Motion based Segmentation for Robot Vision using Adapted EM Algorithm

Wei Zhao and Nico Roos
Department of Knowledge Engineering, Maastricht University, Maastricht, The Netherlands

Keywords: Optical Flow, SIFT Matching, Clustering, Motion Segmentation.

Abstract: Robots operate in a dynamic world in which objects are often moving. The movement of objects may help the robot to segment the objects from the background. The result of the segmentation can subsequently be used to identify the objects. This paper investigates the possibility of segmenting objects of interest from the background for the purpose of identification based on motion. It focusses on two approaches to represent the movements: one based on optical flow estimation and the other based on the SIFT-features. The segmentation is based on the expectation-maximization algorithm. A support vector machine, which classifies the segmented objects, is used to evaluate the result of the segmentation.

1 INTRODUCTION

Studies of visual perception show that human vision is based on seeing changes (Martinez-Conde et al., 2004). In the domain of robot vision, seeing changes also crucial because of the environments are mostly dynamic: robots operate in a dynamically changing world and they may have the capability to move around. We will investigate applicability of detecting changes in robot vision in this paper.

Analysing the changes among the frames can give us a clue of objects in the video (Karasulu and Korukoglu, 2013; Zappella et al., 2008). This is called the moving object detection, which is differs from the objects detection in single image. Detecting an object in single image requires knowledge about object's expressions.

In this paper, a general system is investigated to detect and recognize the objects by their movements. We assume that one object consists of a group of points, and points belong to the same object will have the same movement. The system consists of three main steps. Firstly, we detect all points and their movements in the video sequences, where two methods are investigated. One uses optical flow to estimate the motions of all pixels of an image. The other uses higher level features, e.g. the scale-invariant feature transform (or SIFT) points. Secondly, the points are segmented into different groups based on their movements and scales. These groups of points are possible objects. Segmentation of these points based on their movement is fulfilled by combining the EM algorithm with a divisive hierarchical approach. Finally, a *support vector machine* (SVM) (Boser et al., 1992) is used to evaluate the whether the segmentation results can be recognized accurately as an object.

In the next section, we will briefly review some related work. Section 3, provides some background information about the algorithms that we have applied. Section 4 outlines our approach. Experiments that we used to evaluate our approach are presented in Section 5. Section 6 concludes the paper.

2 RELATED WORK

Detecting objects from images is multi-purpose tasks, where many techniques, such as images segmentation, image processing, machine learning, linear algebra, statistic, etc., are involved in.

Much research has already been done in the area of image segmentation. A high level division of the available techniques are: detecting discontinuities and detecting similarities (Narkhede, 2013). The first category uses edge detection to identify regional boundaries (Narkhede, 2013). The second category consists of techniques such as: thresholding, clustering, motion segmentation and color segmentation (Seerha and Rajneet, 2013; Narkhede, 2013).

In this paper we focus on motion segmentation. Motion segmentation using optical flow and k-means

Zhao, W. and Roos, N.
Motion based Segmentation for Robot Vision using Adapted EM Algorithm.
DOI: 10.5220/0005721606490656
In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2016) - Volume 3: VISAPP*, pages 651-658
ISBN: 978-989-758-175-5

clustering has been proposed by (Wang and Adelson, 1994). Borshukov et al. (Borshukov et al., 1997) improved this method by replacing the k-means clustering with a multistage merging step clustering. Optical flow estimation in combination with the EM algorithm for the purpose of image stabilization has been proposed by (Pan and Ngo, 2005).

(Shi and Malik, 1998) proposed a motion segmentation algorithm that constructs a weighted spatio-temporal graph on image sequence and using normal cuts to find the most salient partitions of the spatio-temporal graph. (Weiss, 1997) presented an algorithm that segments the image sequences by fitting the multiple smooth flow fields to the spatio-temporal data using a variant of the EM algorithm.

We make use of the basic principles of the mentioned approaches: detecting the motion fields and segmenting them into clusters using EM algorithm. But our work differs from previous work in several ways. First, the objects need not to be cut perfectly, just sufficiently consistent to enable object identification. Secondly, the descriptions of clusters are simple and improved gradually.

3 PRELIMINARIES

The approach proposed in this paper makes use of optical flow estimation (OFE), scale-invariant feature transform (SIFT), the expectation-maximization (EM) algorithm and the support vector machine (SVM). In this section we briefly review each of these approaches.

3.1 Optical Flow Estimation

Optical flow estimation is defined as the distribution of apparent velocities of movement of brightness patterns in an image (Horn and Schunck, 1981). The optical flow Estimation is based on the assumption that the intensity of a pixel corresponding with a point on an object, does not change when the object or the camera is moving. Suppose the location of a point is (x,y) at time t and $(x+\Delta x, y+\Delta y)$ at time $t+\Delta t$. Let $I(x,y,t)$ be the intensity of a pixel w.r.t position (x,y) and time t. Based on the assumption of brightness constancy, i.e.,

$$I(x+\Delta x, y+\Delta y, t+\Delta t) = I(x,y,t) \tag{1}$$

Expanding the equation with first-order Taylor series, and using the notation $(u,v) = (\frac{dx}{dt}, \frac{dy}{dt})$, we get:

$$\nabla I(x,y,t) \cdot (u,v,1)^T = 0 \tag{2}$$

Lucas and Kanade(Lucas et al., 1981) proposed an additional assumption that the that neighboring pixels often have the same movement. Given the set of neighbouring points, the optical flow of centroid points of the neighbourhood is able to estimated by solving the optimized problem of Equation 2 over these neighbouring points.

3.2 Scale-Invariant Feature Transform

SIFT is an algorithm to detect and describe local features in images, which was proposed by David Lowe (Lowe, 1999). Unlike the optical flow, SIFT is not a technique for detecting changes. SIFT feature descriptors are some keypoints extracted from a set of reference images. They are invariant to image scaling and rotation, and partially invariant to affine distortion, noise and illumination changes. Because of the scale-invariant properties and the high level feature expression, the movement of segments in the image can be estimated by matching the keypoints between two successive images.

3.3 Expectation-Maximization Algorithm

Both OFE and SIFT can provide motion vectors of point in an image. We assume that points belonging to one object have motion vectors that can be described by an affine transformation. To extract the object from the background, clustering methods are needed to cluster points of these objects. It is also a segmentation task, which aims to segment the image into objects and background based on the motion features of pixels or points. The Expectation-Maximization (EM) algorithm is one of the approaches enables us to do this.

The EM algorithm (Dempster et al., 1977) is an effective and popular technique for estimating parameters of a distribution from a given data set. It aims at determining the most likely values of parameters θ using observed data $\mathbf{x}$ and some hidden variables $\mathbf{Z}$. That is, the most likely parameters θ maximizes the expected expected value of the likelihood $L(\theta;\mathbf{x},\mathbf{Z}) = P(\mathbf{x},\mathbf{Z} \mid \theta)$ over all hidden variables $\mathbf{Z}$, so,

$$\begin{aligned} &\theta = \text{argmax}_{\theta'} L(\theta';\mathbf{x}) \\ &\text{where: } L(\theta;\mathbf{x}) = P(\mathbf{x} \mid \theta) = \sum_{\mathbf{Z}} P(\mathbf{x},\mathbf{Z} \mid \theta) \end{aligned} \tag{3}$$

Equation 3 is a fixed point equation. Given the parameters θ and $\mathbf{x}$ we can determine the probability distribution of the hidden variables $\mathbf{Z}$, and subsequently we can find a maximum likelihood estimate of the parameters θ. The former is called the expectation step

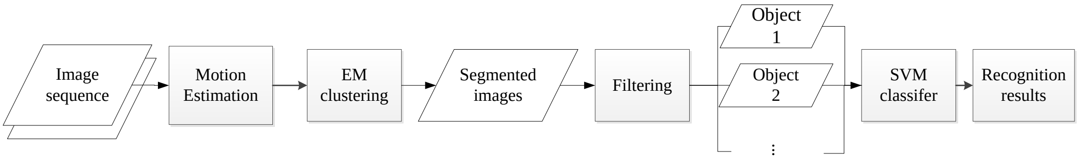


Figure 1: Basic architecture of the detection and recognition approach.

while the latter is called the maximization step. Starting from an initial estimate of θ or $P(\mathbf{x},\mathbf{Z} \mid \theta)$ and repeatedly applying the expectation and maximization step, the EM algorithm will converge to the maximum likelihood parameters θ.

Instead of the probability distribution $P(\mathbf{x},\mathbf{Z} \mid \theta)$ we may determine:

$$\mathbf{z} = \underset{\mathbf{Z}}{\operatorname{argmax}} P(\mathbf{x},\mathbf{Z} \mid \theta) \tag{4}$$

in the expectation step. In the maximization step we determine:

$$\theta = \underset{\theta'}{\operatorname{argmax}} L(\theta';\mathbf{x},\mathbf{z}) \tag{5}$$

4 MOTION-BASED VISION

Although this paper focusses on segmentation, the final goal is the identification of objects a robot is seeing in the world. The results of the segmentation should be evaluated with respect to this goal. Therefore we will present the whole architecture (Figure 1) of the vision system, including the classification of observed objects.

4.1 Optical Flow based Motion Detection

The optical flow is calculated by using the iterative Lucas-Kanade method with pyramids in this system (Bouguet, 2001). The pyramid optical flow estimation allows a high accuracy when the displacements are not too large. However, optical flow estimation often fails to estimate the large displacement due to the constant brightness assumption.

The optical flow computes the displacement of every pixels, so the pixels are selected as points. To improve the computational efficiency, we resized the images to the resolution of 120*160.

4.2 SIFT based Motion Detection

To deal with large movements, we adopt the scale-invariant feature transform (SIFT) to detect the scale-invariant features. The movements of SIFT features can be identified by matching the corresponding features of two frames (Lowe, 2004).

The SIFT algorithm first detects the location of the keypoints in two frames separately, then compute the SIFT descriptor for each keypoint, which is a 128 dimensional feature vector (Lowe, 1999). Keypoints between two images can be matched by using the nearest-neighbours approach. The Euclidean distance between two SIFT feature vectors is used to evaluate the similarity of vectors. A SIFT feature vector D_1 is matched to a SIFT feature D_2 only if the distance satisfy the following two conditions:

- The distance is smaller than some threshold.
- The distance is not greater than the distance of D_1 to all other descriptors.

For the first condition, the ratios between the distance of the nearest neighbor and the second nearest neighbour are calculated. According to Lowe's research (Lowe, 2004), the matches are accepted in which the distance ratio is smaller than 0.8, which will result in a highest accuracy of matching. However, the matched results may still include some incorrect matches due to the imprecision of the SIFT model. RANSAC (Fischler and Bolles, 1981) is used to refine the matching by filtering out the "bad" matches.

A pair of matched vectors denotes the geometric information of the same keypoint in two different images. The movement vector of such keypoint can be obtained by computing the displacement of the coordinates. We can generate a flow field by computing the movement vectors for all matched keypoints.

4.3 Parametric Motion Model of Moving Object

Both the optical flow and SIFT matching can produce a set of movement vectors to denote the geometric transformations of relevant points. The movement of one object can be a combination of a translation, a rotation and a scaling. In other words, an affine transformation (AF) can be used to denote the movement of an object. Assuming that objects will not change their shapes, i.e. the same object looks almost the same in all frames of the sequence, we assume that the displacement of all points in this object will satisfy the affine transformation.

Let $\mathrm{x} = (x,y)^T$ the position of one point in a frame, and let $\mathrm{x}' = (x',y')^T$ be the position of corresponding point in the next frame. This pair (x,x') indicates the movement of one point between 2 frames. Then

$$\mathrm{x}' = A\mathrm{x} + b; \tag{6}$$

where $A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$, $b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$.

The six affine transformation parameters of (A,b) form the parametric motion model of the object.

4.4 Motion based Segmentation

Motion-based segmentation aims to group together the points with the same movement. Since points belonging one object will have the same movement, we may use the movement to identify the points that belong to one object. An affine model with six parameters is used to indicate the movement of an object. We use modified EM algorithm with a recursive division strategy to determine the clusters of points; i.e., the segmentations.

Algorithm 1 gives the main steps of the EM based segmentation algorithm. Each group of points in this algorithm indicates an object.

Algorithm 1: EM-based segmentation algorithm.

Set the number of objects to 1;
Put all points into one group;
repeat
  repeat
    Calculate the parameters (A,b) of the affine transformation of each group of points (see Equation 7);
    Reassign each point to a group (determine the value of variable **z**) based on the error of the point w.r.t. each group;
  until convergence
  if the group with the largest errors given the group parameters (A,b) exceeds the threshold **then**
    Split the group with the largest errors;
    Increase the number of objects by 1;
  end if
until no group can be find to split, or a maximum number of iterarions reached.

There are four key components in this algorithm,

- How to determine the parameters (A,b) of the affine transformation?
- How to determine the best assignment of points for each group?
- How to determine the group to be split?
- How to split the group?

Given a group of points and their locations in two frames, we can obtain the affine parameters by solving Equation 6. In practice, the groups could contain outliers because the segmentation is not perfectly. The parametric motion model (A,b) of the affine transformation of a group G is obtained by solving the optimization problem:

$$\begin{aligned} (A,b) = \mathrm{argmin}_{(A,b)} \textstyle\sum_{(\mathbf{x},\mathbf{x}')\in G} ||\varepsilon||_{l_2} \\ \text{subject to } \varepsilon = \mathrm{x}' - A\mathrm{x} - b \end{aligned} \tag{7}$$

Suppose there are K groups, the division of points is regarded as an optimization problem:

$$\min \sum_{k\in[1,\ldots,K]} E_k \tag{8}$$

where $E_k = \sum_{(x,x')\in G_k} ||\varepsilon||_{l_2}$.

Given a partition of points, each group has an average error $\overline{E}_k = \frac{1}{N_k} E_k$ with respect to its motion model $(A,b)_k$. The group with largest average errors is selected to be split, while the largest error is marked as the error of current partition. The selected group can be split into 2 sub-groups using a bisecting K-means algorithm (Selim and Ismail, 1984). Then the number of groups is increased by 1 and a new partition of the points is computed by solving Equation 8. If the error of the new partition is smaller than the error of the old partition, the current partition is updated by using the new partition and motion model. Otherwise, it means the optimal partition is found and no groups is able to be split, i.e. the iteration comes to an end.

4.5 Segmentation of Sequences

Section 4.4 describe the segmentation based on the movement between two frames. To extend the segmentation to sequences, we need to make use of the historical information. A probability matrix $P_{N\times K}$ is built to indicate the probabilities of each point with regards to each group. Here N denotes the number of points and K is the number of groups. Given a partition $(G_1, G_2, \ldots G_K)$, the probability of point i with regard to group k is estimated:

$$p_{i,k} = 1 - \frac{\varepsilon_{i,k} + \frac{\delta}{K}}{\sum_{j=1}^{K} \varepsilon_{i,j} + \delta} \tag{9}$$

where $\delta = 0.1$, which is used for preventing divided by zero. The EM segmentation computes such a probability matrix for each pair of successive frames. If a point presents in F successively frames, the trajectory of movement vectors has a length of $F-1$. The probability of points w.r.t. the sequence is a combination of the probability computed from last frame pair and the historical probabilities as shown in Equation 10. In Equation 10, the parameter α is a factor to decrease the weight of historical data, which is set as 0.85 in the experiments.

$$p(i,k|v_f,v_{<f}) = \frac{p(i,k|v_f)+\alpha p(i,k|v_{<f})}{1+\alpha} \quad (10)$$

Note that unlike other segmentation algorithms (Shi and Malik, 1998; Borshukov et al., 1997; Elhamifar and Vidal, 2009), we do not require that points are present in all frames.

4.6 Classification

The previous stages results in a segmented image, which separates the moving objects with different movement. For the optical flow based segmentation, the detected regions are groups of pixels, which means the image is divided in to small image patches. SIFT based segmentation result in groups of SIFT features. So in the classification stage, we need to deal with 2 kinds of input data, the pixel level images and the bags of SIFT features.

A Support Vector Machine with an RBF kernel is chosen to fulfill the classification task in this paper.

5 EXPERIMENT

The approach proposed in this paper gives an architecture for detecting and recognizing moving objects from videos. The main component of our approach is the task of motion segmentation. Thus, we will evaluated our approaches in the following ways:

- Evaluate object detection results, with Optical flow based and SIFT based motion segmentation respectively.
- Compare the segmentation results using our method with some control approaches.
- Test the quality of classifications (recognition) when using the result of the segmentation as input.

The segmentation is evaluated on video sequences from three database: the robocup 2014 video [1], CNnet 2014 (Wang et al., 2014) and the Hopkins155 motion database[2]. Figure 2 shows some instances of the images from different sequences. The classification of segmented images is evaluated on results of all above segmentations.

5.1 Objects Detection in Videos

Our approach is examined on the 20 videos mentioned above, whose composition is described in Table1.

Table 1: The composition of tested videos containing different number of objects.

Number of videos	Number of Objects				Camera
	2	3	2-3	2-4	
CNnet2014	2	3	2	3	fixed
Robocup	3	2	3	2	moving

For each video clip which has a frame rate of 24 to 30 fps , a sequence of 30 frames is selected for test. To compare the motion detection results using optical flow and SIFT matching, we tested the accuracy of segmentation results using different frame rates. That is, new sequences are generated from each sequence by selecting frames with an interval of itv (where $itv = 3, 5, 10$), while the original sequence has $itv = 1$. For the sequences with larger intervals, the displacement of points between two frames increases.

The movement of a point is represented by its coordinates in two neighbouring frames, thus there is a set of points $X_f \in \mathbb{R}^{2\times N_f}$ for frames $f = 1,...F$, where N_f is the number of feature points detected in frame f. For the OFE motion detection N_f is fixed for all frames, which is 120×160. For the SIFT detection, trajectories are discontinuous for SIFT points, where N_f varies from 300 to 500 for different frames.

Table 2 shows accuracy of segmentation results with different intervals of sequences.

Table 2: The average accuracy (%) of segmentation using our approach, with different intervals, based on the 20 videos (from CNnet2014 and robocup competition video).

(a) Test with optical flow based motion

Number of objects	Interval of frames			
	1	3	5	10
2	88.2	85.4	78.2	63.6
3	91.0	82.0	81.3	61.6
2-3	79.6	73.2	65.1	59.0
2-4	76.9	69.4	60.1	58.2
all	83.9	77.5	71.2	60.6

(b) Test with SIFT based motion

Number of objects	Interval of frames			
	1	3	5	10
2	97.8	99.5	98.3	97.4
3	98.0	96.9	99.0	98.6
2-3	94.9	99.8	98.2	80.0
2-4	96.7	98.9	92.0	74.1
all	96.9	98.8	96.9	87.5

[1] https://www.youtube.com/watch?v=dhooVgC_0eY
[2] http://www.vision.jhu.edu/data/hopkins155/

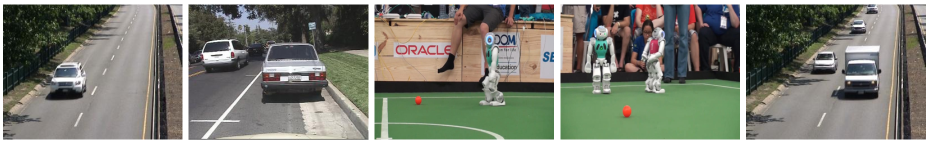
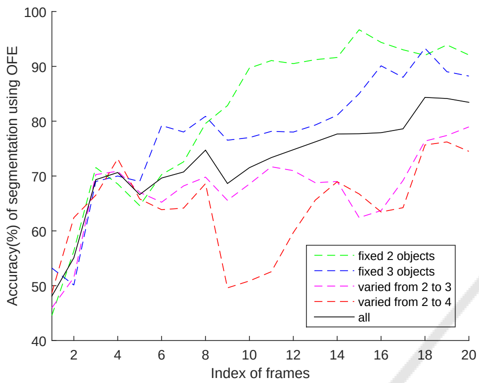
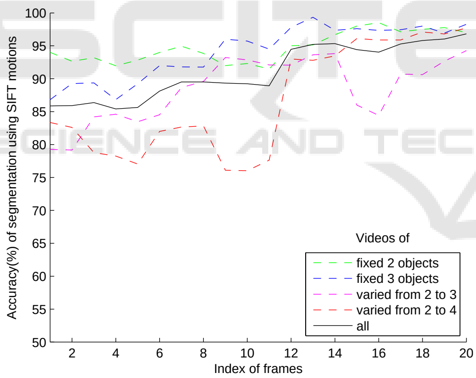


Figure 2: Images from test sequences.

Figure 3a and Figure 3b show the curves of average accuracy of motion segmentation with regards to the number of frames that have been processed. Here $itv = 1$.



(a) optical flow based motion segmentation



(b) SIFT based motion segmentation

Figure 3: Accuracy curves w.g.t. the index of frames, using (a) Optical flow based motions and (b) SIFT based motions. The red line indicates the average accuracy of all video sequences in test. The slash line indicates the average accuracy over sequence with different number of objects, which are drawn with different colors.

From the result, we can make the following conclusions:

1. SIFT based motion segmentation performs better than optical flow based method in 2 ways. Firstly The number of points of SIFT detection are smaller, which requires for less computational resources. Secondly, the accuracy of SIFT based segmentation is always higher than optical flow.
2. For the sequences with changing number of objects, the accuracy fluctuates at the frames where the number of objects changes. Despite the changing number of objects, both methods show an general increasing trend in segmentation accuracy.
3. The approach can deal with fixed number of objects as well as changing number of objects, with a maximum of 4 objects in the test. The performance of segmentation with changing number of objects is slightly worse than the test with fix number of objects.

5.2 Comparison of Motion Segmentation

In this section, a comparison test is described based on the database of Hopkins155, which contains of 155 videos of 29 or 30 frames, each contains 2 or 3 moving objects. The trajectories of feature points are provided by the database (the average number of feature points is 266 for 2 objects, while it's 398 for 3 objects). Only the segmentation part of our approach is evaluated in this section. The objects of "checkerboard" make 3D rotations and translations. The "traffic" sequences contain moving vehicles of outdoor traffic scene. The remaining sequences named "articulated" contain motions constrained by joints, head and face motions, people walking, etc. Over half of the videos are taken using a moving camera.

Our segmentation method named as adapted EM segmentation for motion sequences (AEMS), is compared with the SSC (Elhamifar and Vidal, 2009), LSA (Yan and Pollefeys, 2006), RANSAC (Fischler and Bolles, 1981), and ALC (Rao et al., 2008). Table 3 shows the segmentation accuracy for sequences of Hopkins155, Table 4 shows the accuracy of finding the number of objects in sequence for AEMS.

The SCC outperforms all methods in general. The performance of our approach varies for categories. On average, our method ranks 3rd out of 5 methods. We can also draw the conclusion from the results:

1. AEM performs with a high accuracy of 99% when there are only 2-dimension translations.
2. AEM is sensitive to 3-dimensions rotation and scaling.
3. AEM can find the number of objects automatically, with a high accuracy of 96.2%.

Table 3: Accuracy (%) of motion segmentation using different methods.

(a) sequences with 2 motions.

LSA	RANSAC	ALC	SSC	AEMS
Checkerboard:78 sequences				
97.4	93.5	98.5	98.8	93.4
Traffic:31 sequences				
94.6	97.4	98.4	99.9	99.4
Articulated: 11 sequences				
95.9	92.7	89.3	99.4	93.2
All: 120 sequences				
96.0	94.5	95.4	99.4	95.3

(b) sequences with 3 motions.

LSA	RANSAC	ALC	SSC	AEMS
Checkerboard:26 sequences				
94.2	74.2	94.8	97.0	86.6
Traffic:7 sequences				
74.9	87.2	92.3	99.4	99.1
Articulated: 2 sequences				
92.8	78.6	78.9	98.6	79.6
All: 35 sequences				
87.3	80.0	88.7	98.3	88.4

(c) all sequences.

LSA	RANSAC	ALC	SSC	AEMS
All:155 sequences				
91.6	87.3	92.0	98.8	91.9

Table 4: Accuracy (%) of estimating the number of objects.

Number of objects	Checker-board	Traffic	Articulated
2	92.8	96.6	81.2
3	86.7	98.4	83.6
all	89.9		

4. The AEM methods is not affected by the camera with a 2-dimensional movement, since the video sequences used in the experiment are taken using a fixed or a moving camera.

Note that the comparative methods require trajectories with fixed dimensions, as well as with a fixed number of moving objects. In contrast, our method is able to handle changing numbers of objects, and feature points that presented in only a part of a trajectory.

5.3 Classification

The result of motion segmentation provides groups of points, each of which should represent one object. The next step is to recognize the objects. We classify the segmented results using a SVM classifier. There are two types of segmentation results in section 5.1:

1. optical flow based method provides image fractions because points are pixels from image;
2. SIFT based methods gives the groups of SIFT points, each point is associated with a feature vector. The groups of SIFT features are coded into vectors of the same dimension using the bag of word methods (Csurka et al., 2004).

The sequences contain 8 categories of objects, including cube, conical frustum, curved paper, car, truck, robot, pedestrian, face. We used a training set contains 80 sequences from the total $20+155$ sequences for training and the rest for testing.

The classification results of the 8 categorises are listed in Table 5a. Table 5b shows the classification results of only for cars and trucks, using a classifier that was trained with a different database namely Caltech256 [3].

The results indicate that, segmentation accuracy is sufficient to recognize the objects, for the database tested.

6 CONCLUSIONS

In this paper we proposed an architecture for moving object detection and recognition in video sequences based on detecting changes and clustering movements. We compared two approaches for detecting objects motions. One is based on optical flow motion detection which detect the changes between pixels. The other is based on SIFT which detect the SIFT points and find the motions by matching points between frames. An adapted EM algorithm is used to cluster the moving points, which gives us the segmentation. An SVM is used to identify the segmented objects.

The results shows that higher level-feature (SIFT) has the advantage of a lower computation time and a higher accuracy in segmentation. The main characteristic of our methods has the ability to handle a changing set of feature points. Because of the objects movements, feature points may not be visible in all frames. Moreover, our method can determine the number of objects. Experiment shows that our method perform especially well for the 2D movement.

In the future work, we need to evaluate our method on sequences with more than 4 objects. An extension to a 3D motion model is also needed for applications in robot vision. Last but not least, more research is needed with regard to the other methods of feature detecting and motion extraction.

[3] http://authors.library.caltech.edu/7694/

Table 5: Accuracy (%) of classification using SVM.

(a) All sequences

	robots	cars	trucks	pedestrian	Conical	cube	cylinder	face
OFE	78	92	100	72	98	93	96	98
SIFT	82	89	66	75	95	95	97	99

(b) “cars” and “trucks”

	cars	trucks
OFE	97.9	98.5
SIFT	97.2	96.9

REFERENCES

Borshukov, G. D., Bozdagi, G., Altunbasak, Y., and Tekalp, A. M. (1997). Motion segmentation by multi-stage affine classification. *IEEE Trans. Image Processing*, 6:1591–1594.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM.

Bouguet, J.-Y. (2001). Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel Corporation*, 5.

Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38.

Elhamifar, E. and Vidal, R. (2009). Sparse subspace clustering. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2790–2797. IEEE.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

Horn, B. K. and Schunck, B. G. (1981). Determining optical flow. In *1981 Technical Symposium East*, pages 319–331. International Society for Optics and Photonics.

Karasulu, B. and Korukoglu, S. (2013). Moving object detection and tracking in videos. In *Performance Evaluation Software*, pages 7–30. Springer.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

Lucas, B. D., Kanade, T., et al. (1981). An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679.

Martinez-Conde, S., Macknik, S. L., and Hubel, D. H. (2004). The role of fixational eye movements in visual perception. *Nature Neuroscience*, 5:229 – 240.

Narkhede, H. (2013). Review of image segmentation techniques. *International Journal of Science and Modern Engineering (IJISME)*, 1:5461.

Pan, Z. and Ngo, C.-W. (2005). Selective object stabilization for home video consumers. *IEEE Trans. Consumer Electronics*, 51(4):1074–1084.

Rao, S. R., Tron, R., Vidal, R., and Ma, Y. (2008). Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE.

Seerha, G. K. and Rajneet, K. (2013). Review on recent image segmentation techniques. *International Journal on Computer Science and Engineering (IJCSE)*, 5:109–112.

Selim, S. Z. and Ismail, M. A. (1984). K-means-type algorithms: a generalized convergence theorem and characterization of local optimality. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (1):81–87.

Shi, J. and Malik, J. (1998). Motion segmentation and tracking using normalized cuts. In *Computer Vision, 1998. Sixth International Conference on*, pages 1154–1160. IEEE.

Wang, J. Y. and Adelson, E. H. (1994). Representing moving images with layers. *Image Processing, IEEE Transactions on*, 3(5):625–638.

Wang, Y., Jodoin, P.-M., Porikli, F., Konrad, J., Benezeth, Y., and Ishwar, P. (2014). Cdnet 2014: An expanded change detection benchmark dataset. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 393–400. IEEE.

Weiss, Y. (1997). Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 520–526. IEEE.

Yan, J. and Pollefeys, M. (2006). A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *Computer Vision–ECCV 2006*, pages 94–106. Springer.

Zappella, L., Lladó, X., and Salvi, J. (2008). Motion segmentation: a review. In *Proceedings of the 2008 conference on Artificial Intelligence Research and Development: Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence*, pages 398–407. IOS Press.