Motion Segmentation of Multiple Objects from a Freely Moving Monocular Camera

Rahul Kumar Namdev, Abhijit Kundu, K Madhava Krishna and C. V. Jawahar

***Abstract*— Motion segmentation or segmentation of moving objects is an inevitable component for mobile robotic systems such as the case with robots performing SLAM and collision avoidance in dynamic worlds. This paper proposes an incremental motion segmentation system that efficiently segments multiple moving objects and simultaneously build the map of the environment using visual SLAM modules. Multiple cues based on optical flow and two view geometry are integrated to achieve this segmentation. A dense optical flow algorithm provides for dense tracking of features. Motion potentials based on geometry are computed for each of these dense tracks. These geometric potentials along with optical flow potentials are used to form a graph like structure. A graph based segmentation algorithm then clusters together nodes of similar potentials to form the eventual motion segments. Experimental results of high quality segmentation on different publicly available datasets demonstrate the effectiveness of our method.**

I. INTRODUCTION

Understanding dynamic scenes from moving cameras or cameras mounted on mobile robots is an inevitable component in many applications. For example large city scale mapping with outdoor vehicles, robotic aids in supermarkets, robotic assistance at home and office require moving cameras and sensors interacting with other moving objects, vehicles and people. Often an unavoidable component of such applications is the ability of the moving camera to segment moving objects. A robust system capable of doing so can also go a large way in solving the popular Vision based Simultaneous Localization and Mapping (VSLAM) problem in dynamic environments.

VSLAM involves simultaneously estimating locations of newly perceived landmarks and the location of the moving camera itself while incrementally building a map of an unknown environment. The last decade has seen a significant shift towards vision based SLAM systems [1], [2], [3] from range sensor based systems. However almost all these SLAM approaches assume a static environment, containing only rigid, non-moving objects. The solution to the moving object segmentation and extraction problem will act as a bridge between the static SLAM and its counterpart for dynamic environments.

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 multi view geometric constraints. These constraints can be used for the motion detection task. Those inconsistent with the constraints can be labelled as moving regions or outliers. However the multi view geometric constraints by themselves are unable to segment moving objects completely and the boundaries do not get formed sharp enough.

Rahul Kumar Namdev, Abhijit Kundu, K.M. Krishna and C. V. Jawahar are with IIIT Hyderabad, India.
rahul.namdev@research.iiit.ac.in
abhijit.dgp@gmail.com
{mkrishna,jawahar}@iiit.ac.in

Fig. 1. This Figure shows a result of our algorithm on 3 men sequence. (a) An image from 3 men sequence, in this image the 3 persons are moving. (b) The motion segmentation results from our segmentation algorithm.

In lieu of this we propose a graph based framework for motion segmentation where geometric constraints are further supported by optical flow vectors. The optical flow enhances the segmentation in two ways. Firstly it provides for dense tracking of features and hence a dense segmentation of motion. Secondly it provides motion cues through 2D optical flow values that supplements the two view geometric constraints for a sharper segmentation especially at boundaries. Even with dense tracking, geometric constraints need a wider baseline for their accuracy, which invariably introduces certain sparseness in the segmentation. The optical flow's ability to operate at smaller baselines enables a denser detection apart from segmentation providing for a interpolatory effect over geometric cues.

The algorithm proposed is an incremental motion segmentation algorithm with the aid of an online visual SLAM algorithm. The motion segmentation 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 use efficient geometric constraints that helps in segmenting these degenerate motions.

The segmentation algorithm relies on dense point trajectories [4] calculated using the optical flow from **Classic+NL** algorithm [5] and is not truly real-time due to processing time for computing the optical flow. The reason for using this optical flow algorithm is its high accuracy.

Geometric motion potentials are computed and assigned to the tracks based on geometric constraints. These potentials

along with optical flow potentials are then fed to the graph based segmentation algorithm. The segmentation algorithm is able to reliably segment entire moving objects from the stationary world.

One of the essential contributions of this paper is the use of motion cues to formulate the motion potentials that characterize the node potentials for a graph based motion clustering algorithm. The other essential contributions of this paper is in its judicious integration of multi view geometric and 2D optical flow cues to achieve dense segmentation. The segmentation results are shown for highly challenging outdoor scenes where the area corresponding to moving objects in the image are segmented to very high accuracy. Its ability to work with complex outdoor scenes over large time span is one of the unique features of this work. Many previous efforts such as [6], [7] do not report such results. The closest to this effort is the work on motion cuts [8], [9]. However the results portrayed show only moving object in the foreground. In contrast the current effort is able to segment multiple moving objects from background on more complex outdoor scenes with highly degenerate motion of moving objects. Another difference is that the current algorithm is incremental. Unlike [8], [9] it does not use future images in the pipeline to achieve segmentation of the current.

II. Related Works

The task of moving object segmentation and extraction, is much easier if a stereo sensor is available, which allows additional constraints to be used for detecting independent motion [10], [11], [12], [13]. However the problem is lot more difficult 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. Literature in this area can be loosely divided into four categories. The first category of methods rely on estimating a global parametric motion model of the background. These methods [14], [15], [7] compensate camera motion by 2D homography or affine motion model. Pixels consistent with the estimated model are assumed to be background. 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 [16], [17] based constraints, which forms the second category of approaches. 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 ego motion and structure estimation, the traditional multi-view geometry constrains like epipolar constraint in 2 views or trilinear 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. These constraints are well understood and are now textbook materials [18]. The methods that use such 2-view geometry constraints can be considered as the third category. In this work we use such multi-view geometric cues such as the epipolar constraints and flow vector bound constraint [19] for motion segmentation through formulation of recursive geometric motion potentials.

The fourth category of approaches either use pixel level motion cues such as various forms of optical flows as [20], [21] or appearance cues like color, texture or a combination of both [22]. The methods that use pixel level motion clues inherently tend to suffer from edge effects resulting in various false positives. The approaches that use robust two view geometric constraints for motion potential computations seem rare. The results depicted in these approaches do not show motion segmentation in the presence of highly degenerate motions. In that regard the current work differs in using such multi view geometric constraints to assign motion potentials as well as segmenting multiple moving objects which possess severe degenerate motions over several frames of difficult outdoor sequences. Motion potentials are formed both due to optical flow and geometric constraints. The graph clustering algorithm suitably integrates them to achieve final segmentation. In our earlier works we have detailed independent motion detection of sparse features where camera motion was compensated through odometry in [23], a multibdoy VSLAM framework that enhanced the sparse motion detection in [19]. Whereas in [24] the focus was on motion reconstruction of moving objects through the multibody framework. The current work is different as it achieves dense segmentation through a graph based framework integrating both geometry and optical flow cues.

III. System Overview

Fig. 2 gives an overview of our system. From an image sequence we calculate optical flow and dense feature tracks while simultaneously running a VSLAM system in the background. This VSLAM provides camera ego-motion parameters which are then used to calculate multi-view geometric constraints. These geometric constraints are used to calculate motion potentials due to geometry and these along with the optical flow based motion potentials are given to a graph based clustering algorithm to achieve motion segmentation.

A. Dense Tracking for Segmentation

The dense tracking is a vital part of dense motion segmentation. The requirement for dense correspondence and its subsequent tracking stems from the observation that an accurate segmentation that captures most of the image area underlying the moving object would require such tracks. Dense tracking provides for assignment of edge potentials for edges that connect adjacent nodes or nodes that are spatially proximal. A graph based clustering algorithm can

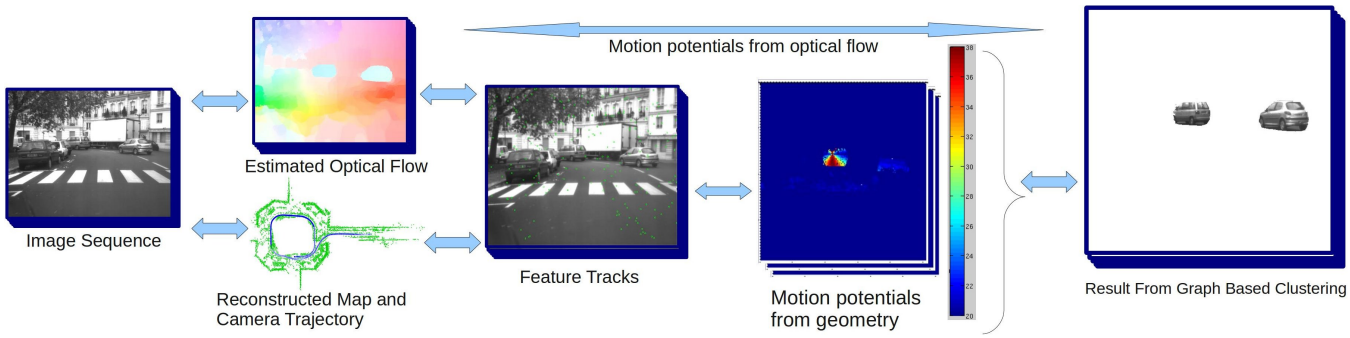

Fig. 2. An overview of our system. Here we calculate dense optical flow over a sequence of images and find dense feature tracks over this dense optical flow. Feedback from VSLAM gives camera motion parameters which are than used to calculate dense geometric motion potentials. These potentials along with optical flow potentials are given to graph based motion clustering algorithm to achieve dense segmentation.

then use such edge potentials to partition the graph into motion clusters.

In this effort we make use of the optical flow implementation provided by [5]. A dense point trajectory tracking scheme from [4] over this optical flow have been used to get dense tracks. This algorithm reasons consistency check for occlusions using forward and backward flow for getting accurate tracks. As mentioned by the authors resulting technique tracks up to three orders of magnitude more points and is 46% more accurate than the KLT tracker. It also provides a tracking density of 48% and has an occlusion error of 3% compared to a density of 0.1% and occlusion error of 8% for the KLT tracker.

B. Visual SLAM

The segmentation algorithm proposed is independent of the SLAM algorithm used. However, we chose the bundle adjustment visual SLAM [25], [3], [26] framework over the filter based approaches [1], [27] because of the accuracy benefits [28]. Our Visual SLAM implementation closely follows to that of [25], [3]. In brief, a 5-point algorithm [29] with RANSAC is used to estimate the initial epipolar geometry, and subsequent pose is determined with 3-point resection [30]. Some of the frames are selected as key-frames, which are then used to triangulate 3D points. The set of 3D points and the corresponding key frames are used in by the bundle adjustment process to iteratively minimize re-projection error. The bundle adjustment is initially performed over the most recent key frames, 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 optimizes this estimate by bundle adjustment. The most important application of this VSLAM module is to provide camera ego-motion parameters which are used to calculate efficient geometric constraint. The ego-motion parameter can be computed based on principle of two view geometry alone. However the benefits of the bundle adjustment optimization procedure are provided through accurate description of camera ego-motion parameters entailed for robust segmentation. Thus VSLAM is an integral part of the segmentation pipeline.

C. Geometric Constraints

For segmenting/detecting highly difficult degenerate motions we need to use strong geometric constraint. For this purpose we have used a combination of commonly used epipolar constraint along with Flow Vector Bound constraint introduced by us in [19], [23] . Epipolar constraint alone is not able to handle degenerate motions, for example if a point moves in the epipolar plane then epipolar constraint will fail to work. In this case Flow vector bound works well. Flow vector bound can take care of the highly degenerate cases where moving object is closely followed by the camera. These geometric constraints are explained in detail in [19], [23]. The section on motion potential that follows makes use of these geometric constraints.

D. Assigning Motion Potentials

Computing Epipolar Potentials Let p_n and p_{n+1} be the images of a same 3D point, X in two views or images I_n and I_{n+1}. Let the epipolar line in I_{n+1}, corresponding to p_n be l_{n+1}. If the 3D point is stationary then p_{n+1} should ideally lie on l_{n+1} and hence $l_{n+1} \cdot p_{n+1} = 0$. But if a point is not stationary, the dot product $l_{n+1} \cdot p_{n+1}$ is a measure of the deviation of the point from the line. Thus higher the dot product larger is the potential that the point is moving. Similarly if l_n is the epipolar line corresponding to p_{n+1} in I_n the epipolar potential for frame n is computed as $epp_n = \frac{l_{n+1} \cdot p_{n+1} + l_n \cdot p_n}{2k_1}$, where $k_1 > 1$ is a scaling constant, which is useful in providing spatial smoothness to Epipolar potentials. A very small value of k_1 does not provide any smoothness, where as a very large value of k_1 makes the contribution of Epipolar potential in graph based clustering insignificant. We have used a value of $k_1 = 10$, throughout all of our experiments.

Computing Flow Vector Bound Potentials

For calculating Flow vector bound potentials we compute for a pair of corresponding points p_n and p_{n+1} the maximum and minimum flow possible, d_{max} and d_{min} corresponding to minimum and maximum possible depth of the world point. If the actual flow vector fv computed as the pixel distance between p_n and p_{n+1} lies in the interval bound $[d_{min}, d_{max}]$ the flow vector potential is computed as $fvbp = \frac{fv}{c_1}$. This is

a case when the point is not moving and contribution of its flow vector in graph based clustering should be minuscule; because of this a very high value of $c_1 = 100$ has been used throughout all the experiments.

If the flow vector fv *does not* lie in the interval $[d_{min}, d_{max}]$ then the potential is computed as $fvbp = \frac{fv}{c_2} + c_3$. This is the case when the point is moving. Similar to k_1, c_2 provides for spatial smoothness and a value of $c_2 = 10$ has been used in all experiments. The purpose of c_3 is to make a clear distinction between a $fvbp$ of moving point and $fvbp$ of a non-moving point to the graph based clustering. A value of $c_3 = 5$ has been used throughout all the experiments.

E. Recursive Potential Estimation

Recursive potential estimation seeks to factor in the temporal dimension of the geometric potentials in the eventual potential calculation. If over multiple views the epipolar potential is observed to be high for a track then the potential that it is a moving object is higher than if a track acquires such a potential for the first time. By the same token if the epipolar or flow vector potential turns out lower over multiple instances the potential that the object is moving is much lower than if such a low potential is observed on the track first time.

The following recursive definition was found to work well for the eventual motion potential, due to Epipolar potential, Mep_n

$$\text{Mep}_n = \begin{cases} Mep_{n-1} + \theta & epp_n > \lambda, Mep_{n-1} > \lambda, \\ & Mep_{n-1} < M \\ Mep_{n-1} - \theta & epp_n < \eta \& Mep_{n-1} < \eta, \\ & Mep_{n-1} > 0 \\ \frac{\sum_{i-k}^{n} epp_i}{k} & \text{otherwise} \end{cases} \quad (1)$$

The above recursive definition adds to the motion potential due to epipolar constraint if the current epipolar potential and previous motion potential are above a certain threshold, λ, and less than a maximum threshold value M. Similarly potentials are subtracted by a value θ if both current epipolar potential and previous motion potentials are less than a threshold η and greater than zero. In all other cases a moving average over the last k instances constitutes the motion potential due to epipolar potential.

Similar recursive definitions characterize the motion potential due to flow vector bound potential denoted by $Mfvp$.

The utility of such a recursive definition of motion potentials can be seen in a Fig. 3. Here regions of low motion potentials (indicating stationary areas) become regions of high motion potentials as color changes from shades of blue to brown with time. The scenario corresponds to complex case of two cars moving in front of the moving camera through a highly degenerative motion sequence. At first instance very few parts of the car have a high motion potential. With time most areas of the car are covered as a consequences of recursive estimation of potentials.

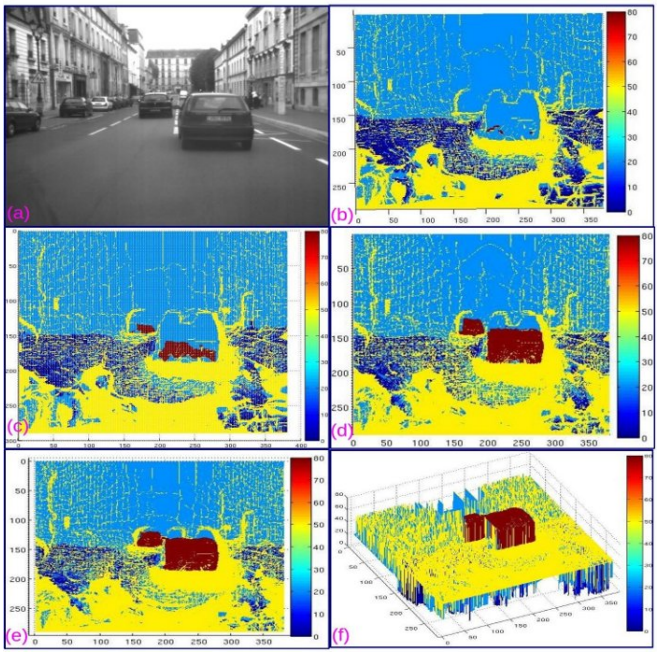

Fig. 3. In this figure we have shown intermediate results of recursive motion potentials due to flow vector bound. (a) An Image from Versailles Rond dataset. (b), (c), (d) and (e) shows that regions of low motion potentials become regions of high motion potentials as color changes from shades of blue to brown with time. (f) is just a 3D version of the (e) showing that all the points which are in shades of brown have probability values greater than .5(multiplied by 100) and the points which are in shades of blue have probability less than .5.

F. Optical Flow Potentials

The optical flow potentials are a transformation of optical flow vectors to a color space. This transformation is the Flow field color coding given at [31].

G. Graph-Based Recursive Clustering

Our graph based clustering algorithm has its base in Efficient graph based image segmentation algorithm [32] but modified suitably for segmentation in videos as well as in the manner in which it integrates multiple cues; in this case the geometry and optical flow cues. Let $G = (V, E)$ be an undirected graph with vertices $v_i \in V$, the set of elements to be segmented, and edges $(v_i, v_j) \in E$ corresponding to pairs of neighbouring vertices. Each edge $(v_i, v_j) \in E$ has corresponding weights $w_p((v_i, v_j)) \forall p \in U$ where U is the set of different properties(epipolar potentials, fvb potentials and optical flow potentials) used for segmentation. Each of these weights is a non-negative measure of some kind of dissimilarity between neighbouring elements v_i, v_j. In the case of motion segmentation elements in V are pixels and weights are W_{geo} and W_{of}, where W_{geo} measures the dissimilarity due to geometric motion potentials and W_{of} is the measure of dissimilarity in optical flow potentials.

In our graph based clustering algorithm we want results of our algorithm to be a set of connected components (or regions) in the graph $G = (V, E)$ such that each connected component has similar nodes (nodes with similar properties)

grouped together and this set of connected components forms a partition of the V.

1) We sort E into $t = (o_{geo_1}, ..., o_{geo_m})$, by non-decreasing geometric edge weights, where each geometric edge weight is defined as
$Wgeo((v_i, v_j)) = \sqrt{w_{epi}((v_i, v_j))^2 + w_{fvb}((v_i, v_j))^2}$
where
$w_{fvb}((v_i, v_j))$ = |fvbpotential(v_i)-fvbpotential(v_j)|
$w_{epi}((v_i, v_j))$ = |epipotential(v_i)-epipotential(v_j)|
2) Initially each vertex $v_i \in V$ is a cluster in it self.
3) New clusters are formed from previous clusters if vertices v_i and v_j connected by the qth edge in ordering i.e. $o_{geo_q} = (v_i, v_j)$ is such that v_i and v_j are in two different clusters than,
if ($w(o_{geo_q})$ <internal difference of both clusters)
group the clusters
else
if ($w_o f(o_q) < \alpha_k$)
group the clusters
else
do nothing.
Internal difference of the clusters is defined to be the maximum edge weight in the minimum spanning tree of the cluster, $MST(C, E)$. That is $Int(C) = \max_{e \in MST(C,E)} w(e)$.
More formally, let C_1 and C_2 are two different clusters including v_i and v_j the clusters will be merged if $w(o_{geo_q}) \leq Int(C_1, C_2)$ or $w(o_{of_q}) \leq \alpha_k$ (α_k is threshold for difference in optical flow potential between two pixels).
4) To enforce temporal consistency the following is done. The algorithm considers a node, $v_i(t)$, at time instant t and records the cluster $C_j(t)$ with which it is associated at that instant. The clusters to which the same node v_i was associated in previous n instances are also looked at. If there is an uninterrupted sequence of atleast k instances where the node belonged to the same cluster, C_k, and if the last instant of that uninterrupted sequence is no earlier than three instances before the current instant, t, then $v_i(t)$ is still associated with C_k. This is done to ensure continuity with the past. Else $v_i(t)$ is associated to C_j. A more formal procedure invoking a decaying term in the edge potential due to optical flow was tried that did not give as effective results as above. So it was decided to continue to maintain temporal consistency in the above fashion.

IV. Experimental Results and Analysis

A. Results

We validate our algorithm using our own Lab dataset and several other publicly available datasets.

3 Men Sequence: This dataset was created using a Point Grey firewire camera. This outdoor dataset has 3 moving people in some part of IIIT-Hyderabad campus. Fig. 4 shows that our method can efficiently segment multiple moving objects. Fig. 4(a) shows an image from 3 men sequence.

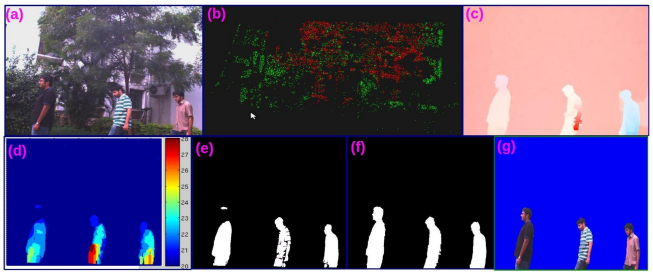

Fig. 4. These are the results on the 3 men sequence. This dataset was collected from a Point Grey firewire Camera.(a) An image of 3 men sequence. (b) Sparse map of the scene. (c) optical flow map. (d) Recursive geometric motion potentials map. (e) Results from only geometry potentials. (f) Final results calculated from geometry potentials and optical flow potentials. (g) Extracted moving people.

Fig. 4(b) shows the sparse map of the scene generated by Visual SLAM module. Fig. 4(c) shows optical flow map and Fig. 4(d) shows the estimated recursive geometric motion potentials. Fig. 4(e) shows the segmentation due to only these motion potentials. Fig. 4(f) shows the motion segmentation results from the coherence of the optical flow potentials and geometric potentials. This figure clearly explains that segmentation results from coherence of optical flow and geometric are better than the segmentation results due to only geometry. Fig. 4(g) shows us the extracted moving objects.

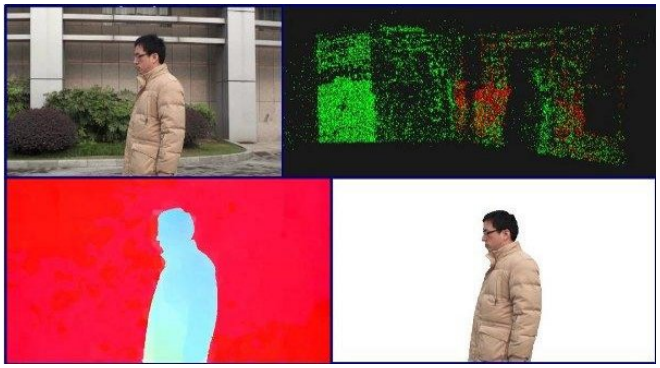

Fig. 5. Results on Rotation sequence. TOP LEFT: A scene involving a moving person from Rotation sequence. TOP RIGHT: The map of the scene generated by VSLAM. BOTTOM LEFT: Optical flow map. BOTTOM RIGHT: The extracted moving person.

Motion-Cut dataset: This is the same dataset which was used in [8]. We have shown our results on 2 different sequences namely Rotation Sequence and Cube Sequence out of many sequences in this dataset. Fig. 5 shows results on the Rotation sequence. Fig. 6 shows the results on the cube sequence.

Versailles Rond Dataset: This is an urban outdoor sequence taken from a fast moving car, with multiple number of moving objects appearing and leaving the scene. Some parts of this dataset have upto 4 moving cars and we are able to segment all of them to a robust accuracy. Only left of the stereo image pairs has been used. Fig. 7 shows the motion segmentation and reconstruction results.

CamVid Dataset: We tested our system on some dynamic

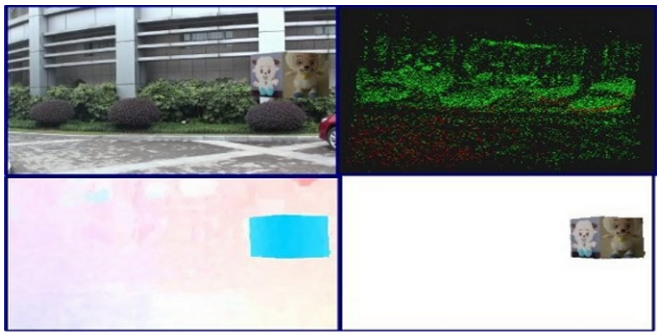

Fig. 6. Results on Cube sequence. This sequence has a moving cube. TOP LEFT: A scene involving a moving object from Cube sequence.TOP RIGHT: The map of the scene generated by VSLAM. BOTTOM LEFT: Optical flow map. BOTTOM RIGHT: The extracted moving object.

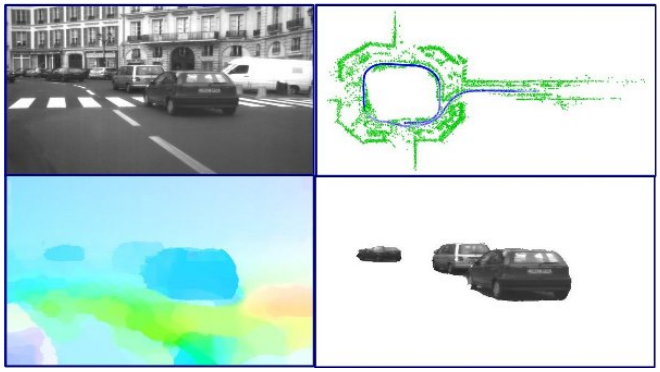

Fig. 7. Results on a dynamic part of Versailles Rond Dataset. This image has 3 moving cars. TOP LEFT: A scene involving 3 moving cars. TOP RIGHT: The map generated by VSLAM. BOTTOM LEFT: Optical flow map. BOTTOM RIGHT: The three extracted moving cars.

parts of the publicly available CamVid dataset [33], [34]. This a road sequence involving a camera mounted on a moving car. The results on this sequence is highlighted in Fig. 8. Fig. 8(a) shows an image of the scene. Fig. 8(b) is the recovered optical flow map. Fig. 8(c) is the result of clustering over geometric motion potentials only. Fig. 8(d) is the final segmentation result due to integration of optical flow potentials and geometric potentials. Fig. 8(e) is the final extracted moving object. It should be noted that in the current result some part of the moving car is occluded by a pole and we are able to successfully recover both parts of the car, whereas the part corresponding to the pole is correctly classified as the non moving background.

B. Analysis

In this section we describe how by judiciously integrating geometry and 2D optical flow cues we are able to overcome both larger errors and smaller inaccuracies in motion segmentation. The integration process essentially exploits the coherence of optical flow and geometry cues as well as temporal coherence to provide for a more consistent segmentation closer to ground truth.

Preservation of Motion Coherence: Fig. 4(e) shows the clusters formed by the graph based clustering algorithm purely based on geometric cues. The degenerate motion of

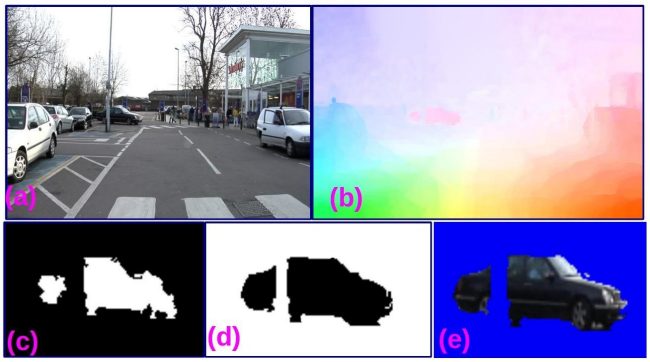

Fig. 8. Results on a some dynamic part of CamVid Dataset. (a) An image from CamVid dataset. This image has 1 moving car and some part of that car is being occluded by a pole. (b) Recovered optical flow Map. (c) Zoomed version of result from geometry only. (d) Zoomed version of final segmentation result. (e) Zoomed version of final extracted moving object.

pixels between the head and neck of the person on the left results in parts of these pixels having a potential similar to the static background. This results in the moving person getting segmented into more than one object. On the other hand the optical flow map is shown in Fig. 4(c). The coherence in optical flow potentials result in a single cluster formed across the face. The integration step explained in step 3 of the algorithm in III-G results in an eventual segmentation shown in figure Fig. 4(f), wherein clusters separated by geometric cues are merged together if the 2D motion cues from optical flow indicate a large degree of coherence amongst them.

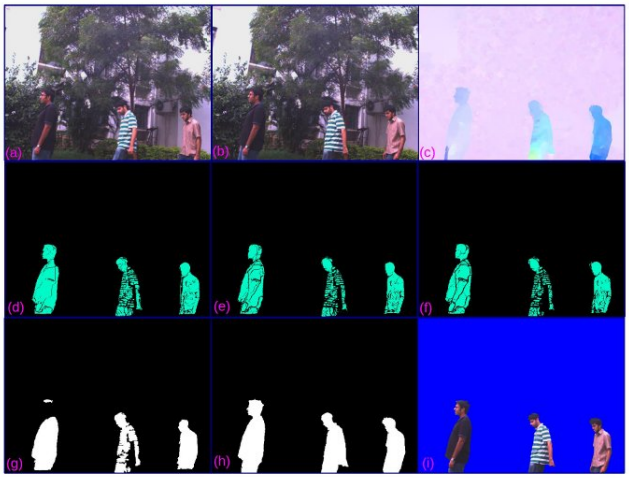

Fig. 9. This figure shows an explanation of decreasing no. of tracks over wider baseline. (a), (b) are two images from 3 men sequence separated by 4 images in between. (c) most recent optical flow map. (d), (e), (f) corresponds to tracks on the 4th image, 5th image and 6th and final image. (g) is result due to geometric cues only. (h) results from coherence of geometry and optical flow and (i) is the final extracted moving objects.

Maintaining Segmentation at Narrow Baselines: The geometric cues entail a wider baseline for their efficacy. Despite the denseness of segmentation the tracks obtained over wider baselines tend to be less dense than those obtained over narrow baselines. As a result there is an ambiguity

to cluster separated pixels together; one such segmentation result is shown in Fig. 9. Fig. 9(a), (b) shows 2 images from 3 men sequences separated by 4 images in between. (c) shows the very recent optical flow map. Fig. 9(d), (e), (f) shows that number of tracks decreases as baseline becomes wider. This can be seen through increased appearance of background pixels (black) onto the foreground objects. Here tracks only on the moving objects have been shown. The ambiguity regarding whether or not to merge the various separated clusters resembling salt and pepper noise is resolved through motion coherence cues from optical flow. Fig. 9(g) shows result of clustering on geometric motion potentials only. Fig. 9(h) shows the eventually segmented motion clusters. Fig. 9(i) shows the extracted moving objects.

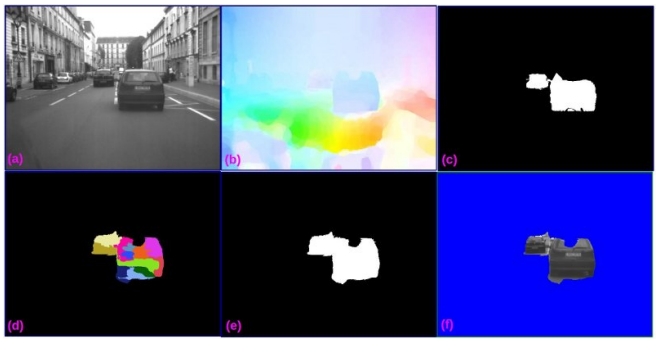

Fig. 10. This figure gives an explanation geometric coherence. (a) An image from the Versailles Rond dataset. (b) Optical flow map. (c) Motion segmentation result from the geometric potentials only. (d) clustering over the recovered optical flow map. Cluster only on two moving cars have been shown here. (e) This image shows the result of geometric coherence along with optical flow. (f) Final extracted moving cars.

Geometric Coherence: As much as optical flow cues provide for coherence of 2D motion, geometric cues can provide coherence where optical flow need not. For example Fig.10(d) shows clustering over optical flow map of the image 10(a). Clusters formed only on two moving cars have been shown here. It should be noted that multiple clusters are formed on the two cars due to lack of coherence in 2D flow vectors. However the geometric potentials are similar across both the cars as they primarily portray the satisfaction or not of the geometric constraints. Thus the clusters formed by geometry are as shown in Fig.10(c), while the overall segmented motion clusters are shown in Fig.10(e) as a result of the integration proposed in step 3 of the segmentation algorithm III-G.

Temporal Coherence: The temporal coherence is portrayed by Fig. 11. In this result Fig. 11(a) is an image from CamVid dataset. (b) is the recovered optical flow map. (c), (d) and (e) are three consecutive results for frames previous to current frame shown in Fig. 11(a). (f) is the intermediate result of motion segmentation algorithm without temporal coherence. (g) is the result of recursive clustering III-G step 4. In this result the violet clusters are supposed to be part of non-moving background. (h) is the final result of recursive clustering and (i) shows the final extracted moving cars. Parts of non moving background (white) in 11(c), 11(d), 11(e) gets

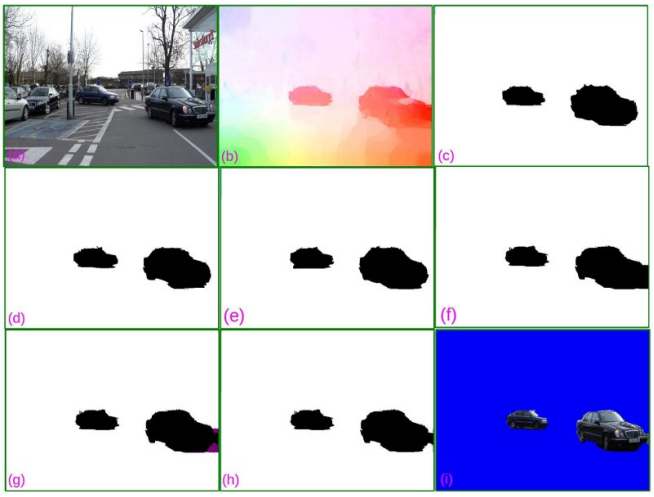

Fig. 11. This figure shows an explanation of temporal coherence. (a) An image from CamVid dataset. (b) Recovered optical flow map. (c), (d) and (e) are motion segmentation results on last three frames. (f) clustering on the current optical flow. (g) is the result from step 4 of the recursive clustering algorithm III-G . In this result the clusters which are in violet are supposed to be the part of non moving background. (h) gives the final result from the temporal recursive clustering. (i) shows the final extracted foreground.

classified as moving in 11(f). Enforcing temporal consistency as described in step 4 of algorithm mentioned in section III-G results in parts of the background correctly classified in figure 11(h).

C. *Datasets used:*

The datasets used in this paper are complex in various ways. For example the three men dataset has at various places parts of the moving foreground merge with the background if one would use state of the art grabcut algorithms to segment foreground based on color. Figure 12 shows a grabcut result on figure 4(a), where the persons in the middle and left are not segmented accurately as the foreground, indicating how the foreground blends with the background in this dataset. Such a situation is not to be found for example in the dataset over which results were shown in [8], where grabcut results match the results portrayed in [8]. These experiments are not shown here for brevity of space. Also the results shown for Versailles Rond which is a grayscale dataset portray a very difficult degenerate case of the camera tailing the moving cars. The motion cues used in this paper is able to segment such objects with significant accuracy. The CamVid dataset consists of pedestrians moving slowly at very far distances from the camera, which are difficult to detect and segment at narrow baselines. The uploaded video shows segmentation of such moving objects at far distances.

V. System Details

The system is implemented in MATLAB with the aid of C++ and OpenCV. We have used the **Classic+NL** version of the optical flow mentioned in [5]. We tested our implementation on standard laptop (Intel Core i7) and in our implementation the total computational time is 7 minutes for each frame averagely. The computational cost is mostly in

Fig. 12. This figure shows an explanation of how grabcut (color consistency) fails on our 3 men dataset.(a) Rectangle in green is the bounding box for grabcut algorithm. Blue shows stroke for background and red shows stroke for foreground. (b) Result from grabcut. (c) shows result for grabcut on the left most person.

the optical flow calculation step which itself takes around 6.5 minutes. We also tested our system using other optical flow algorithms which are real time. Even with these optical flows, we are able to get very high quality segmentation. The reason for using **Classic+NL** optical flow is its use of median filtering and preservation of motion information even at boundaries of object.

VI. Conclusions

This paper presented a method for dense segmentation of multiple moving objects from a moving monocular camera. The method integrates optical flow and geometry cues to provide for a dense segmentation through a graph based clustering algorithm that has been modified to work on videos as well as to aptly integrate multiple cues. The paper describes how both these cues are able to complement each other to segment moving objects performing difficult degenerate motions. The results have been shown on five different outdoor datasets, where accurate segmentations showcase the efficacy of the method. The authors opine that this is the first such result to show dense motion segmentation with a single moving camera on various outdoor datasets with multiple moving objects. A fully perspective model of the camera has been used and there are no restrictions on the kind of motion to be professed either by the camera or the objects. Such a segmentation method goes hand in hand with VSLAM systems that reconstruct not only the stationary world but also the dynamic objects thereby providing for a robust scene interpretation and mapping. Such systems would find immense use in outdoor autonomous navigation and collision avoidance.

VII. Acknowledgements

This work was supported by Department of Science and Technology (DST) India.

References

[1] A. Davison, I. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *(TPAMI)*, vol. 29, no. 6, pp. 1052–1067, 2007.

[2] 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.

[3] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *(ISMAR)*, 2007.

[4] K. K. Narayanan Sundaram, Thomas Brox, "Dense Point Trajectories by GPU-accelerated Large Displacement Optical Flow," in *(ECCV)*, 2010.

[5] M. J. B. Deqing Sun, Stefan Roth, "Secrets of optical flow estimation and their principles," in *(CVPR)*, 2010.

[6] R. V. Rita C., Andrea P., "Real-time motion segmentation from moving cameras," *ELSEVIER*, 2004.

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

[8] W. X. Guofeng Zhang, Jiaya Jia, T.-T. Wong, P.-A. Heng, and H. Bao, "Moving Object Extraction with a Hand-held Camera," in *(ICCV)*, 2007.

[9] W. H. Guofeng Zhang, Jiaya Jia and H. Bao, "Robust bilayer segmentation and motion/depth estimation with a handheld camera," *(TPAMI)*, 2011.

[10] A. Ess, K. Schindler, B. Leibe, and L. van Gool, "Improved multi-person tracking with active occlusion handling," in *ICRA Workshop on People Detection and Tracking*, May 2009.

[11] A. Talukder and L. Matthies, "Real-time detection of moving objects from moving vehicles using dense stereo and optical flow," in *(IROS)*, 2004.

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

[13] Z. Chen and S. Birchfield, "Person following with a mobile robot using binocular feature-based tracking," in *(IROS)*, 2007.

[14] J. Wang and E. Adelson, "Layered representation for motion analysis," in *(CVPR)*, 1993.

[15] S. Pundlik and S. Birchfield, "Motion segmentation at any speed," in *(BMVC)*, 2006.

[16] M. Irani and P. Anandan, "A unified approach to moving object detection in 2D and 3D scenes," *(TPAMI)*, vol. 20, no. 6, pp. 577–589, 1998.

[17] 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," *(TPAMI)*, vol. 29, no. 9, pp. 1627–1641, 2007.

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

[19] A. Kundu, K. M. Krishna, and C. V. Jawahar, "Realtime motion segmentation based multibody visual slam," in *(ICVGIP)*, 2010.

[20] S. M. Chang M., Teklap A, "Simultaneous motion estimation and segmentation," *IEEE Transactions on Image Processing*, 1997.

[21] T. G. Grinias I, "Motion segmentation and tracking using a seeded region growing method," *Proceedings of European Signal Processing Conference*, 1998.

[22] B. P. Gelgon M, "A region-level motion-based graph representation and labeling for tracking a spatial image partition," *Pattern Recognition*, 2000.

[23] A. Kundu, K. M. Krishna, and J. Sivaswamy, "Moving object detection by multi-view geometric techniques from a single camera mounted robot," in *(IROS)*, 2009.

[24] A. Kundu, K. M. Krishna, and C. V. Jawahar, "Realtime multibody visual slam with a smoothly moving monocular camera," in *(ICCV)*, 2011.

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

[26] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *(CVPR)*, 2004.

[27] 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.

[28] H. Strasdat, J. Montiel, and A. Davison, "Real-Time Monocular SLAM: Why Filter?" in *(ICRA)*, 2010.

[29] D. Nister, "An efficient solution to the five-point relative pose problem," *(TPAMI)*, vol. 26, no. 6, pp. 756–770, 2004.

[30] B. Haralick, C. Lee, K. Ottenberg, and M. Nolle, "Review and analysis of solutions of the three point perspective pose estimation problem," *(IJCV)*, vol. 13, no. 3, pp. 331–356, 1994.

[31] [Online]. Available: http://vision.middlebury.edu/flow/

[32] D. P. H. Pedro F. Felzenszwalb, "Efficient graph-based image segmentation," *(IJCV)*, 2004.

[33] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *ECCV (1)*, 2008, pp. 44–57.

[34] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognition Letters*, vol. xx, no. x, pp. xx–xx, 2008.