Chapter 3
Moving Object Detection from Mobile Platforms Using Stereo Data Registration

Angel D. Sappa[1], David Gerónimo[1,2], Fadi Dornaika[3,4], Mohammad Rouhani[1], and Antonio M. López[1,2]

[1] Computer Vision Center
Universitat Autònoma de Barcelona, 08193, Bellaterra, Barcelona, Spain
[2] Computer Science Department
Universitat Autònoma de Barcelona, 08193, Bellaterra, Barcelona, Spain
[3] University of the Basque Country, San Sebastian, Spain
[4] IKERBASQUE, Basque Foundation for Science, Bilbao, Spain
{asappa,dgeronimo,rouhani,antonio}@cvc.uab.es,
fadi_dornaika@ehu.es

Abstract. This chapter describes a robust approach for detecting moving objects from on-board stereo vision systems. It relies on a feature point quaternion-based registration, which avoids common problems that appear when computationally expensive iterative-based algorithms are used on dynamic environments. The proposed approach consists of three main stages. Initially, feature points are extracted and tracked through consecutive 2D frames. Then, a RANSAC based approach is used for registering two point sets, with known correspondences in the 3D space. The computed 3D rigid displacement is used to map two consecutive 3D point clouds into the same coordinate system by means of the quaternion method. Finally, moving objects correspond to those areas with large 3D registration errors. Experimental results show the viability of the proposed approach to detect moving objects like vehicles or pedestrians in different urban scenarios.

1 Introduction

The detection of moving objects in dynamic environments is generally tackled by first modelling the background. Then, foreground objects are directly obtained by performing an image subtraction (e.g., [14], [15], [32]). An extensive survey on motion detection algorithms can be found in [21]. In general, most of the approaches assume stationary cameras, which means all frames are registered in the same coordinate system. However, when the camera moves, the problem becomes intricate since it is unfeasible to have a unique background model. In such a case, moving object detection is generally tackled by compensating the camera motion so that all frames from a given video sequence, obtained from a moving camera/platform, are referred to the same reference system (e.g., [7], [27]).

Moving object detection from a moving camera is a challenging problem in computer vision, having a number of applications in different domains: mobile robots

M.R. Ogiela and L.C. Jain (Eds.): Computational Intelligence Paradigms, SCI 386, pp. 25–37.
springerlink.com

[26]; aerial surveillance [35] [34]; video segmentation [1]; vehicles and driver assistance [15], [24]; just to mention a few. As mentioned above, the underlying strategy in the solutions proposed in the literature essentially relies on the compensation of the camera motion. The difference between them lie on the sensor (i.e., monocular/stereoscopic) or on the use of prior-knowledge of the scene together with visual cues. For instance, [26] uses a stereo system and predicts the depth image for the current time by using ego-motion information and the depth image obtained at the previous time. Then, moving objects are easily detected by comparing the predicted depth image with the one obtained at the current time. The prior-knowledge of the scene is also used in [35] and [34]. In these cases the authors assume that the scene is far from the camera (monocular) and the depth variation of the objects of interest is small compared to the distance (e.g., airborne image sequences). In this context camera motion can be approximately compensated by a 2D parametric transformation (a 3x3 homography). Hence, motion compensation is achieved by warping a sequence of frames to a reference frame, where moving objects are easily detected by image subtraction like in the stationary camera cases.

A more general approach has been proposed in [1] for segmenting videos captured with a freely moving camera, which is based on recording complex background and large moving non-rigid foreground objects. The authors propose a region-based motion compensation. It estimates the motion of the camera by finding the correspondence of a set of salient regions obtained by segmenting successive frames. In the vehicle on-board vision systems and driver assistance fields, the compensation of camera motion has also attracted researchers' attention in recent years. For instance, in [15] the authors present a simple but effective approach based on the use of GPS information to roughly align frames from video sequences. A local appearance comparison between the aligned frames is used to detect objects. In the driver assistance context, but by using an onboard stereo rig, [24] introduce a 3D data registration based approach to compensate camera motion from two consecutive frames. In that work, consecutive stereo frames are aligned into the same coordinate system; then moving objects are obtained from a 3D frame subtraction, similar to [26]. The current chapter proposes an extension of [24], by detecting misregistration regions according to an adaptive threshold from the depth information.

The remainder of this chapter is organized as follows. Section 2 introduces related work in the 3D data registration problem. Then, Section 3 presents the proposed approach for moving object detection. It consists of three stages: *i*) 2D feature point detection and tracking; *ii*) robust 3D data registration; and *iii*) moving object detection through consecutive stereo frame subtraction. Experimental results in real environments are presented in Section 4. Finally, conclusions and future works are given in Section 5.

2 Related Work

A large number of approaches have been proposed in the computer vision community for 3D Point registration during the last two decades (e.g., [3], [4], [22]). 3D

data point registration aims at finding the best transformation that places both the given data set and corresponding model set into the same reference system. The different approaches proposed in the literature can be broadly classified into two categories, depending on whether an initial information is required (*fine registration*) or not (*coarse registration*); a comprehensive survey of registration methods can be found in [23]. The approach followed in the current work for moving object detection lies within the fine rigid registration category.

Typically, the fine registration process consists in iterating the following two stages. Firstly, the correspondence between every point from the current data set and the model set shall be found. These correspondences are used to define the residual of the registration. Secondly, the best set of parameters that minimizes the accumulated residual shall be computed. These two stages are iteratively applied until convergence is reached. The Iterative Closest Point (ICP)—originally introduced by [3] and [4]—is one of the most widely used registration techniques using this two-stage scheme. Since then, several variations and improvements have been proposed in order to increase the efficiency and robustness (e.g., [25], [8], [5]).

In order to avoid the point-wise nature of ICP, which makes the problem discrete and non-smooth, different techniques have been proposed: *i*) probabilistic representations are used to describe both data and model set (e.g. [31], [13]); *ii*) in [8] the point-wise problem is avoided by using a distance field of the model set; *iii*) an implicit polynomial (IP) is used in [36] to fit the distance field, which later defines a gradient field leading the data points towards that model set; *iv*) implicit polynomials have been also used in [28] to represent both the data set and model set. In this case, an accurate pose estimation is computed based on the information from the polynomial coefficients.

Probabilistic-based approaches avoid the point-wise correspondence problem by representing each set by a mixture of Gaussians (e.g., [13], [6]); hence, registration becomes a problem of aligning two mixtures. In [13] a closed-form expression for the L_2 distance between two Gaussian mixtures is proposed. Instead of Gaussian mixture models, [31] proposes an approach based on multivariate t-distributions, which is robust to large number of missing values. Both approaches, as all mixture models, are highly dependent on the number of mixtures used for modelling the sets. This problem is generally solved by assuming a user defined number of mixtures or as many as the number of points. The former one needs the points to be clustered, while the latter one results in a very expensive optimization problem that cannot handle large data sets or could get trapped in local minimum when complex sets are considered.

The non-differentiable nature of ICP is overcome by using a derivable distance transform—Chamfer distance—in [8]. A non-linear minimization (Levenberg - Marquardt algorithm) of the error function, based on that distance transform, is used for finding the optimal registration parameters. The main disadvantage of [8] is the precision dependency on the grid resolution, where the Chamfer distance transform and discrete derivatives are evaluated. Hence, this technique cannot be directly applied when the point set is sparse or unorganized.

On the contrary to the previous approaches, [36] proposes a fast registration method based on solving an energy minimization problem derived from an implicit polynomial fitted to the given model set [37]. This IP is used to define a gradient flow that drives the data set to the model set without using point-wise correspondences. The energy functional is minimized by means of a heuristic two step process. Firstly, every point in the given data set moves freely along the gradient vectors defined by the IP. Secondly, the outcome of the first step is used to define a single transformation that represents this movement in a rigid way. These two steps are repeated alternately until convergence is reached. The weak point of this approach is the first step of the minimization that lets the points move independently in the proposed gradient flow. Furthermore, the proposed gradient flow is not smooth, specially close to the boundaries.

Most of the algorithms presented above have been originally proposed for registering overlapped sets of points corresponding to the 3D surface of a single rigid object. Extensions to a more general framework, where the 3D surfaces to be registered correspond to different views of a given scene, have been presented in the robotic field (e.g., [30, 18]). Actually, in all these extensions, the registration is used for the simultaneous localization and mapping (*SLAM*) of the mobile platform (i.e., the robot). Although some approaches differentiate static and dynamic parts of the environment before registration (e.g., [30], [33]), most of them assume that the environment is static, containing only rigid, non-moving objects. Therefore, if moving objects are present in the scene, the least squares formulation of the problem will provide a rigid transformation biased by the motions in the scene.

Independently to the kind of scenario to be tackled (partial view of a single object or whole scene), 3D registration algorithms are computationally expensive, which prevents their use in real time applications. In the current work a robust strategy that reduces the CPU time by focusing only on feature points is proposed. It is intended to be used in ADAS (Advanced Driver Assistance Systems) applications, in which an on-board camera explores the current scene in real time. Usually, an exhaustive window scanning approach is adopted to extract regions of interests (ROIs), needed in pedestrian or vehicle detection systems. The concept of consecutive frame registration for moving object detection has been explored in [11], in which an active frame subtraction for pedestrian detection from images of moving cameras is proposed. In that work, consecutive frames were not registered by a vision based approach but by estimating the relative camera motion using vehicle speed and a gyrosensor. A similar solution has been proposed in [15], but by using GPS information.

3 Proposed Approach

The proposed approach combines 2D detection of key points with 3D registration. The first stage consists in extracting a set of 2D feature points at a given frame and track it through the next frame; 3D coordinates corresponding to each of these 2D feature points are later on used during the registration process, where the rigid displacement (six degrees of freedom) that maps the 3D scene associated with frame

(n) into the 3D scene associated with frame ($n+1$) is computed (see Figure 1). This rigid transform represents the 3D motion of the camera between frame (n) and frame ($n+1$). Finally, moving objects are detected by computing the difference between the 3D coordinates of points represented in the same coordinate system. Before going into details in the stages of the proposed approach a brief description of the used stereo vision system is given.

3.1 System Setup

A commercial stereo vision system (Bumblebee from Point Grey[1]) is used to acquire the 3D information of the scene in front of the host vehicle. It consists of two Sony ICX084 Bayer pattern CCDs with 6mm focal length lenses. Bumblebee is a pre-calibrated system that does not require in-field calibration. The baseline of the stereo head is 12cm and it is connected to the computer by an IEEE-1394 interface. Right and left color images (Bayer pattern) were captured at a resolution of 640×480 pixels. After capturing each right-left pair of images, a dense cloud of 3D data points $\mathbf{P}^n$ is computed by using a 3D reconstruction software at each frame n. The right intensity image $\mathbf{I}^n$ is used during the feature point detection and tracking stage.

3.2 Feature Detection and Tracking

As previously mentioned, the proposed approach is intended to be used on on-board vision systems for driver assistance applications. Hence, due to real time constraint, it is clear that the whole cloud of points cannot be used to find the rigid transformation that maps two consecutive frames to the same reference system. In order to tackle this problem, an efficient approach that relies only on the use of a reduced set of points from the given image $\mathbf{I}^n$ is proposed. Feature points, $f^n_{i(u,v)} \subset \mathbf{I}^n$, far away from the camera position ($P^n_{i(x,y,z)} > \delta$) are discarded in order to increase registration accuracy[2] ($\delta = 15$ m in the current implementation).

The proposed approach does not depend on the technique used for detecting feature points; actually, two different approaches have been tested: one based on the Harris corner points [10] and another on SIFT features [16]. In the first case, once feature points have been selected a tracking window W_T of (9×9) pixels is set. Feature points are tracked by minimizing the sum of squared differences between two consecutive frames by using an iterative approach [17]. In the second case SIFT features [16] are detected in the extreme of difference of Gaussians in a scale-space representation and described as histograms of gradient orientations. In this case, following [16], a function based on the corresponding histograms distance is used to match the features in consecutive frames (the public implementation of SIFT in [29] has been used).

[1] `www.ptgrey.com`

[2] Stereo head data uncertainty grows quadratically with depth [19].

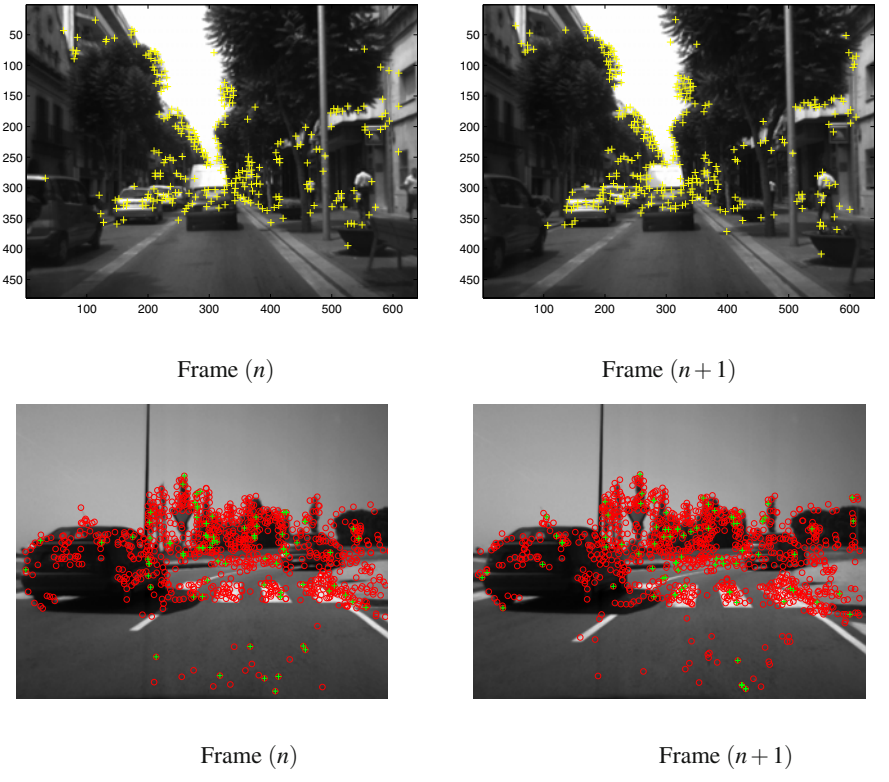

Fig. 1 Feature points detected and tracked through consecutive frames: (top) using Harris corner detector; $(bottom)$ using SIFT detector and descriptor

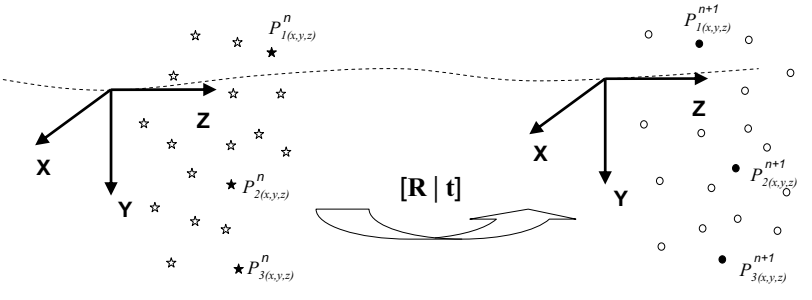

Fig. 2 Illustration of feature points represented in the 3D space, together with three couples of points used for computing the 3D rigid displacement: $[\mathbf{R}|\mathbf{t}]$—RANSAC-like technique

3.3 Robust Registration

The set of 2D-to-2D point correspondences obtained in the previous stage, is easily converted to a set of 3D-to-3D points since for every frame we have a quasi dense 3D reconstruction that is rapidly provided by Bumblebee. In the current approach, contrary to Iterative Closest Point (ICP) based algorithms, the correspondences between the two point sets are known; hence, the main challenge that should be faced during this stage is the fact that feature points can belong to static or moving objects in the scene. Since the camera is moving there are no additional clues to differentiate them easily. Hence, the use of a robust RANSAC-like technique is proposed to find the best rigid transformation that maps the 3D points of frame (n) into their corresponding in frame ($n+1$). The closed-form solution provided by unit quaternions [12] is chosen to compute this 3D rigid displacement, with rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ between the two sets of vertices. The proposed approach works as follows:

Random sampling. Repeat the following three steps K times (in our experiments K was set to 100):

1. Draw a random subsample of 3 different pairs of feature points $(P^n_{i(x,y,z)}, P^{n+1}_{i(x,y,z)})_k$, where $P^n_{i(x,y,z)} \in \mathbf{P}^n$, $P^{n+1}_{i(x,y,z)} \in \mathbf{P}^{n+1}$ and $i = \{1,2,3\}$.
2. For this subsample, indexed by k ($k = 1,....,K$), compute the 3D rigid displacement $D_k = [\mathbf{R}_k|\mathbf{t}_k]$ that minimizes the residual error $\sum_{i=1}^{3} |P^{n+1}_{i(x,y,z)} - \mathbf{R_k} P^n_{i(x,y,z)} - \mathbf{t_k}|^2$. This minimization is carried out by using the closed-form solution provided by the unit quaternion method [12].
3. For this solution D_k, compute the number of inliers among the entire set of pairs of feature points according to a user defined threshold value.

Solution

1. Choose the best solution, i.e., the solution that has the highest number of inliers. Let D_q be this solution.
2. Refine the 3D rigid displacement $[\mathbf{R}_q|\mathbf{t}_q]$ by using the whole set of couples considered as inliers, instead of the corresponding 3 pairs of feature points. A similar unit quaternion representation [2] is used to minimize: $\sum_{i=1}^{\#inliers} |P^{n+1}_{i(x,y,z)} - \mathbf{R_q} P^n_{i(x,y,z)} - \mathbf{t_q}|^2$.

3.4 Frame Subtraction

The best 3D rigid displacement $[\mathbf{R}_q|\mathbf{t}_q]$ computed above with inliers 3D feature points is representing the camera motion. Thus, it will be used for detecting moving regions after motion compensation. First, the whole set of 3D data points at frame (n) is mapped by:

$$\widehat{P}^{n+1}_{i(x,y,z)} = \mathbf{R}_q P^n_{i(x,y,z)} + \mathbf{t}_q \ , \tag{1}$$

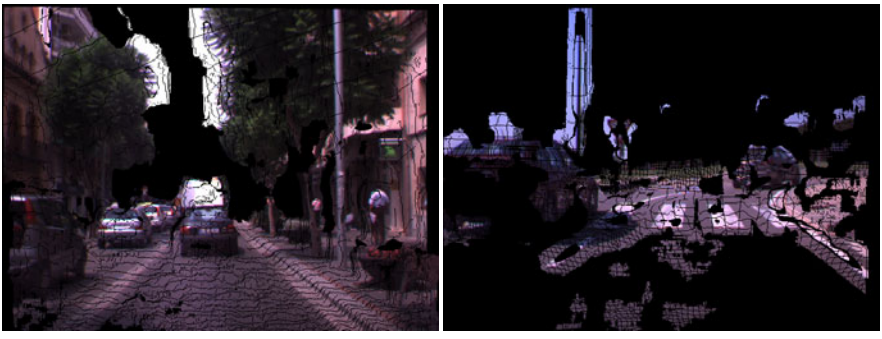

Fig. 3 Synthesized views representing frames (n) (from Fig. 1$(left)$) in the coordinate systems of frames $(n+1)$, by using their corresponding rigid displacements: $[\mathbf{R_q}|\mathbf{t_q}]$

where $\widehat{P}^{n+1}_{i(x,y,z)}$ denotes the mapping of a given point from frame n into the next frame. Note that for static 3D points we ideally have $\widehat{P}^{n+1}_{i(x,y,z)} = P^{n+1}_{i(x,y,z)}$.

Once the whole set of points $\mathbf{P}^n$ has been mapped, we can also synthesize the corresponding 2D view $(\widehat{I}^{n+1}_{(u,v)})$ as follows:

$$\widehat{u}_i^{n+1} = (round)\left(u_0 + f\frac{\widehat{x}_i^{n+1}}{\widehat{z}_i^{n+1}}\right) \quad , \tag{2}$$

$$\widehat{v}_i^{n+1} = (round)\left(v_0 + f\frac{\widehat{y}_i^{n+1}}{\widehat{z}_i^{n+1}}\right) \quad ,$$

where f denotes the focal length in pixels, (u_0, v_0) represents the coordinates of the camera principal point, and $(\widehat{x}_i^{n+1}, \widehat{y}_i^{n+1}, \widehat{z}_i^{n+1})$ correspond to the 3D coordinates of

Fig. 4 $(left)$ $D_{(u,v)}$ map of moving regions, from frames (n) and $(n+1)$ presented in Fig. 1(top). $(right)$ Image difference between these consecutive frames: $(|\mathbf{I}^{(n)} - \mathbf{I}^{(n+1)}|)$ to illustrate their relative displacement.

the mapped point (1). Figure 3 shows two synthesized views obtained after mapping frames (n) (Fig. 1*(left)*) with their corresponding $[\mathbf{R_q}|\mathbf{t_q}]$.

A *moving region map*, $D_{(u,v)}$, is then computed using the difference between the synthesized scene and the actual scene as follows:

$$D_{(u,v)} = \begin{cases} 0, & \text{if} \quad |\widehat{P}^{n+1}_{i(x,y,z)} - P^{n+1}_{i(x,y,z)}| < \tau_i \\ (\widehat{I}^{n+1}_{(u,v)} + I^{n+1}_{(u,v)})/2, & \text{otherwise} \end{cases}, \tag{3}$$

where, τ_i is a threshold directly related with the depth to the camera (since the accuracy of the stereo rig decreases with the depth, the value of τ increases to compensate that loss of accuracy). Image differences are used in the above map just to see the correlation between intensity differences and 3D coordinate differences of mapped points (i.e., a given point in frame (n) with its corresponding one in frame $(n+1)$). Figure 4*(left)* presents the map of moving regions, $D_{(u,v)}$, resulting from the frame $(n+1)$ (Fig. 1$(right)$) and the synthesized view corresponding to frame

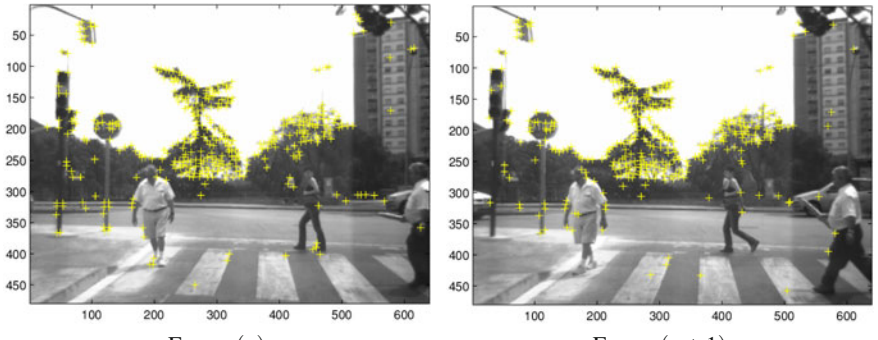

Fig. 5 Feature points detected and tracked through consecutive frames

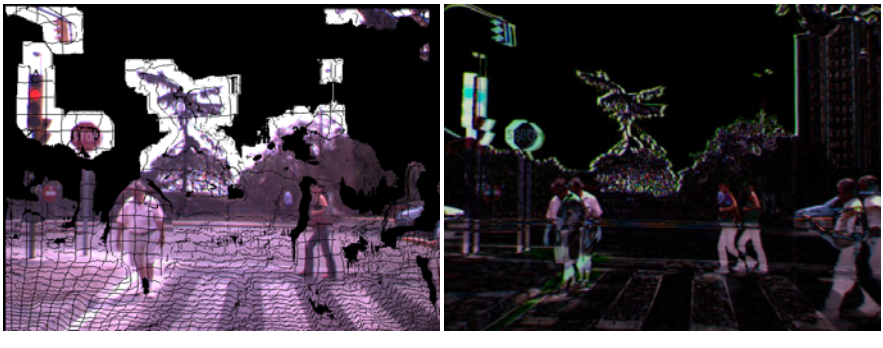

Fig. 6 $(left)$ Synthesized view of frame (n) (Fig. 5$(left)$). $(right)$ Difference between consecutive frames: $(|\mathbf{I}^{(n)} - \mathbf{I}^{(n+1)}|)$ to illustrate their relative displacement (pay special attention at the traffic lights and stop signposts)

(n) (see Figure 3). Additionally, Fig. 4*(right)* illustrates the raw image difference between the two consecutive frames ($|\mathbf{I}^{(n)} - \mathbf{I}^{(n+1)}|$).

4 Experimental Results

Experimental results in real environments and different vehicle speeds are presented in this section. In all the cases large error regions correspond to both moving objects and misregistered areas. Several video sequences were processed on a 3.2 GHz Pentium IV PC. Experimental results presented in this chapter correspond to video sequences recorded at 10 fps. In other words the elapsed time between two consecutive frames is about 100 ms.

The proposed algorithm takes, on average, 31 ms for registering consecutive frames by using about 300 feature points. Fig. $1(top)$ shows two frames of a crowded urban scene. This scene is particularly interesting since a large set of feature points over surfaces moving at different speed have been extracted. In this case, the use of classical ICP based approaches (e.g., [18]) would provide a wrong scene registration since points from static and moving objects are considered together. The synthesized view obtained from frame (n) is presented in Fig. $3(left)$. The quality of the registration result can be appreciated in the map of moving regions presented in Fig. 4*(left)*. Particularly interesting is the lamp post region, where there is a perfect registration between the 3D coordinates of these pixels. Large errors at the top of trees or further away regions are mainly due to depth uncertainty, which as mentioned before grows quadratically with depth [19]. Wrong moving regions mainly correspond to hidden areas in frame (n) that are unveiled in frame $(n+1)$. Figure $4(right)$ presents the difference between consecutive frames ($|\mathbf{I}^{(n)} - \mathbf{I}^{(n+1)}|$)

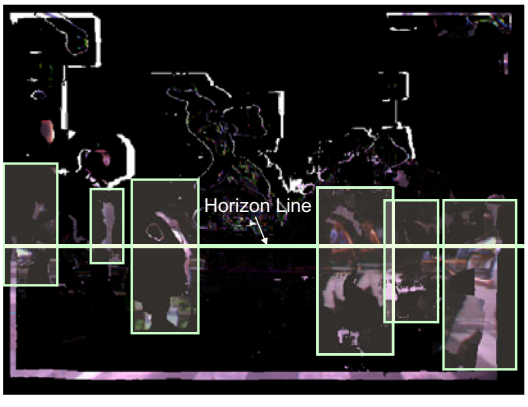

Fig. 7 Map of moving regions ($D_{(u,v)}$) obtained from the synthesized view ($\widehat{\mathbf{I}}^{n+1}$) (Fig. $6(left)$) and the corresponding frame ($\mathbf{I}^{n+1}$) (Fig. $5(right)$)—bounding boxes are only illustrative and have been placed using the information of horizon line position as in [9]

to highlight that although these frames (Fig. 1*(top)*) look quite similar there is a considerable relative displacement between them.

A different scenario is shown in the two consecutive frames presented in Fig. 5. In that scene, the car is reducing the speed to stop for a red light, three pedestrian are crossing the street. Although the vehicle is reducing the speed there is a relative displacement between these consecutive frames (see Fig. 6$(right)$). The synthesized view of frame (n), using the computed 3D rigid displacement, is presented in Fig. 6$(left)$. Finally, the corresponding moving regions map is depicted in Fig. 7. Bounding boxes enclosing moving objects can provide a reliable information to select candidate windows to be used by a classification process (e.g., a pedestrian classifier). In this case, the number of windows would greatly decrease compared to other approaches in the literature, such as 10^8 windows in an exhaustive scan [20] or 2,000 windows in a road uniform sampling [9].

5 Conclusions

This chapter presents a novel and robust approach for moving object detection by registering consecutive clouds of 3D points obtained by an on-board stereo camera. The registration process is only applied over two small sets of 3D points with known correspondences by using key point features extraction and a RANSAC-like technique based on the closed-form solution provided by the unit quaternion method. Then, a synthesized 3D scene is obtained after mapping the whole set of points from the previous frame to the current one. Finally, a map of moving regions is generated by considering the difference between current 3D scene and the synthesized one.

As future work more evolved approaches for combining registered frames will be studied. For instance, instead of only using consecutive frames, temporal windows including more frames are likely to help filtering out noisy areas. Furthermore, color information of each pixel could be used during the estimation of the moving region map.

Acknowledgment. This work was supported in part by the Spanish Ministry of Science and Innovation under Projects TRA2010-21371-C03-01, TIN2010-18856 and Research Program Consolider Ingenio 2010: MIPRCV (CSD2007-00018).

References

1. Amir, S., Barhoumi, W., Zagrouba, E.: A robust framework for joint background/foreground segmentation of complex video scenes filmed with freely moving camera. Pattern Analysis and Applications 46(2), 175–205
2. Benjemaa, R., Schmitt, F.: A solution for the registration of multiple 3D point sets using unit quaternions. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407, pp. 34–50. Springer, Heidelberg (1998)
3. Besl, P., McKay, N.: A method for registration of 3D shapes. IEEE Trans. on Pattern Analysis and Machine Intelligence 14(2), 239–256 (1988)

4. Chen, Y., Medioni, G.: Object modelling by registration of multiple range images. Image Vision Comput. 10(3), 145–155 (1992)
5. Chetverikov, D., Stepanov, D., Krsek, P.: Robust Euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm. Image and Vision Computing 23(1), 299–309 (2005)
6. Chui, H., Rangarajan, A.: A feature registration framework using mixture models. In: MMBIA 2000: Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis, pp. 190–197 (2000)
7. Dahyot, R.: Unsupervised camera motion estimation and moving object detection in videos. In: Proc. of the Irish Machine Vision and Image Processing, Dublin, Ireland (August 2006)
8. Fitzgibbon, A.: Robust registration of 2D and 3D point sets. Image and Vision Computing 21(13), 1145–1153 (2003)
9. Gerónimo, D., Sappa, A.D., López, A., Ponsa, D.: Adaptive image sampling and windows classification for on–board pedestrian detection. In: Proc. Int. Conf. on Computer Vision Systems, Bielefeld, Germany (2007)
10. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proc. of The Fourth Alvey Vision Conference, Manchester, UK, pp. 147–151 (1988)
11. Hashiyama, T., Mochizuki, D., Yano, Y., Okuma, S.: Active frame subtraction for pedestrian detection from images of moving camera. In: Proc. IEEE Int. Conf. on Systems, Man and Cybernetics, Washington, USA, pp. 480–485 (October 2003)
12. Horn, B.: Closed-form solution of absolute orientation using unit quaternions. Journal of the Optical Society of America A 4, 629–642 (1987)
13. Jian, B., Vemuri, B.: A robust algorithm for point set registration using mixture of Gaussians. In: 10th IEEE International Conference on Computer Vision, Beijing, China, October 17-20, pp. 1246–1251 (2005)
14. Kastrinaki, V., Zervakis, M., Kalaitzakis, K.: A survey of video processing techniques for traffic applications. Image and Vision Computing 21(4), 359–381 (2003)
15. Kong, H., Audibert, J., Ponce, J.: Detecting abandoned objects with a moving camera. IEEE Transactions on Image Processing 19(8), 2201–2210 (2010)
16. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 2(60), 91–110 (2004)
17. Ma, Y., Soatto, S., Kosecká, J., Sastry, S.: An Invitation to 3D Vision: From Images to Geometric Models. Springer, New York (2004)
18. Milella, A., Siegwart, R.: Stereo-based ego-motion estimation using pixel tracking and iterative closest point. In: Proc. IEEE Int. Conf. on Mechatronics and Automation, USA (January 2006)
19. Oniga, F., Nedevschi, S., Meinecke, M., To, T.: Road surface and obstacle detection based on elevation maps from dense stereo. In: Proc. IEEE Int. Conf. on Intelligent Transportation Systems, Seattle, USA, pp. 859–865 (September 2007)
20. Oren, M., Papageorgiou, C., Sinha, P., Osuna, E., Poggio, T.: Pedestrian detection using wavelet templates. In: Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition, Puerto Rico (June 1997)
21. Radke, R., Andra, S., Al-Kofahi, O., Roysam, B.: Image change detection algorithms: A systematic survey. IEEE Trans. on Image Processing 14(3), 294–307 (2003)
22. Restrepo-Specht, A., Sappa, A.D., Devy, M.: Edge registration versus triangular mesh registration, a comparative study. Signal Processing: Image Communication 20(9-10), 853–868 (2005)
23. Salvi, J., Matabosch, C., Fofi, D., Forest, J.: A review of recent range image registration methods with accuracy evaluation. Image Vision Computing 25(5), 578–596 (2007)

24. Sappa, A.D., Dornaika, F., Gerónimo, D., López, A.: Registration-based moving object detection from a moving camera. In: Proc. on Workshop on Perception, Planning and Navigation for Intelligent Vehicles, Nice, France (September 2008)
25. Sharp, G., Lee, S., Wehe, D.: ICP registration using invariant features. IEEE Trans. Pattern Anal. Mach. Intell. 24(1), 90–102 (2002)
26. Shimizu, S., Yamamoto, K., Wang, C., Satoh, Y., Tanahashi, H., Niwa, Y.: Moving object detection by mobile stereo omni-directional system (SOS) using spherical depth image. Pattern Analysis and Applications (2), 113–126
27. Taleghani, S., Aslani, S., Saeed, S.: Robust moving object detection from a moving video camera using neural network and kalman filter. In: Iocchi, L., Matsubara, H., Weitzenfeld, A., Zhou, C. (eds.) RoboCup 2008. LNCS, vol. 5399, pp. 638–648. Springer, Heidelberg (2009)
28. Tarel, J.-P., Civi, H., Cooper, D.B.: Pose estimation of free-form 3D objects without point matching using algebraic surface models. In: Proceedings of IEEE Worshop Model Based 3D Image Analysis, Mumbai, India, pp. 13–21 (1998)
29. Vedaldi, A.: An open implementation of the SIFT detector and descriptor. Technical Report 070012, UCLA CSD (2007)
30. Wang, C., Thorpe, C., Thrun, S.: Online simultaneous localization and mapping with detection and tracking of moving objects: theory and results from a ground vehicle in crowded urban areas. In: Proc. IEEE Int. Conf. on Robotics and Automation, Taipei, Taiwan, pp. 842–849 (September 2003)
31. Wang, H., Zhang, Q., Luo, B., Wei, S.: Robust mixture modelling using multivariate t-distribution with missing information. Pattern Recogn. Lett. 25(6), 701–710 (2004)
32. Wange, L., Yung, N.: Extraction of moving objects from their background based on multiple adaptive thresholds and boundary evaluation. IEEE Transactions on Intelligent Transportation Systems 11(1), 40–51 (2010)
33. Wolf, D., Sukhatme, G.: Mobile robot simultaneous localization and mapping in dynamic environments. Autonomous Robots 19(1), 53–65 (2005)
34. Yu, Q., Medioni, G.: Map-enhanced detection and tracking from a moving platform with local and global data association. In: Proc. IEEE Workshops on Motion and Video Computing, Austin, Texas (February 2007)
35. Yu, Q., Medioni, G.: A GPU-based implementation of motion detection from a moving platform. In: Proc. IEEE Workshops on Computer Vision and Pattern Recognition, Anchorage, Alaska (June 2008)
36. Zheng, B., Ishikawa, R., Oishi, T., Takamatsu, J., Ikeuchi, K.: A fast registration method using IP and its application to ultrasound image registration. IPSJ Transactions on Computer Vision and Applications 1, 209–219 (2009)
37. Zheng, B., Takamatsu, J., Ikeuchi, K.: An adaptive and stable method for fitting implicit polynomial curves and surfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(3), 561–568 (2010)