Realtime Moving Object Detection from a Freely Moving Monocular Camera

Abhijit Kundu, C. V. Jawahar and K Madhava Krishna

***Abstract*—Detection of moving objects is a key component in mobile robotic perception and understanding of the environment. In this paper, we describe a realtime independent motion detection algorithm for this purpose. The method is robust and is capable of detecting difficult degenerate motions, where the moving objects is followed by a moving camera in the same direction. This robustness is attributed to the use of efficient geometric constraints and a probability framework which propagates the uncertainty in the system. The proposed independent motion detection framework integrates seamlessly with existing visual SLAM solutions. The system consists of multiple modules which are tightly coupled so that one module benefits from another. The integrated system can simultaneously detect multiple moving objects in realtime from a freely moving monocular camera.**

I. Introduction

SLAM involves simultaneously estimating locations of newly perceived landmarks and the location of the robot itself while incrementally building a map of an unknown environment. Over the last decade, SLAM has been one of the most active research fields in robotics and excellent results have been reported by many researchers [1]; predominantly using laser range-finder sensors to build 2-D maps of planar environments. Though accurate, laser range-finders are expensive and bulky, so lot of researchers turned to cameras which provide low-cost, full 3-D and much richer intuitive "human-like" information about the environment. So last decade also saw a significant development in vision based SLAM systems [2], [3], [4]. But almost all these SLAM approaches assume a static environment, containing only rigid, non-moving objects. Moving objects are taken as noise sources and filtered out. Though, this may be a feasible solution in less dynamic environments, but it becomes unavoidable as the environment becomes more and more dynamic. Also accounting for both the static and moving objects provides richer information about the environment. A robust solution to the SLAM problem in dynamic environments will expand the potential for robotic applications, especially in applications which are in close proximity to human beings and other robots. As put by [5], robots will be able to work not only for people but also with people.

The solution to the moving object detection and segmentation problem will act as a bridge between the static SLAM and its counterpart for dynamic environments. But, motion detection from a freely moving monocular camera is an ill-posed problem and a difficult task. The moving camera causes every pixel to appear moving. The apparent pixel motion of points is a combined effect of the camera motion, independent object motion, scene structure and camera perspective effects. Different views resulting from the camera motion are connected by a number of multiview geometric constraints. This constraints can be used for the motion detection task. Those inconsistent with the constraints can be labeled as moving regions or outliers.

We propose a realtime independent motion detection algorithm with the aid of an online visual SLAM algorithm. The moving object detection is robust and is capable of segmenting difficult degenerate motions, where the moving objects is followed by a moving camera in the same direction. We introduce efficient geometric constraints that helps in detecting these degenerate motions and a probability framework that recursively updates feature probability and takes into consideration the uncertainty in camera pose estimation. The final system integrates independent motion detection with visual SLAM. We introduce several feedback paths between these modules, which enables them to mutually benefit each other. A full perspective camera model is used, and we do not have any restrictive assumptions on the camera motion or environment. Unlike many of the existing works, the proposed method is online and incremental in nature and scales to arbitrarily long sequences. We also describe how this system can be used to constrain and speed-up object detection algorithms, where detection of specific object category like person is required. Finally we show experimental results of this algorithm on real image datasets.

II. Related Works

The task of moving object detection and segmentation, is much easier if a stereo sensor is available, which allows additional constraints to be used for detecting independent motion [6], [7], [8]. However the problem is very much ill-posed for monocular systems. The problem of motion detection and segmentation from a moving camera has been a very active research area in computer vision community. The multiview geometric constraints used for motion detection, can be loosely divided into four categories. The first category of methods used for the task of motion detection, relies on estimating a global parametric motion model of the background. These methods [9], [10], [11] compensate camera motion by 2D homography or affine motion model and pixels consistent with the estimated model are assumed to be background and outliers to the model are defined as moving regions. However, these models are approximations which only holds for the restricted cases of camera motion and scene structure.

The problems with 2D homography methods led to plane-parallax [12], [13] based constraints. The "planar-parallax" constraints, represents the scene structure by a residual displacement field termed parallax with respect to a 3D reference plane in the scene. The plane-parallax constraint was designed to detect residual motion as an after-step of 2D homography methods. Also they are designed to detect motion regions when dense correspondences between small baseline camera motions are available. Also, all the planar-parallax methods are ineffective when the scene cannot be approximated by a plane.

Though the planar-parallax decomposition can be used for egomotion estimation and structure, the traditional multi-view geometry constrains like epipolar constraint in 2 views or trillinear constraints in 3 views and their extension to N views have proved to be much more effective in scene understanding as in structure from motion (SfM) and visual SLAM. This constraints are well understood and are now textbook materials [14].

In realtime monocular visual SLAM systems, moving objects have not yet been dealt properly. We found the following three works for visual SLAM in dynamic environments: a work by Sola [15] and two other recent works of [16] and [17]. Sola [15] does an observability analysis of detecting and tracking moving objects with monocular vision. He proposes a BiCamSLAM [15] solution with stereo cameras to bypass the observability issues with mono-vision. In [16], a 3D object tracker runs parallel with the monocular camera SLAM [2] for tracking a predefined moving object. This prevents the visual SLAM framework from incorporating moving features lying on the moving object. But the proposed approach does not perform moving object detection; so moving features apart from those lying on the tracked moving object can still corrupt the SLAM estimation. Also they used a model based tracker, which can only track a previously modeled object with manual initialization. The work by Migliore *et al.* [17] maintains two separate filters: a monoSLAM filter [2] with the static features and a bearing only tracker for the the moving features. As concluded by Migliore *et al.*, the main disadvantage of their system is the inability to obtain an accurate estimate of the moving objects in the scene. This is due to the fact that they maintain separate filters for tracking each individual moving feature, without any analysis of the structure of the scene; which for e.g can be obtained from clustering points belonging to same moving object or performing same motion. This is also the reason that they are not able to use the occlusion information of the tracked moving object, for extending the lifetime of features as in [16].

Previously in [18], we used robot odometry to estimate the camera motion, which was then used to detect independently moving objects in the scene. In this work we extend that work to freely moving monocular camera, without any aid from odometry or IMU kind of devices.

III. System Overview

In the first step, sparse salient features are detected and tracked through the image sequence. An online visual SLAM algorithm estimates the camera trajectory and 3D structure using the feature tracks. Between any two views, relative camera motion and locations of features is used to evaluate the geometric constraints, as detailed in Sec. IV-A. A recursive Bayes filter is used to compute the probability of the feature being stationary or dynamic through the geometric constraints. The present probability of a feature being dynamic is fused with the previous probabilities in a recursive framework to give the updated probability of the features. The probabilities also take care of uncertainty in pose estimation by the visual SLAM. Features with high probabilities of being dynamic are either mismatched features arising due to tracking error or features belonging to some independently moving objects. This residual feature tracks are then clustered into independently moving entities, using spatial proximity and motion coherence.

A. Feature Tracking

In order to detect moving objects, we should be able to get feature tracks on the moving bodies also. This is challenging as different bodies are moving at different speeds. Thus, contrary to conventional SLAM, where the features belonging moving objects are not important, we need to pay extra caution to feature tracking in this scenario. In each image, a number of salient features (FAST corners [19]) are detected, at different image pyramidal levels while ensuring the features are sufficiently spread all over the image. A patch is generated on these feature locations and are matched across images on the basis of zero-mean SSD scores to produce feature tracks. A number of constraints is used to improve feature matching. When a match is found, we try to match that feature backward in the original image. Matches, in which each point is the other's strongest match are only taken as valid. 3D reconstruction by visual SLAM enables the use of additional constraints. For the 3D points, whose depth is computed from the visual SLAM module, the 1D epipolar search is reduced to just around the projection of the 3D point on the image with predicted camera pose. Also with the knowledge of camera relative pose and depth of a feature, an affine warp can be performed on the image patches to maintain view invariance from the patch's first and current observation.

B. Visual SLAM Framework

The method proposed is independent of the SLAM algorithm used. However, we chose the bundle adjustment visual SLAM [20], [4], [21] framework over the filter based approaches [2], [22]. Apart from accuracy benefits [23], the bundle adjustment visual SLAM methods extracts as much correspondence information as possible compared to very sparse map (about 10-30 features per frame) in filter based approaches. Our implementation closely follows to that of [20], [4]. In brief, a 5-point algorithm [24] with RANSAC is used to estimate the initial epipolar geometry, and subsequent pose is determined with 3-point resection [25]. Some of the frames

are selected as key-frames, which are used to triangulate 3D points. The set of 3D points and the corresponding keyframes are used in by the bundle adjustment process to iteratively minimize reprojection error. The bundle adjustment is initially performed over the most recent keyframes, before attempting a global optimization. The whole algorithm is implemented as two-threaded process, where one thread performs tasks like camera pose estimation, key-frame decision and addition, another back-end thread performs optimizes this estimate by bundle adjustment.

IV. Independent Motion Detection

Using camera relative motion and feature tracks, the task is to assign each feature tracks a probability of being dynamic or static. Efficient geometric constraints are used to form these probabilistic fitness scores. With each new frame, the probabilities of feature being dynamic is fused with the previous probabilities in a recursive framework to give the updated probability of the features. Features with high probability of being dynamic are assigned to one of the independently moving objects.

A. Geometric Constraints

Epipolar constraint is the commonly used constraint that connects two views. Reprojection error or its first order approximation called Sampson error, based on the epipolar constraint is used throughout the structure and motion estimation by the visual SLAM module. Basically they measure how far a feature lies from the epipolar line induced by the corresponding feature in the other view. Though these are the gold standard cost functions for 3D reconstruction, it is not good enough for independent motion detection. If a 3D point moves along the epipolar plane formed by the two views, its projection in the image move along the epipolar line. Thus in spite of moving independently, it still satisfies the epipolar constraint. This is depicted in Fig. 1. This kind of degenerate motion, is quite common in real world scenarios, e.g camera and a object are moving in same direction as in camera mounted in car moving through a road, or camera-mounted robot following behind a moving person. To detect degenerate motion, we make use of the camera motion and 3D structure, to estimate a bound in the position of the feature along the epipolar line. We describe this as Flow Vector Bound (FVB) constraint.

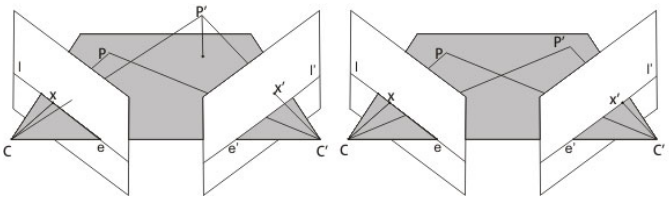

Fig. 1. LEFT: The world point P moves non-degenerately to $P^{'}$ and hence $x^{'}$, the image of $P^{'}$ does not lie on the epipolar line corresponding to x. RIGHT: The point P moves degenerately in the epipolar plane to $P^{'}$. Hence, despite moving, its image point lies on the epipolar line corresponding to the image of P.

1) Flow Vector Bound (FVB) Constraint:: For a general camera motion involving both rotation and translation R, t, the effect of rotation can be compensated by applying a projective transformation to the first image. This is achieved by multiplying feature points in view 1 with the infinite homography $H = KRK^{-1}$ [14]. The resulting feature flow vector connecting feature position in view2 to that of the rotation compensated feature position in view1, should lie along the epipolar lines. Now assume that our camera translates by t and p_n,p_{n+1} be the image of a static point X. Here p_n is normalized as $p_n = (u, v, 1)^T$. Attaching the world frame to the camera center of the 1st view, the camera matrix for the views are $K[I|0]$ and $K[I|t]$. Also, if z is depth of the scene point X, then inhomogeneous coordinates of X is $zK^{-1}p_n$. Now image of X in the 2nd view, $p_{n+1} = K[I|t]X$. Solving we get, [14]

$$p_{n+1} = p_n + \frac{Kt}{z} \tag{1}$$

Equation 1 describes the movement of the feature point in the image. Starting at point p_n in I_n it moves along the line defined by p_n and epipole, $e_{n+1} = Kt$. The extent of movement depends on translation t and inverse depth z. From equation 1, if we know depth z of a scene point, we can predict the position of its image along the epipolar line. In absence of any depth information, we set a possible bound in depth of a scene point as viewed from the camera. Let z_{max} and z_{min} be the upper and lower bound on possible depth of a scene point. We then find image displacements along the epipolar line, d_{min} and d_{max}, corresponding to z_{max} and z_{min} respectively. If the flow vector of a feature, does not't lie between d_{min} and d_{max}, it is more likely to be an image of an independent motion.

The structure estimation from visual SLAM module helps in reducing the possible bound in depth. Instead of setting z_{max} to infinity, known depth of the background enables in setting a more tight bound, and thus better detection of degenerate motion. The depth bound is adjusted on the basis of depth distribution along the particular frustum.

The probability of satisfying flow vector bound constraint $P(FVB)$. can be computed as

$$P(FVB) = \frac{1}{1 + \left(\frac{FV - d_{mean}}{d_{range}}\right)^{2\beta}} \tag{2}$$

Here $d_{mean} = \frac{d_{min} + d_{max}}{2}$ and $d_{range} = \frac{d_{max} - d_{min}}{2}$. d_{min} and d_{max} are the bound in image displacements, The distribution function is similar to a Butterworth bandpass filter. $P(FVB)$ has a high value if the feature lies inside the bound given by FVB constraint, and the probability falls rapidly as the feature lies outside the bound. Larger the value of β, more rapidly it falls. In our implementation, we used $\beta = 10$.

B. Computing Independent Motion Probability

In this section we describe a recursive formulation based on Bayes filter to derive the probability of a world point and

hence its projected image point being classified as stationary or dynamic. The motion noise and image pixel noise if any are bundled into a Gaussian probability distribution of the epipolar lines as derived in [14] and denoted by $EL^i = N(\mu_l{}^i, \sum l^i)$ where EL^i refers to the set of epipolar lines corresponding to image point i, and N ($\mu_l{}^i$, $\sum l^i$) refers to the standard Gaussian probability distribution over this set.

Let $p_n{}^i$ be the ith point in image I_n. The probability that $p_n{}^i$ is classified as stationary is denoted as $P(p_n{}^i|I_n, I_{n-1}) = P_{n,s}(p^i)$ or $P_{n,s}{}^i$ in short, where the suffix s signifying static. Then, with Markov approximation and recursive probability update of a point being stationary given a set of images can be derived as

$$P(p_n{}^i|I_{n+1}, I_n, I_{n-1}) = \eta_s{}^i P_{n+1,s}{}^i P_{n,s}{}^i \tag{3}$$

Here $\eta_s{}^i$ is normalization constant that ensures the probabilities sum to one.

The term $P_{n,s}{}^i$ can be modeled to incorporate the distribution of the epipolar lines EL^i. Given an image point $p_{n-1}{}^i$ in I_{n-1} and its corresponding point $p_n{}^i$ in I_n then the epipolar line that passes through $p_n{}^i$ is determined as $l_n{}^i = e_n \times p_n{}^i$. The probability distribution of the feature point being stationary or moving due to epipolar constraint is defines as

$$P_{EP,s}{}^i = (2\pi \sum_t)^{-0.5} exp(-0.5(l_n{}^i - \mu_n{}^i)^\tau \sum l^{-1}(l_n{}^i - \mu_n{}^i)) \tag{4}$$

However this does not take into account the misclassification arising due to degenerate motion explained in previous sections. To overcome this the eventual probability is fused as a combination of epipolar and flow vector bound constraints as

$$P_{n,s}{}^i = \alpha \cdot P_{EP,s}{}^i + (1-\alpha) \cdot P_{FVB,s}{}^i \tag{5}$$

where, α balances the weight of each constraint. A χ^2 test is performed to detect if the epipolar line $l_n{}^i$ due to the image point is satisfying the epipolar constraint. When Epipolar constraint is not satisfied, α takes a value close to 1 rendering the FVB probability inconsequential. As the epipolar line $l_n{}^i$ begins indicating a strong likelihood of satisfying epipolar constraint, the role of FVB constraint is given more importance, which can help detect the degenerate cases.

An analogous set of equations characterize the probability of an image point being dynamic that are not delineated due to brevity of space. In our implementation, the envelope of epipolar lines [14] is generated by a set of F matrices distributed around the mean obtained from of the R,t transformation between two frames as estimated by the visual SLAM. Hence a set of epipolar lines corresponding to those matrices are generated and characterized by the sample set, $EL_{ss}{}^i = \left(\hat{l}_1{}^i, \hat{l}_2{}^i .....\hat{l}_q{}^i\right)$ and the associated probability set, $P_{EL} = \left(w\hat{l}_1{}^i, w\hat{l}_2{}^i .....w\hat{l}_q{}^i\right)$ where each $w\hat{l}_j{}^i$ is the probability of that line belonging to the sample set $EL_{ss}{}^i$ computed through usual Gaussian procedures. Then the probability that an image point $p_n{}^i$ is static is given by,

$$P_{n,s}{}^i = \sum_{j=1\rightarrow q} \alpha_j \cdot P_{EP,\hat{l}_j{}^i}{}^S \cdot p_n{}^i + (1-\alpha_j) \cdot P_{FVB,\hat{l}_j{}^i}{}^S \cdot p_n{}^i \cdot w\hat{l}_j{}^i \tag{6}$$

where, $P_{EP,\hat{l}_j{}^i}{}^S$ and $P_{FVB,\hat{l}_j{}^i}{}^S$ are the probabilities of the point being stationary due to the respective constraints with respect to the epipolar line $\hat{l}_j{}^i$.

C. Clustering Independent Motions

Features with high probabilities of being dynamic are either belongs to tracking outliers or potential moving objects. We adopt a simple move-in-unison model to cluster. Spatial proximity and motion coherence is used to cluster these feature tracks into independently moving entities. By motion coherence, we use the heuristic that the variance in the distance between features belonging to same object should change slowly in comparison. These features of spatial proximity and motion coherence are then used in an agglomerative clustering framework to divide the dynamic features into moving entities.

D. Feedback to Visual SLAM

Features lying over the independently moving objects are not used in the structure and motion estimation by the visual SLAM module. In spite of the use of robust estimators like RANSAC [26], independently moving objects can give rise to incorrect initial SfM estimate and lead the bundle adjustment to converge to a local minima. The feedback also results in less number of outliers in the visual SLAM process. Thus the structure and motion estimate is more well conditioned and less number of RANSAC iterations is needed [26]. Apart from improvement in the camera motion estimate, the knowledge of the independent foreground objects coming from motion segmentation helps in the data association of the features, which is currently being occluded by that object. For the foreground independent motions, we form a convex-hull around the tracked points clustered as an independently moving entity. Existing 3D points lying inside this region is marked as not visible and is not searched for a match. This prevents 3D features from unnecessary deletion and reinitialization, just because it was occluded by an independent motion for some time.

V. Experimental Results

The system is implemented as threaded processes in C++ and runs in realtime at the average of 22Hz. The Independent motion detection module takes around 10ms for each image of 512x284 resolution and with 3 independently moving bodies.

A. Results of Moving Object Detection

The system has been tested on a number of real image datasets, with varying number and type of moving entities. Moving object detection results in the three sequences are discussed next.

Moving Box Sequence: This is same sequence as used in [16]. A previously static box is being moved in front of the camera which is also moving arbitrarily. However unlike [16],

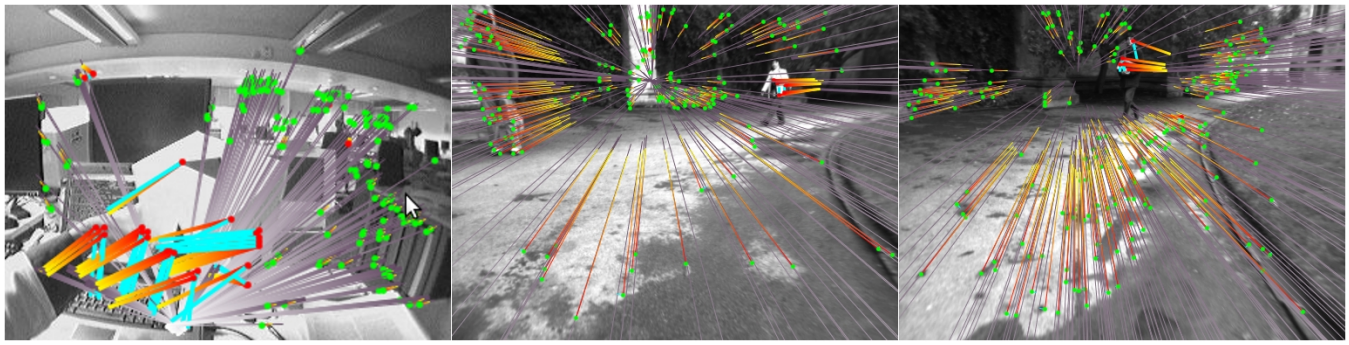

Fig. 2. Epipolar lines in Grey, flow vectors after rotation compensation is shown in orange. Cyan lines show the distance to epipolar line. Features detected as independently moving are shown as red dots. Note the near-degenerate independent motion in the middle and right image. However the use of FVB constraint enables efficient detection of degenerate motion.

our method does not uses any 3D model, and thus can work for any previously unseen object. As shown in Fig. 3 our algorithm reliably detects the moving object just on the basis of motion constraints. The difficulty with this sequence is that the foreground moving box is nearly white and thus provides very less features. This sequence also highlights the detection of previously static moving objects.

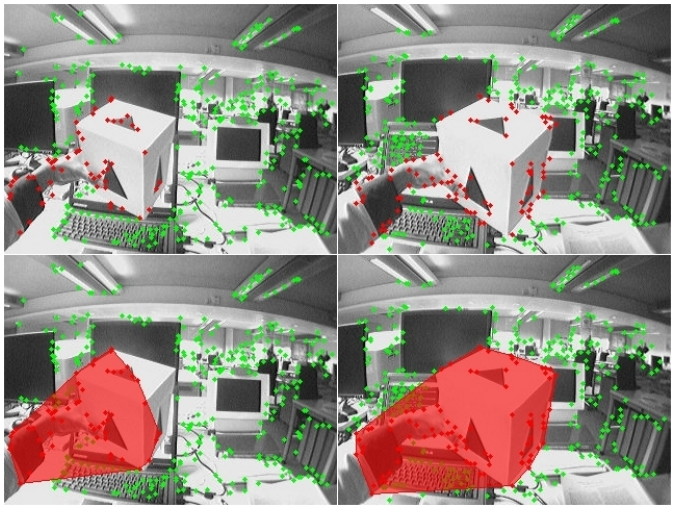

Fig. 3. Results from the Moving Box Sequence

New College Sequence: We tested our results on some dynamic parts of the publicly available New College dataset [27]. Only left of the stereo image pairs has been used. In this sequence, the camera moves along an roughly circular campus path, and three moving persons passes by the scene. Fig. 4 depicts the motion segmentation results for this sequence.

B. Detection of Degenerate Motions

Fig. 2 shows an example of degenerate motion detection, as the flow vectors on the moving person almost move along epipolar lines, but they are being detected due to usage the FVB constraint. This results verifies system's performance for

Fig. 4. Independent Motion detection results from the New College Sequence.

arbitrary camera trajectory, degenerate motion and changing number of moving entities.

C. Person detection

Some applications demand people to be explicitly detected from other moving objects. We use "part-based" representations [28], [29] for person detection. The advantage of the part-based approach is that it relies on body parts and therefore it is much more robust to partial occlusions than the standard approach considering the whole person. We model our implementation as described in [28]. Haar-feature based cascade classifiers was used to detect different human body parts, namely upper body, lower body, full body and head and shoulders. These detectors often leads to many false alarms and missed detections. Bottom-left image of Fig. 5 depicts the false detections, by this individual detectors. A probabilistic combination [28] of these individual detectors gives a more robust person detector. But running four Haar-like-feature based detectors on the whole image takes about 400ms, which is very high for realtime implementation. We use knowledge of motion regions as detected by our method, to

reduce the search space of part detectors. This greatly reduces the computations and the time taken is mostly less than 40ms. Also the detections have less false positives.

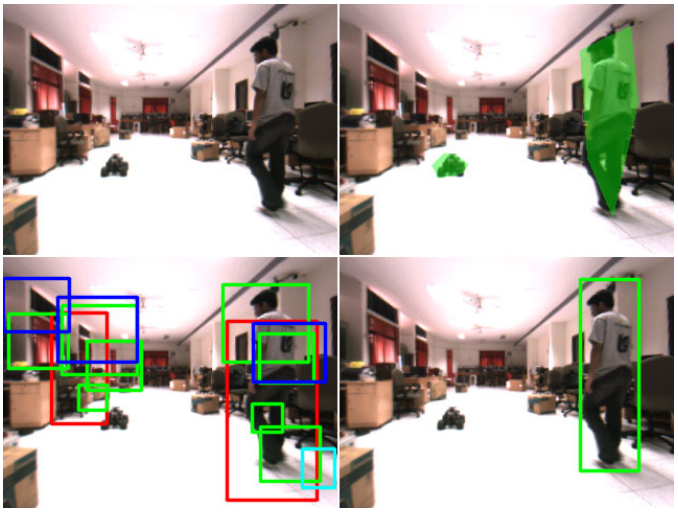

Fig. 5. TOP LEFT: A scene involving a moving toy car and person from the indoor sequence. TOP RIGHT: Detected moving regions are overlaid in green. BOTTOM LEFT: Haar classifier based body part detectors. BOTTOM RIGHT: Person detected by part-based person detection over image regions detected as moving.

VI. Conclusions

This paper presents a realtime moving object detection algorithm from a single freely moving monocular camera. An online visual SLAM algorithm running simultaneously estimates the camera egomotion. Multiview geometric constraints were explored to successfully detect various independent motion including degenerate motions. A probabilistic framework in the model of a recursive Bayes filter was developed that assigns probability of a feature being stationary or moving based on geometric constraints. Uncertainty in camera pose estimation is also propagated into this probability estimation. Unlike many existing methods, the proposed methods works with a full perspective camera model, and have no restrictive assumptions about camera motion and environment. The method presented here can find immediate applications in various robotics applications involving dynamic scenes.

References

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics. 2005*. MIT Press.

[2] A. Davison, I. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 29, no. 6, pp. 1052–1067, 2007.

[3] J. Neira, A. Davison, and J. Leonard, "Guest editorial, special issue in visual slam," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 929–931, October 2008.

[4] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, 2007.

[5] C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous localization, mapping and moving object tracking," *The International Journal of Robotics Research (IJRR)*, vol. 26, no. 9, pp. 889–916, 2007.

[6] A. Talukder and L. Matthies, "Real-time detection of moving objects from moving vehicles using dense stereo and optical flow," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[7] M. Agrawal, K. Konolige, and L. Iocchi, "Real-time detection of independent motion using stereo," in *IEEE Workshop on Motion and Video Computing*, 2005.

[8] Z. Chen and S. Birchfield, "Person following with a mobile robot using binocular feature-based tracking," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007.

[9] J. Wang and E. Adelson, "Layered representation for motion analysis," in *Computer Vision and Pattern Recognition (CVPR)*, 1993.

[10] S. Pundlik and S. Birchfield, "Motion segmentation at any speed," in *Proceedings of British Machine Vision Conference (BMVC)*, 2006.

[11] B. Jung and G. Sukhatme, "Real-time motion tracking from a mobile robot," *International Journal of Social Robotics*, pp. 1–16.

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

[13] 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 transactions on pattern analysis and machine intelligence (TPAMI)*, vol. 29, no. 9, pp. 1627–1641, 2007.

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

[15] J. Sola, "Towards visual localization, mapping and moving objects tracking by a mobile robot: a geometric and probabilistic approach," Ph.D. dissertation, LAAS, Toulouse, 2007.

[16] S. Wangsiripitak and D. Murray, "Avoiding moving outliers in visual SLAM by tracking moving objects," in *International Conference on Robotics and Automation (ICRA)*, 2009.

[17] D. Migliore, R. Rigamonti, D. Marzorati, M. Matteucci, and D. G. Sorrenti, "Avoiding moving outliers in visual SLAM by tracking moving objects," in *ICRA'09 Workshop on Safe navigation in open and dynamic environments*, 2009.

[18] A. Kundu, K. Krishna, and J. Sivaswamy, "Moving object detection by multi-view geometric techniques from a single camera mounted robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, 2009.

[19] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, pp. 105–119, 2010.

[20] E. Mouragnon, F. Dekeyser, P. Sayd, M. Lhuillier, and M. Dhome, "Real time localization and 3d reconstruction," in *Computer Vision and Pattern Recognition (CVPR)*, 2006.

[21] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *Computer Vision and Pattern Recognition (CVPR)*, 2004.

[22] J. Civera, A. Davison, and J. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 932–945, 2008.

[23] H. Strasdat, J. Montiel, and A. Davison, "Real-Time Monocular SLAM: Why Filter?" in *International Conference on Robotics and Automation (ICRA)*, 2010.

[24] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 26, no. 6, pp. 756–770, 2004.

[25] B. Haralick, C. Lee, K. Ottenberg, and M. Nolle, "Review and analysis of solutions of the three point perspective pose estimation problem," *International Journal of Computer Vision*, vol. 13, no. 3, pp. 331–356, 1994.

[26] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[27] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman, "The new college vision and laser data set," *The International Journal of Robotics Research (IJRR)*, vol. 28, no. 5, p. 595, 2009.

[28] Z. Zivkovic and B. Krose, "Part based people detection using 2D range data and images," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007.

[29] B. Wu and R. Nevatia, "Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors," *International Conference on Computer Vision (ICCV)*, 2005.