An EM based approach for motion segmentation of video sequence

Wei Zhao
Department of Data Science and Knowledge Engineering
Maastricht University
Maastricht, Netherlands
wei.zhao@maastrichtuniversity.nl

Nico Roos
Department of Data Science and Knowledge Engineering
Maastricht University
Maastricht, Netherlands
roos@maastrichtuniversity.nl

ABSTRACT

Motions are important features for robot vision as we live in a dynamic world. The detection of moving objects is crucial for mobile robots and computer vision systems. This paper investigates an architecture for the segmentation of moving objects from image sequences. Objects are represented as groups of SIFT feature points. Instead of tracking the feature points over a sequence of frames, the movements of feature points between two successive frames are used. The segmentation of motions of each pair of frames is based on the expectation-maximization algorithm. The segmentation algorithm is iteratively applied over all frames of the sequence and the results are combined using Bayesian update.

Keywords

Motion segmentation, EM algorithm, Bayesian update, SIFT feature, trajectory clustering

1 INTRODUCTION

Moving object detection is an important issue in the field of computer vision and one of the basic tasks of video processing. It differs from the class-specific object detection [YP06, FPZ03] and static object detection [FGMR10, POP98], which focus on building models of objects or background. Moving object detection is based on the assumption that foreground objects are usually accompanied by unique motion patterns [HFH07]. Techniques of moving object detection are widely used in different areas, such as video surveillance systems [JT12], robot navigation [JS04, CSSF07], unmanned aerial vehicles [RCTdC$^+$12], and so on. In general, they consist of three main steps: motion detection, motion segmentation and object classification.

The motion detection can be achieved by tracking feature points [WKSL13], or estimating the optical flow between frames to recover the motion of each image pixel [CSSF07]. Motion segmentation aims at dividing the points (or pixels) into a set of groups according to their motion coherence [BBAT97, VH04, RCTdC$^+$12, JT12, ZR16]. The segmention results are groups of features points, or regions of images, which are processed by an object classification algorithm.

The approach proposed in this paper aims at segmenting moving objects in image sequences (videos). There are four steps in our approach: feature extraction, motion detection, motion segmentation, and combining segmentations of multiple frames, where the third and fourth steps are the main steps of our approach. Feature extraction and motion detection are realized by technique of scale-invariant feature transform [Low99]. The feature points are segmented into different groups based on their movements between pairs of image frames, using an adapted EM algorithm. The segmentations of multiple frame pairs are combined using Baysian update. The resulting groups of feature points are either moving objects in the scene or background regions. These groups of feature points can be processed by a classification algorithm. We evaluated our work in two ways: the accuracy of the segmentation and the computational efficiency.

A brief review of some related work will be given in the next section. The general architecture of the proposed model and the details of the segmentation algorithm of our approach are described in Section 4 . Experiments that we used to evaluate our approach are presented in Section 5. Section 6 concludes the paper.

2 RELATED WORK

Detecting and tracking moving objects are challenging issues in the field of computer vision and a very important step in video processing. Numerous approaches of video-based object detection and tracking are proposed for different domains of use. Approaches for detecting similarities are widely used for moving object detection, which is related to techniques such as: optical flow, feature tracking, data clustering and segmentation [SR13]. In this paper, we focus on the motion segmentation techniques.

Wang and Andelson used optical flow for motion estimation and k-means clustering for segmenting [WA94]. Shi and Malik [SM98] construct a weighted spatio-temporal graph on an image sequence and use normal cuts for motion segmentation.

Jung and Sukhatme [JS04] proposed a moving object detection system for mobile robots. They subtract the background by estimating the motion model of the camera. Pan and Ngo proposed to combine optical flow estimation with the EM algorithm [PN05] for the purpose of image stabilization.

Vidal and Hartley proposed a motion segmentation algorithm for trajectory clustering by using generalized principal component analysis (GPCA) to cluster projected data [VH04]. Jung and Sukhatme [JS04] proposed a moving object detection system for mobile robots, where a probabilistic model accompanied with an adaptive particle filter and an EM algorithm is used for detecting the moving foreground objects. Elhamifar and Vidal use the sparse representation to cluster trajectories from multiple linear or affine subspaces [EV09].

3 PRELIMINARIES

The approach proposed in this paper makes use of scale-invariant feature transform (SIFT), affine transformation, expectation-maximization (EM) and Bayesian update. In the section, a brief review of the techniques used in our paper is provided.

3.1 Scale-Invariant Feature Transform

SIFT is an algorithm to detect and describe local features in images, which was proposed by [Low99]. It is proved to be an efficient and robust way of detecting points of interests, which is useful in object detection and recognition. The SIFT feature are invariant to image scaling and rotation, and robust to large amounts of pixel noise [Low04]. Because of the scale-invariant properties and the high level feature expression, SIFT features are easy track in video sequences. Moreover, object recognition based on SIFT feature performs well [Low04].

3.2 Affine Transformation

The motions of objects in 3D space are projected to 2D images by camera in daily life videos. In a very short period, the changes of objects due to the 3D motions will be small and can be ignored. Thus the points belonging to one object can be assumed have the same 2D motions in frames. In that case, an affine transformation model is able to describe the movement of an object. If a point is detected at position x in one frame and at position x' in the next frame, then Equation 1 is assumed to hold for all points belonging to the same object.

$$\mathrm{x}' = A\mathrm{x} + b; \tag{1}$$

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

3.3 Expectation-Maximization algorithm

The EM algorithm [DLR77] is an effective and popular technique for estimating parameters of a distribution from given data set.

Given the observed data $\mathbf{x}$ associated with a set of unobserved latent data or missing values $\mathbf{Z}$, and a vector of unknown parameters θ, the maximum likelihood estimate (MLE) of the unknown parameters is determined by maximizing the expected value of the likelihood function $L(\theta;\mathbf{x},\mathbf{Z}) = P(\mathbf{x},\mathbf{Z} \mid \theta)$.

Two steps are iteratively applied to find the MLE of the marginal likelihood until convergence,

E-step Given the parameters θ and the data $\mathbf{x}$ we can determine the probability distribution of the hidden variables $\mathbf{Z}$.

M-step Find a maximum likelihood estimate of the parameters.

$$\begin{aligned} &\theta = \mathrm{argmax}_{\theta'} L(\theta';\mathbf{x}) \\ &\text{where: } L(\theta;\mathbf{x}) = P(\mathbf{x} \mid \theta) = \sum_{\mathbf{Z}} P(\mathbf{x},\mathbf{Z} \mid \theta) \end{aligned} \tag{2}$$

In the application, we make use an adapted version to find hidden variables and parameters θ. Instead of the probability distribution $P(\mathbf{x},\mathbf{Z} \mid \theta)$ we determine:

$$\mathbf{z} = \operatorname*{argmax}_{\mathbf{Z}} P(\mathbf{x},\mathbf{Z} \mid \theta) \tag{3}$$

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

$$\theta = \operatorname*{argmax}_{\theta'} L(\theta';\mathbf{x},\mathbf{z}) \tag{4}$$

4 METHOD

We proposed a new approach for the segmentation of moving objects from video sequences. Fig.1 gives the architecture of our approach.

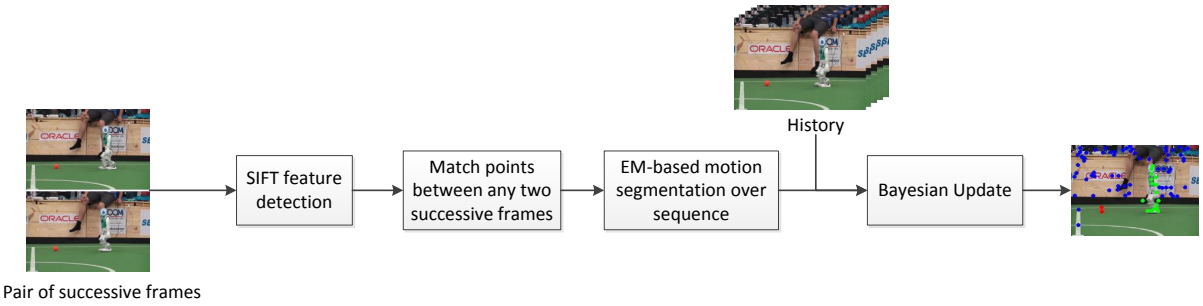


Figure 1: Architecture of our approach for video segmentation

We detect the SIFT feature points in each frame in the video sequence. The feature points of two successive frames are matched using the algorithm suggested by Lowe [Low04]. The movements of the matched feature points between two frames are subsequently obtained.

The movement of a point over multiple frames can be represented as a trajectory, which is a vector consisting of the positions of the point in multiple frames. Trajectories can be generated for "continuity" feature points, which means they appear in all frames of the sequence [SWY+09, WKSL13]. However, for many points, the "continuity" doesn't hold because of occlusion or 3D rotation of objects. Thus many feature points are excluded when requiring full trajectories over a sequence, which reduces the segmentation quality and increase the difficulty of recognition in the next step.

We investigate an segmentation algorithm making use feature points of both "continuity" and "discontinuity". An EM based segmentation algorithm is iteratively applied to segment feature points for each pair of successive frames. The segmentations are iteratively refined frame by frame using Bayesian update.

4.1 SIFT based motion detection

We detect the SIFT key points in each frame of the video sequence using the approach proposed by Lowe [Low04]. The movements of SIFT features can be identified by matching the corresponding features of two frames using the nearest-neighbours approach. The similarities of two features points are evaluated by computing the Euclidean distance between the feature 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.

RANSAC [FB81] is used to refine the matching by filtering out the incorrect matches due to the imprecision of the SIFT model.

The movement vector of a matched point can be obtained by computing the displacement of the coordinates of matched the features, which denotes the position change of the same point in two different images. A motion flow field is determined by computing the movement vectors for all matched points. A motion field is generated between each pair of neighbouring frames.

4.2 Parametric Motion Model

An affine model of 6 parameters is used for representing the parametric motion model of an object. The affine model is estimated iteratively for movements between a pair of neighboring frames. Given the movement of any 3 points of the object, (A,b) can be computed. However, in our approach, the segments of moving points could contain outliers because the segmentation is not perfect. Moreover, the observed movements of points can contain noise. So, given a set of pairs of feature points G, the parameters of affine model for one object can be estimated by solving the optimization problem:

$$\begin{gathered}(A,b) = \text{argmin}_{(A,b)} \sum_{(\mathbf{x},\mathbf{x}')\in G} ||\varepsilon||_{l_2} \\ \text{where } \varepsilon = \mathrm{x}' - A\mathrm{x} - b\end{gathered} \tag{5}$$

In some situations, the number of points belonging to an object is less than 3. For example, for a small rolling ball, SIFT can only detect 1 or 2 feature points on the ball. In this case, we assume the affine transformation degenerates to translation for one point, and a combination of translation and scaling for 2 points. The matrix A is reformulated as Equation 6.

$$A = \begin{cases} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, & \text{for group of 1 point} \\ \begin{pmatrix} a_{11} & 0 \\ 0 & a_{22} \end{pmatrix}, & \text{for group of 2 points} \\ \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, & \text{for group of 3 or more points} \end{cases} \tag{6}$$

4.3 EM-based Motion Segmentation

Given the points and their movements between two frames, an EM-based segmentation algorithm is used to segment the motion field into groups of points each representing an object. Algorithm 1 gives the main steps of the EM based segmentation algorithm.

Algorithm 1 EM-based motion segmentation algorithm

```
Initialize the points in one group
repeat
  repeat
    Using EM algorithm to estimate the best parameters of affine motion model, and the assignment of points
  until convergence
  if the group with the largest errors given the affine parameters 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
```

In this algorithm, there are 3 components to be noticed:

Estimating affine parameters

Given a partition of points, the affine parameters of each group can be estimated by Equation 5 as discussed in Section 4.2.

Re-partitioning of points

Re-partitioning of points by reassigning the points to the groups, when the affine models are known. Suppose there are K groups, the division of points is regarded as an optimization problem:

$$\min \sum_{k\in[1,\ldots,K]} \mathscr{E}_k \tag{7}$$

where $\mathscr{E}_k = \sum_{(x,x')\in G_k} ||\varepsilon||_{l_2}$, and $\varepsilon = \mathrm{x}' - A\mathrm{x} - b$

Splitting

There are two aspects to be considered:

1. How to determine the group to be split?

   Given a partition of points, each group has an average error $\overline{\mathscr{E}_k} = \frac{1}{N_k}\mathscr{E}_k$ with respect to its motion model $(A,b)_k$. We choose the group with the largest average error to split.

2. How to split the selected group?

   We split the group with largest $\mathscr{E}_k$ using a bisecting K-means algorithm [SI84]. Once the group is split, a new partition of points and the corresponding motion models are computed. If the largest error of the new partition decreases, the current partition is updated by using the new partition and models. Otherwise, it means the optimal partition is found and no groups can be split, i.e. the iteration comes to an end.

4.4 Segmentation of trajectories

The EM-based segmentation algorithm in Section 4.3 deals with the temporal movements between two frames. It is extended to a video sequences using Bayesian update.

Given an image sequence of $T+1$ frames $f_0, f_1, \ldots f_T$, a segmentation is determined for each pair of successive frames (f_{i-1}, f_i). For each pair of frames, we estimated the probability $p(e|i,k)$ of the evidence e given the assignment of feature point i to a group k. Here the evidence is the error of the motion vector of a feature point with respect to the affine transformation of each group. We assume that the probability $p(e|i,k)$ is a decreasing function of the relative error of point i with respect to group k given K different groups. Equation 8 formalizes the computation of $p(e|i,k)$.

$$p(e|i,k) = 1 - \frac{\varepsilon_{i,k} + \frac{\delta}{K}}{\sum_{j=1}^{K} \varepsilon_{i,j} + \delta} \tag{8}$$

where $\delta = 0.1$, which is used for preventing dividing by zero.

Assuming that the evidence $E_t = (e_1, \ldots e_t)$ over t $(0 < t < T)$ pairs of frames in the sequence is independent, we may use Bayesian update to determine the probability that point i belongs to group k given all evidence E_t:

$$P(i,k|E_t) = \frac{P(E_t|i,k)}{P(E_t)} P(i,k) \tag{9}$$

where $P(i,k) = \frac{1}{K}$ and

$$\begin{aligned} P(E_t|i,k) &= P(e_1, \ldots e_t|i,k) \\ &= P(E_{t-1}|i,k) \cdot p(e_t|i,k) \end{aligned} \tag{10}$$

$$P(E_t) = \sum_{k=1}^{K} P(E_t|i,k) \tag{11}$$

5 EXPERIMENTS AND RESULTS

In this section, we will compare the segmentation results using our approach with some control approaches. Since our approach aims at dealing with long term motions, trajectory clustering algorithms of motion segmentation such as SSC [EV09], GPCA [VH04] and LSA [YP06] are used for comparison. The segmentation is evaluated on video sequences from three data bases: the robocup 2014 video [1], CNnet

[1] $https://www.youtube.com/watch?v=dhooVgC_0eY$

Figure 2: Images from videos used in experiment

2014 [WJP+14] and the Hopkins155 motion database[2]. There are videos of some indoor objects, moving pedestrian, moving cars, and robot soccer. Fig.2 shows some instance of the videos. Videos from Hopkins155 have a frame rate of 15 fps, while frame rate of videos from the other two data base are about 24 to 30 fps. Sequences of 30 frames are used in the experiments.

We will evaluate the approaches in the following ways:

- Evaluate the performance of segmentation algorithm on the data of motion trajectories.
- Evaluate the segmentation results w.r.t. all detected features.
- Evaluate the computing times.

Comparing with the other methods, our approach has the 3 unique characteristics:

- Our method includes the function of detecting feature points and their motions, while the comparison approaches are pure clustering algorithms for detected motion trajectories.
- Our method can deal with missing points in some frames, which doesn't hold for most of the comparison approaches because they require that the motion trajectories (the input data) are of the same length.
- Our method can determine the number of groups, while the other methods need the number of groups as an input.

Due to the differences of our method and the comparison methods, we designed two experiments to evaluate them. First, we will compare the performance of our motion segmentation algorithm on the full trajectories of the feature points provided by the data set Hopkins155, which will be discussed in Section 5.1. In this experiment, we don't detect feature points and their motions.

In the second experiment, we will evaluate our method using the original videos. The feature points and their motions are detected first. The segmentation quality over all detected SIFT features are evaluated, which will be discussed in Section 5.2.

All experiments are run on Matlab 2014a, with a computer of Intel Core i5 at 3.1GHz and 4GB of RAM.

[2] *http://www.vision.jhu.edu/data/hopkins155/*

5.1 Motion segmentation over trajectories

The experiment in this section runs on the Hopkins155 dataset. The codes for compared methods are from the site of hopkins155.

The Hopkins155 dataset contains 155 videos of 29 or 30 frames, each containing 2 or 3 moving objects. Each object is represented by a group of feature points. There are 266 to 398 feature points provided for each video, as well as the ground truth segmentation of the feature points. In these videos, the background is regarded as one object. Points from the background indicate the movement of the camera. The trajectory data $X \in \mathbb{R}^{2F\times N}$ is provided for each video, where F is the number of frames, N is the number of feature points. Each row of X is a trajectory of one feature point.

The videos are divided into 3 categories, the category named "checkerboard" contains several objects covered with a uniform checker board surface, which make 3D rotations and translations. The "traffic" sequences contain moving vehicles in outdoor traffic scenes. 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 adapted EM segmentation using Bayesian update for motion sequences (AEM-b), is applied to the trajectories for segmenting the given feature points. The results are measured by the percentage of points that are clustered correctly, compared with the ground-truth clustering provided by the Hopkins155 dataset.

Table 1 presents the accuracies of segmentation results for sequences of different categories and number of motions. Each motion indicates an moving object (the background is also regarded as an object moving with the camera). The result of RANSAC for the same sequence can vary in each operation because of the statistical nature of RANSAC. We take the average results by running the algorithm 1,000 times for each sequence, and the threshold is set to 0.005.

The results in Table 1 show that SSC outperforms all methods in general, while our method ranks 2nd out of 5 methods on average. The average difference with SSC varies between 0.4% and 15%.

Since out approach especially performs worse than SSC for the 'checkerboard' category, we analyzed these segmentation results in more detail. Our method per-

forms 7% to 15% worse than SSC in this category. Table 2 presents the segmentation results for different types of camera movements. The tables shows that our method can achieve 99.8% segmentation accuracy for the videos with a static camera. However, our method is not good in dealing with the videos taken by a rotating camera. More specifically, when we look into the details of the segmentation results of this category, our method performs very bad (under 70%) for the videos where the displacement of camera (both rotating and translating) is large compare with the displacements of objects.

We also investigated whether our method is able to identify the correct number of moving objects in the videos. Table 3 shows the accuracy of identifying the correct number of objects given different numbers of moving objects.

We can draw the following conclusion from the results:

- AEM-b performs well for the traffic videos, where the major motions are 2d translations.
- AEM-b is able to find the number of objects automatically, with a high accuracy of 96.2%.
- AEM-b is not good at dealing with the 'checkerboard' videos, especially when the camera is rotating.
- AEM-b doesn't consider the relative position of feature points. Points apart from each other but with similar movements could be mis-clustered.

5.2 Motion Segmentation over detected points

In this section, we will apply the SIFT motion detection discussed in Section 4 directly to the original videos from CNet and robocup 2014. The SIFT features and their movements are generated frame by frame. For our method, we will apply the process of Figure 1, which will make use of the feature points existed in any two successive frames. Because the comparison approaches can only deal with trajectories of the same length for different lengths, we will detect the SIFT feature points existed in all frames, which will be result in a matrix of trajectories having the format of the data from Hopkins155 dataset.

The number of feature points in the trajectories matrix will decrease as the length of sequence increases. For each video, we test the methods using sequences of different lengths, which varies from 2 frames to 30 frames. Figure 3 shows the average number of feature points for different sequence lengths, with respect to different lengths of sequences. The blue line indicates all detected feature points, the red line is the number of feature points utilized by our method, and the green

LSA	RANSAC	GPCA	SSC	AEM-b
Checkerboard:78 sequences				
93.91	92.01	79.11	98.4	91.5
Traffic:31 sequences				
98.6	92.14	73.2	99.4	99.0
Articulated: 11 sequences				
96.9	90.45	72.5	98.9	92.0
All: 120 sequences				
95.4	91.9	77.0	98.8	93.5

(a) Sequences with 2 motions.

LSA	RANSAC	GPCA	SSC	AEM-b
Checkerboard:26 sequences				
68.1	72.23	80.4	97.4	83.9
Traffic:7 sequences				
80.2	88.28	53.1	99.2	98.9
Articulated: 2 sequences				
83.2	76.98	78.9	98.9	84.4
All: 35 sequences				
71.3	75.7	74.9	97.9	87.0

(b) Sequences with 3 motions.

LSA	RANSAC	GPCA	SSC	AEM-b
All:155 sequences				
90.0	88.2	76.5	98.5	92.1

(c) All sequences.

Table 1: Accuracy (%) of motion segmentation for different settings.

LSA	RANSAC	GPCA	SSC	AEM-b
Static camera: 20 sequences				
92.2	92.2	89.4	99.6	99.8
Translating camera: 20 sequences				
79.9	81.8	76.9	99.1	96.5
Rotating camera: 24 sequences				
71.0	76.4	62.7	98.0	80.1
Rotating and translating camera: 40 sequences				
94.2	93.4	83.2	97.5	90.4

Table 2: Segmentation of checkerboard videos according to the movement of camera.

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

Table 3: Accuracy (%) of estimating the number of objects.

line shows the number of points utilized in the trajectories. It is clearly that our method can make use of more points in each pair of frames. The number of utilized feature points remains stable with growing length of sequences in our method, while it decreases sharply for trajectories.

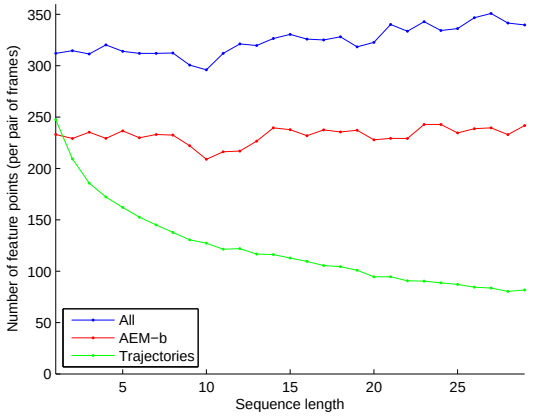


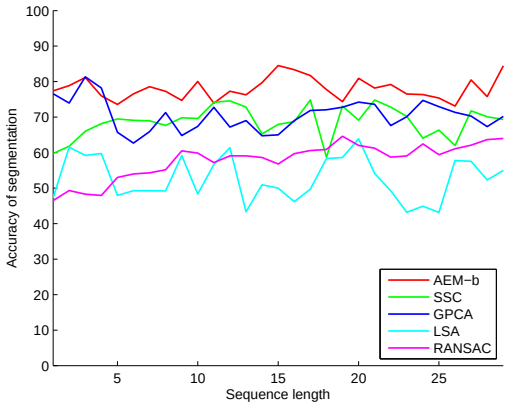
Figure 3: Number of feature points used in different methods.

Figure 4a shows the segmentation accuracy of all methods with respect to all trajectories of the specified sequence lengths. Figure 4b shows the segmentation accuracy with respect to all feature points. Because the comparison methods are all using the trajectories as inputs, their segmentation accuracies w.r.t. all feature points decrease when sequences getting longer. From the Figure 4a and 4b, we can draw the following conclusions:

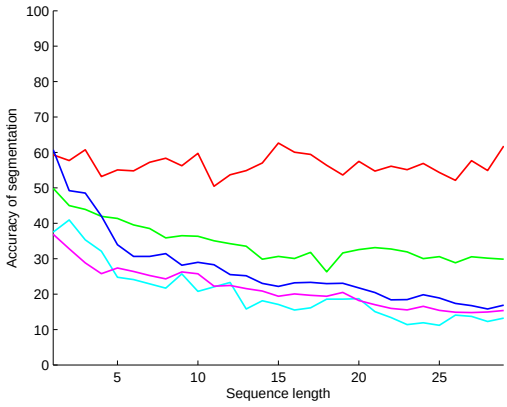
- For the original videos, our method provides a solution including the feature detection, motion estimation and segmentation, which can make use of more feature points. The other methods require a separate step of building trajectories, which will lead to a loss of feature points.
- Our method can achieve higher accuracy of segmentation in the videos from CNet and robocup 2014, where the movement of feature points are not as accurate as them from the Hopkins155 dataset. For the latter one, the movements of features points are detected by a special tracker.
- Our method can always make use of the most feature points even the length of sequence increases. More feature points will be helpful to improve the accuracy of object recognition in the next step

5.3 Computing time

Table 4 and 5 give the average computing time for sequences with a length of 30 frames for different dataset.



(a) Accoracy of segmentation.



(b) Accuracy of segmentation w.r.t. all feature points.

Figure 4: Accuracy curves w.r.t. lengths of sequences, compare to (a) the utilized points (b) all feature points.

Although RANSAC and GPCA have the lowest computation times, their segmentation accuracy is also lower. Moreover, the performance of RANSAC is not stable as mentioned in Section 5.1. Our method has an average computation time of 0.3s, which is smaller than LSA and SSC.

	LSA	RANSAC	GPCA	SSC	AEM-b
Number of points			330		
Hopkins155	4.32s	0.09s	0.14s	3.8s	0.31s

Table 4: Computing time (seconds per 30 frames) of segmentation stage of Hopkins155 dataset.

	LSA	RANSAC	GPCA	SSC	AEM-b
Number of points	78	78	78	78	235
CNnet	0.94s	0.01s	0.04s	0.68s	0.19s
RobotCup	0.91s	0.01s	0.04s	0.66s	0.20s

Table 5: Computing time (seconds per 30 frames) of segmentation stage of CNet and robocup2014 videos.

	LSA	RANSAC	GPCA	SSC	AEM-b
Hopkins155	13.0	0.27	0.42	11.4	0.9
CNnet	12.1	0.19	0.51	8.72	0.8
RobotCup	11.7	0.19	0.51	8.46	0.9
all	12.2	0.2	0.5	9.5	0.9

Table 6: Computing time (ms per point per 30 frames) of segmentation stage.

For experiment two, we only consider the computation time of the segmentation stage, which means the computing time of feature detecting and motion estimation is not taken into consideration.

In Figure 3 we can see that the average number of points utilized in trajectory clustering for a 30 frame sequences is about 80, while it is about 240 in our method. That means our method processes three times more points compared to the other methods in this experiment. Nevertheless, our method is faster than SSC and LSA. Table 6 shows the average computation time per feature point. Taking the difference in the number of feature points in to account, our method is ten times faster than SSC, fourteen times faster than LSA, three times slower that RANSAC, and two times slower than GPCA.

6 CONCLUSION

We proposed an approach for segmenting video frames into groups of feature points based on their motions. In the proposed method, SIFT feature points and their movements are detected using Lowe's algorithm [Low04], an adapted EM algorithm is applied with a recursive division strategy for segmenting the feature points according to their motions. The segmentation is iteratively applied for each pair of frames in the sequence, and combined with Bayesian update to generate segmentation results over all frames. The characteristics of our method are as follows

- Because our method processes pairs of frames iteratively, it can deal with arbitrary length of video sequence.
- The EM algorithm with a division strategy can determine the number of moving objects in the frames.
- Bayesian update combines the results of a sequence of frames.
- Our method can handle the problem of missing points in any frames, because it does not track feature points over sequence of frames. We only consider the feature points in neighbouring frames in each step of the segmentation.

Results shows that our method performs well in trajectory segmentation, and has an average accuracy of 92.1% in general. It is especially successful for videos of translation. However, it performs not well the displacements of objects are small compared to the displacement caused by the moving camera.

Our approach does not require that all trajectories of feature points have the same length, which means that it can deal with the data with missing points. This property makes our approach more flexible than other approaches.

Experiments also show that the computational cost of our method is reasonable. On the one hand, it performs better than the methods which are faster. On the other hand, it is ten times faster than the methods perform better (actually only the SSC) in the segmentation stage giving the trajectories of feature points (provided by Hopkins155 dataset). In general, our method proposes an efficient way to deal with motion segmentation of video sequences in a dynamic environment.

The first drawback of our method is that it can not deal very well when the movement of camera is significant compare with the movements of the objects. Secondly, our method does not consider the position relationships of points, so some points being far away from an object but having similar movements will be misclassified, which is not a big problem for SSC. Thirdly, the performance of our method drops too much when the number of moving objects increases, compare to the best one (SSC).

In the future, we will do more experiments to evaluate the robustness of our methods in varying conditions. The motion model should be made more robust for camera movements. Exploring whether different types of feature points influence the segmentation is also worth investigating. Last but no least, we will investigate its applicability in real time for mobile robots.

7 REFERENCES

[BBAT97] Georgi D. Borshukov, Gozde Bozdagi, Yucel Altunbasak, and A. Murat Tekalp. Motion segmentation by multi-stage affine classification. *IEEE Trans. Image Processing*, 6:1591–1594, 1997.

[CSSF07] Eliete Maria de Oliveira Caldeira, Hans Jörg Andreas Schneebeli, and Mário Sarcinelli-Filho. An optical flow-based sensing system for reactive mobile robot navigation. *Sba: Controle & Automação Sociedade Brasileira de Automatica*, 18(3):265–277, 2007.

[DLR77] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[EV09] Ehsan Elhamifar and René Vidal. Sparse subspace clustering. In *Computer Vision and*

Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 2790–2797. IEEE, 2009.

[FB81] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[FGMR10] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

[FPZ03] Robert Fergus, Pietro Perona, and Andrew Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–264. IEEE, 2003.

[HFH07] Shih-Shinh Huang, Li-Chen Fu, and Pei-Yung Hsiao. Region-level motion-based background modeling and subtraction using mrfs. *Image Processing, IEEE Transactions on*, 16(5):1446–1456, 2007.

[JS04] Boyoon Jung and Gaurav S Sukhatme. Detecting moving objects using a single camera on a mobile robot in an outdoor environment. In *International Conference on Intelligent Autonomous Systems*, pages 980–987, 2004.

[JT12] Kinjal A Joshi and Darshak G Thakore. A survey on moving object detection and tracking in video surveillance system. *IJSCE, ISSN*, pages 2231–2307, 2012.

[Low99] David G Lowe. 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, 1999.

[Low04] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[PN05] Zailiang Pan and Chong-Wah Ngo. Selective object stabilization for home video consumers. *IEEE Trans. Consumer Electronics*, 51(4):1074–1084, 2005.

[POP98] Constantine P Papageorgiou, Michael Oren, and Tomaso Poggio. A general framework for object detection. In *Computer vision, 1998. sixth international conference on*, pages 555–562. IEEE, 1998.

[RCTdC+12] Gonzalo R Rodríguez-Canosa, Stephen Thomas, Jaime del Cerro, Antonio Barrientos, and Bruce MacDonald. A real-time method to detect and track moving objects (datmo) from unmanned aerial vehicles (uavs) using a single camera. *Remote Sensing*, 4(4):1090–1111, 2012.

[SI84] Shokri Z Selim and Mohamed A Ismail. K-means-type algorithms: a generalized convergence theorem and characterization of local optimality. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (1):81–87, 1984.

[SM98] Jianbo Shi and Jitendra Malik. Motion segmentation and tracking using normalized cuts. In *Computer Vision, 1998. Sixth International Conference on*, pages 1154–1160. IEEE, 1998.

[SR13] Gurjeet Kaur Seerha and Kaur Rajneet. Review on recent image segmentation techniques. *International Journal on Computer Science and Engineering (IJCSE)*, 5:109–112, 2013.

[SWY+09] Ju Sun, Xiao Wu, Shuicheng Yan, Loong-Fah Cheong, Tat-Seng Chua, and Jintao Li. Hierarchical spatio-temporal context modeling for action recognition. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2004–2011. IEEE, 2009.

[VH04] René Vidal and Richard Hartley. Motion segmentation with missing data using powerfactorization and gpca. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–310. IEEE, 2004.

[WA94] John YA Wang and Edward H Adelson. Representing moving images with layers. *Image Processing, IEEE Transactions on*, 3(5):625–638, 1994.

[WJP+14] Yi Wang, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, Yannick Benezeth, and Prakash Ishwar. Cdnet 2014: An expanded change detection benchmark dataset. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 393–400. IEEE, 2014.

[WKSL13] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *International journal of computer vision*, 103(1):60–79, 2013.

[YP06] Jingyu Yan and Marc Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *Computer Vision–ECCV 2006*, pages 94–106. Springer, 2006.

[ZR16] W Zhao and N Roos. Motion based segmentation for robot vision using adapted em algorithm. In *Proceedings of the 11th International Conference on Computer Vision Theory and Applications (VISIGRAPP 2016)*, pages 649–656, 2016.