Registration-based moving object detection from a moving camera

Angel D. Sappa, Fadi Dornaika, David Gerónimo and Antonio López

Abstract— **This paper presents 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 stages. Initially, feature points are extracted and tracked through consecutive frames. Then, a RANSAC based approach is used for registering two 3D point sets with known correspondences by means of the quaternion method. Finally, the computed 3D rigid displacement is used to map two consecutive frames into the same coordinate system. Moving objects correspond to those areas with large registration errors. Experimental results, in different scenarios, show the viability of the proposed approach.**

I. INTRODUCTION

In general, moving object detection algorithms assume stationary cameras, which means all frames are registered in the same coordinate system. Therefore typical approaches reduce to background modelling and subtraction (see [1] and [2] for an extensive survey). However, when the camera moves, the problem becomes intricate since it is unfeasible to have a background model. In such a case, moving object detection is generally tackled by using prior-knowledge of the scene together with visual cues. In the current paper the use of *3D image registration* will be explored in order to align consecutive stereo frames into the same coordinate system; then, a 3D frame subtraction is performed to find regions with large misregistration, which theoretically would correspond to moving objects.

A large number of approaches have been proposed in the literature for 3D point registration. Most of these approaches are based on the well-known ICP (*Iterative Closest Point*) algorithm [3], or adaptations of it such as LM-ICP (*Levenberg-Marquardt ICP*) [4], TrICP (*Trimmed ICP*) [5]. All these algorithms have been originally proposed for registering overlapped sets of points corresponding to the 3D surface of a 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., [6], [7], [8], [9]). 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 ([7], [10]), 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.

On the contrary to the robotic field, where the objective is simultaneous localization and mapping, the proposed robust registration aims at detecting moving objects in the scene. It is intended to be used in ADAS (Advanced Driver Assistance Systems) applications, where 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 many object (e.g. pedestrian or vehicle) detection systems. More evolved approaches, focussing on ROIs extracted from vertical surfaces (e.g., v-disparity based [11]) have also been proposed in the literature. Unfortunately, both of the previous approaches could become computationally expensive when thousands of ROIs are extracted. The concept of consecutive frame registration for moving object detection has been recently explored in [12], where 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.

The current paper presents a robust quaternion-based solution for registering dense clouds of 3D points with known sparse correspondences [13], [14], obtained from sequential stereo images. Images are taken from a moving vehicle on an urban scenario containing static and moving objects. The use of additional information, such as inertial sensors [15], [16] or vehicle speed [12] is not required in the proposed approach.

The remainder of this paper is organized as follows. Section II briefly describes the feature point detection and tracking algorithms. Section III presents the proposed robust registration approach. Finally, the frame subtraction technique used to detect moving objects is described. Experimental results in real environments are presented in Section V. Finally, conclusions and future works are given in Section VI.

II. EXTRACTION OF 3D POINT SETS

The first stage of the proposed approach consists in extracting a set of 2D feature points at a given frame and track

This work was partially supported by the Government of Spain under MEC project TRA2007-62526/AUT and research programme Consolider Ingenio 2010: MIPRCV (CSD2007-00018). First author was supported by The Ramón y Cajal Program; Third author was supported by BES-2005-8864 grant.

A. D. Sappa, David Gerónimo and Antonio López are with Computer Vision Center, 08193 Bellaterra, Barcelona, Spain {asappa, dgeronimo, antonio}@cvc.uab.es

F. Dornaika is with the French National Geographical Institute, 94165 Saint-Mandé, France fadi.dornaika@ign.fr

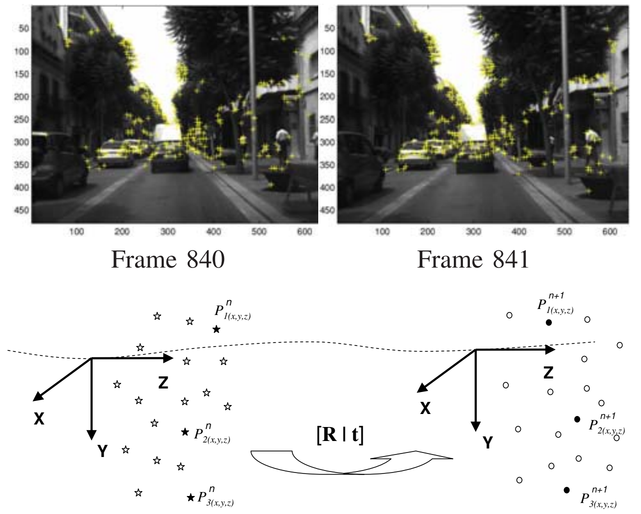


Fig. 1. (top) Feature points detected and tracked through consecutive frames. (bottom) Illustration of feature points represented in the 3D space, together with three couples of points used for computing the 3D rigid displacement: $[\mathbf{R}|\mathbf{t}]$—RANSAC-like technique.

it through the next frame; the 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). We can note that this rigid transform is simply the 3D motion of the camera between frame (n) and frame $(n+1)$. Before going into details in the feature point detection and tracking algorithms a brief description of the used stereo vision system is given.

A. System Setup

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

B. Feature Detection and Tracking

The proposed algorithm starts by selecting a set of feature points, using Harris corner detector [17], in a given image $\mathbf{I}^n$. Feature points, $f^n_{i(u,v)} \subset \mathbf{I}^n$, further 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). More elaborated descriptors, such as SIFT-based feature extraction [19], could be used without affecting the rest of the proposed algorithm.

After selecting a set of feature points and setting a tracking window W_T (9×9 pixels in the current implementation) an iterative feature tracking algorithm is used [20]. Feature points are tracked by minimizing the sum of squared differences between two consecutive frames.

III. Robust Registration

The set of 2D-to-2D point correspondences obtained by tracking, is easily converted to a set of 3D-to-3D points since for every frame we have a quasi dense 3D reconstruction. In the current approach, contrary to 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 could belong to static or moving objects in the scene. Since the camera is moving there are no additional clues to differentiate them easily. In the current work the use of a robust RANSAC-like technique is proposed to find the best rigid transform that maps the 3D points of frame (n) into their corresponding in frame $(n+1)$. The closed-form solution provided by unit quaternions 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 [13].
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 [14] is used to minimize: $\sum_{i=1}^{\#inliers} |P^{n+1}_{i(x,y,z)} - \mathbf{R_q} P^n_{i(x,y,z)} - \mathbf{t_q}|^2$.

IV. 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

[1] [www.ptgrey.com]

[2] Stereo head data uncertainty grows quadratically with depth [18].

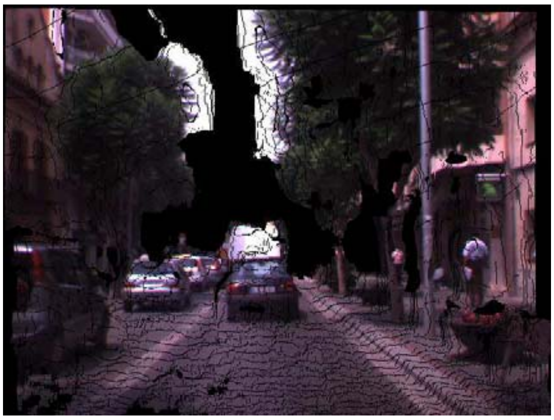


Fig. 2. Synthesized view representing frame (840) in the coordinate system of frame (841), by using the computed 3D rigid displacement: $[\mathbf{R_q}|\mathbf{t_q}]$.

after motion compensation. First, the whole set of 3D data points at frame (n) is mapped by:

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

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 ideally we 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 as follows:

$$\widehat{u}^{n+1}_i = (round)\left(u_0 + f\frac{\widehat{x}^{n+1}_i}{\widehat{z}^{n+1}_i}\right) \ , \qquad (2)$$
$$\widehat{v}^{n+1}_i = (round)\left(v_0 + f\frac{\widehat{y}^{n+1}_i}{\widehat{z}^{n+1}_i}\right)$$

where f denotes the focal length in pixels, (u_0, v_0) represents the coordinates of the camera principal point, and $(\widehat{x}^{n+1}_i, \widehat{y}^{n+1}_i, \widehat{z}^{n+1}_i)$ correspond to the 3D coordinates of the mapped point (1). Fig. 2 shows an illustration of the synthesized view obtained after mapping frame (840) (Fig. 1*(left)*) with its 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:

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

where, τ is a user defined threshold directly related to the camera frame rate (in the current implementation it has been empirically set to 0.1 meters, assuming a 10fps frame rate). 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 3*(top)* presents the map of moving regions resulting from the frame (841) (Fig. 1$(right)$) and the synthesized view corresponding to frame (840) (see Figure 2). Additionally, the difference between the consecutive frames, $(|\mathbf{I}^{840} - \mathbf{I}^{841}|)$, is presented in Figure 3*(bottom)* just to show the relative motion between them.

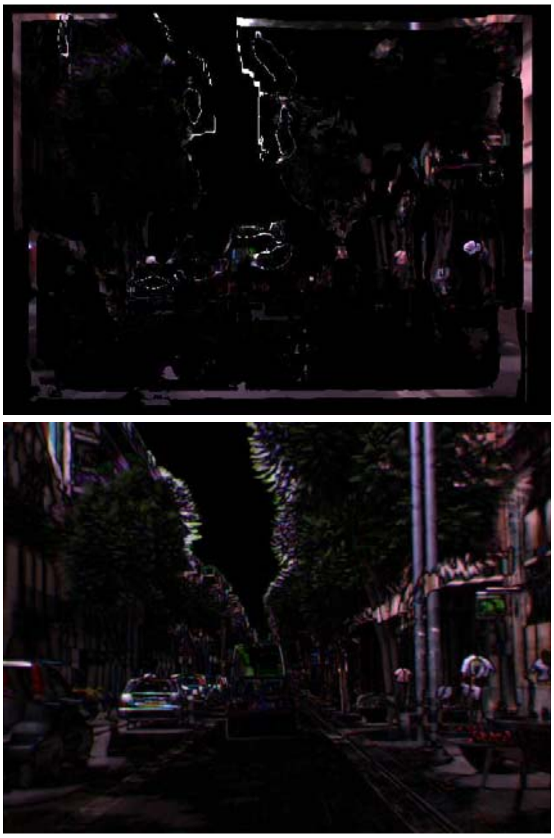
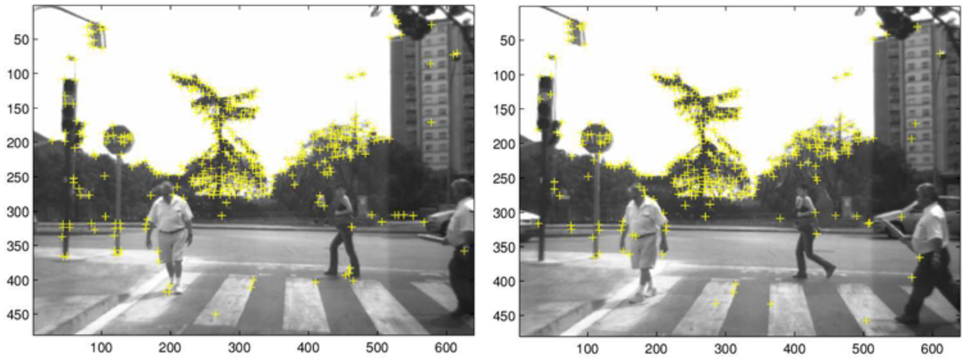


Fig. 3. (top) $D_{(u,v)}$ map of moving regions, from frames (840) and (841) presented in Fig. 1. $(bottom)$ Difference between these consecutive frames: $(|\mathbf{I}^{840} - \mathbf{I}^{841}|)$ to illustrate their relative displacement.

V. EXPERIMENTAL RESULTS

Experimental results with real environments and different vehicle speeds are presented. 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 with a non-optimized C++ code. Although the stereo head can work at a frame rate near to 30 fps experimental results presented in this paper 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 took, on average, 31 ms for registering consecutive frames by using about 300 feature points. Fig. 1 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., [9]) would provide a wrong scene registration. The synthesized view obtained from (840) is presented in Fig. 2. The quality of the registration result can be appreciated in the map of moving regions presented in Fig. 3*(top)*, in particular pay attention at the lamp post, where there is



Frame 298

Frame 299

Fig. 4. Feature points detected and tracked through consecutive frames.

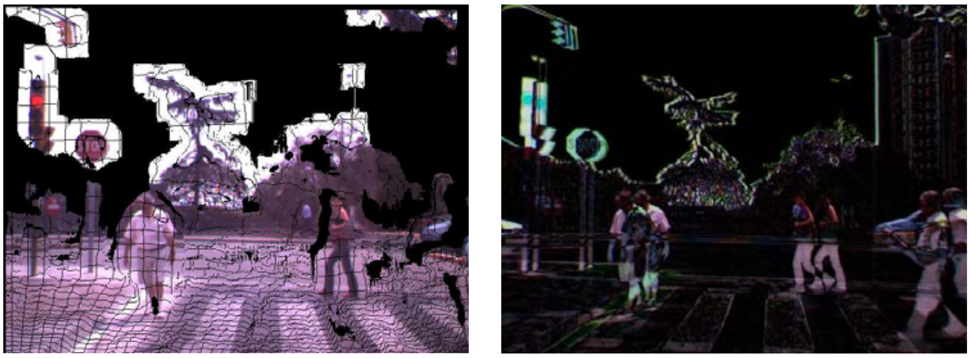


Fig. 5. $(left)$ Synthesized view of frame (298) (Fig. 4$(left)$). $(right)$ Difference between consecutive frames: $(|\mathbf{I}^{298} - \mathbf{I}^{299}|)$ to illustrate their relative displacement (pay special attention at the traffic lights and stop signposts).

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 [18]. Wrong moving regions mainly correspond to hidden areas in frame (n) that are unveiled in frame $(n+1)$. Fig. 3$(bottom)$ presents the difference between consecutive frames $(|\mathbf{I}^{840} - \mathbf{I}^{841}|)$ 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. 4. In that scene, the car is reducing the speed to stop for a red light, three pedestrian are crossing the street. Although the vehicle was reducing the speed there is a relative displacement between these consecutive frames (see Fig. 5$(right)$). The synthesized view of frame (298), using the computed 3D rigid displacement, is presented in Fig. 5$(left)$. Finally, the corresponding moving regions map is depicted in Fig. 6. Bounding boxes enclosing moving objects can provide a reliable information to select ROIs to be used by a classification process (e.g., a pedestrian classifier). In this case, the number of ROIs would greatly decrease compared to other approaches in the literature, such as 10^8 ROIs in an exhaustive scan [21] or 2,000 ROIs in a road uniform sampling [22] (see Fig. 7).

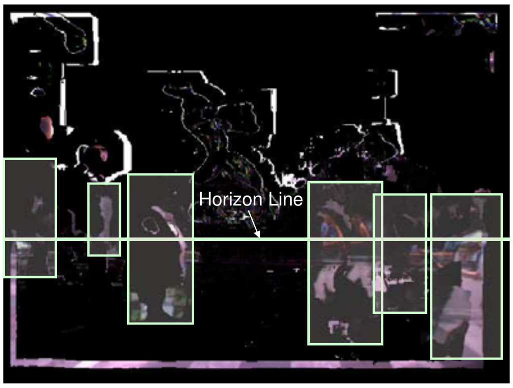


Fig. 6. Map of moving regions ($D_{(u,v)}$) obtained from the synthesized view ($\widehat{\mathbf{I}^{299}}$) (Fig. 5$(left)$) and the corresponding frame ($\mathbf{I}^{299}$) (Fig. 4$(right)$)—bounding boxes are only illustrative and have been placed using the information of horizon line position as in [22].

VI. CONCLUSIONS

This paper 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 a RANSAC-like technique based on the closed-form solution provided by the unit quaternion method. Then, a synthesized 3D scene

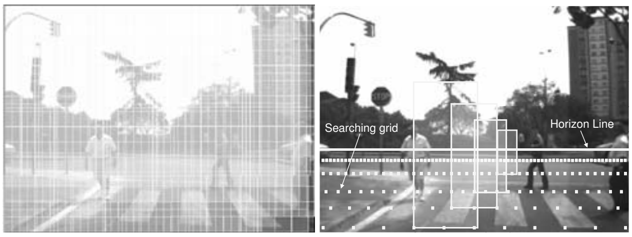


Fig. 7. ($left$) Candidate windows obtained from an exhaustive scan, about 10^8 windows [21]. ($right$) Candidate windows obtained by using a uniform grid based approach, about 2,000 windows [22].

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 synthesized one.

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

References

[1] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis. A survey of video processing techniques for traffic applications. *Image and Vision Computing*, 21(4):359–381, April 2003.

[2] R. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: A systematic survey. *IEEE Trans. on Image Processing*, 14(3):294–307, March 2003.

[3] P. Besl and N. McKay. A method for registration of 3-d shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1988.

[4] A. Fitzgibbon. Robust registration of 2d and 3d point sets. *Image and Vision Computing*, 21(13):1145–1153, December 2003.

[5] D. Chetverikov, D. Stepanov, and P. Krsek. Robust euclidean alignment of 3d point sets: the trimmed iterative closest point algorithm. *Image and Vision Computing*, 23(1):299–309, March 2005.

[6] M. García and A. Solanas. 3d simultaneous localization and modeling from stereo vision. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 847–853, New Orleans, USA, April 2004.

[7] C. Wang, C. Thorpe, and S. Thrun. 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*, pages 842–849, Taipei, Taiwan, September 2003.

[8] C. Wang and C. Thorpe. Simultaneous localization and mapping with detection and tracking of moving objects. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 2918–2924, Washington, USA, May 2002.

[9] A. Milella and R. Siegwart. Stereo-based ego-motion estimation using pixel tracking and iterative closest point. In *Proc. IEEE Int. Conf. on Mechatronics and Automation*, page 21, New York, USA, January 2006.

[10] D. Wolf and G. Sukhatme. Mobile robot simultaneous localization and mapping in dynamic environments. *Autonomous Robots*, 19(1):53–65, July 2005.

[11] R. Labayrade, D. Aubert, and J. Tarel. Real time obstacle detection in stereovision on non flat road geometry through 'V-disparity' representation. In *Proc. IEEE Intelligent Vehicles Symposium*, pages 646–651, Versailles, France, June 2002.

[12] T. Hashiyama, D. Mochizuki, Y. Yano, and S. Okuma. Active frame subtraction for pedestrian detection from images of moving camera. In *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics*, pages 480–485, Washington, USA, October 2003.

[13] B. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4:629–642, April 1987.

[14] R. Benjemaa and F. Schmitt. A solution for the registration of multiple 3d point sets using unit quaternions. In *Proc. of 5th European Conference on Computer Vision*, volume 1407 of *Lecture Notes in Computer Science*, Freiburg, Germany, June 1998. Springer.

[15] J. Lobo and J. Dias. Vision and inertial sensor cooperation using gravity as a vertical reference. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(12):1597–1608, 2003.

[16] J. Lobo and J. Dias. Relative pose calibration between visual and inertial sensors. *International Journal of Robotics Research*, 26(6):561–575, 2007.

[17] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. of The Fourth Alvey Vision Conference*, pages 147–151, Manchester, UK, September 1988.

[18] F. Oniga, S. Nedevschi, M. Meinecke, and T. To. Road surface and obstacle detection based on elevation maps from dense stereo. In *Proc. IEEE Int. Conf. on Intelligent Transportation Systems*, pages 859–865, Seattle, USA, September 2007.

[19] S. Se and M. Brady. Road feature detection and estimation. *Machine Vision and Applications*, 14(3):157–165, July 2003.

[20] Y. Ma, S. Soatto, J. Kosecká, and S. Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer-Verlag New York, 2004.

[21] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Puerto Rico, June 1997.

[22] D. Gerónimo, A.D. Sappa, A. López, and D. Ponsa. Adaptive image sampling and windows classification for on–board pedestrian detection. In *Proc. Int. Conf. on Computer Vision Systems*, Bielefeld, Germany, 2007.