On Combining Visual SLAM and Dense Scene Flow to Increase the Robustness of Localization and Mapping in Dynamic Environments

Pablo F. Alcantarilla, José J. Yebes, Javier Almazán, Luis M. Bergasa

***Abstract*— In this paper, we introduce the concept of dense scene flow for visual SLAM applications. Traditional visual SLAM methods assume static features in the environment and that a dominant part of the scene changes only due to camera egomotion. These assumptions make traditional visual SLAM methods prone to failure in crowded real-world dynamic environments with many independently moving objects, such as the typical environments for the visually impaired. By means of a dense scene flow representation, moving objects can be detected. In this way, the visual SLAM process can be improved considerably, by not adding erroneous measurements into the estimation, yielding more consistent and improved localization and mapping results. We show large-scale visual SLAM results in challenging indoor and outdoor crowded environments with real visually impaired users. In particular, we performed experiments inside the Atocha railway station and in the city-center of Alcalá de Henares, both in Madrid, Spain. Our results show that the combination of visual SLAM and dense scene flow allows to obtain an accurate localization, improving considerably the results of traditional visual SLAM methods and GPS-based approaches.**

I. Introduction

Autonomous navigation is of extreme importance for those who suffer from visual impairment problems. Without a good autonomy, visually impaired people depend on other factors or other persons to perform typical daily activities. Within this context, a system that can provide robust and accurate localization of a visually impaired user in urban city-like environments or indoor ones is much more than desirable.

Nowadays, most of the commercial solutions for visually impaired localization and navigation assistance are based on the Global Positioning System (GPS). However, these solutions are not suitable enough for the visually impaired community mainly for two reasons: the low accuracy in urban-environments (errors about the order of several meters) and signal loss due to multi-path effect or line-of-sight restrictions. Moreover, GPS does not work if an insufficient number of satellites are directly visible. Therefore, GPS cannot be used in indoor environments.

All the authors are with the Department of Electronics, University of Alcalá. Alcalá de Henares, Madrid, Spain. e-mail: pablo.alcantarilla,javier.yebes,javier.almazan, bergasa@depeca.uah.es

Computer vision-based approaches offer substantial advantages with respect to GPS-based systems and constitute a promising alternative to address the problem. By means of visual SLAM techniques [1], [2], it is possible to build an incremental map of the environment, providing at the same time the location and spatial orientation of the user within the environment. In addition, compared to other sensory modalities computer vision can also provide a very rich and valuable perception information of the environment such as for example obstacle detection [3] or 3D scene understanding [4].

Most of visual SLAM systems in the literature [5], [6] need to assume static features in the environment and that a dominant part of the scene changes only with the camera egomotion. As a result, these approaches are prone to failure in crowded scenes with many independently moving objects. Even though some of the outliers can be detected by geometric constraints validation, one needs to take special care about not introducing any outlier in the 3D reconstruction process or otherwise the estimated map and camera trajectory can diverge considerably from the real solution.

In this paper, we will show how traditional stereo visual SLAM algorithms can be improved by means of the detection of moving objects, thanks to a dense scene flow [7] representation of the environment. In this way, visual SLAM algorithms can obtain more robust and accurate localization and mapping results in crowded and dynamic environments with many independent moving objects. The rest of the paper is organized as follows: In Section II, we briefly review the main components of our stereo visual SLAM system. In Section III, the formulation of the dense scene flow and detection of moving objects are described. Then, we show in Section IV, how visual SLAM and dense scene flow can be combined in order to detect moving objects in the image and avoid adding erroneous measurements into the SLAM process. Finally, in Section V, we show main conclusions and experimental results considering challenging environments with many independent moving objects.

II. Stereo Visual SLAM

In this section, we briefly review the main components of our stereo visual SLAM system. Our visual SLAM system uses a stereo camera as the only sensor and it

is based on a combination of stereo visual odometry [8] and a hierarchical structure and motion refinement by means of Bundle Adjustment (BA) [9].

A. Stereo Visual Odometry

We estimate the relative camera motion by matching detected features between two consecutive frames. Features are detected by means of Harris corner detector [10] at different scale levels. We detect features only for the left image of the stereo pair. Then, we find the correspondences of the 2D features in the right image by accessing the disparity map and compute the 3D coordinates of the point by means of standard stereo geometry equations.

For each detected 2D feature in the left image we also extract a MU-SURF [11] descriptor vector of length 64. Once we have computed the features descriptors, we find the set of putatives matches between two consecutive frames by matching their associated list of descriptors vectors. Then, we estimate the relative camera motion using the standard three-point algorithm in a Random Sample Consensus (RANSAC) [12] setting. The resulting relative camera motion is translated to a global coordinate frame and then used by the mapping management module.

B. Mapping Management

By means of stereo visual odometry, we estimate the camera motion between consecutive frames. When the accumulated motion in translation or rotation is higher than a fixed threshold we create a new keyframe. This keyframe, will be optimized later in a local BA procedure. In the local BA process, 3D points and camera poses are refined simultaneously through the sequence. Similar to [13] we use a sliding window BA approach over the last N keyframes (e.g. 10), reducing the computational complexity.

We perform an intelligent management of features into the map, in order to produce an equal distribution of feature locations over the image. While adding a new feature to the map, we also store its associated appearance descriptor and 3D point location. Then, we try to match the feature descriptor against the detected new 2D features on a new keyframe by matching their associated descriptors in a high probability search area. In this way, we can create for a map element, *feature tracks* that contain the information of the 2D measurements of the feature (both in left and right views) in several keyframes. Then, this information is used as an input in the local BA procedure. Features are deleted from the map when the mean re-projection error per frame in the 3D reconstruction is higher than a fixed threshold (e.g. 3 pixels). Notice here that we avoid adding erroneous new features by discarding those that are located on moving objects. In Section IV, we will explain how to identify areas in the image that belong to moving objects, thanks to a dense scene flow representation.

By means of appearance based methods, loop closure situations can be detected and the residual error in the 3D reconstruction can be corrected by means of pose-graph optimization techniques such as Smoothing and Mapping (SAM) [14]. After the pose-graph optimization, the solution can be further refined in a global BA procedure.

III. Dense Scene Flow and Detection of Moving Objects

One of the advantages of stereo vision against monocular one, is that we can exploit the information from four images at once, obtaining dense disparity maps (between the left and right stereo views at each frame) and dense 2D optical flow correspondences (between two consecutive frames). Since for every pixel that has a valid disparity value we know its 3D position (with respect to the camera coordinate frame) and the associated dense 2D optical flow (between two consecutive images), a dense scene flow [7] representation can be obtained, describing the 3D motion of the world points. Fig. 1 depicts an example of the four images that can be used in order to compute a dense scene flow representation of the environment.

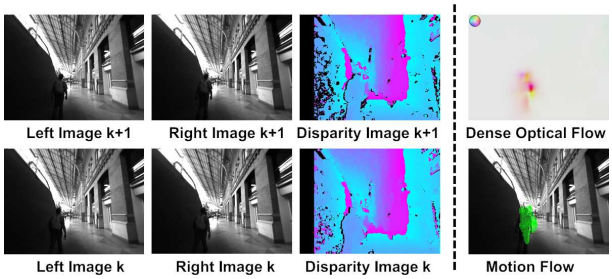


Fig. 1. Process of computing a dense scene flow representation. Best viewed in color.

Scene flow was introduced in [7], and should be considered as an essential algorithm for studying 3D motion in scenes. Scene flow describes the 3D motion of the points in the scene, whereas optical flow describes the 2D motion of the pixels in the image. Recently, scene flow techniques have been proposed for intelligent vehicles applications [15], [16]. The work of Wedel et al. [16] can be considered as the main reference for computing dense scene flow from stereo images. In this work, the authors proposed a variational framework for estimating dense stereo correspondences and dense 2D optical flow correspondences between consecutive images and also how dense scene flow estimates can be used for moving objects segmentation. In [15], a sparse scene flow representation of the scene is obtained in order to detect moving objects in road-traffic urban environments. Those adjacent points that describe a similar scene flow are clustered and considered to belong to the same rigid object.

However, scene flow computation considering a moving stereo pair with 6-Degrees of Freedom (DoF) and crowded urban scenarios with many independently moving objects is more challenging than for common intelligent vehicle applications. In these kind of Advanced Driver Assistance Systems (ADAS), it is possible to use the prior information from inertial sensors to compensate for camera egomotion. Despite of this, some approaches neglect the effect of camera rotation [17] or do not perform any kind of egomotion compensation [15].

In stereo visual SLAM applications, it is necessary to consider the camera rotation and translation for egomotion compensation for a reliable scene flow computation. SLAM and scene flow computation from a stereo camera in highly crowded environments can be a difficult task. These scenarios are extremelly challenging since we can have fast camera motion, changes in lighting conditions, motion blur and many independently moving objects such as pedestrians that on occasions can almost cover the entire image view. For example, Fig. 2 depicts few samples of typical environments where visually impaired users have to deal with during navigation tasks.

Fig. 2. These three images depict some examples of the difficult challenging scenes that we can have in real-world crowded environments for the visually impaired.

By means of dense scene flow estimates, we can derive motion likelihoods that can be used to segment moving objects, aiding the visual SLAM process. To the best of our knowledge, this is the first time that dense scene flow has been used in the context of visual SLAM for dealing with moving objects in crowded and highly dynamic scenarios aiding the visual SLAM estimation. In Section III-A we will derive the set of equations that are necessary in order to obtain a dense scene flow representation. Finally, we will show in Section III-B, how to obtain motion likelihoods by means of a dense scene flow representation, and how these likelihoods can be used to identify moving objects in the image. In this way, we obtain more robust camera egomotion estimates and delete those features located on moving objects from the SLAM process, yielding superior 3D reconstruction results.

A. Scene Flow Computation

Given dense disparity maps between the two images of a stereo rig and dense optical flow estimates between two consecutive images, we can estimate a dense 3D scene flow, describing the 3D motion of world points. Using this information and considering that the images are rectified and undistorted, the 3D motion vector associated to two correspondent points can be computed considering the following equations:

$$\begin{pmatrix} X_{t+1} \\ Y_{t+1} \\ Z_{t+1} \end{pmatrix} = \frac{B}{u'_R - u'_L} \cdot \begin{pmatrix} u'_L - u_0 \\ v' - v_0 \\ f \end{pmatrix} \quad (1)$$

$$\begin{pmatrix} X_t \\ Y_t \\ Z_t \end{pmatrix} = \frac{B}{u_R - u_L} \cdot \mathbf{R} \cdot \begin{pmatrix} u_L - u_0 \\ v - v_0 \\ f \end{pmatrix} - \mathbf{T} \quad (2)$$

where Eq. 1 describes the coordinates of a 3D point at time instant $t+1$, and Eq. 2 describes the 3D coordinates of a 3D point at time t referenced to the camera coordinate frame at time $t+1$. $\mathbf{R}$ and $\mathbf{T}$ are respectively the rotation matrix and the translation vector of the camera between the two time steps. Notice that if the camera is stationary, the rotation matrix is equal to an identity matrix and the translation vector components are zero. B is the baseline of the stereo rig, f is the camera focal length and u_0, v_0 are the coordinates of the camera principal point.

Considering the above two equations, the 3D translation or motion vector $\mathbf{M}$ can be expressed as follows:

$$\mathbf{M} = \left[\begin{pmatrix} X_{t+1} \\ Y_{t+1} \\ Z_{t+1} \end{pmatrix} - \begin{pmatrix} X_t \\ Y_t \\ Z_t \end{pmatrix} \right] \quad (3)$$

In this work, we have used the dense optical flow method described in [18], for obtaining dense 2D correspondences between consecutive frames $\{(u_L, v) \rightarrow (u'_L, v')\}$. This algorithm computes the 2D motion between consectutive frames by means of minimizing a cost function that approximates each neighborhood of both frames by quadratic polynomials. In addition, this algorithm is included in the OpenCV library [1] and exhibits good performance. Notice here that other advanced variational optical flow methods could have been employed [19], [20], but the derivation of the scene flow and residual motion likelihoods remain the same.

B. Detection of Moving Objects by Motion Likelihoods

Once we have computed the 3D motion vector for each pixel in the image, it is necessary to take into account the uncertainties of the scene flow vector in order to derive robust motion likelihoods. If we try to segment objects based on the modulus of the 3D motion vector, the segmentation is prone to errors due to measurement noise and depth uncertainty in the stereo reconstruction process. Therefore, it is much more robust to take all the uncertainties of the problem into account and derive a metric based on the Mahalanobis distance. By means of the Mahalanobis distance, a metric can be derived in order to identify possible moving objects in the scene [15], [16].

[1] Available from http://sourceforge.net/projects/opencvlibrary/

First, we need to define the uncertainties of our measurements. Then, the resulting error of the 3D motion vector can be computed by linear error propagation and the Jacobian of the motion vector with respect to the measurements. Let us denote the scene flow vector of measurements $\mathbf{z_{SF}}$ as:

$$\mathbf{z_{SF}} = (u'_L, u'_R, v', u_L, u_R, v, t_x, t_y, t_z, q_x, q_y, q_z)^t \quad (4)$$

where (u'_L, u'_R, v') are the image coordinates for a given stereo point at time instant $t+1$ and (u_L, u_R, v) are the image coordinates for the same corresponding point at time instant t. The set of parameters (t_x, t_y, t_z) and (q_x, q_y, q_z) represent respectively the 3D translation vector and the rotation matrix parametrized by means of a unit quaternion between the two time instants. This translation vector and rotation matrix can be obtained directly from the visual odometry procedure described in Section II-A.

The covariance of the scene flow $\mathbf{\Sigma_{SF}}$ is obtained as:

$$\mathbf{\Sigma_{SF}} = \mathbf{J_{SF}} \cdot \mathbf{S_{SF}} \cdot \mathbf{J_{SF}}^t \quad (5)$$

where $\mathbf{J_{SF}}$ is the Jacobian of the scene flow with respect to the vector of measurements $\mathbf{z_{SF}}$ and $\mathbf{S_{SF}}$ is the measurement noise matrix. We consider a pixelic standard deviation of ± 1 pixel for all the pixelic values that are involved in the measurement scene flow $\mathbf{z_{SF}}$. Regarding the translation and orientation variances, these quantities can be obtained from the visual odometry estimation, which is formulated as a nonlinear least squares minimization. When the sum of squares represents the goodness of fit of a nonlinear model to observed data, there are several approximations to obtain the covariance matrix of the estimated regression coefficients. These approximations estimate the Hessian of a function in the neighbourhood of a solution by means of the Jacobian product $J(x)^t \cdot J(x)$ being $J(x)$ the Jacobian matrix of the function $f(x)$, thereby avoiding to compute or approximate any second-order derivatives. For more information about how to compute these covariances estimates we recommend the reader to check the following references [21], [22].

For a given pixel in the image (u, v), we can evaluate the associated Mahalanobis distance of the 3D motion vector in order to compute a residual motion likelihood:

$$\xi_{motion}(u, v) = \sqrt{(\mathbf{M}^t \cdot \mathbf{\Sigma_{SF}}^{-1} \cdot \mathbf{M})} \quad (6)$$

Assuming a stationary world and Gaussian error propagation, Eq. 6 can be used to identify possible outliers or moving points. Stationary points will exhibit low residual motion likelihoods, whereas moving points will yield higher deviations from zero. Then, by thresholding on the residual motion likelihood we can identify those parts in the scene that are static or that belong to moving objects. In this way, we can identify those points in the image that are not static, deleting them from the SLAM process yielding more robust 3D reconstruction results. The squared Mahalanabois distance $\xi_{motion}(u, v)$ follows a χ^2 distribution, and outliers can be identified by thresholding according to this distance.

IV. Combining Visual SLAM and Dense Scene Flow

Visual odometry is a key component in our visual SLAM algorithm, since we use the visual odometry information as an initialization for the structure and motion in the reconstruction. Visual odometry assumes that a dominant part of the scene changes only due to camera egomotion. There can be some situations in crowded and dynamic environments where some visual odometry correspondences declared as inliers in the RANSAC step, will belong to moving objects, yielding wrong and inconsistent camera pose estimates. Those outliers can be detected if we have some prior information about the position of the moving objects in the image. Therefore, in order to use the information from the dense scene flow representation, we obtain a more robust visual odometry estimate by means of a two-step approach:

1) First, we obtain visual odometry estimates between two consecutive images. With the resulting camera egomotion and associated uncertainty, we build a dense scene flow representation of the environment. Even though the visual odometry estimate can be corrupted due to the presence of some outliers, we can use this visual odometry estimate as an initialization for building an approximate dense scene flow representation.
2) Second, from the dense scene flow representation and derived residual motion likelihoods, we detect those possible visual odometry inliers that are located on moving objects and discard those from the set of correspondences. Then, we re-estimate visual odometry without the discarded set of correspondences. In this way, we can obtain more robust visual odometry estimates that will be used to create consistent priors on the 3D structure and camera motion that will be incorporated in the visual SLAM estimation.

Fig. 3 depicts one comparison of visual odometry with and without moving objects detection by means of the residual motion likelihoods obtained from the dense scene flow. Even though RANSAC can detect most of the outliers or wrong correspondences, in extremelly challenging scenarios where moving objects can cover almost the whole image view there can be some remaining correspondences declared as inliers that belong to moving objects. By means of the dense scene flow representation, these areas can be identified and visual odometry can be re-estimated without the wrong set of correspondences, improving considerably the egomotion results.

Once we have computed the residual motion likelihoods for every pixel in the current image, we can create

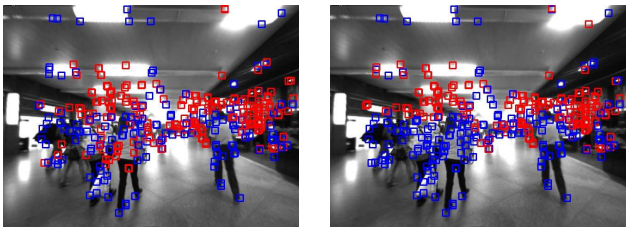


Fig. 3. Visual odometry in the presence of moving objects. Inliers are depicted in red, whereas outliers are depicted in blue. (a) Without moving objects detection (b) With moving objects detection. Best viewed in color.

a moving objects image mask. Those pixels that have a residual motion likelihood higher than a fixed threshold are discarded from the visual SLAM process, in order to avoid adding erroneous measurements. According to our experiments, we can only detect moving objects in a reliable way up to a distance of approximately 5 m. Detection of objects in a far range is more difficult due to the high errors and nonlinearities introduced by the stereo reconstruction and wrong optical flow estimates due to the small size of the objects. Fig. 4 depicts two different examples of satisfactory detection of moving objects, one from an indoor dataset and the other one from an outdoor dataset. Notice how there are no tracked features (depicted in red) on the moving objects, since these features are discarded from the SLAM process.

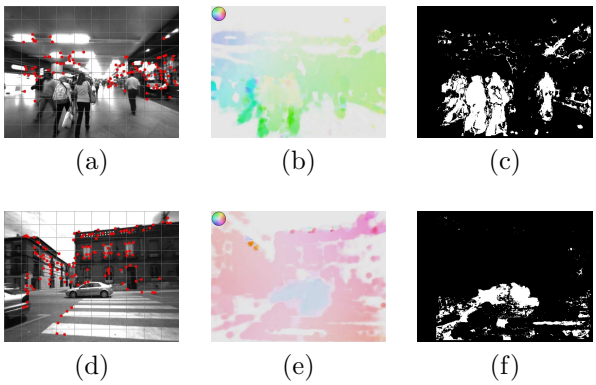


Fig. 4. Detection of moving objects by residual motion likelihoods. First row, experiment inside Atocha railway station. Second row, experiments in the city of Alcalá de Henares. (a,d) Original image with the tracked SLAM features in red (b,e) Dense optical flow image (c,f) Mask of moving objects. For the dense optical flow images, the color encodes the direction and the saturation encodes the magnitude of the flow. For the mask of moving objects images, a pixel of white color means that the pixel belongs to a moving object. Best viewed in color.

One of the problems of the scene flow computation is that the algorithm can detect some pixels in the image as moving objects, when in fact those pixels belong to static points due to measurement noise or optical flow problems. In general, most dense optical flow methods are not able to handle properly real non-artificial scenarios and textureless regions yielding constant image flow estimates over these areas. These constant image flow estimates in textureless regions do not correspond to the real observed flow. However, in visual SLAM applications this is not a big problem, since in textureless regions there are no detected 2D features at all, and even if some features are detected, these features are difficult to be tracked successfully during a large number of frames. Fig. 5 depicts one example in which some static points located in the floor are detected as moving points. Notice also that in these textureless areas there are no detected 2D features.

Fig. 5. Problems of dense scene flow estimation in textureless areas. (a) Image with the tracked features depicted in red (b) Dense optical flow (c) Mask of moving objects. Notice that in the textureless areas there are no features of interest. Best viewed in color.

V. Results and Discussion

Our vision-based system aid for the visually impaired consists of a stereo camera connected through a fireware cable to a small laptop for recording and processing the images. Fig. 6 depicts one image of our vision-based system aid for the visually impaired.

Fig. 6. The stereo camera system is attached to chest of the visually impaired user by means of a non-invasive orthopedic vest. Then the camera is connected to a small laptop by means of a fireware cable.

We conducted large-scale visual SLAM experiments with visually impaired users in highly dynamic environments, with many independently moving objects such as pedestrians or cars. We performed experiments inside the Atocha railway station (Madrid, Spain) and in a crowded area of the city center of Alcalá de Henares (Madrid, Spain). In these experiments, we were mainly interested in evaluating the performance of visual SLAM approaches in these kind of crowded environments, to

know if visual SLAM approaches can be used successfully in future navigation applications for the visually impaired. For this purpose, the visually impaired user received several indications before the start of the sequence about going from one starting point to a final destination.

For the mentioned experiments, we have used the Bumblebee2 stereo camera sold by Point Grey Research[2]. This commercial stereo rig provides highly accurate camera calibration parameters and also stereo rectification and dense depth map generation on-chip. The camera baseline is 12 cm and the horizontal field of view is of 100°. The image resolution was 640×480 pixels and the acquisition frame rate was about 15 frames per second, considering B&W images.

Fig. 7 depicts some image views from different viewpoints of the sparse 3D point map from the Atocha railway station and Alcalá de Henares sequences, using the visual SLAM algorithm with moving objects detection by means of residual motion likelihoods. The final Atocha sparse 3D reconstruction comprises of 65,584 3D map points and 2060 camera poses that correspond to the set of reconstructed keyframes. In contrast, the final Alcalá 3D reconstruction comprises of 64,360 3D map points and 1483 camera poses.

(a) (b)

Fig. 7. Image views of the sparse 3D point cloud reconstruction from the Atocha railway station (a) and Alcalá de Henares (b) sequences. Each 3D point is depicted by they grey image value when the point was added to the map for first time.

A. Atocha Railway Station Experiment

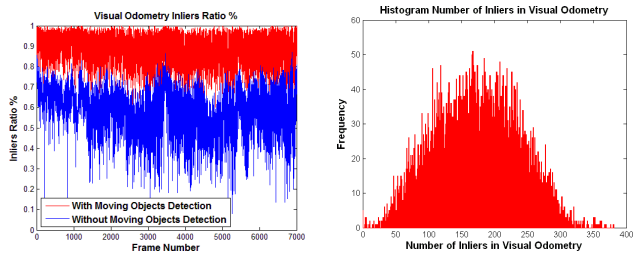
We performed a sequence in which a visually impaired user entered into the Atocha railway station and had to go to the entrance of the underground station, wich is located inside the railway one. Then from the entrance of the underground station, the user had to come back to the same starting place of the route in order to close the loop and correct the accummulated drift in the trajectory. The total length of the route (round trip) was approximately 647 m. The sequence comprises of a total number of 7,109 stereo frames and the total length in time of the experiment was 11 minutes. Fig. 8 depicts some image samples from the experiment inside the Atocha railway station.

[2]For more information, please check the following url: http://www.ptgrey.com/products/stereo.asp

(a) (b) (c)

Fig. 8. (a) Start of the route (b) One image sample inside the railway station (c) End of the route: Entrance to the underground station.

Fig. 9(a) depicts a comparison of the inliers ratio with respect to the stereo visual odometry step for the Atocha sequence. As it can be observed when we incorporate the moving objects detection (MOD) module into the SLAM system, the final inliers ratio in the visual odometry estimation increases considerably. This is due to the fact that thanks to the dense scene flow representation and the derived motion likelihoods, we are able to identify possible areas in the image that may belong to moving objects. With this information we can re-estimate again visual odometry without the wrong set of correspondences, yielding improved egomotion estimates. In contrast, Fig. 9(b) depicts the histogram of the number of inliers in the visual odometry estimation. As it can be observed, there are several frames in which the number of inliers in the visual odometry estimation is below 50. Those situations correspond to images where almost the whole image view is covered by moving objects. Notice that as a default option we tried to extract 400 stereo features per frame in order to find correspondences with the previous frame for the visual odometry estimation.

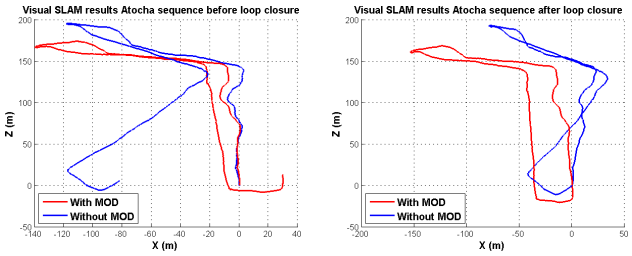


(a) (b)

Fig. 9. Visual odometry results considering moving objects detection, Atocha sequence. (a) Comparison of the inliers ratio in visual odometry when using the moving objects detection module (b) Histogram of the number of inliers in visual odometry with moving objects detection by residual motion likelihoods. Best viewed in color.

Fig. 10(a) depicts the trajectory performed by the visually impaired user in the Atocha sequence before the loop closure correction, considering the visual SLAM algorithm with (in red) and without (in blue) moving objects detection. In contrast, Fig. 10(b) depicts the same comparison after the loop closure correction by means of pose-graph optimization techniques and a subsequent global BA optimization. It can be observed that

the obtained camera trajectory considering the moving objects detection is more similar to the real camera trajectory and fits to the real shape of the Atocha railway station. However, without the moving objects detection the estimated camera trajectory is completely inconsistent with the real-performed trajectory.



(a) (b)

Fig. 10. Comparison of visual SLAM estimated camera trajectories, Atocha sequence. (a) Before loop closure (b) After loop closure. Best viewed in color.

At the same time we grabbed the sequence, we employed a wheel odometer for estimating the total length in m of the trajectory. Then, we compared the estimated total length of the wheel odometer with respect to the estimated trajectory lengths obtained in the visual SLAM experiments. The estimated total length of the trajectory considering the wheel odometer was 647.00 m, whereas for the visual SLAM was 646.07 m and 641.37 m, with and without the detection of moving objects respectively. As we can observe, the estimated length of the visual SLAM algorithm with moving objects detection is very close to the ground truth length with a difference of about 1 m in a total trajectory of 647 m length.

B. Alcalá de Henares Experiment

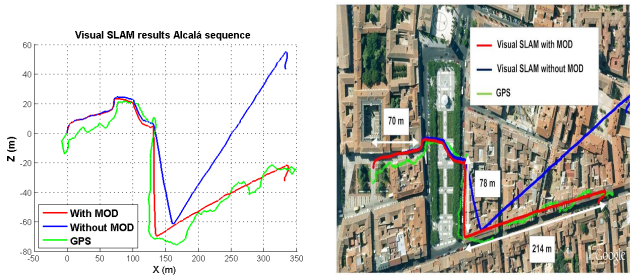
For this scenario, we mapped an area of special cultural interest in the city-center of Alcalá de Henares. The trip starts at the facade of the University of Alcalá and finishes at the Cervantes house, passing through the Mayor street. This street is the most crowded one in the city of Alcalá and it is a very popular commercial area. Lining the street there are arcades supported on columns dating from the 19th century. The total length of the route was approximately 447 m. The sequence comprises of a total number of 5,592 stereo frames and the total length in time of the experiment was approximately 10 minutes. From the facade of the University, the user passed Cervantes Square and walked through one of the sides of the square, which is an arcade supported on columns, and then the user rotated and headed to Mayor street going in a straight way for approximately 214 m. Then, the user crossed the street and finally reached Cervantes house. Fig. 11 depicts some image samples of the conducted experiments in Alcalá.

Fig. 12(a) depicts the trajectory performed by the visually impaired user in the Alcalá de Henares sequence, considering the visual SLAM algorithm with (in red) and

(a) (b) (c)

Fig. 11. (a) Start of the route: Facade of the University of Alcalá (b) Mayor street (c) End of the route: Cervantes house.

without (in blue) the moving objects detection module, and also the estimated trajectory by means of a commercial GPS (in green). As we can observe, the visual SLAM without the moving objects detection module is not able to estimate the real camera trajectory, showing higher errors when estimating the two big rotations that are present in the sequence. In contrast, we can appreciate how by means of the moving objects detection module, the estimated trajectory is in correspondence with the real trajectory. GPS estimated trajectory is also similar to the one obtained with our visual SLAM and moving objects detection module, however there are some places in the sequence where the standard deviation of GPS measurements is big, in ocassions higher than 10 m. These situations correspond to areas where the user was walking close to buildings, or when the user was walking through the arcades in Mayor street or one of the sides of Cervantes Square. In those areas, GPS is prone to failure due to low satellite visibility conditions. Fig. 12(b) depicts the same comparison but displaying results onto an aerial image view of the sequence.



(a) (b)

Fig. 12. Comparison of visual SLAM and GPS estimated camera trajectories, Alcalá de Henares sequence. (a) Visual SLAM with and without moving objects detection and GPS (b) Aerial image view of the sequence. Best viewed in color.

The estimated total length of the trajectory considering the wheel odometer was 447.00 m, whereas for the visual SLAM was 449.79 m and 451.54 m, with and without the detection of moving objects respectively.

VI. Conclusions and Future Work

In this paper, we have shown that it is possible to obtain accurate visual SLAM results in extremely challenging large-scale environments with many independently moving objects. This is possible, due to the detection of moving objects in the image by means of a dense scene

flow representation and from derived residual motion likelihoods. When this object detection module is added to the visual SLAM pipeline, we can improve considerably visual odometry estimates and consequently, we obtain more accurate localization and mapping results in highly dynamic environments. We think that our results can be improved considerably in the next future from better dense scene flow representations [23], [20].

We are also interested in improving the capabilities of visual SLAM and SfM approaches in order to deal with moving objects in the scene. We think that the combination of a robust dense scene flow representation and the tracking of the moving objects [24] can yield a very robust visual SLAM method that can be used in challenging and crowded environments.

The visual SLAM module explained in this paper is an important part of a mobility system towards the autonomous navigation of the visually impaired. Once a persistent map of the environment is created by means of visual SLAM, this map can be used for localization [25] or topological navigation [26] purposes. Given a prior map of the environment and an estimate of the localization of the user within the environment, navigation commands can be computed and transmitted by audio devices to the visually impaired users. We are doing experiments with *audio bone conducting*, which is a non-invasive technology that allows visually impaired users to listen to other important sound sources in the environment (e.g. vehicles) while receiving navigation commands.

VII. Acknowledgments

This work was supported in part by the Community of Madrid under grant CM: S-0505/DPI/000176 (RoboCity2030 Project) and by the Spanish Science and Innovation Ministry under grant TRA2011-29001-C04-01 (ADD-Gaze Project). The authors would like to thank ONCE and Technosite for their support during the experiments with visually impaired users.

References

[1] J. M. Saéz, F. Escolano, and A. Peñalver, "First steps towards stereo-based 6DOF SLAM for the visually impaired," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, San Diego, USA, 2005.

[2] V. Pradeep, G. Medioni, and J. Weiland, "Robot vision for the visually impaired," in *CVAVI10, IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, CA, USA, 2010.

[3] J. M. Saéz and F. Escolano, "Stereo-based aerial obstacle detection for the visually impaired," in *European Conference on Computer Vision (ECCV) / Workshop on Computer Vision Applications for the Visually Impaired (CVAVI)*, Marselle, France, 2008.

[4] A. Geiger, M. Lauer, and R. Urtasun, "A generative model for 3D urban scene understanding from movable platforms," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, USA, June 2011.

[5] K. Konolige and M. Agrawal, "FrameSLAM: from bundle adjustment to real-time visual mapping," *IEEE Trans. Robotics*, vol. 24, no. 5, pp. 1066–1077, Oct 2008.

[6] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid, "REAL: A system for large-scale mapping in constant-time using stereo," *Intl. J. of Computer Vision*, 2010.

[7] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade, "Three-dimensional scene flow," in *Intl. Conf. on Computer Vision (ICCV)*, 1999.

[8] M. Kaess, K. Ni, and F. Dellaert, "Flow separation for fast and robust stereo odometry," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Kobe, Japan, 2009.

[9] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment – a modern synthesis," in *Vision Algorithms: Theory and Practice*, ser. LNCS, W. Triggs, A. Zisserman, and R. Szeliski, Eds. Springer Verlag, Sep 1999, pp. 298–375.

[10] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Fourth Alvey Vision Conference*, 1988, pp. 147–151.

[11] M. Agrawal, K. Konolige, and M. R. Blas, "CenSurE: Center Surround Extremas for realtime feature detection and matching," in *Eur. Conf. on Computer Vision (ECCV)*, 2008.

[12] R. Bolles and M. Fischler, "A RANSAC-based approach to model fitting and its application to finding cylinders in range data," in *Intl. Joint Conf. on AI (IJCAI)*, Vancouver, Canada, 1981, pp. 637–643.

[13] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Generic and real-time structure from motion using Local Bundle Adjustment," *Image and Vision Computing*, vol. 27, pp. 1178–1193, 2009.

[14] F. Dellaert and M. Kaess, "Square Root SAM: Simultaneous localization and mapping via square root information smoothing," *Intl. J. of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, Dec 2006.

[15] P. Lenz, J. Ziegler, A. Geiger, and M. Roser, "Sparse scene flow segmentation for moving object detection in urban environments," in *IEEE Intelligent Vehicles Symposium (IV)*, Baden-Baden, Germany, June 2011.

[16] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, and D. Cremers, "Stereoscopic scene flow computation for 3D motion understanding," *Intl. J. of Computer Vision*, vol. 35, no. 1, pp. 29–51, 2011.

[17] A. Wedel, A. Meißner, C. Rabe, U. Franke, and D. Cremers, "Detection and segmentation of independently moving objects from dense scene flow," in *Proceedings of the 7th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, 2009, pp. 14–27.

[18] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Proceedings of the 13th Scandinavian Conference on Image Analysis*, Gothenburg, Sweden, 2003, pp. 363–370.

[19] T. Pock, M. Urschler, C. Zach, B. Reinhard, and H. Bischof, "A duality based algorithm for TV-L1-Optical-Flow image registration," in *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2007.

[20] T. Müller, J. Rannacher, C. Rabe, and U. Franke, "Feature- and depth-supported modified total variation optical flow for 3D motion field estimation in real scenes," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[21] J. Wolberg, *Prediction Analysis*. Van Nostrand, 1967.

[22] Y. Bard, *Nonlinear Parameter Estimation*. Academic Press, 1974.

[23] C. Rabe, T. Müller, A. Wedel, and U. Frank, "Dense, robust, and accurate motion field estimation from stereo image sequences in real-time," in *Eur. Conf. on Computer Vision (ECCV)*, 2010, pp. 582–595.

[24] A. Ess, B. Leibe, K. Schindler, and L. V. Gool, "Robust multiperson tracking from a mobile platform," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 31, no. 1, pp. 1831–1846, 2009.

[25] P. Alcantarilla, S. Oh, G. Mariottini, L. Bergasa, and F. Dellaert, "Learning visibility of landmarks for vision-based localization," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Anchorage, AK, USA, 2010, pp. 4881–4888.

[26] A. Ranganathan and F. Dellaert, "Online probabilistic topological mapping," *Intl. J. of Robotics Research*, 2010.