Motion Segmentation by Learning Homography Matrices from Motor Signals

Changhai Xu
Department of Computer Science
University of Texas at Austin
1 University Station, Austin, TX 78712
changhai@cs.utexas.edu

Jingen Liu and Benjamin Kuipers
Computer Science and Engineering
University of Michigan
2260 Hayward Street, Ann Arbor, MI 48109
{liujg,kuipers}@umich.edu

***Abstract*—Motion information is an important cue for a robot to separate foreground moving objects from the static background world. Based on the observation that the motion of the background (from the robot's egocentric view) has stronger correlation to the robot's motor signals than the motion of foreground objects, we propose a novel method to detect foreground moving objects by clustering image features according to their motion consistency with motor signals.**

Corner/edge features are detected and tracked across adjacent frames. The errors between the estimated feature locations based on motor signals and their actual tracked locations are calculated. The features are clustered into background/foreground using Expectation-Maximization on these errors. Labeled features are then used for pixel-level image segmentation with the Active Contours and Graph-based Transduction techniques.

Unlike pixel-level background subtraction methods, the proposed approach does not require a large number of frames for background model construction, and does not suffer from accumulated image registration error for dynamic cameras. In contrast to existing sparse feature based foreground/background separation methods, our approach clusters features in only one dimensional space instead of a higher dimensional space, and there is no need to search for parameters in an affine or homography transformation space or motion trajectory space.

***Keywords*-motion segmentation; motor signals; background subtraction; moving object detection.**

I. Introduction

In order for an intelligent robot to understand and interact with its environment (such as manipulating an object), it must be capable of learning high-level object models. We are developing the Object Semantic Hierarchy (OSH) [26] which is a hierarchical computational model of the background world and the foreground objects, consisting of multi-layer representations. To construct the object models in the OSH, a fundamental problem is to separate out moving objects in the pixel-level sensory input from the static background world.

Motion segmentation based on background subtraction has been widely used for foreground segmentation. Statistical pixel-level background models such as Pfinder [25], Gaussian Mixture Model (GMM) [22], and their variations [14], [10], [16] have achieved many successful applications in areas such as traffic monitoring and visual surveillance. These methods usually require a large number of frames to build a stable background model. In the dynamic camera case, the robot may not have enough frames to build such pixel-level background models before the visual field changes significantly. In other words, the robot has to detect moving objects based on only very few frames. In addition, when a robot is manipulating an object, the object may take up a large portion of its field of view (as shown in row 5 of Fig. 4). As a result, pixel-level background models can completely fail because the most frequently observed intensity/color values may come from foreground pixels instead of background pixels, and sometimes a part of the real background may not be observed at all due to permanent occlusion by moving objects. Another issue with pixel-level background models is that for dynamic cameras the model can be very noisy around edges due to accumulated image registration error.

(a) (b)

Figure 1. The hardware used in our experiments where a webcam is mounted on a pan-tilt unit (PTU).

Tracking features and then clustering them into foreground/background is another classic way for motion segmentation. In general, a set of affine/homography parameters [17], [8], [21] or trajectory parameters [18] need to be estimated by iterative linear regression [8] or RANSAC [18], [17]. Our method is also based on feature clustering, however, it is significantly different from these existing works. Our system takes advantage of motor signals besides visual information for foreground segmentation. Without using motor signals, it would be impossible to distinguish background and foreground when the foreground objects take up more than half of the visual field if no prior knowledge is available. More importantly, the motion segmentation

978-0-7695-4362-8/11 $26.00 © 2011 IEEE
DOI 10.1109/CRV.2011.49

process in [8], [21], [18] is carried out in a two or higher dimensional space; by taking advantage of motor signals, we will show that the proposed method can cluster tracked features in a one dimensional space. In addition, compared to the work in [18] where *dense features* are tracked in a 30-frame window, our approach can separate foreground objects by tracking *sparse features* for a very small number (around 5) of frames, which allows the robot to quickly adapt to an environment.

Based on our observation that the background motion (from the robot's egocentric view) has stronger correlations to the robot's motor signals than the motion of foreground moving objects, we propose a system to detect moving objects by measuring the consistency between visual change and motor signal change. Fig. 1 shows our experimental setup, which is a webcam mounted on a pan-tilt unit (PTU). Note that this setup is different from a general pan-tilt camera in that the optical center of the webcam will have translations when the PTU moves beyond the horizontal plane. The input to the system includes both images and corresponding pan/tilt positions.

Our method segments foreground moving objects from the static background based on only a small number of adjacent frames. Across these frames, the robot in general has small translations (but possibly large rotations) with respect to its environment. In this case, the visual change of the static background can be well approximated by homography transformations [9]. The relation between motor signal change and visual change is automatically learned as a mapping function from motor signal changes to these homography transformations, in a one-time manner in our system.

With the learned relation between motor signal change and visual change, the robot can precisely predict the motion patterns of background features among adjacent frames. In contrast, the actual motion patterns of foreground (i.e., moving objects) features will be obviously different from the predictions because they have independent motions from the robot. As a result, we can cluster image features into background and foreground by measuring the discrepancy between their actual motions and predictions.

In the ideal case, we can simply cluster the features with zero discrepancy as background and others as foreground, because the discrepancy between the predicted background feature locations and their actual tracked locations will be zero. However, due to asynchronous sensor/motor readings and noisy feature tracking results, it is infeasible to pre-select a threshold to separate the features. Additionally, the systematic errors may be different under different camera poses, which makes it impossible to find a single threshold to separate the foreground and background features. Therefore, in our system we automatically determine the threshold on-line by fitting Gaussian mixture models using the Expectation-Maximization algorithm, and cluster tracked sparse features into background and foreground.

After the sparse features are labeled, we apply the Active Contours (AC) method [12], [15] to segment the dense foreground. The AC method runs in real-time, but it is not robust to noisy labels and may not have good segmentation on boundaries. We then apply the Graph-based Transduction (GBT) method to smoothly transfer the labels from sparse features to the unlabeled pixels [13], [5]. In other words, by treating the labeled sparse features as training samples and the unlabeled pixels as test data, the GBT method classifies the unlabeled pixels into background and foreground in a semi-supervised manner. One advantage of using GBT is that it can transfer pixel labels by exploring the distribution of pixel features in an image such that the labeling is robust to noisy initial labels. The GBT learning has been widely explored in the machine learning and image segmentation areas [3], [28], [5]. In our work, we use the Spectral Graph Transducer [13], [19] as our transductive classifier.

To summarize, the major contributions of our paper are: (i) To the best of our knowledge, we are the first to use motor signals for motion segmentation by clustering feature prediction errors in one dimensional space. (ii) Expectation-Maximization algorithm is applied to automatically select the error threshold to cluster sparse features into background and foreground. (iii) Graph-based Transduction is used to transfer foreground/background labels from sparse features to dense pixels.

II. Related Work

Background subtraction is a popular technique for detection of moving objects in image sequences and many works [25], [22], [14], [6] have been published in this area. In comparison to these works where pixel-level models are used, we use sparse corner/edge features and these features are more robust to lighting variations than ordinary pixels.

An edge-based method was proposed by Hossain *et al.* [11] where connected edge pixels are traced to form edge segments for background modeling. Yokoyama and Poggio [27] presented a contour-based background model for moving object detection. Contours are approximated as line segments and background line segments are subtracted in the new input image to separate out foreground objects. The authors did not show how to handle non-linear contour segments. In our method, we also consider contour fragment features, which can be either straight or curved.

To deal with pan-tilt camera images, a background model can be constructed by image mosaicing. Distinctive local features can be used to calculate the geometric transformations between images and stitch these images to get a panoramic view [2], [23], [1], [24]. This method requires a highly-textured background, and may result in blurred edges due to accumulated image registration error. In contrast, our motion segmentation method does not need to maintain a background model across a long sequence of images, and hence does not accumulate registration error.

Criminisi *et al.* [4] fused motion and color cues with temporal and spatial priors in a probabilistic framework to achieve real-time foreground detection. This method deals with only small camera motions, and needs different hand-labeled segmentations for different environments. Furthermore, the weights of different cues need to be hand-tuned.

III. Relation Learning from Motor Signal Change to Visual Change

Our system takes every few adjacent frames as input and separates the foreground objects from the background. Across this small number of frames, the robot's translation is very small and the position changes of the background features can be well approximated by homography transformations [9].

We will describe our motion segmentation approach based on the experimental setup shown in Fig. 1, where the motor signals are two dimensional pan and tilt positions. Although this experimental setup is specific, the reasoning of this paper is very general and can be easily extended to other setups, for example, with higher dimensional motor signals.

Let $u = \{u_1, u_2\}$ be the motor signal change, where u_1 and u_2 are the pan and tilt position changes respectively. Let H be the visual change that corresponds to the motor signal change u, where H is a 3x3 homography matrix. Our goal here is to learn the relation f between the motor signal change u and the visual change H.

The relation f can be constructed by hand for an ideal pan-tilt camera. But for the webcam in Fig. 1, manual construction of f can be complicated. This is because the optical center of the webcam will change while the pan-tilt unit moves beyond the standard horizontal plane (the standard horizontal plane is the pan plane where there is zero tilt). In addition, if the camera's principle axis is not exactly parallel to the standard horizontal plane, manual construction will include large systematic errors. Another issue with manual construction of f is that it will have to estimate intrinsic camera parameters from camera calibration.

To avoid camera calibration, systematic errors, and complexity of manual construction, we learn the function f automatically in our system. Due to the observation that f is an invariant and hence independent of the environment, we learn it in an environment which has good textures in order for different frames of images to be well registered and has no objects moving in it. Note that this learning process needs to be taken for only one time and the learned relation will remain the same in any other environment. The robot collects a set of images plus the corresponding motor signals in this environment. It then estimates the homography transformation H based on matched image features and learns a mapping function f from motor signal change u to visual change H without human intervention.

The homography transformation H is calculated based on tracked local point features using the KLT method [20]. Since H has 8 degrees of freedom, it can only be obtained up to a scale factor (from four or more pairs of corresponding points) [9]. So we need to normalize H in order to learn a continuous function between u and H. Across a small number of frames, the robot translation is zero or very small, and the homography matrix is equivalent or close to a pure rotation matrix multiplied by a scale factor. In addition, the camera pan and tilt will not have extremely large changes across a few frames, and the last element of the rotation matrix will be guaranteed to be non-zero. Thus we normalize H such that its last element is always 1.

Each mapping relation from the motor signal change to an element in the homography matrix is fitted as a polynomial (a non-linear relation). In our experiments, the fitting error becomes very small when the degree of the polynomials grows to three. Let V^H denote the stacked 8-dimensional vector of H. The 10-dimensional motor signal vector V^u is defined as

$$V_t^u = [u_1^3, u_2^3, u_1^2 u_2, u_1 u_2^2, u_1^2, u_2^2, u_1 u_2, u_1, u_2, 1]^T.$$

For each two frames that are captured within a small number of time steps, we obtain a pair of V^u and V^H. Suppose we have a number of n pairs of these vectors, denoted by $\{V_k^u, V_k^H\}$ $(k = 1, ..., n)$. We stack all V_k^u as rows in a $n \times 10$ matrix A^u, and stack all V_k^H as rows in a $n \times 8$ matrix B^H. Then the relation function f between motor signal changes and homography matrices is learned as third order bivariate polynomials from the following equation,

$$A^u f = B^H \tag{1}$$

where f is a 10×8 matrix. Given any motor signal change, we are able to get the predicted visual change from f.

After f is learned in one environment, it can be used in any other arbitrary environment without re-learning.

IV. Sparse Background/Foreground Feature Classification

For a reference frame I_t, we detect two types of features: corners and edges. Corner features are detected by the KLT method, and edge features are extracted by the Canny detector. The locations of the corners and sampled edge points form our sparse feature set P_t. These sparse features are tracked in I_t's neighboring frames I_{t+k} $(k = \{-M, ..., -1, 1, ..., M\})$. The tracked feature set in frame I_{t+k} are denoted as P_{t+k}.

Given the motor signals at two frames t and $t + k$, the homography matrix between the two frames is calculated from

$$V_k^H = V_k^u f \tag{2}$$

where V_k^H can be unstacked to get the homography matrix H_k.

From frame I_t to I_{t+k}, the background features should be consistent with the transformation H_k, while the foreground features will violate this transformation. Thus we can classify the features based on the errors between the actual tracked feature locations and their estimated locations predicted from H_k. For each feature P_i tracked from I_t to I_{t+k}, the error is defined as

$$d_{i,t+k} = \min(\eta_d, \|\hat{P}_{i,t+k} - P_{i,t+k}\|). \quad (3)$$

where η_d is a bounding constant to avoid large errors from incorrectly tracked features, $P_{i,t+k}$ is P_i's tracked location in frame $t+k$, $\hat{P}_{i,t+k} \propto H_k P_{i,t}$ is P_i's predicted location in frame $t+k$, and the last element of $\hat{P}_{i,t+k}$ is normalized to 1 (such that the error is measured directly in the image space).

We then cluster the tracked features based on the error set $\{d_{i,t+k}\}$ $(i = 1, 2, ..., N_p)$. Note that this clustering process is taken in only one dimensional space. To avoid distractions from incorrectly tracked features, we assign $\{d_{i,t+k}\}$ a maximum limit η_d (10 pixels in our experiments). Due to asynchronous sensor/motor readings, inaccurate parameter estimation in f, and noisy feature tracking results, it is difficult to pre-determine a threshold to divide $\{d_{i,t+k}\}$ into two groups. We use the EM algorithm to fit a two-component Gaussian mixture model (corresponding to background/foreground) on $\{d_{i,t+k}\}$. The model is described by

$$G_{t+k}(d) = \sum_{j=\{bg,fg\}} w^j_{t+k} g(d; \mu^j_{t+k}, \sigma^j_{t+k}) \quad (4)$$

where $g(.)$ is the normal distribution, and $w^{bg}_{t+k} + w^{fg}_{t+k} = 1$. Here the superscripts bg and fg correspond to background and foreground respectively. At each frame t, the two Gaussian components are initialized with the Gaussians estimated in frame $t-1$. We take the gaussian component with the smaller mean as the background distribution.

Those features with a high average of likelihood from Eq. 4 across frames I_{t+k} $(k = \{-M, ..., -1, 1, ..., M\})$ are classified as background features and others as foreground features in frame I_t. In our experiments, we set $M = 3$.

V. Dense Foreground Segmentation

Given frame I_t, let the set of classified sparse features be $P = \{(x_1, l_1), \ldots, (x_{N_p}, l_{N_p})\}$ (N_p is the number of labeled features), where $x_i \in R^5$ is a pixel feature vector (consisting of HSV color and 2D location) and $l_i \in \{+1, -1\}$ is the foreground/background label. Our goal is to classify the remaining unlabeled pixels $U = \{x_{N_p+1}, \ldots, x_{N_p+N_u}\}$ into background/foreground, where N_u is the number of unlabeled pixels. We apply two approaches to achieve this goal: Active Contours (AC) and Graph-based Transduction (GBT).

Foreground Segmentation by the AC method: Given sparse foreground features $P^{fg} \subset P$, we first cluster them into groups based on their 2D location distance. Any cluster approaches can be used. In this paper, we adopt the k-means algorithm. Clusters with a small number of features are identified as outliers and are filtered out. We then find the convex hull for each cluster, initialize an AC model with the convex hull, and fit it to image edges. The AC model uses piecewise splines to represent objects, and fits the splines to object boundaries by minimizing a sum of two energy terms: Internal Energy and External Energy. The Internal Energy accounts for boundary smoothness, and the External Energy evolves the model to fit with observed image edges (Please see [12], [15] for details). This method is very efficient in computation and works well for many applications. However, since our AC model is initialized with a convex hull, it may never converge to the real object boundaries when the object shapes are non-convex. In addition, the weights for the energy terms in the AC model are hard to be tuned. As a result, small non-boundary edges around the object boundaries can cause serious distractions. Hence, we further propose using the GBT method to classify the unlabeled pixels.

Foreground Segmentation by the GBT method: By treating labeled pixels P as training data and unlabeled pixels U as test data, we formulate the foreground segmentation problem as a binary classification problem via transductive learning. We aim at finding a transductive classifier $f(x) \in \{+1, -1\}$ in the feature space to classify the test data. The basic idea of transductive learning is to train a classifier, which has not only small training error on the training data but also highly consistent outputs to the distribution of the test data. This works well for the situation with a small number of training examples and a large amount of test data. Our foreground segmentation problem has a good fit with this situation. In our work, we choose graph as a tool to analyze the data distribution structure.

Let us define a graph G with P and U as vertices and adjacent weight matrix W. Each entry $w(x_i, x_j)$ of W is defined by $\mathcal{K}(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$, where $\mathcal{K}(x_i, x_j)$ is a symmetric function. We seek a function $f(x)$ that projects the graph vertices onto $\{+1, -1\}$ such that we have low training error on P and precise label assignments (clustering) on $P + U$. In other words, we integrate graph clustering and classification targets together. The objective function is,

$$\min_{\mathbf{f}} \mathbf{f}^T L \mathbf{f} + \lambda(\mathbf{f} - \mathbf{b})^T C(\mathbf{f} - \mathbf{b}), \quad (5)$$
$$\text{subject to} \quad \mathbf{f}^T \mathbf{1} = 0 \text{ and } \mathbf{f}^T \mathbf{f} = n$$

where n is the pixel number of an image, $\mathbf{b} \in R^n$ with each dimension $\mathbf{b}(i) = \sqrt[2]{(n_-/n_+)}$ for positive labeled data and $\mathbf{b}(i) = -\sqrt[2]{(n_+/n_-)}$ for negative data (n_+ and n_- are the numbers of positive and negative labeled data), Laplacian

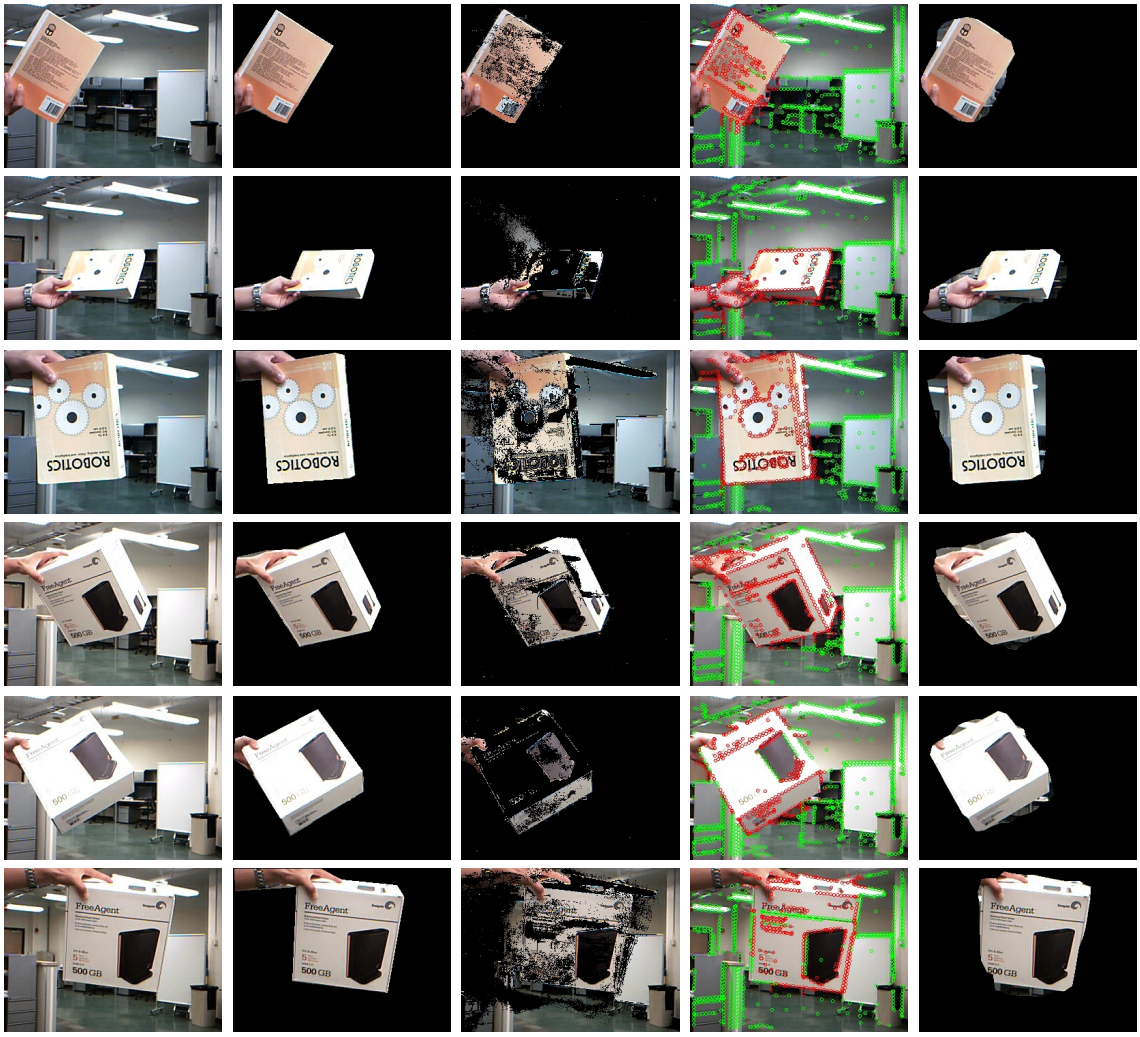

Figure 2. Detection results for static camera case (best viewed in pdf). The columns show the original images, ground truth, detection results by GMM, classified sparse features (green and red represent background and foreground respectively) by MSMS, and segmentation results by MSMS-AC. The GMM method tends to produce black holes on the foreground objects and spreading misclassified pixels over the whole image when illumination condition changes. In contrast, the MSMS method gives better results in these situations.

matrix $L = D - W$ with $D_{ii} = \sum_{x_j} w(x_i, x_j)$, and C is a diagonal matrix assigning penalty to any misclassification of the training examples. The first term measures the discontinuity of the graph bi-partition and the second term computes the training errors on the labeled data. The parameter λ controls the tradeoff between training error and clustering quality. We adopt the Spectral Graph Transducer [13], [19] as our transductive classifier.

VI. Evaluation

The relation between motor signal change and visual change is learned from different sensor positions. In the 2D motor signal space $\{u_1, u_2\}$, we draw 32 evenly distributed rays shooting out from the point $(u_1, u_2) = (0, 0)$. On each ray, we select 16 evenly spaced points including the point $(0, 0)$. Thus we have $32 \times 15 + 1$ different points, and at each such point we collect an image. The transformations between each two significantly overlapped images are obtained by tracked KLT features. We use the motor signal changes and the calculated image transformations to learn f in Eq. 1. This learning process is performed once and no user intervention is needed.

To evaluate the proposed motor signal based motion segmentation (MSMS) method, we collected five datasets. Two of them were used to evaluate the static camera case, and the remaining three were used for the dynamic camera case. For each dataset, we manually labeled the foreground objects every five frames as the ground truth. We tracked features in its 6 neighboring frames (i.e., $M = 3$). For quantitative evaluation, the detection accuracy is defined as A_I/A_U, where A_I and A_U are the areas of the intersection

and union between the detected foreground and the ground truth respectively.

A. Static Camera Case

First we compare the performance of MSMS with GMM (Gaussian Mixture Model) when the camera is static. We use the GMM implementation [14] in OpenCV. Since the camera is static, there is no motor signal change and the homography transformation H remains constant as an identity matrix. Thus the step of learning f is excluded in the system. We apply only the sparse feature classification and dense pixel segmentation steps, and test the system on a "book" dataset and a "hard-drive box" dataset. Both datasets have significant illumination changes because the webcam's light auto-adjust function is enabled. The light auto-adjust function usually takes effect when the object moves from very close to the camera to far away, or vice versa.

Fig. 2 illustrates some visual results for qualitative comparison on the "book" and "hard-drive box" datasets, and Fig. 3 (a) shows the average foreground detection accuracy. It clearly shows that MSMS is superior to GMM (with more than 20% improvement). This is because the GMM method often produces a large number of noisy pixels spreading over the whole image when the illumination changes due to objects moving close to or far away from the camera and reflections on the objects. Moreover, GMM needs sufficient frames to learn a stable pixel-level background model. In contrast, the proposed MSMS method is more robust to illumination changes, and more importantly it only needs a few frames to detect foreground and background sparse features.

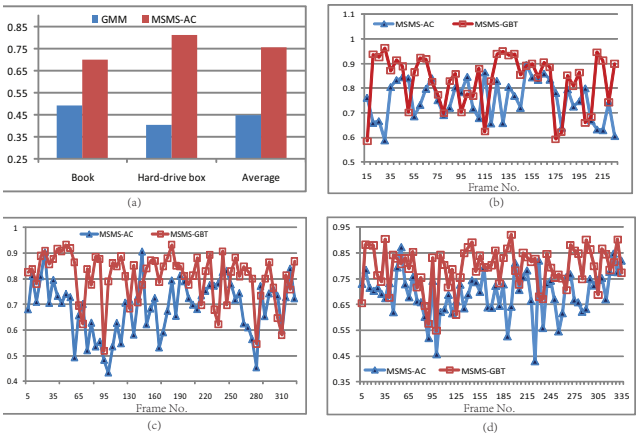

Figure 3. Quantitative evaluation results. (a) shows the average detection accuracy in static camera case. (b)-(d) show the detection accuracy for the "tea-box","football", and "toy-pig" datasets in dynamic camera case. MSMS-AC and MSMS-GBT represent the results for motor signal based motion segmentation with AC and GBT.

Table I
DETECTION ACCURACY COMPARISON FOR DYNAMIC CAMERA CASE

Accuracy	tea box	football	toy pig	average
RANSAC-AC	63.6	61.1	64.5	63.1
RANSAC-GBT	69.7	65.5	68.1	67.8
MSMS-AC	75.3	68.0	69.3	70.9
MSMS-GBT	82.9	79.5	79.5	80.6

B. Dynamic Camera Case

To further demonstrate the performance of the MSMS method, we conduct experiments on three datasets ("tea-box", "football", and "toy-pig") captured from a dynamic camera. The camera has significant viewpoint change, and the foreground objects have large translation, rotation, and scale change. For the sake of computational efficiency in the GBT method, we over segment an image into super-pixels (about 1,200 super-pixels for each image on average) using the image segmentation method proposed in [7], and neighboring super-pixels are connected to build a graph.

Fig. 4 shows some typical images of the detected objects for qualitative evaluation. The AC method is simple and fast, but it may miss boundary details. For example, as shown in column (f), the AC method fails to segment part of the hand because there are few or no detected features on it. This is because its performance relies on the quality of sparse feature detection (column (e)). Moreover, in row 3 the AC method extracts extra regions from the background because its initialized shape by a convex hull is significantly different from the real object boundary. In order to preserve more details on the boundaries, we further apply the GBT method in our system. From column (g), we can see the GBT method segments the moving objects with better boundaries, since it makes use of the distribution of pixel features for segmentation.

We conduct a set of baseline experiments using RANSAC which directly fits a homography transformation to the background features. The sparse features that do not fit the homography are classified as foreground. Then we use the AC and GBT methods with the same setting to segment the foreground. The results are shown in Fig. 4 column (c) (classified sparse feature) and column (d) (segmentation results). Comparing the classified sparse features obtained by RANSAC (column (c)) and MSMS (column (e)), we can see the RANSAC method misclassifies many background features that are close to the moving object into foreground. As a result, many background regions are segmented as foreground (as shown in column (d)).

Table I illustrates the quantitative results for both the RANSAC and MSMS methods with the AC and GBT foreground segmentation. On average, the MSMS-GBT improves the performance about 13% over RANSAC-GBT, and about 10% over MSMS-AC.

Fig. 5 shows some detection results for a non-rigid

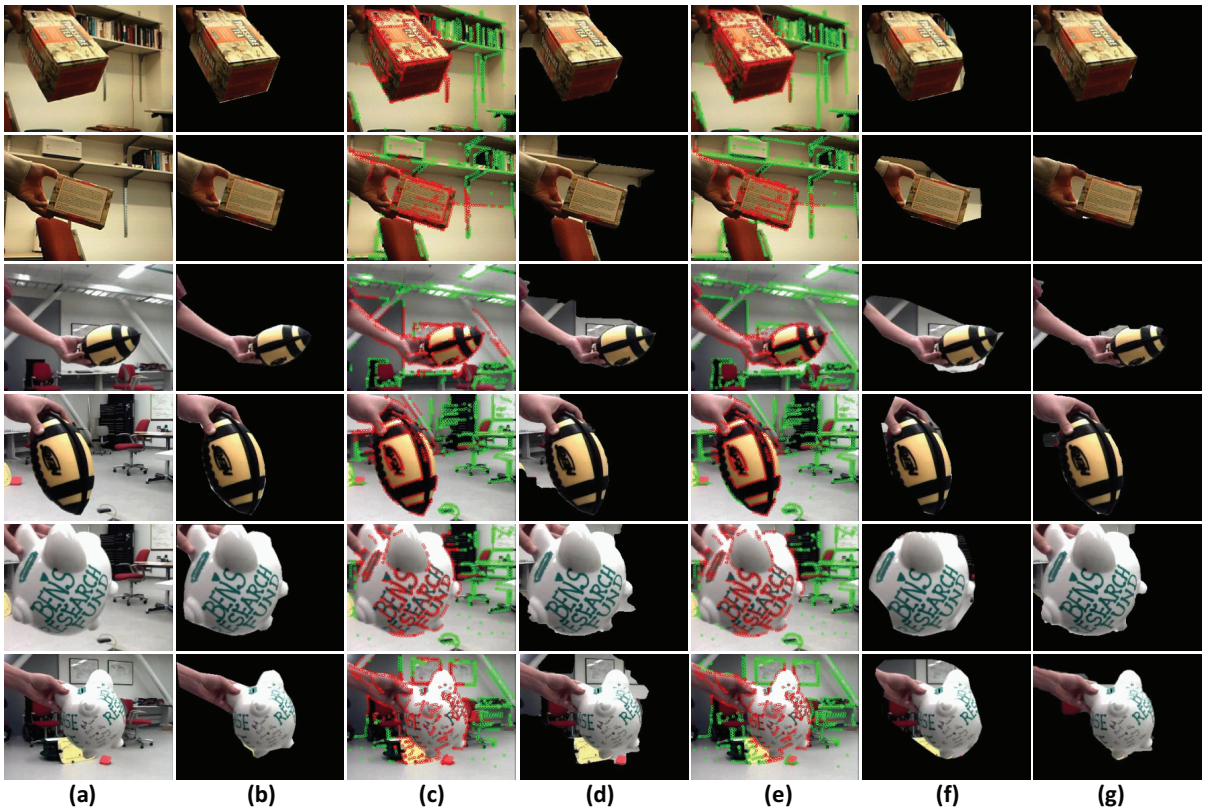

Figure 4. Detection results for dynamic camera case (best viewed in pdf). Collums (a)-(g) show the original images, ground truth, classified sparse features by RANSAC (background shown as green and foreground red), segmentation results by RANSAC-GBT, classified sparse features by MSMS, segmentation results by MSMS-AC, and segmentation results by MSMS-GBT, respectively.

Figure 5. Detection results by MSMS-GBT for a walking person (dynamic camera case).

object: a walking person. Our motion segmentation approach performs well in this video, where the foreground object has large depth and shape change.

VII. Conclusion

We have presented a motion segmentation approach using motor signals. The approach clusters sparse image features into background and foreground in only one dimensional space. It is robust to illumination changes, adapts fast to the environment, and does not suffer from accumulated image registration error.

Currently in our system we use temporal information only for a few frames to help foreground object detection. We will investigate how longer temporal information can be integrated into the system. We will also set up an experimental environment for a mobile robot and evaluate the performance of the proposed approach. A combination of the RANSAC and MSMS methods will be investigated to improve segmentation accuracy.

Acknowledgment. This work has taken place in the Intelligent Robotics Labs at the University of Texas at Austin

and at the University of Michigan. Research in the Intelligent Robotics Labs is supported in part by grants from the National Science Foundation (IIS-0713150 to UT Austin and CPS-0931474 to UM) and from the TEMA-Toyota Technical Center to UM.

REFERENCES

[1] A. Bevilacqua, L. Di Stefano, and P. Azzari, "An effective real-time mosaicing algorithm apt to detect motion through background subtraction using a PTZ camera," in *IEEE Conference on Advanced Video and Signal Based Surveillance*, 2005, pp. 511–516.

[2] M. Brown and D. Lowe, "Recognising panoramas," in *Proceedings of the Ninth IEEE International Conference on Computer Vision*, 2003, p. 1218.

[3] O. Chapelle, B. Scholkopf, and A. Zien, *Semi-supervised learning*. The MIT Press, 2006.

[4] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov, "Bilayer Segmentation of Live Video," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2006, p. 60.

[5] O. Duchenne, J. Audibert, R. Keriven, J. Ponce, and F. Segonne, "Segmentation by transduction," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[6] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," *European Conference on Computer Vision*, pp. 751–767, 2000.

[7] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[8] M. Han, W. Xu, and Y. Gong, "Video object segmentation by motion-based sequential feature clustering," in *Proceedings of the 14th annual ACM international conference on Multimedia*, 2006, p. 782.

[9] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[10] E. Hayman and J. Eklundh, "Statistical background subtraction for a mobile observer," in *Proceedings of the Ninth IEEE International Conference on Computer Vision*, vol. 1, 2003, pp. 67–74.

[11] M. Hossain, M. Dewan, and O. Chae, "Moving object detection for real time video surveillance: an edge based approach," *IEICE Transactions on Communications*, vol. 90, no. 12, pp. 3654–3664, 2007.

[12] M. Isard and A. Blake, "CONDENSATION - conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[13] T. Joachims, "Transductive learning via spectral graph partitioning," in *International Conference on Machine Learning*, vol. 20, no. 1, 2003, p. 290.

[14] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *European Workshop on Advanced Video-based Surveillance Systems*, vol. 1, no. 3, 2001.

[15] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International journal of computer vision*, vol. 1, no. 4, pp. 321–331, 1988.

[16] S. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler, "Tracking groups of people," *Computer Vision and Image Understanding*, vol. 80, no. 1, pp. 42–56, 2000.

[17] X. Ren and C. Gu, "Figure-ground segmentation improves handled object recognition in egocentric video," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3137–3144.

[18] Y. Sheikh, O. Javed, and T. Kanade, "Background Subtraction for Freely Moving Cameras," in *International Conference on Computer Vision*, 2009.

[19] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[20] J. Shi and C. Tomasi, "Good features to track," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600, 1994.

[21] J. Sivic, F. Schaffalitzky, and A. Zisserman, "Object level grouping for video shots," *International Journal of Computer Vision*, vol. 67, no. 2, pp. 189–210, 2006.

[22] C. Stauffer and W. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, 2000.

[23] R. Szeliski, "Image alignment and stitching: A tutorial," *Foundations and Trends in Computer Graphics and Vision*, vol. 2, p. 104, 2006.

[24] P. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.

[25] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-Time Tracking of the Human Body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, 1997.

[26] C. Xu and B. Kuipers, "Towards the object semantic hierarchy," *International Conference On Development and Learning*, pp. 39–45, 2010.

[27] M. Yokoyama and T. Poggio, "A contour-based moving object detection and tracking," in *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005, pp. 271–276.

[28] X. Zhu, "Semi-supervised learning literature survey," *Computer Science, University of Wisconsin-Madison*, 2006.