The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Detection of Moving Objects by Statistical Motion Analysis

Chunrong Yuan, Isabell Schwab, Fabian Recktenwald and Hanspeter A. Mallot

Abstract— **In this work we present a new approach for the detection of moving objects observed by a mobile camera, which is a critical issue related to autonomous robot navigation as well as driver/pilot assistance systems. In order to separate individual object motions from the self-motion of the observing camera, we implement a linear method to recover the full set of 3D motion parameters undergone by the camera. Based on the recovered camera motion and reconstructed depth information of the detected scene points, a criterion has been derived to determine how well individual scene points agree with the estimated camera motion. The classification of scene points is achieved by statistical analysis of the probability distribution function of the points' motion characteristics. After the initial classification, the identified dynamic scene points are further clustered into different objects by taking into account the underlying geometric distribution in the image. The approach is unique in that it can detect moving objects using a single pair of images and is completely automated. Several experiments have been carried out in challenging environments using two different hardware setups. A comparative study shows that the proposed classification method generates fewer false alarms compared to a standard one.**

I. INTRODUCTION

If an imaging sensor is moving in an environment consisting of rigid objects, the observed image displacements are the result of two different kinds of motion: camera egomotion and independent object motion. While the motions of static scene points are caused entirely by camera motion, the observed 2D motions of dynamic scene points are generated by the joint effect of both kinds of motion. The goal of moving object detection is to know whether there exist any object motions and eventually to separate them from the camera-induced motion. In other words, we need to classify the scene points into different categories based on their 2D motion patterns appearing in the image sequence.

Potential application areas of such visual motion detection include robot navigation, pilot or driver assistance, tracking and surveillance, etc. Particularly for an autonomous robot navigating in the real world, it is essential to know whether there are any dynamic objects maneuvering in its visual field which could lead to possible collisions. In order to avoid them, it is critical that they could be detected at a large distance so that there is substantial time left for conducting the necessary avoidance strategy in the control loop.

Like [1], most of the availabe approaches deal with single-frame motion segmentation. Among those using multiple frames, a major difference lies in the parametric modeling of the underlying constraint used for motion detection. Under the assumption of constant camera motion, the authors of [2] have derived a simple constraint extracted from three frames to detect scene points whose 2D motion changes between frames. In [3] and [4], 2D homography has been used for establishing a constraint between a pair of viewed images. However, the success of such an approach depends on the existence of a dominant plane (e.g. the ground plane) in the viewed scene.

Another possibility is to use geometric constraints among multiple views. The approach proposed by [5] uses the trilinear constraint over three views. Scene points are clustered into different groups, where each group agrees with a different trilinear constraint. A manual threshold is set to decide the support of each group. A multibody trifocal tensor based on three views is applied in [6], where the EM (Expectation and Maximization) algorithm is used to refine the constraints as well as their support iteratively. Correspondences among the three views, however, are selected manually, with equal distribution between the static and dynamic scene points. An inherent problem shared by such approaches is their inability to deal with dynamic objects that are either small or moving at a distance. Under such circumstances it would be difficult to estimate the parametric model of object motion, since not enough scene points may be detected from the dynamic objects.

A further possibility is to recover the 3D motion parameters undergone by the camera and to find scene points whose motion is incompatible with the camera egomotion. In the work of [7], it is assumed that both the camera and the object are just translating. Hence the recovered egomotion parameters contain only the translational part. The authors of [8] proposed a method based on the recovery of the focus of expansion. In [9], although both translation and rotation parameters are recovered from 2D image displacements, the constraint used for the detection of moving objects is only based on the translational part. In [10], it is assumed that both the egomotion and location of the camera are known.

In this paper, we present a new approach for automatic detection of independent motion under challenging real–world situations. It does not make any restriction on the motion of camera or objects. Using a calibrated camera mounted on a mobile platform, we are able to recover the 5-DOF (degree of freedom) egomotion parameters based on image pairs taken by the mobile camera. A single motion constraint is derived from the estimated egomotion parameters. Unlike the cited approaches, our constraint takes full advantage of both the translation and the rotation parts of the egomotion parameters and no extra knowledge about camera location is required.

This work was supported by the European Commission for the research projects μDrones and CURVACE with the contract number FP6-2005-IST-6-045248 and FP7-2009-ICT-237940 respectively

The authors are with the Chair for Cognitive Neuroscience, University of Tübingen, Germany

The remainder of the paper is organized as follows. We outline in section 2 the theoretical basics of 3D motion and structure estimation. In section 3, we present our approach on statistical motion analysis. Experimental evaluation is shown in section 4. Section 5 summarizes the whole paper.

II. Motion and Structure Estimation

Suppose that the 2D image displacement $\boldsymbol{v} = [u, v]^{\mathrm{T}}$ of a point $p(x, y)$ is caused solely by a 3D rigid motion between the observing camera and the scene and let's denote the motion parameter with a rotation vector $\boldsymbol{r} = [r_x, r_y, r_z]^{\mathrm{T}}$ and a translation vector $\boldsymbol{t} = [t_x, t_y, t_z]^{\mathrm{T}}$, the following two equations hold for a camera with unit focal length [11]:

$$u = \frac{t_x - x t_z}{Z} + [-r_x xy + r_y(x^2 + 1) - r_z y] \ , \tag{1}$$

$$v = \frac{t_y - y t_z}{Z} + [-r_x(y^2 + 1) + r_y xy + r_z x] \ . \tag{2}$$

The scalar Z is the depth of the 2D image point p. Without the knowledge of the exact scene depth, it is only possible to recover the direction of $\boldsymbol{t}$. For this reason, the recovered motion parameters have five degrees of freedom.

Like many problems in computer and robot vision, recovery of camera motion parameters from 2D image displacement is an ill–posed problem. For linear/nonlinear approximation of the solution, several approaches exist [12]. Our approach for independent motion separation is based on a linear algorithm proposed by [13]. Here we give a brief summary of the theory. For details please refer to the original paper.

Let $\boldsymbol{m}$ denote a vector with unit length starting from the optical center of the camera and pointing at the image point p, and $\dot{\boldsymbol{m}}$ denote the image displacement vector projected from the image plane onto the unit sphere centered at the camera optical center, we may then define a vector $\dot{\boldsymbol{m}}^*$ as

$$\dot{\boldsymbol{m}}^* = \boldsymbol{m} \times \dot{\boldsymbol{m}} \ , \tag{3}$$

where $\times$ represents the cross product. Using some algebraic manipulation, it is found that the solution for $\mathbf{t}$ is equal to the least eigenvector of a matrix $\boldsymbol{A} = (A_{ij})$ where $i, j = 1$ to 3 and

$$A_{ij} = L_{ij} - \sum_{k,l,m,n=1}^{3} M_{ikl} N^{-1}_{klmn} M_{jmn} \ , \tag{4}$$

$$L_{ij} = \int_{\Omega} \dot{\boldsymbol{m}}^*_i \dot{\boldsymbol{m}}^*_j d\Omega \ , \tag{5}$$

$$M_{ijk} = \int_{\Omega} \dot{\boldsymbol{m}}^*_i \boldsymbol{m}_j \boldsymbol{m}_k d\Omega \ , \tag{6}$$

$$N_{ijkl} = \int_{\Omega} \boldsymbol{m}_i \boldsymbol{m}_j \boldsymbol{m}_k \boldsymbol{m}_l d\Omega \ , \tag{7}$$

and $\boldsymbol{L} = (L_{ij})$, $\boldsymbol{M} = (M_{ijk})$, $\boldsymbol{N} = (N_{ijkl})$ are tensors.

Once $\boldsymbol{t}$ is recovered, the solution for $\boldsymbol{r}$ is given as

$$\boldsymbol{r} = \frac{1}{2}[Tr(\boldsymbol{K}) + 3\boldsymbol{t}^{\mathrm{T}} \boldsymbol{K} \boldsymbol{t}]\boldsymbol{t} - 2\boldsymbol{K}\boldsymbol{t} \ , \tag{8}$$

where Tr is the trace of a matrix and matrix $\boldsymbol{K} = (K_{ij})$ is defined as

$$K_{ij} = - \sum_{k,l,m=1}^{3} N^{-1}_{ijkl} M_{mkl} \boldsymbol{t} \ . \tag{9}$$

If $(\boldsymbol{t}, \boldsymbol{r})$ is a solution, then $(-\boldsymbol{t}, \boldsymbol{r})$ is also a solution. The correct one can be chosen based on the cheirality constraint, by assuring positive scene depth calculated as

$$Z(\boldsymbol{m}) = \frac{1 - (\boldsymbol{m}^{\mathrm{T}} \boldsymbol{t})^2}{\boldsymbol{m}^{\mathrm{T}}(\boldsymbol{r} \times \boldsymbol{t}) - \dot{\boldsymbol{m}}^{\mathrm{T}} \boldsymbol{t}} \ . \tag{10}$$

III. Statistical Motion Analysis

Once the 3D motion and structure parameters have been estimated, they can be used as a motion constraint for identifying those scene points whose motion deviates from the estimated one. By projecting the 3D motion of scene points onto the image plane, an algebraic distance measure can be calculated. Under a given 3D motion with parameter $(\boldsymbol{t}, \boldsymbol{r})$, the motion of a static scene point with depth $Z(\boldsymbol{m})$ leads to the observation of a projected 2D image displacement represented as

$$\boldsymbol{u} = -\boldsymbol{r} \times \boldsymbol{m} - \frac{(\boldsymbol{I} - \boldsymbol{m}\boldsymbol{m}^{\mathrm{T}})\boldsymbol{t}}{Z(\boldsymbol{m})} \ . \tag{11}$$

Sine the measured image displacement is $\boldsymbol{v}$, a criterion can be defined based on the distance between the projected and measured image displacement as

$$J(\boldsymbol{m}) = \|\boldsymbol{u} - \boldsymbol{v}\| \ . \tag{12}$$

A. Statistical Classification

The smaller the distance $J(\boldsymbol{m})$ is, the more likely is the point a static one. Now the problem becomes finding a threshold k so that points with $J(\boldsymbol{m}) < k$ can be regarded as static points.

A threshold can be found by analyzing the probability distribution of the distance measurements. We first quantize the distance set $\{J(\boldsymbol{m})\}$, calculated from all available scene points, into $L+1$ levels, ranging form 0 to L units. Following that, a distance histogram $h(j)$, $j \in [0, L]$ can be calculated. If $h(j)$ is a multimodal histogram with at least two peaks, a threshold k can be found with $0 < k < L$.

Suppose that the static scene points belong to class Ω_0 and that the dynamic points as well as outliers belong to Ω_1, a probability distribution of the distance measurements can be computed as

$$\rho_i = \frac{h_i}{N} \ , \tag{13}$$

where N is the total number of available points to be classified. Since points belonging to the same class have similar distribution, the threshold k can be computed automatically by maximizing the inter–class difference, which can be done by going through the following steps:

1) Calculate the probability of Ω_0 and Ω_1

$$\rho(\Omega_0) = \sum_{i=0}^{k} \rho_i \tag{14}$$

$$\rho(\Omega_1) = \sum_{i=k+1}^{L} \rho_i \tag{15}$$

2) Estimate the mean value of Ω_0 and Ω_1

$$\mu(\Omega_0) = \sum_{i=0}^{k} \frac{i\rho_i}{\rho(\Omega_0)} \tag{16}$$

$$\mu(\Omega_1) = \sum_{i=k+1}^{L} \frac{i\rho_i}{\rho(\Omega_1)} \tag{17}$$

3) Compute the mean of the whole data set

$$\mu = \sum_{i=0}^{L} i\rho_i \tag{18}$$

4) Set the inter–class difference function as

$$dk = \rho(\Omega_0)(\mu(\Omega_0) - \mu)^2 + \rho(\Omega_1)(\mu(\Omega_1) - \mu)^2 \tag{19}$$

5) Threshold is found as

$$\kappa = \underset{k}{\operatorname{argmax}} \{dk\} \ . \tag{20}$$

With a simple linear search, the threshold κ can be found which maximizes dk. If κ is equal to 0 or L, this indicates that the histogram $h(j)$ has a unimodal distribution. In this case, there exist no moving objects in the environment.

B. Motion Clustering

With the automatically calculated threshold, the set of candidate dynamic points can be determined. The remaining problem is to remove outliers and know how many moving objects exist. Our method is to first cluster the points into as many subsets as possible. Since the number of moving objects is unknown, we use a modified k–means algorithm for point clustering. Similar to [14], the idea is to first cluster the points into several subsets based on the geometric location of these points in the input image. This splitting process continues until each subset can be fit by a unimodal distribution. After the splitting process, outliers, which have a relative large distance to their nearest subsets, can hence be identified. In order to overcome possible over–segmentation, a merging process is carried out. The merging of two subsets is based on the distance between their geometric centers. Through the above split–and–merge process, moving objects can be segmented and outliers are filtered out automatically.

IV. Experiments

For the purpose of evaluation, we have carried out experiments in both indoor and outdoor environments, where the illumination changes constantly. Tests are done using two different kinds of setup. Video frames are captured by a perspective camera mounted either on a ground vehicle or a micro UAV (unmanned air vehicle). Fig. 1 shows the two setups. In both cases, images taken by the cameras are transmitted online to the laptop on which our algorithm is running. The intrinsic parameters of the two cameras have been calibrated separately in advance. The locations and orientations of the cameras, i.e. the extrinsic camera parameters, can be changed online after camera calibration.

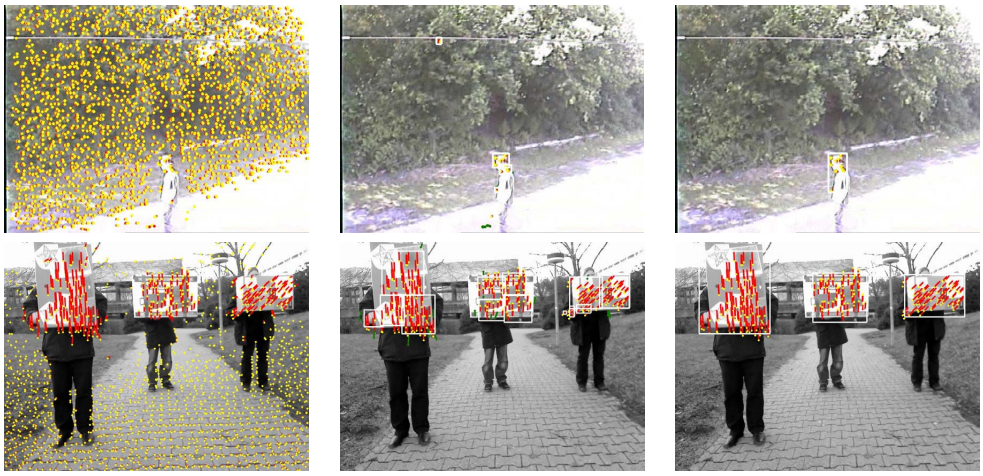
Fig. 1. Experimental setups: A ground vehicle with a USB camera and an AR–100 UAV with onboard camera.

For each pair of images $(\boldsymbol{f}^t, \boldsymbol{f}^{t+1})$ captured, we first apply the pyramidal Lucas–Kanade algorithm [15] and obtain an initial image displacement set $\{\boldsymbol{v}_i\}$ together with a set of corresponding points $(\boldsymbol{p}_i, \boldsymbol{q}_i)$. Due to changes in illumination conditions as well as noise in the image formation process, some of the initially calculated vectors may not be correct. In order to filter out the incorrect ones, we use the point set $\boldsymbol{q}_i$ to calculate a backward image displacement between frame $\boldsymbol{f}^{t+1}$ and $\boldsymbol{f}^t$, resulting in another vector set $\{\hat{\boldsymbol{v}}_i\}$. A correctly calculated displacement should satisfy

$$e_i = \|\boldsymbol{v}_i + \hat{\boldsymbol{v}}_i\| = 0 \ . \tag{21}$$

We keep only those points with $e_i < 0.1$ pixel. The advantage of using this criterion is that it holds regardless of the magnitude of the motion vector $\boldsymbol{v}$. From the remaining points, we use RANSAC together with the linear method introduced in section 2 to recover the 3D motion parameters undergone by the camera. Considering the fact that moving objects usually come from a distance and hence occupy only relatively small areas compared with the static background, the use of RANSAC is justified. With the motion and structure parameters estimated, scene points can be classified into two classes: Ω_0 with points whose motion parameters agree with the estimated camera motion parameters and Ω_1 with points whose motion parameters are incompatible with the camera motion. Points belonging to Ω_1 are further clustered into different moving objects or outliers.

In Fig. 2 two examples are shown. In column (a) of Fig. 2 we show the image $\boldsymbol{f}^t$ together with the set of detected image displacement vectors. After having estimated the 3D motion parameters, those points whose motion parameters do not agree with the camera motion are identified and clustered into several subsets, as is shown in Fig. 2 column (b). At the same time, outliers are detected, which are visualized as green vectors. In Fig. 2 column (c) we show the final segmentation results with the subsets correctly merged and outliers removed. Shown in Fig. 3 are the two distance histograms $h(j)$, calculated respectively from the two examples. It is evident that they are multimodal histograms having more than one peak.



(a) (b) (c)

Fig. 2. Examples of automatic detection of moving objects. Images on the top and bottom rows are captured by the aerial and ground vehicles, respectively.

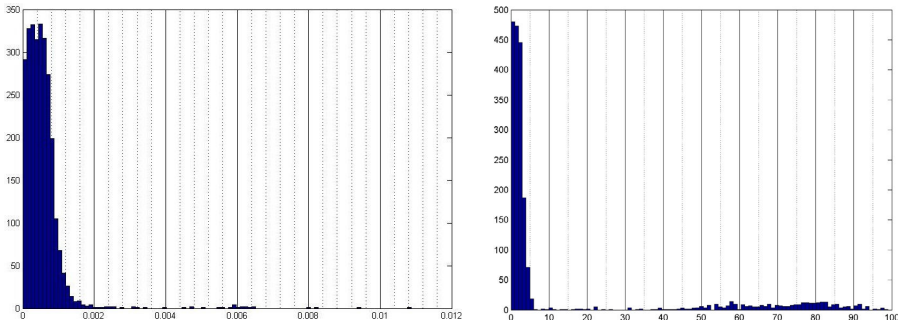


Fig. 3. Example histograms $h(j)$ for the images appearing in Fig. 2.

As demonstrated by the above examples, the algorithm is capable of detecting both fast and slowly moving objects. The distance between the objects and the camera can be either small or large. In the first example (top row of Fig. 2), both the person and the camera are moving to the left. The detection is successful despite the fact that the person moves slowly and far away.

For the purpose of evaluation, a comparative study has been carried out. In particular, we compare our automatic thresholding algorithm against the minimum error thresholding method proposed in [16]. The results are shown in Fig. 4. On the left column we show the original image together with the finally detected image displacement vectors. From these vectors, one can perceive the movement of the camera and objects. Shown in the second column are the detection results using our approach as described above. The third column shows the results of replacing our optimal thresholding with the minimum error thresholding method (called reference approach in the following). In the rightmost column we show the ground truth obtained by manual segmentation.

As can be seen from Fig. 4 (a), our approach works better than the reference approach, since a larger part of the moving person has been detected. In the next two examples, images are captured in an indoor environment. While the camera is moving forward, the object has side movement in Fig. 4 (b) and parallel movement (approaching the camera) in Fig. 4 (c). Both methods work fine for the example shown in Fig. 4 (b). For the example shown in Fig. 4 (c), the reference approach generates two false alarms. Due to the shadow of the object, our approach has generated a slightly bigger boundary of the moving object.

In the example shown in Fig. 4 (d), both the camera and the object are moving in parallel in the same direction. This is a particularly difficult situation. While the reference approach fails to detect the object at all, two parts of the object have been segmented using our approach.

Another example in Fig. 4 (e) shows three objects moving while the camera is moving toward them. Each of the two persons on the left is pushing an object toward the camera, while the third person is shifting an object to the right. Using our approach, all the moving objects have been identified, with the object on the right be split into two objects. Errors due to false alarms can be observed using the reference approach. While the first moving object has been ignored totally by the reference approach, a part of it (the moving arm) has been identified using our approach.

A similar example with three objects moving in front of a forward moving camera is shown in Fig. 4 (f). Using our algorithm, two of the three objects have been detected. The reference approach is able to detect all of them. However, a

Fig. 4. A comparative study with six examples.

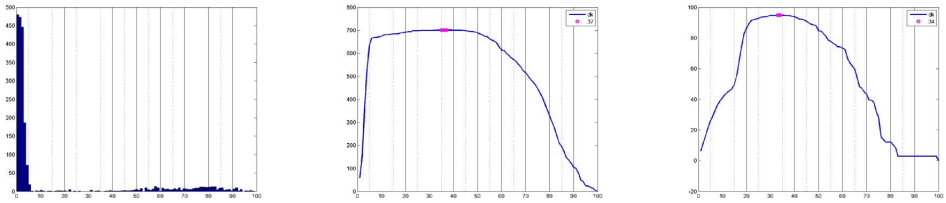


Fig. 5. The histogram $h(j)$ and the two different curves of dk.

false alarm has occurred.

In the last example shown in Fig. 4 (f), the moving object on the left is missed due to its relatively insignificant motion. The moving object on the right is detected entirely. The partial detection of the moving object in the middle indicates that some error might have occurred in the motion separation process.

In order to know the source of the error, we have plotted the distribution of the distance histogram $h(j)$ as well as the corresponding inter–class difference function dk in Fig. 5. As can be seen, the optimal threshold has been set by our algorithm at histogram bin 37. However, we can observe that the value of dk increases quickly before reaching bin 7. It then increases only slightly or even stagnates before the maximal value is achieved at bin 37. For comparison, we show on the right of Fig. 5 another dk function obtained with the data shown in the second example of Fig. 2. This curve is totally different (without flat part). The flat part on the curve dk shown in the middle of Fig. 5 is relatively big. This as well as the sparse density in the histogram $h(j)$ between bin 8 and bin 37 suggests the need of further thresholding of the histogram in case of a flat dk function. Regarding the distance distribution $h(j)$ between the bins 0 to 37, a further threshold can be calculated using our approach, resulting in a new threshold at bin 12. If we use this new threshold, all three moving objects can be identified and further outliers can be removed.

V. Conclusion

We have presented in this paper a statistical approach for motion analysis. The goal is to detect objects maneuvering in the visual field of a moving observer. Our method is based on a motion constraint established by recovering the full 3D motion parameters undergone by the camera. The detection of dynamic scene points is done by statistical analysis of the probability distribution of their motion characteristics. Further segmentation of individual moving objects is done via geometry-based split–and–merge process. Experimental evaluation has shown that the approach works well in most cases. While estimation of the egomotion parameters and detection of independent motion can be achieved in real-time, the computation complexity of the geometrical segmentation of individual moving objects remains a bottleneck for video-rate implementation. For further performance improvement, the detection of 2D image displacements and the estimation of 3D motion and structure parameters can be enhanced by increasing the density of point correspondences and enlarging the field–of–view of the camera. Based on our earlier work [17], we are currently carrying out research using omni-directional cameras with enlarged field-of-view for better estimation of the camera egomotion. In addition to comparative study with similar approaches, we also plan to integrate our approach within a SLAM framework for camera–based mapping in dynamic environments.

References

[1] R. Zabih and V. Kolmogorov, "Spatially coherent clustering with graph cuts," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2004)*, 2004, pp. 437–444.

[2] A. Argyros, M. Lourakis, P. Trahanias, and S. Orphanoudakis, "Fast visual detection of changes in 3d motion," in *IAPR Workshop on Machine Vision Application (MVA'96)*, 1996.

[3] M. Irani and P. Anadan, "A unified approach to moving object detection in 2D and 3D scenes." *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 6, pp. 577–589, 1998.

[4] C. Yuan, G. Medioni, J. Kang, and I. Cohen, "Detecting motion regions in the presence of a strong parallax from a moving camera by multiview geometric constraints," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, pp. 1627–1641, 2007.

[5] P. Torr, A. Zissermann, and D. Murray, "Motion clustering using the trilinear constraint over three views," in *Workshop on Geometrical Modeling and Invariants for Computer vision*, 1995.

[6] R. Hartley and R. Vidal, "The multibody trifocal tensor: motion segmentation from 3 perspective views," in *Int. Conf. Computer Vision and Pattern Recognition (CVPR 2004)*, 2004, pp. 769–775.

[7] J. Clarke and A. Zisserman, "Detecting and tracking of independent motion," *Image and Vision Computing*, vol. 14, no. 8, pp. 565–572, 1996.

[8] N. Lobo and J. Tsotsos, "Computing egomotion and detecting independent motion from image motion using collinear points," *Computer Vision and Image Understanding*, vol. 64, no. 1, pp. 21–52, 1996.

[9] W. MacLean, A. Jepson, and R. Frecker, "Recovery of egomotion and segmentation of independent object motion using the EM algorithm," in *British Machine Vision Conference (BMVC'96)*, 1996.

[10] J. Klappstein, F. Stein, and U. Franke, "Detectablility of moving objects using correpondences over two and three frames," in *DAGM Symposium 2007*, 2007, pp. 112–121.

[11] A. Bruss and B. Horn, "Passive navigation," *Computer Vision, Graphics and Image Processing*, vol. 21, pp. 3–20, 1983.

[12] T. Tian, C. Tomasi, and D. Heeger, "Comparison of approaches to egomotion computation," in *Int. Conf. Computer Vision and Pattern Recognition (CVPR'96)*, 1996.

[13] K. Kanatani, "3-D interpretation of optical flow by renormalization," *Int. Journal of Computer Vision*, vol. 11, no. 3, pp. 267–282, 1993.

[14] G. Hamerly and C. Elkan, "Learning the k in k-means," in *Neural Information Processing Systems*. MIT Press, 2003.

[15] T. Kanade and B. Lucas, "An iterative image registration technique with an application to stereo vision," in *Int. Joint Conf. Artificial Intelligence (IJCAI'81)*, 1981.

[16] J. Kittler and J. Illingworth, "Minimum error thresholding," *Pattern Recognition*, vol. 19, no. 1, pp. 41–47, 1986.

[17] C. Yuan, F. Recktenwald, and H. A. Mallot, "Visual steering of UAV in unknown environments," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2009)*, 2009, pp. 3906–3911.