust estimation of the camera motion
 217 and a measurement of the uncertainty associated with this solution. In [27]
 218 and [28], the authors proposed a derivation of the covariance matrix using
 219 the following model:

$$\Sigma_{\Theta} = \left(\frac{\partial g}{\partial \Theta} \right)^{-1} \left(\frac{\partial g}{\partial \mathbf{x}} \right)^T \Sigma_{\mathbf{x}} \left(\frac{\partial g}{\partial \mathbf{x}} \right) \left(\frac{\partial g}{\partial \Theta} \right)^{-T}, \quad (5)$$

220 where $g(\mathbf{x}, \Theta) = \frac{\partial F(\mathbf{x}, \Theta)}{\partial \Theta}$ is the gradient vector of $F(\Theta, \mathbf{x})$ with respect to
 221 Θ , and where $\Sigma_{\mathbf{x}}$, such as defined in Eq. (4), is the covariance matrix of the
 222 measured features at previous and current frames. The partial derivatives,
 223 $\frac{\partial g}{\partial \Theta}$ and $\frac{\partial g}{\partial \mathbf{x}}$ can be computed according to Eq. (5). Alg. (1) presents how the
 224 ego-motion and its associated uncertainty can be computed.

Algorithm 1 Ego-motion estimation and error propagation

Require: - Stereo image pairs at previous and current frames;
 - Covariance matrix of matched features;

Ensure:

- Relative pose Θ (\mathbf{R} and \mathbf{tr}) and its covariance Σ_{Θ} ;

```

1: ► Features extraction, tracking and matching in four images;
2: ► Compute the 3D point at previous frame using Eq. (??);           ▷
   RANSAC process to remove outliers;
3: for  $i = 1$  do  $N$                                      ▷  $N$  is maximum RANSAC times
4:   ► Randomly select 3 matched features pairs;
5:   ►  $\text{iter} = 0$ ;
6:   while  $\text{iter} < 100 \parallel \text{Gauss-Newton increment} > \xi$  do
7:     ► Compute Jacobian matrix and residual matrix;
8:     ► Update  $\Theta$  using Gaussian-Newton iteration approach ;
9:   end while
10:  ► Record  $\Theta$  and inliers indexes if we have more inliers than before;
11: end for
12: ► Refine the final parameters using all the inliers;
13: ► Compute the covariance matrix  $\Sigma_{\Theta}$  using Eq. (5);
14: ► return  $\Theta$  and  $\Sigma_{\Theta}$ 

```

225 3.2. Moving Pixel Detection

226 At the beginning of Section 3, the RIMF has been proposed to detect
 227 moving pixels. In order to compute the RIMF, the GIMF should be estimated
 228 first. In addition, the uncertainty of RIMF can also be computed from the
 229 ego-motion and disparity map uncertainties.

230 3.2.1. Global Image Motion Flow

231 The GIMF is used to represent the image motion flow caused by the
 232 camera motion. Given a pixel position $\mathbf{p}_{t-1} = (u_{t-1}, v_{t-1}, 1)^T$ in the previ-
 233 ous image frame, we can predict its image location $\mathbf{p}_t = (u_t, v_t, 1)^T$ in the

current frame according to Eq. (1). Theoretically, the image location correspondences of a 3D static point in the current frame can be predicted by its depth information in the previous frame and the relative motion information of the camera only. However, this prediction stands only when the 3D point comes from static objects, and it does not hold for dynamic objects. Finally, the GIMF $\mathbf{g} = (g_u, g_v)^T$ for an image point $(u, v)^T$ caused by the camera motion can be expressed as:

$$\mathbf{g} = (g_u, g_v)^T = (u_t - u_{t-1}, v_t - v_{t-1})^T. \quad (6)$$

3.2.2. RIMF Computation

Then, assuming that the MOF estimated between the previous and current frame at point (u, v) is $\mathbf{m} = (m_u, m_v)^T$, the RIMF $\mathbf{q} = (q_u, q_v)^T$ is computed as:

$$\mathbf{q} = \mathbf{g} - \mathbf{m} = (g_u - m_u, g_v - m_v)^T. \quad (7)$$

Ideally, the RIMF should be zero for a static point, while it should be greater than zero for moving points. Simply comparing the RIMF absolute difference to a fixed threshold does not lead to satisfying results to differentiate moving pixels from static ones, because points with different 3D world locations have different image motions. Moreover, the estimated uncertainty, e.g. on camera motion or pixel depth, have a different influence on the image points. Ignoring these uncertainties could lead to a large number of false positive detections. The uncertainty of the RIMF mainly comes from four parts. The first and the most important one is the uncertainty from the camera motion estimation because it has a global influence on each pixel according to Eq. (1). In addition, it affects differently the pixels at different locations. The second influence part is the error of the depth estimation and the third comes from the optical flow estimation process. The last one is the pixel location noise which results directly from the image noise (image rectification, camera intrinsic and extrinsic calibration, digital image quantization, etc).

3.2.3. Motion Likelihood Estimation

As mentioned above, a fixed threshold does not lead to a satisfying solution to detect moving pixels. In order to handle this problem, the uncertainty of RIMF is propagated from the sensors to the final estimation using a first order Gaussian approximation. As in Eq.(7), the RIMF is a function of camera motion Θ , the pixel location (u, v) at previous frame, the disparity d and the measured optical flow (m_u, m_v) . The uncertainty of the measured optical

flow is not considered in this work because it only affects the detection results locally. Based on the forward covariance propagation framework in [5], the RIMF covariance can be calculated by using a first-order approximation as below:

$$\Sigma_{RIMF} = \mathbf{J}\Sigma\mathbf{J}^T, \quad (8)$$

where \mathbf{J} represents the Jacobian matrix with respect to each input variable (e.g. the camera motion Θ , the pixel position (u, v) and the disparity value d in the previous frame) and $\Sigma = \text{diag}(\Sigma_{\Theta}, \Sigma_o)$ is the covariance matrix of all the input variables. The covariance matrix of the camera motion is Σ_{Θ} , and that of the disparity values in the estimation process is $\Sigma_o = \text{diag}(\sigma_u^2, \sigma_v^2, \sigma_d^2)$, where σ_u and σ_v are the variances that are used to describe the pixel quantization error of the camera and σ_d . In [18], the authors proposed that the uncertainty of the disparity map could also be considered as an approximate standard Gaussian Distribution and its variance can be linearly approximated by:

$$\sigma_d(u, v) = \sigma_0 + \gamma U_d(u, v), \quad (9)$$

where σ_0 and γ are two constant parameters, and where $U_d(u, v)$ is the uncertainty on the disparity value at position (u, v) . Here, the matching cost is used as a confidence measure of the disparity value (further details can be found in [29]). Compared to the variance of each parameter in Σ , the covariances between the ego-motion parameters, position and the disparity are negligible and the estimation process is difficult.

Based on the Σ_{RIMF} estimated above, we can compute the likelihood of a flow vector to be moving. Assuming a stationary world and a Gaussian error propagation, a flow vector is assumed to follow a Gaussian distribution with zero mean and covariance matrix Σ_{RIMF} . Deviations from this assumption can be detected by testing this null hypothesis via a goodness-of-fit. Alternatively, the Mahalanobis distance [30] associated to the RIMF vector can be computed:

$$\mu_{\mathbf{q}} = \sqrt{\mathbf{q}^T \Sigma_{RIMF}^{-1} \mathbf{q}}, \quad (10)$$

where \mathbf{q} is the RIMF vector at a certain image location defined in Eq. (7). Since $\mu_{\mathbf{q}}^2$ is χ^2 -distributed, the RIMF motion likelihood $\xi(m)$ of RIMF vector can be computed according to its $\mu_{\mathbf{q}}$ value.

In Fig. (3), the sub-figures (a),(b) are the motion likelihood images resulting from the Mahalanobis distance $\mu_{\mathbf{q}}$. Green pixels are detected as static

299 and red as moving. In sub-figure 3a, two cyclists come from the opposite di-
 300 rection of the host vehicle and a pedestrian moves in the same direction as
 301 the vehicle and all three have been well detected as moving. The shadow
 302 of the moving car in the glass window has also been detected. In sub-figure
 303 3b, all the moving pedestrians have been detected, but false positives on the
 304 ground are due to MOF errors.

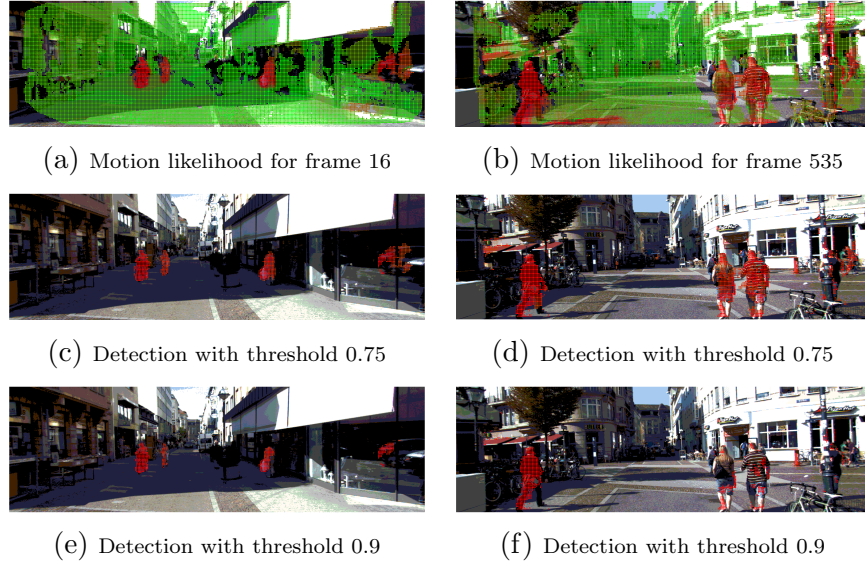


Figure 3: Motion likelihood calculation by using MIMF and moving pixel detection using different thresholds.

305 4. Multi-Cues for Motion Segmentation

306 A likelihood threshold can be applied to the motion likelihood image so
 307 as to distinguish between moving and static pixels. However, detection noise
 308 may pervade the process because of the imperfect MOF. Fig. (3) shows some
 309 detection results using different thresholds. For example, the motion likeli-
 310 hood estimation at frame 16 (sub-figure 3a) is good and all the moving objects
 311 have been well detected, no matter which thresholds are used. Despite that
 312 the motion likelihood at frame 535 (sub-figure 3b) is also well estimated, it is
 313 still noisy on the edge of static objects due to crude estimates of the optical
 314 flow. A lower threshold results in both high true positives and high false
 315 positives; conversely, a higher threshold may result in a poor detection rate.
 316 An optimal threshold that suits all situations cannot be determined.

317 4.1. Segmentation Problem

318 To effectively separate the motion foreground from the background, a
 319 segmentation step is adopted here. Usually, the segmentation of image into
 320 moving and stationary parts can be considered as a problem of assigning
 321 binary labels to each pixel: $l(\mathbf{x}) = 1$ if \mathbf{x} is moving, otherwise $l(\mathbf{x}) = 0$.
 322 Several constraints should be considered for segmentation. First, pixels with
 323 high motion likelihood should be detected as moving. Second, adjacent pixels
 324 with similar appearance and distance should share the same label; otherwise,
 325 their labels should be different. By considering all the constraints, an energy
 326 function can be built as

$$E(L) = E_r(L) + \lambda E_b(L) \quad (11)$$

327 where $L = \{l_1, l_2, \dots, l_p\}$ is a binary vector, p is the number of the pixels in
 328 the image, l_i is a binary label for each pixel. Here, E_r and E_b stand for the
 329 region and boundary terms and λ is used to balance their influences.

330 The region term E_r captures the likelihood that the pixels belong to the
 331 moving foreground or static background. The motion likelihood of each pixel
 332 can be used to build the region term as

$$E_r = - \sum_{\mathbf{x} \in \Omega} \{l(\mathbf{x})\xi_m(\mathbf{x}) + (1 - l(\mathbf{x}))\xi_s(\mathbf{x})\}, \quad (12)$$

333 where Ω represents the image domain, ξ_m is the motion likelihood and ξ_s is
 334 a fixed prior likelihood describing the belief of points being static. Here we
 335 assume that all the image pixels share the same stationary likelihood ξ_s since
 336 no prior information is available.

337 The boundary term E_b is used to encourage similar neighboring pixels to
 338 be assigned the same label. In order to obtain robust segmentation results, we
 339 apply both color [31, 32, 18] and depth information together for building E_b .
 340 Since moving objects usually have a significant depth difference with their
 341 lateral background, the boundary depth similarity can be defined as:

$$B^d(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\sigma(|z(\mathbf{x}_i) - z(\mathbf{x}_j)|) + \alpha), \quad (13)$$

342 where $z(x_i)$ and $z(x_j)$ represent the depth values at the point x_i and x_j .
 343 Note that $B(\cdot)$ is positive function, monotonically decreasing according
 344 to depth difference, in which α and σ are two parameters that control the
 345 descent speed and peak value respectively. A bigger value gives a higher

penalization cost to the depth difference, while the value controls the change of cost with the increase of absolute depth difference. Here we empirically set $\alpha = 0$ and $\sigma = \sqrt{2}$ for all experiments. Similarly, the color similarity B^c is also measured as Eq. (13) by considering the color difference. Finally, the boundary term is expressed as

$$E_b = \sum_{\Omega} \sum_{\hat{\mathbf{x}} \in N_4(\mathbf{x})} (B^d(\hat{\mathbf{x}}, \mathbf{x}) + B^c(\hat{\mathbf{x}}, \mathbf{x})) |l(\hat{\mathbf{x}}) - l(\mathbf{x})|, \quad (14)$$

where $N_4(\mathbf{x})$ is the 4-neighborhood of a pixel \mathbf{x} .

4.2. Graph-Cut for Motion Segmentation

The minimization of this problem can be solved in the graph-cut framework. A minimum cut, or *min-cut*, is the cut with minimal cost that can be computed using min-cut/max-flow algorithms [33, 31]. In Eq. (11), λ is used to balance the influence between the region and boundary terms. Clearly, the segmentation results heavily depend on the weight parameter λ . For a low value of λ , the segmentation is mainly on the motion likelihood of a single pixel whereas a high value of λ results in only small or no segment at all. In our experiments, we have tested different values: $\lambda \in \{0.25, 0.5, 0.75, 2.0, 5.0\}$. The segmentation results show that small λ result in some error detection, while high λ result in small regions (such as in (d), (e) and (f)). We finally chose $\lambda = 0.5$ for our experiments, which gives good results compared to the other values tested. In order to save computer memory and to improve the processing speed in the graph-cut algorithm, a down-sampling technique is used. We take one pixel out of four in both rows and columns. Fig. (4) displays some of the segmentation results obtained using our approach.

4.3. Bounding Box Generation

A bounding box should be generated around each moving object. Additionally, some erroneously detected pixels (e.g., shadows) should also be eliminated. In our approach, we mainly focused on a cubic detection space of 30 m (longitudinal), 20 m (lateral) and 3 m (height) in front of the vehicle. In this limited subspace, a density map is constructed by projecting all the detected 3D moving points onto the xOz plane. The density map is associated with an accumulation buffer. A cell in the accumulation buffer covers an area of 50 cm \times 50 cm on the xOz plane. The weights that the points

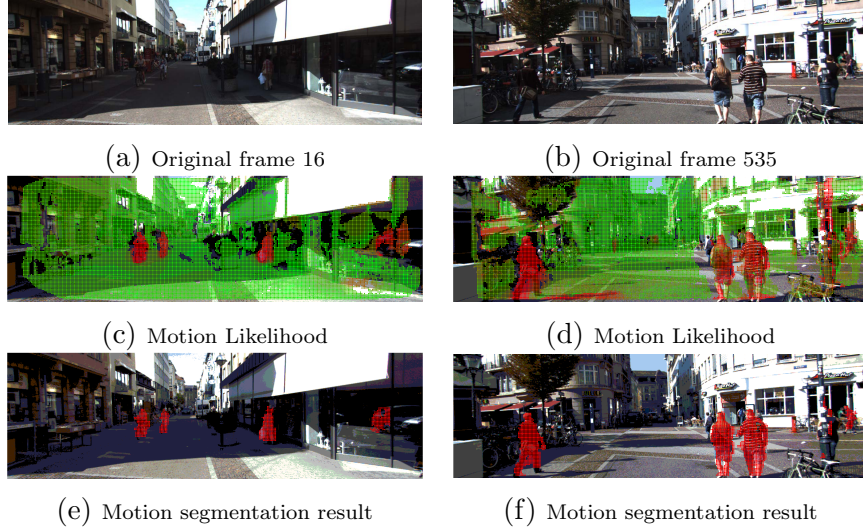


Figure 4: Graph-cut based segmentation in different frames.

378 add to the density map have a Gaussian distribution, with the maximum
 379 at the center cell and decreasing in the neighboring cells. Because points
 380 become sparser as we move away from the camera, the diameter of the patch
 381 increases gradually with the distance. The size of the patch p is defined by
 382 the following strategy (as shown in sub-figure 5a):

$$p = \begin{cases} 1 \times 1 \text{ cell} & z < 10 \text{ m} \\ 2 \times 2 \text{ cells} & 10 \text{ m} < z < 15 \text{ m} \\ 4 \times 4 \text{ cells} & 15 \text{ m} < z < 25 \text{ m} \\ 6 \times 6 \text{ cells} & 25 \text{ m} < z < 31 \text{ m} \end{cases} . \quad (15)$$

383 After obtaining the density map, an empirical threshold is chosen so as to
 384 remove sparse points, that could be misdeteected image pixels (e.g., shadow
 385 or objects borders). Here, a patch will be emptied if its amount of points
 386 is below this threshold (e.g. 50). The false alarms at objects boundary are
 387 usually due to the error on the measured optical flow (smoothing constraint).
 388 Sub-figure 5b shows some ROI generation results relying on the grid-based
 389 method. Based on this approach, the shadow can be easily removed, such
 390 as in 5b-(c). In 5b-(c), each color corresponds to one rough clustering in the
 391 disparity map.

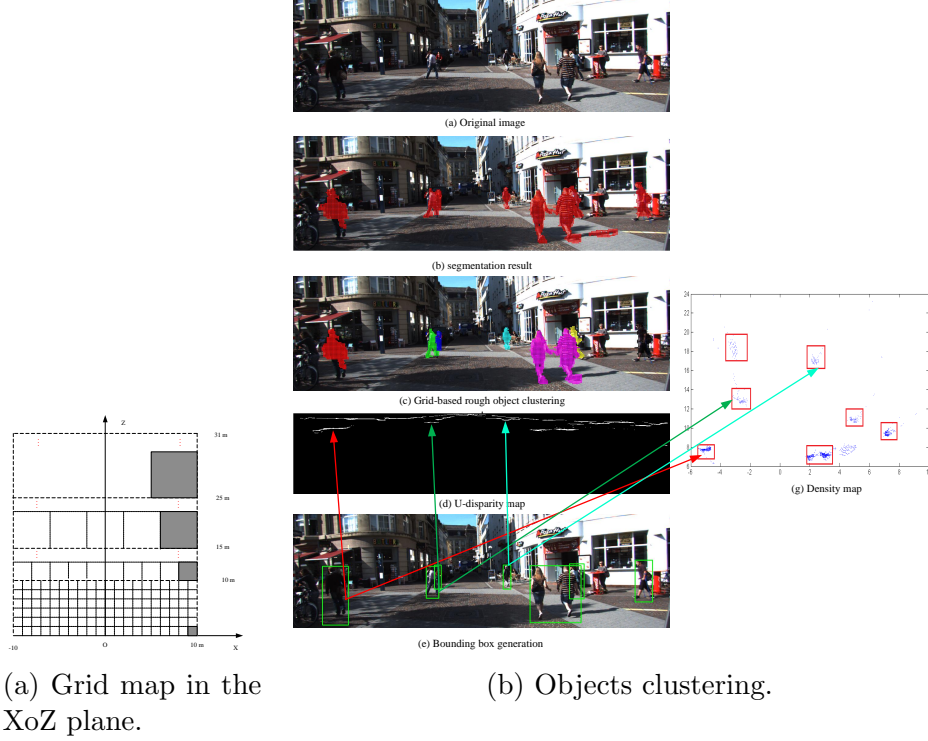


Figure 5: Bounding boxes generation from moving pixels segmentation.

4.3.1. U-Disparity Map Based ROI Generation

In each cluster, the bounding box can be generated for every moving objects for the next recognition step. Region growing is used to remove redundancies and to integrate part detection using the dense disparity map. U-V disparity maps [34, 35], which are two variants of the classical disparity map, are often used for road and obstacle detection. The U-disparity map has the same width as the original image, which is formed by recording the number of the pixels who share the same disparity value along each image column.

In the U-disparity map, an upright object will form a horizontal line because of similar disparity value. Conversely, each white horizontal line represents a corresponding upright object. This information can be effectively used to determine the width of the objects. After getting the width of the bounding box, region growing [36] is applied to the neighborhood of the clustering group pixels based on the disparity value. The pixels whose

407 disparity values are between the minimum and maximum disparity value of
 408 each cluster are considered to belong to the same object. The final bounding
 409 boxes of the moving objects are shown in 5b-(e).

410 4.3.2. V-Disparity Map-Based Cluster Reduction

411 According to [37], the real world height of the objects may be estimated
 412 by:

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

413 Here, h_i and h_c are respectively the heights of the i^{th} object and of the camera
 414 in the world coordinate frame; θ is the camera tilt angle and f is the camera
 415 focal length; z is the depth of the object; y_0 and y_i are the horizontal position
 416 and top of the object in the image coordinate. Assuming that moving objects
 417 are not higher than 3 m, some obvious false positives may be filtered. For
 418 this purpose, the horizontal position is first computed using the V-disparity
 419 map. Then, the actual height of the objects h_i is calculated using Eq. (16).
 420 Finally we retain only the objects whose height is between 0.75 m and 3 m,
 421 because the height of most moving objects is in this range. Detailed steps
 422 can be found in Alg. (2).

Algorithm 2 Bounding Box Generation and Cluster Reduction

Require: - Objects Bounding box;

- Camera height h_c , camera tilt angle θ and camera focal length f ;
- The distance of objects to the camera z ;
- Horizon position y_0 ;

Ensure:

- Real world height of the objects h_i ;
-

- 1: ► Compute the U- and V- disparity maps;
 - 2: ► According to its disparity value, each moving pixel may be assigned to different upright objects using the U-disparity map;
 - 3: ► Compute the horizontal line defined by y_0 and the camera tilt θ from the V-disparity map;
 - 4: ► Calculate the real world height h_i of the objects using the horizontal line y_0 , the camera height h_c and the tilt angle θ as in Eq. (16);
 - 5: ► Keep the detected objects for which h_i is between 0.75 m and 3 mm.
-

423 5. Experimental Results

424 Several real image sequences from the KITTI dataset² have been chosen
425 to test the effectiveness of our system. More details about the sensor setup
426 and data information can be found in [38, 39]. The actual object labels and
427 locations have been provided in some of these sequences, which can be used
428 to evaluate our moving object detection algorithm. The video sequences are
429 acquired from a SVS installed on the roof of a vehicle. Five different video
430 sequences (fps = 10) in the raw data are chosen for our evaluation. These
431 sequences are captured in the inner city streets, which includes moving vehi-
432 cles, pedestrians, cyclists etc. In the inner city sequence, the host vehicle was
433 driven at a low speed (about 15 km/h) because of complex road conditions.

434 Before presenting the experimental results, we review our detection algo-
435 rithm. First, dense disparity and optical flow [25] are computed before the
436 moving objects detection steps. At the same time, the relative camera pose
437 between two consecutive frames and its covariance are estimated based on
438 sparse feature detection and tracking. The standard deviation of the fea-
439 tures in Eq. (4) is empirically set to $\Sigma = \text{diag}[1.0, 1.0]$ pixel. Ideally this
440 value should be changed depending on the situation. In order to compute the
441 variance of disparity in Eq. (9), we empirically set $\sigma_0 = 0.25$ and $\gamma = 0.075$.

442 5.1. Quantitative Evaluation

443 Due to the difficulty of finding the moving object detection and segmenta-
444 tion benchmark in the real traffic scene, we try to construct our own ground
445 truth based on the existing KITTI dataset.

446 5.1.1. Moving Object Segmentation Evaluation at Pixel Level

447 In the “Scene Flow Evaluation 2015 benchmark”, the ground truth of
448 some moving objects (pixel level) have been provided for the training im-
449 ages. We can use this dataset to evaluate our depth-aided moving object
450 segmentation approach at the pixel level. The precision (P), recall (R) and
451 F-measure (F) are usually computed to measure the performance of the sys-
452 tem; they are defined as

$$R = \frac{tp}{tp + fn}, P = \frac{tp}{tp + fp} \text{ and } F = \frac{2R * P}{R + P}. \quad (17)$$

²<http://www.cvlibs.net/datasets/kitti/>

453 The true positive (tp) amount represents the number of pixels that have
 454 been correctly detected as moving. False positives (fp) are static pixels that
 455 have been mis-detected as moving. False negatives (fn) are the moving
 456 pixels that have not been detected. The training dataset consists of 200
 457 image groups and each group includes four images: two stereo image pairs
 458 at current and next time instant. However, some dynamic objects have been
 459 labeled in this training dataset, such as trucks, pedestrians and cyclists.
 460 Finally, only 164 image groups, where the moving objects have been fully
 461 labeled, have been used for our evaluation.

462 Some examples of the moving object segmentation results are displayed
 463 in Fig. 6. Table 1 displays the moving detection results with or without seg-
 464 mentation. The results show that it is hard to determine an optimal motion
 465 likelihood threshold with reasonable precision and recall. A lower thresh-
 466 old gives a high recall value while many false positives have been generated.
 467 Inversely, a high threshold will reduce the recall value. By using segmenta-
 468 tion, the detection accuracy can be increased: the recall, precision and the
 469 F-measure are significantly improved compared to when using a fixed thresh-
 470 old. In addition, compared to [20], the segmentation results have been also
 471 improved, likely by adding the color information into the graph-cut frame-
 472 work.

Methods	Recall (R)	Precise (P)	F-measure (F)
Fixed threshold (0.5)	0.7568	0.4993	0.6016
Fixed threshold (0.7)	0.7338	0.6073	0.6646
Fixed threshold (0.9)	0.6315	0.7007	0.6643
Graph-cut with depth	0.7274	0.6997	0.7133
Graph-cut with depth + color	0.7641	0.6959	0.7284

Table 1: Moving objects segmentation evaluation on the KITTI dataset. Different thresh-
 olds have been chosen for evaluation, while we set $\xi_s = 0.65$ for the whole evaluation.

473 5.1.2. Moving Objects Detection Evaluation at Bounding Boxes Level

474 In the “Raw dataset” category, 2D bounding boxes of the moving objects
 475 have been provided for several sequences with tracklets. We also used these
 476 sequences to evaluate our system at the bounding box level. The ground
 477 truth of the moving objects are generated by labeling them manually from

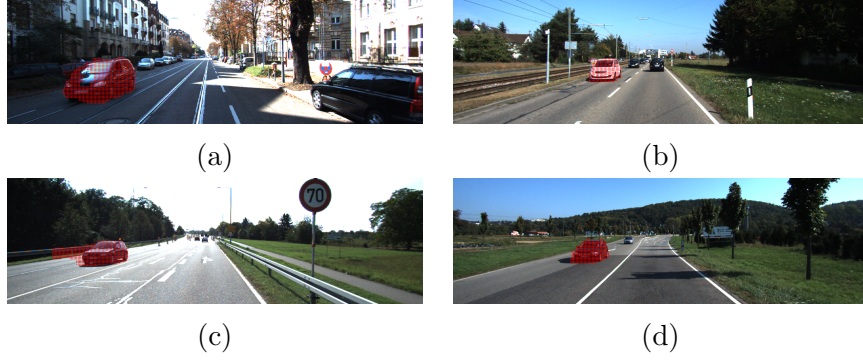


Figure 6: Moving object segmentation on the KITTI "Scene Flow Evaluation 2015 benchmark". Sub-figures (a),(b) and (c) display three segmentation examples, the detected moving pixels are drawn in red. In (a), the moving car has been well detected. In (b), the black one right in front of the camera has not been detected because the car locates nearly at the point of epipole. In (c), some background pixels have been detected as moving due to the occlusion in the next frame. In (d), the silvery car has not been detected because it is out of the detection range.

478 the tracklets for each frame. Here, only the moving objects whose distance
 479 is less than 30m are considered. We employ the PASCAL challenge [40]
 480 measure to evaluate the detection results:

$$score = \frac{area(BB_g\{i\} \cap BB_d\{j\})}{area(BB_g\{i\} \cup BB_d\{j\})}, \quad (18)$$

481 where BB_d and BB_g are the detected and ground truth bounding boxes
 482 of the objects. An object is considered to be correctly detected only when
 483 $BB_d\{i\}$ and $BB_g\{j\}$ share a sufficient overlap area. A threshold $score$ is
 484 set to determine this overlap area: we chose $score = 0.5$ as in the PASCAL
 485 challenge [40]. The precision, recall and F-measure are also computed to
 486 measure the performance of the system. In this case, tp represents the num-
 487 ber of real moving objects bounding boxes have been correctly detected in
 488 the whole sequence, fp stands for static objects that have been misdetected
 489 as moving, and fn are the moving objects that have not been detected. The
 490 true static objects are not taken into account because our algorithm focuses
 491 on detecting moving objects only. Finally, 6 typical image sequences in the
 492 "City" category of the "Raw dataset" are taken for our evaluation. Some
 493 detection results of these sequences are shown in Fig. 7.

494 Fig. 7-(a) shows the detection results in sequence 5. This sequence is
 495 captured around the corner of a quiet city street. The moving van and

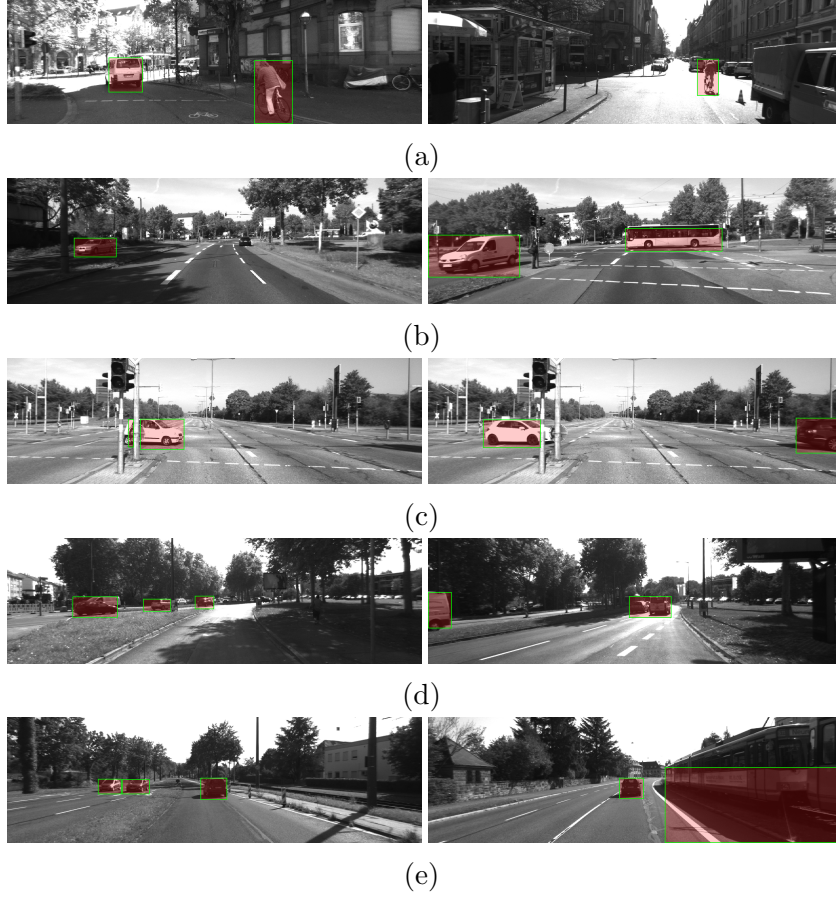


Figure 7: Moving object detection on different sequences. Sub-figures (a), (b), (c), (d) and (e) display the detection results, respectively on sequences 05, 11, 17, 51 and 56 in the category of "City" in the raw data.

the cyclist have been detected in nearly all the frames. The good detection performance benefits from the low camera speed and the relative simple street environment. The van in the right image has not been detected due to the far distant. Fig. 7-(b) and (c) give results of sequences 11 and 17 respectively.

In order to highlight the advantage of our proposal to consider the camera pose and disparity uncertainty, we take the method presented in [15] (which does not consider these uncertainties) as the baseline. In contrast with our proposed method, the RIMF is directly used to detect moving objects. We transform the RIMF into motion likelihood $\mu_{\mathbf{q}}$ as $\mu_{\mathbf{q}} = 1 - \exp(-|\mathbf{q}|)$ and take this value as the input of the segmentation step. The motion likelihood in-

creases with the increasing of the RIMF. In order to achieve fair comparison, we keep all the other steps and parameters as the same as in the proposed approach.

Method	Evaluation	05	11	17	18	51	56
Method [15]	Recall	0.781	0.895	0.961	0.910	0.967	0.949
	Precision	0.383	0.675	0.961	0.843	0.556	0.510
	F-measure	0.513	0.770	0.961	0.876	0.706	0.664
Our method	Recall	0.898	0.919	1.00	0.933	0.968	0.787
	Precision	0.690	0.696	1.00	0.849	0.680	0.768
	F-measure	0.780	0.792	1.00	0.890	0.799	0.777

Table 2: Moving object detection evaluation on different "Raw" data sequences.

Tab. 2 illustrates the quantitative evaluation of the two approaches with the ground truth on six different image sequences. For a clear comparison, we have highlighted the best results in blue for each sequence. From the table, we can see that the detection results have been greatly improved for all the sequences. Taking the uncertainties into account improves the detection rate and reduces the false alarm rate.

5.2. Detection Results on KITTI Sequences

Besides the evaluation results mentioned above, we also tested our system on other scenarios in the KITTI dataset. Fig. 8a shows the detection results in the campus sequence (Campus sequence 37). During this sequence, the camera turned from left to right at a high speed: the vehicle direction changes nearly of 90 degrees in 4.3 seconds. The experimental results show that our algorithm can work well in this situation. The cyclists behind the trees far from the camera can be detected. In Fig. 8a, the red rectangle highlights the undetected moving objects. The cyclist has not been detected by our algorithm because it does not appear in the right camera and the 3D points cannot be reconstructed in the disparity map. Two pedestrians at the left boundary of the second image have also been included in one rectangle because they are not separable in the disparity space.

We also tested our algorithm on a suburban highway sequence (Road sequence 16) and the detection results are displayed in Fig. 8b. On the highway, both the ego-vehicle and the other vehicles move at a high speed, about 60 km h⁻¹. The frame rate of image sequence is 10 frames per second.



(a) Results on a campus sequence.

(b) Results on a suburban road.

Figure 8: Detection results on KITTI dataset.

532 In this case, the dense optical flow approach does not work well because
 533 of the high changes between two successive frames. Then, sparse feature
 534 tracking and matching between two stereo frames can be used for detecting
 535 moving objects. A lower threshold is set in the feature extraction step to
 536 make sure that we can obtain enough features on the moving objects. The
 537 driving vehicles coming from the opposite direction were detected at a range
 538 of 40 m, which remains sufficient for an appropriate reaction of the driver.
 539 The white car moving in front of the camera was also properly detected even
 540 as it moves in the same direction as the ego vehicle.

541 An interesting thing is that the proposed method can also detect forward
 542 moving objects even if it stands in the center of Field-Of-View and has exactly
 543 the same speed with the ego-vehicle, because the MOF is zero in this case,
 544 while the GIMF caused by camera motion is not. Therefore, the forward
 545 vehicle can be detected because the final RIMF is not zero. On the contrary,
 546 the proposed method can also consider the distant background existing in
 547 the center of FOV as static object because both the MOF and GIMF will be
 548 zero in this case.

549 The last sequence we tested is taken in a crowded street³. The host vehicle

³A video of the detection results can be found at

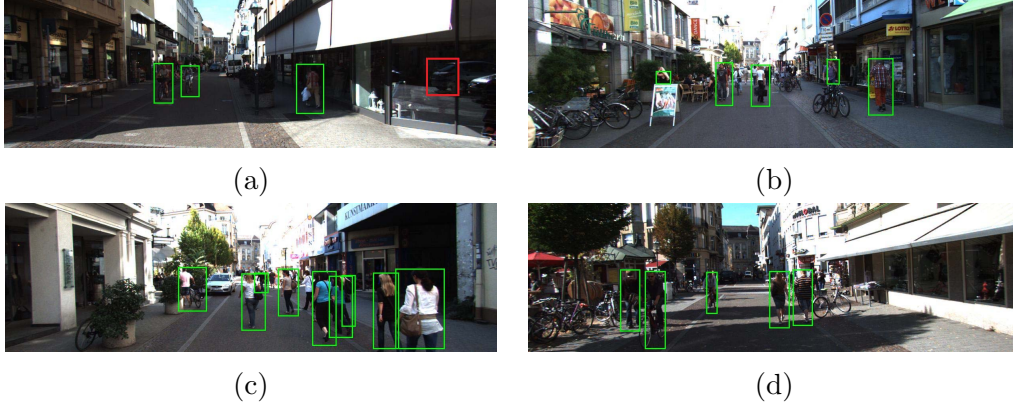


Figure 9: Detection results on a crowd inner city street.

550 moves slowly, which makes detecting moving objects easier. Slowly moving
 551 objects can well be detected by our approach, even when they move on the
 552 epipolar plane. Note that the algorithm also detected partially occluded
 553 objects because dense disparity and optical flow maps are used. Some false
 554 negative and false positive detection happen in the real image sequences, as
 555 displayed in Fig. 9, due to reflections on windows in the scene.

556 5.3. Computational Time

557 All the experiments have been realized on a standard laptop (Intel i7,
 558 4 Core) with the Matlab R2015a processing environment. When the dense
 559 flow is used, the total average computational time is about 165 seconds for
 560 each frame. The dense optical flow and disparity map calculation step takes
 561 about 150 seconds. Around 10 seconds are spent on the motion likelihood
 562 computation, 4 seconds on the graph-cut based segmentation and 1 second
 563 on the bounding boxes generation. Computing ego-motion and estimating
 564 the uncertainty only takes about 0.25 seconds. Although our Matlab imple-
 565 mentation is not real-time, it is also faster when compared to [41] (7 minutes
 566 per frame) and further accelerations could be achieved by C/C++ imple-
 567 mentation with parallel/GPU computing.

<https://www.youtube.com/watch?v=mfSJnCoyLxc>.

568 6. Conclusion and Future Works

569 In this paper, an approach has been proposed to detect moving objects
570 from two consecutive stereo frames. The ego-motion uncertainty is estimated
571 through a first-order error propagation model that is used to obtain the mo-
572 tion likelihood for each pixel. Pixels with a high motion likelihood and a sim-
573 ilar depth are detected as moving based on a graph-cut motion segmentation
574 approach. Additionally, a fast recognition of moving objects becomes possi-
575 ble based on the segmentation results. Detection results in several different
576 real video sequences show that our proposed algorithm is robust with respect
577 to global (camera motion) and local (optical flow) noise. Furthermore, our
578 approach works with all image pixels and arbitrarily moving objects (includ-
579 ing partially occluded) can be detected. Without any tracking strategies, our
580 detection approach gives a high recall rate and also exhibits an acceptable
581 precision rate in several public sequences.

582 However, the computational complexity of the proposed method is an im-
583 portant concern. This is mainly due to computation of the motion likelihood
584 for every image pixel and the segmentation using the graph-cut algorithm.
585 GPU-based algorithms could be used to overcome this weakness [42]. In ad-
586 dition, the performance of MOD highly relies on the results of dense optical
587 flow and disparity maps. However, their estimation in a complex dynamic
588 environment (including other moving objects) often becomes very difficult.

589 Acknowledgments

590 This work was carried out within the framework of the Equipex ROBO-
591 TEX (ANR-10- EQPX-44-01). Dingfu Zhou was sponsored by the China
592 Scholarship Council for 3.5 year’s PhD study at HEUDIASYC laboratory in
593 University of Technology of Compiegne.

594 References

- 595 [1] S. Song, M. Chandraker, and C. Guest, “High accuracy monocular sfm and
596 scale correction for autonomous driving,” *Pattern Analysis and Machine In-*
597 *telligence, IEEE Transactions on*, vol. 38, no. 4, pp. 730–743, 2016.
- 598 [2] R. Mur-Artal, J. Montiel, and J. D. Tardos, “Orb-slam: a versatile and accu-
599 rate monocular slam system,” *Robotics, IEEE Transactions on*, vol. 31, no. 5,
600 pp. 1147–1163, 2015.

- 601 [3] S. Y. Elhabian, K. M. El-Sayed, and S. H. Ahmed, "Moving object detection
602 in spatial domain using background removal techniques-state-of-art," *Recent
603 patents on computer science*, vol. 1, no. 1, pp. 32–54, 2008.
- 604 [4] Z. Zivkovic, "Improved adaptive gaussian mixture model for background sub-
605 traction," in *Pattern Recognition, Proceedings of the 17th International Con-
606 ference on*, vol. 2, pp. 28–31, IEEE, 2004.
- 607 [5] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*.
608 Cambridge University Press, 2003.
- 609 [6] A. Kundu, K. M. Krishna, and J. Sivaswamy, "Moving object detection by
610 multi-view geometric techniques from a single camera mounted robot," in
611 *Intelligent Robots and Systems, International Conference on*, pp. 4306–4312,
612 2009.
- 613 [7] A. Kundu, K. M. Krishna, and C. Jawahar, "Realtime multibody visual slam
614 with a smoothly moving monocular camera," in *Computer Vision (ICCV),
615 IEEE International Conference on*, pp. 2080–2087, IEEE, 2011.
- 616 [8] M. Irani and P. Anandan, "A unified approach to moving object detection in
617 2d and 3d scenes," *Pattern Analysis and Machine Intelligence, IEEE Trans-
618 actions on*, vol. 20, no. 6, pp. 577–589, 1998.
- 619 [9] A. Bugeau and P. Perez, "Detection and segmentation of moving objects in
620 complex scenes," *Computer Vision and Image Understanding*, vol. 113, no. 4,
621 pp. 459–476, 2009.
- 622 [10] L. Wang and R. Yang, "Global stereo matching leveraged by sparse ground
623 control points," in *Computer Vision and Pattern Recognition (CVPR), Con-
624 ference on*, pp. 3033–3040, IEEE, 2011.
- 625 [11] H. Hirschmuller, "Accurate and efficient stereo processing by semi-global
626 matching and mutual information," in *Computer Vision and Pattern Recog-
627 nition, IEEE Computer Society Conference on*, vol. 2, pp. 807–814, 2005.
- 628 [12] S. Moqqaddem, Y. Ruichek, R. Touahni, and A. Sbihi, "Objects detection and
629 tracking using points cloud reconstructed from linear stereo vision," *Current
630 Advancements in Stereo Vision*, p. 161, 2012.
- 631 [13] B. Kitt, B. Ranft, and H. Lategahn, "Detection and tracking of indepen-
632 dently moving objects in urban environments," in *Intelligent Transportation
633 Systems, 13th International IEEE Conference on*, pp. 1396–1401, IEEE, 2010.

- [14] P. Lenz, J. Ziegler, A. Geiger, and M. Roser, “Sparse scene flow segmentation for moving object detection in urban environments,” in *Intelligent Vehicles Symposium (IV)*, IEEE, pp. 926–932, 2011.
- [15] A. Talukder and L. Matthies, “Real-time detection of moving objects from moving vehicles using dense stereo and optical flow,” in *Intelligent Robots and Systems, Proceedings. International Conference on*, vol. 4, pp. 3718–3725, IEEE, 2004.
- [16] V. Romero-Cano and J. I. Nieto, “Stereo-based motion detection and tracking from a moving platform,” in *Intelligent Vehicles Symposium, IEEE*, pp. 499–504, IEEE, 2013.
- [17] C. Rabe, T. Müller, A. Wedel, and U. Franke, “Dense, robust, and accurate motion field estimation from stereo image sequences in real-time,” in *European conference on computer vision*, pp. 582–595, Springer, 2010.
- [18] A. Wedel and D. Cremers, *Stereo scene flow for 3D motion analysis*. Springer Verlag London Limited, 2011.
- [19] D. Zhou, V. Frémont, and B. Quost, “Moving objects detection and credal boosting based recognition in urban environments,” in *Cybernetics and Intelligent Systems (CIS), Conference on*, pp. 24–29, IEEE, 2013.
- [20] D. Zhou, V. Fremont, B. Quost, and B. Wang, “On modeling ego-motion uncertainty for moving object detection from a mobile platform,” in *Intelligent Vehicles Symposium Proceedings*, pp. 1332–1338, IEEE, 2014.
- [21] P. Lenz, J. Ziegler, A. Geiger, and M. Roser, “Sparse scene flow segmentation for moving object detection in urban environments,” in *Intelligent Vehicles Symposium (IV)*, pp. 926–932, IEEE, 2011.
- [22] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O’Reilly Media, Inc., 2008.
- [23] C. Liu, *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Massachusetts Institute of Technology, 2009.
- [24] A. Geiger, M. Roser, and R. Urtasun, “Efficient large-scale stereo matching,” in *Asian Conference on Computer Vision*, pp. 25–38, Springer, 2010.
- [25] C. Vogel, K. Schindler, and S. Roth, “3d scene flow estimation with a piecewise rigid scene model,” *International Journal of Computer Vision*, vol. 115, no. 1, pp. 1–28, 2015.

- 667 [26] A. Geiger, J. Ziegler, and C. Stiller, “Stereoscan: Dense 3d reconstruction in
668 real-time,” in *Intelligent Vehicles Symposium, IEEE*, pp. 963–968, 2011.
- 669 [27] R. M. Haralick, “Propagating covariance in computer vision,” in *Performance
670 Characterization in Computer Vision*, pp. 95–114, Springer, 2000.
- 671 [28] J. C. Clarke, “Modelling uncertainty: A primer,” *University of Oxford. Dept.
672 Engineering science, Tech. Rep*, vol. 2161, p. 98, 1998.
- 673 [29] X. Hu and P. Mordohai, “A quantitative evaluation of confidence measures for
674 stereo vision,” *Pattern Analysis and Machine Intelligence, IEEE Transactions
675 on*, vol. 34, no. 11, pp. 2121–2133, 2012.
- 676 [30] P. Mahalanobis, “On the generalised distance in statistics,” in *Proceedings of
677 the National Institute of Science of India*, pp. 49–55, 1936.
- 678 [31] Y. Y. Boykov and M.-P. Jolly, “Interactive graph cuts for optimal boundary
679 & region segmentation of objects in ND images,” in *Computer Vision. IEEE
680 International Conference on*, vol. 1, pp. 105–112, 2001.
- 681 [32] C. Rother, V. Kolmogorov, and A. Blake, “Grabcut: Interactive foreground
682 extraction using iterated graph cuts,” in *ACM Transactions on Graphics
683 (TOG)*, vol. 23, pp. 309–314, 2004.
- 684 [33] D. Greig, B. Porteous, and A. H. Seheult, “Exact maximum a posteriori
685 estimation for binary images,” *Journal of the Royal Statistical Society. Series
686 B (Methodological)*, pp. 271–279, 1989.
- 687 [34] R. Labayrade, D. Aubert, and J.-P. Tarel, “Real time obstacle detection in
688 stereovision on non flat road geometry through” v-disparity” representation,”
689 in *Intelligent Vehicle Symposium*, vol. 2, pp. 646–651, IEEE, 2002.
- 690 [35] Z. Hu and K. Uchimura, “UV-disparity: an efficient algorithm for stereovision
691 based scene analysis,” in *Intelligent Vehicles Symposium.*, pp. 48–54, 2005.
- 692 [36] R. Adams and L. Bischof, “Seeded region growing,” *Pattern Analysis and
693 Machine Intelligence, IEEE Transactions on*, vol. 16, no. 6, pp. 641–647,
694 1994.
- 695 [37] D. Hoiem, A. A. Efros, and M. Hebert, “Putting objects in perspective,”
696 *International Journal of Computer Vision*, vol. 80, no. 1, pp. 3–15, 2008.

- 697 [38] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving?
698 the kitti vision benchmark suite,” in *Computer Vision and Pattern Recognition, Conference on*, pp. 3354–3361, IEEE, 2012.
- 700 [39] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The
701 kitti dataset,” *International Journal of Robotics Research (IJRR)*, 2013.
- 702 [40] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman,
703 “The pascal visual object classes (voc) challenge,” *International journal of*
704 *computer vision*, vol. 88, no. 2, pp. 303–338, 2010.
- 705 [41] R. K. Namdev, A. Kundu, K. M. Krishna, and C. Jawahar, “Motion segmenta-
706 tion of multiple objects from a freely moving monocular camera,” in *Robotics*
707 *and Automation, International Conference on*, pp. 4092–4099, IEEE, 2012.
- 708 [42] Z. Yang, Y. Zhu, and Y. Pu, “Parallel image processing based on cuda,” in
709 *Computer Science and Software Engineering, International Conference on*,
710 vol. 3, pp. 198–201, IEEE, 2008.