Towards Implementation of Far Background Tracker for Vision-Based Robot Navigation

Branko Karan
Institute of Technical Sciences
Serbian Academy of Sciences and Arts
Belgrade, Serbia
branko.karan@itn.sanu.ac.rs

Abstract—**Visual tracking of stationary points positioned at a large distance from moving robot provides a good basis for determining robot orientation and it may speed-up operation of structure from motion and navigation algorithms. This paper considers an implementation of the tracker that builds a far background model by assigning each visually tracked point a degree of membership that express an extent to which an apparent motion of corresponding image patch is in accordance to motion of projections of points at infinity. An experimental implementation of the tracker is described together with illustrative examples of its operation.**

Keywords—robot vision; vision-based robot navigation; background extraction

I. INTRODUCTION

This paper deals with motion-based detection and tracking of stationary distant visual features. This task is recognized as important both in everyday life as well as in robotics and computer vision.

Keeping salient distant visual features in the field of view is significant part of everyday routine that is regularly performed by humans in all tasks involving some kind of balancing. Although humans possess significant ability to detect orientation directly, using vestibular system placed in inner ear, the sense of balance is normally result of joint operation of both eyes and inner ears [1]. Visual feedback obtained by tracking far features is a normal prerequisite for keeping orientation and ability to balance without it is regarded as an exceptional skill.

In robotics and computer vision, it is well recognized that stationary distant visual features make a good basis for determining orientation of camera since their projection on image depends only on orientation and not on position of the camera [2]. Besides, estimation of orientation may accelerate pairing of visual keypoints detected in successive image frames: if an estimation of orientation is available, it may be used to constrain search region when looking for matching keypoints. Finally, after detecting image regions that are densely covered by distant features, such regions may be excluded from consideration when looking for near features that are of interest in e.g. navigation and collision avoidance.

This work was supported in part by the Serbian Ministry of Education, Science and Technology Development under projects III-44008 and TR-35003.

In the context of this work, by far background we assume a union of regions, laying on a sphere centered in the camera, and being tightly covered by polar projections of stationary objects positioned far from the camera. During motion of the camera, images of far background objects in front of the camera remain practically unchanged. Their position and orientation are however subject to change, aside small deformations that are consequence of the fact that camera projection is made on image plane and not on image sphere. Viewing angle of objects in far background does not depend on translatory movement of the camera. Besides, it is usually relatively easy to keep salient points of far background in visual field of the camera, making them ideal visual inputs for maintenance of desired orientation.

The proposed approach for extracting background from a sequence of images is based on using monocular vision sensor. Alternative techniques are also possible, e.g. using a gyroscope to determine change in orientation of the camera and afterward using this change to align successive frames in order to find best matching regions. However, a possibility of extracting background using only a camera is attractive because it eliminates a need to invest into additional sensory equipment. It should be noted here that use of stereo camera should not be considered as a serious advantage over monocular camera, at least not in outdoor situations, since the parallax that may be detected by the stereo rig is normally significantly lower than changes that could be found by comparing images obtained from possibly distant positions during motion of the camera.

Distant points may also be detected using available a priori knowledge on robot environment. However, this approach is applicable only in specific situations, contrary to motion-based approach that is more general in nature.

Background detection and tracking is of primary importance in outdoor motion. Furthermore, the suggested motion-based tracker is specially envisioned for situations where motion of the camera is subject to fast dynamics, whereby dynamics of rotation is significantly larger than translation. This is a typical situation with cameras mounted on humanoid and other walking robots as well as with the platforms moving on rough terrain.

Far background is a real-world approximation to ideal objects at infinite distance. This fact brings an idea to introduce a degree of membership as an indicator of how much the apparent motion of considered image patch may

be treated as apparent motion of a stationary point at infinity. The membership degree is conveniently modeled by a continual function that may be complex in the sense that it does not depend only on the actual state but also on tracking history of considered point as well as on apparent motion of surrounding points.

Assignment of membership degrees to tracked points is also a heart idea of the proposed tracker. Once estimated, membership degrees are employed to enhance estimation of the orientation of the robot. In turn, the estimated orientation is afterward used to update memberships.

The rest of the paper is organized as follows. The next section enlists some recent research activities that may be brought in relation to the problem of far background tracking. Basic relations and motivation for introducing membership function to describe degree of membership of particular point to far background are described in detail in section III. Afterward, section IV briefly outlines the operation of the proposed tracker, whereas section V describes results of an experimental application of the tracker to few typical outdoor situations. The final section considers possible improvements and further work on the tracker.

II. Related Work

Projective geometry of very far points, i.e. points at infinity, is a well-developed field in computer vision [2,3]. However, it seems that there were no attempts so far for systematic extracting and tracking of far features for the purpose of vision-based robot navigation.

Building a model of far background may be regarded as a special case of structure from motion algorithms. Contemporary approaches to structure from motion problem using monocular camera may be roughly divided into two groups: visual SLAM systems, with representative realizations [4,5], and bundle adjustment systems, with notable realizations [6,7,8,9]. What seems common to all approaches is the same treatment of all tracked points, irrespectively on their distance from observer. The outcome is that far visual features, although their images display practically no parallax with translatory motion of the camera, influent the process of determining relative position of the camera with respect to near objects. Similarly, the process of determining orientation of the camera is influenced by near objects, although their position in image depends not only on orientation, but also on relative position with respect to the camera. Notable exceptions are works [5,10,11] that propose using inverse depth parameterization for distant features. Contrary to their approach, no attempt is made in this work to estimate inverse depth for far features. The parallax detected during motion of the camera is used here instead only as a criterion for determining membership of the point to far background.

An important degenerate case that has been treated by many authors is pure rotational motion of the camera. In the case of rotational motion, all points behave as points at infinity and they can be conveniently located using e.g. sphere coordinates. According to examined literature, this approach was used exclusively in tasks where rotational motion of the camera was employed to generate a mosaic, i.e. a spherical image obtained by assembling individual images taken from different views, e.g. [12,13,14].This approach is extended here to the case of general camera motion, whereby an additional classifier is employed to detect points for which the rotational model is justified.

Yet another important field that may be related to this work is background extraction, a task characteristic in e.g. surveillance applications [15] and video editing [16]. However, the notion of "background" in these applications is different with respect to the notion used in this work. Additionally, both surveillance and video editing normally consider only the case of stationary or possibly quasi-stationary camera. On the contrary, this work is concentrated on approaches that may deal with fast dynamics of camera motion, what is a usual case with vehicle-mounted cameras and, especially, with cameras mounted on bipeds and other walking robots.

III. Far Background Model

A. Points at Infinity

A point at very large distance from the observer may be approximately regarded as a point at infinity. It is mathematically conveniently described using homogeneous coordinates $[x;y;z;0]^T$, where x,y,z are coordinates of a point laying on a ray along which the point is viewed from the origin of coordinate frame (i.e. from the center of the camera). Normalized image coordinates of such a point are $[x/z;y/z]^T$ and they do not change during translatory camera motion.

In the context of this work, far background is simply a set of infinity points. All infinity points belong to the plane at infinity π_∞. Projective geometry of intersection of π_∞ with 3D objects, is well developed in classic computer vision literature, e.g. [2,3]. However, for the purpose of this work, only few elementary relations will suffice.

To begin, describe a point at infinity using unit vector $\mathbf{e}=[e_x;e_y;e_z]^T$ along the ray passing through the point (note that value of $\mathbf{e}$ is uniquely determined from normalized image coordinates whenever the point is in the field of view of the camera). When the point is seen from two camera positions, related by rotation matrix $\mathbf{R}$, then the unit vectors $\mathbf{e},\mathbf{e}'$ corresponding to the first and second camera position satisfy:

$$\mathbf{e}'=\mathbf{R}\cdot\mathbf{e} \tag{1}$$

By combining this relation for three points in general relative position:

$$\begin{bmatrix}\mathbf{e}'_1 & \mathbf{e}'_2 & \mathbf{e}'_3\end{bmatrix}=\mathbf{R}\cdot\begin{bmatrix}\mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3\end{bmatrix} \tag{2}$$

or equivalently:

$$\mathbf{E}'_{3\times3}=\mathbf{R}\cdot\mathbf{E}_{3\times3} \tag{3}$$

with $\mathbf{E}_{3\times3}=\begin{bmatrix}\mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3\end{bmatrix}$ and $\mathbf{E}'_{3\times3}=\begin{bmatrix}\mathbf{e}'_1 & \mathbf{e}'_2 & \mathbf{e}'_3\end{bmatrix}$, relative rotation may be computed as:

$$\mathbf{R}=\mathbf{E}'_{3\times3}\cdot\mathbf{E}^{-1}_{3\times3}. \tag{4}$$

Relation (4) may also be used to test whether three tracked points belong to far background: provided that camera has performed both translatory and rotational motion between two views, then it is sufficient to check whether the matrix obtained by (4) is a rotation matrix. Additionally, if an estimate of the rotation matrix is available, then relation (1) may be used to check whether single point is a member of far background.

Accuracy in determining orientation from tracked infinity points can be improved by involving a larger number of points. If there are data about *n* points at infinity available, then relation (2) may be extended to:

$$\left[\mathbf{e}'_1 \ \ldots \ \mathbf{e}'_n\right] = \mathbf{R} \cdot \left[\mathbf{e}_1 \ \ldots \ \mathbf{e}_n\right] \tag{5}$$

By substituting $\mathbf{E} = \left[\mathbf{e}_1 \ \ldots \ \mathbf{e}_n\right]$, $\mathbf{E}' = \left[\mathbf{e}' \ \ldots \ \mathbf{e}'_n\right]$ the matrix equation is obtained:

$$\mathbf{E}' = \mathbf{R} \cdot \mathbf{E} \tag{6}$$

and it can be solved in the least square sense after post-multiplication by $\mathbf{E}^T$:

$$\mathbf{E}' \cdot \mathbf{E}^T = \mathbf{R} \cdot \mathbf{E} \cdot \mathbf{E}^T \tag{7}$$

or, in developed form:

$$\sum_i \mathbf{e}'_i \cdot \mathbf{e}_i^T = \mathbf{R} \cdot \sum_i \mathbf{e}_i \cdot \mathbf{e}_i^T \tag{8}$$

yielding the solution:

$$\mathbf{R} = \left(\sum_i \mathbf{e}'_i \cdot \mathbf{e}_i^T \right) \cdot \left(\sum_i \mathbf{e}_i \cdot \mathbf{e}_i^T \right)^{-1} \tag{9}$$

B. Far Background Membership

Relations (8–9) are rigorously valid only when applied to points at infinity. However, they will remain valid if the sums were made over *all* tracked points with the addition of scalar factors $w_i \in \{0,1\}$ indicating whether the corresponding point belongs to far background:

$$\sum_i w_i \cdot \mathbf{e}'_i \cdot \mathbf{e}_i^T = \mathbf{R} \cdot \sum_i w_i \cdot \mathbf{e}_i \cdot \mathbf{e}_i^T \tag{10}$$

$$\mathbf{R} = \left(\sum_i w_i \cdot \mathbf{e}'_i \cdot \mathbf{e}_i^T \right) \cdot \left(\sum_i w_i \cdot \mathbf{e}_i \cdot \mathbf{e}_i^T \right)^{-1} \tag{11}$$

Moreover, relations (10–11) would stay valid for *any* values $w_i \in (0,1]$ assigned to infinity points, as long as non-infinity points would have assigned zero weights.

On the other hand, infinity points are only a useful idealization of real 3D points. A consequence is that neither (9) nor (11) would yield an exact rotation matrix. Instead, the resulting matrix is deformed and it is necessary to extract rotation part from it. In computer vision community, standard procedures for extracting rotational part from a matrix are QR decomposition and SVD decomposition [17]. In context of this work, SVD is preferred approach because it gives a better basis for estimating deformations superimposed on rotation matrix. The decomposition takes the form:

$$\mathbf{R} = \mathbf{L} \cdot \mathbf{D} \cdot \mathbf{U}^T \tag{12}$$

where $\mathbf{L}$ and $\mathbf{U}$ are rotation matrices and $\mathbf{D}$ is diagonal. Now, an estimate:

$$\hat{\mathbf{R}} = \mathbf{L} \cdot \mathbf{U}^T \tag{13}$$

may be adopted with diagonal elements of $\mathbf{D}$ serving as a measure of stretching deformations along corresponding axes. For a good estimate, all diagonal elements must be close to unity.

A related important aspect is that points that better approximate infinity points should be given higher membership value so that their contribution to resulting rotation estimate becomes enlarged. Here, the notion of "better approximation" primarily means satisfactory outcome of tests such as (1) and (4) over long time.

C. Fuzziness of Membership Function

"Good estimates" of membership values should be related to geometry constrains, but they should not mechanistically depend only on them. Several things should be emphasized here. First, the relations such as (1) and (4) provide only necessary and not sufficient conditions. For example, it is quite possible that a considered point is not stationary at all and it is moved instead in such a manner that the tests become fulfilled. Second, the outcome of a test must be evaluated in the context of outcomes obtained for other points. For example, if camera motion is rotational, then *all* points would satisfy the tests. It is important to recognize such a situation, especially when having in mind that, with very large frequency of frames taken by the camera, successive images may be well approximated as being a result of pure rotation. Third, a history must be also taken into account, because far points may be temporarily occluded by near objects. Finally, it is important to realize that the quality of estimate (11) does not necessarily depend much on actual values of memberships, as long as the values assigned to near points are kept near zero.

IV. PRINCIPAL OPERATION OF THE TRACKER

The proposed tracker operates by updating current model of far background and estimate of current orientation of the camera according to apparent motion of keypoints (salient features) identified in sequence of images obtained by the camera. In its simplest form, the model of far background is merely a set of tuples $\left(w_i, \mathbf{e}_i, \mathbf{d}_i\right)$ associated to tracked keypoints, where w_i denotes degree of membership, $\mathbf{e}_i$ represents a ray along which the keypoint has been detected for the first time, and $\mathbf{d}_i$ is a descriptor representing image characteristics of an image patch in vicinity of the point. Rotational motion of the camera may be represented by the current rotational matrix $\mathbf{R}$. Initially, the model may be filled with all keypoints identified in current image whereby memberships may be set to some neutral value, say 0.5.

The update takes place when the next frame is acquired by the camera. Here, the first step consists in extracting keypoints from input image, computing keypoint descriptors, and matching the extracted descriptors to descriptors of saved keypoints. These tasks are time consuming and error prone and therefore have received much

attention in computer vision community. Many efforts have been done to develop robust, yet computationally efficient methods [18]. Two frequently emphasized attributes are scale and affine invariance, which are often attained at the price of increase in computational time. A fortunate characteristic of patches in far background is that affine and scale invariance are of less importance for them, thus offering a possibility of using faster methods, such as [19,20].

Once the correspondence between model and newly detected keypoints has been established, an estimate of camera rotation can be obtained from relations (11–13). Provided that percentage of correct correspondences is high, this estimate would be close to the actual rotation and thus it could be employed in (1) to compute angles φ_i between expected $\hat{\mathbf{R}} \cdot \mathbf{e}$ and detected orientation $\mathbf{e}'$ of paired keypoints:

$$\varphi_i = \measuredangle(\mathbf{e}', \hat{\mathbf{R}} \cdot \mathbf{e}) \tag{14}$$

The distribution of angular deviations φ_i can be further used to remove outliers and recompute the rotation estimate.

The final step of the operation cycle consists in updating membership degrees. This step should be performed only when distribution of angular differences is sufficiently wide — for example, when its standard deviation σ_φ is larger than some prescribed limit φ_w:

$$\sigma_\varphi > \varphi_w \tag{15}$$

In other words, update in membership degrees should be done only when the motion of the camera is rich enough so that distinction between near and far keypoints can be accurately made. Once this precondition is fulfilled, a border between near and far features should be established. This border is fuzzy in nature and it depends on nature of robotic task and speed of robot motion.

In one simple form, membership degree may be implemented as a decreasing function of angular deviation (14). This is justified by the expectation that far features would display low deviations whereas the near features would be characterized by larger deviations. In the experimental implementation that is outlined in the next section, the following function has been employed:

$$w_i = \frac{1}{1 + \left(\varphi_i / \varphi_0\right)^2} \tag{16}$$

Here, the threshold φ_0 is a value of deviation for which the membership function attains value ½. Features with deviations lower than the threshold would have increased values of membership. On the other side, with increase of the deviation, the membership degree gets lowered, so that it becomes practically negligible at $\varphi_i = 3\varphi_0$.

Initial tests have shown that operation of the tracker did not depend much on the actual shape of membership function, as long as the function was continuous and with a significant change in slope around a threshold point. However, results *were* sensitive to the choice of threshold angles φ_w and φ_0.

V. Experimental Implementation

To verify the approach, an experimental implementation of the tracker has been made. The implementation has been done in OpenCV environment [21] using its 2D Features Framework [22]. The operation of the tracker closely follows description given in Sect. IV. Feature extraction has been performed using the state of the art FAST detector [23], whereas SIFT descriptor [24] was employed for feature matching. Although far from being the fastest, SIFT descriptor was chosen because it was demonstrated to outperform other descriptors in terms of rate of outliers [18].

The tracker was applied to several open-air video sequences obtained by a low-cost camera built in a cellular phone. The camera provided a low-resolution MPEG-4 compressed CIF video with 352×288 pixels in images at the frame rate of 15fps. Aside from noise introduced by the compression, output images were additionally blurred due to lag induced by its CMOS sensor.

Fig. 1 shows excerpts from three typical sequences. The displayed images were taken in one second intervals and they are overlaid with small circles marking detected far features with membership degrees equal or larger than 0.5; additionally, to provide an illustration of obtained rotation estimates, coordinate axes of a frame corresponding to initial orientation of the camera are drawn in the center of each image (*x*-axis is shown in red, *y*-axis in blue, and *z*-axis in yellow).

The results have been obtained by selecting FAST detector parameters so that the number of tracked features is maintained between 200 and 500. After initial matching of features obtained using SIFT descriptors, matches for which angular deviations φ_i were larger than $2\sigma_\varphi$ were marked as outliers. Values of threshold angles φ_w and φ_0 were practically found by trial-and-error. The results shown in Fig. 1 were obtained by φ_w set to 10% of the viewing angle of the camera, whereas φ_0 was set to $\varphi_w/2$.

The obtained results seem satisfactory in spite of significant camera noise. However, closer inspection reveals an instability in tracked features that is outside the scope of the basic tracking mechanism of the proposed background tracker. The instability is an outcome of inability of employed pure descriptor-based matcher to recognize correct matches in the presence of noise. As a consequence, a chain to earlier detected keypoints is lost and it further leads to drifts in orientation as it is visible in second and third video sequence.

VI. Conclusion

The approach presented in this paper seems promising. However, further work is necessary to make the proposed tracker operational. First, feature matching has to be made more robust. Here, the readily available estimate of orientation of the camera may be of help. To this end, the orientation estimate provided by the tracker should be more

tightly integrated into matching algorithm, instead of serving as a simple external criterion to get rid of outlier matches. This could also result in decreased dependency on complex feature descriptors and lead to computationally more efficient implementation. The second direction would be in development of more sophisticated background membership criteria that would result in more accurate estimates and better covering of background regions.

REFERENCES

[1] J. Goldberg, V. Wilson, K. Cullen, D. Angelaki, D. Broussard, J. Buttner-Ennever, K. Fukushima, and L. Minor, *The Vestibular System: A Sixth Sense.* Oxford University Press, 2012.

[2] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision,* 2. ed. Cambridge University Press, 2004.

[3] O. Faugeras, *Three-Dimensional Computer Vision*. MIT Press, 1993.

[4] A.J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proc. 2003 IEEE Int. Conf. Computer Vision*, Nice, France, Oct. 2003, pp. 1403-1410.

[5] E. Eade and T. Drummond, "Scalable monocular SLAM," in *Proc. 2006 IEEE Conf. Computer Vision and Pattern Recognition*, New York, June 2006, pp. 469-476.

[6] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proc. 2004 IEEE Conf. Computer Vision and Pattern Recognition,* Washington, DC, 27 June - 2 July 2004, pp. I-652 - I-659.

[7] G. Klein and D.W. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. Int. Symp. Mixed and Augmented Reality ISMAR 2007,* Nara, Japan, Nov. 2007, pp. 225-234.

[8] 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.

[9] H. Strasdat, J.M.M. Montiel, and A.J. Davison, "Scale drift-aware large scale monocular SLAM," in *Proc. Robotics: Science and Systems,* Zaragoza, Spain, June 2010.

[10] J.M.M. Montiel, J. Civera, and A.J. Davison, "Unified inverse depth parametrization for monocular SLAM," in *Proc. Robotics: Science and Systems,* Philadelphia, USA, Aug. 2006.

[11] N. Trawny, and S.I. Roumeliotis, "A unified framework for nearby and distant landmarks in bearing-only SLAM," in *Proc. 2006 IEEE Int. Conf. Robotics and Automation,* Philadelphia, USA, Aug. 2006, pp. 1923-1929.

[12] D. Capel and A. Zisserman, "Automated mosaicing with super-resolution zoom," in *Proc. 1998 IEEE Conf. Computer Vision and Pattern Recognition,* Santa Barbara, USA, June 1998, pp. 885-891.

[13] J. Civera, A.J. Davison, J.A. Magallón, and J.M.M. Montiel, "Drift-free real-time sequential mosaicing," *Int. J. Computer Vision,* Vol. 81, pp. 128-137, Feb. 2009.

[14] S. Lovegrove and A.J. Davison, "Real-time spherical mosaicing using whole image alignment," in *Proc. 2010 European Conf. Computer Vision,* Heraklion, Crete, Sept. 2010, Springer, 2010, pp. 73-86.

[15] M. Cristani, M. Farenzena, D. Bloisi, and Vittorio Murino, "Background subtraction for automated multisensor surveillance: A comprehensive review," in *EURASIP J. Advances in Signal Processing,* Art. No. 343057, 2010.

[16] J. Wang and M.F. Cohen, "Image and video matting: A survey," in *Foundations and Trends in Computer Graphics and Vision,* Vol. 3, pp. 97-175, 2007.

[17] R. Szeliski, *Computer Vision: Algorithms and Applications,* Springer, 2011.

[18] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," in *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615-1630, Oct. 2005.

[19] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. 2011 IEEE Int. Conf. Computer Vision,* Barcelona, Nov. 2011, pp. 2564-2571.

[20] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast retina keypoint, in *Proc. 2012 IEEE Conf. Computer Vision and Pattern Recognition,* Providence, USA, June 2012, pp. 510-517.

[21] G. Bradski, A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library,* O'Reilly Media, 2008.

[22] *OpenCV v2.4.2 Documentation, 2D Features Framework,* online at http://docs.opencv.org/modules/features2d/doc/features2d.html

[23] E. Rosten and T. Drummond, "Machine Learning for High-speed Corner Detection," in *Proc. 2006 European Conf. Computer Vision,* Graz, May 2006, Springer, 2006, pp. 430-443.

[24] D. G. Lowe, *"Distinctive Image Features from Scale-Invariant Keypoints,"* in Int. J. Computer Vision, vol. 60, pp. 91-110, 2004.

Figure 1. Detected far background points in three sample video sequences