# Motion Matrix Classification for Moving Camera Segmentation and Tracking

Isaac Zhang

College Engineering & Comp Sci
Australia National University
u7344258@anu.edu.au

## Abstract

The previous moving object segmentation method usually uses motion vectors and optical flow to describe motions. Motion vectors perform great for motions in a 2 dimensional plane, and can be easily classified into foreground motions and background motions with clustering algorithms. However, this vector based approach performs bad as the actual motions in the video is much more complicated. Our proposed algorithm provides a novel way to use motion matrices which contains motion information for each sub-regions of the frame to describe motions. The motion matrix was obtained by comparing local feature descriptors between frames. More specifically, each descriptor is associated with a motion matrix compute with its nearest matched descriptors. Then each motion matrix can be classified with a simple neural network classifier. These motion matrices can be then used as motion cues for moving object segmentation and object tracking. Compared to the method by clustering motions obtained by the optic flow or motion vectors, our method is more robust as homography motion matrices containing motion information in 3D space and invariant to rotations. We conduct extensive experiments on DAVIS-16 dataset [1]. to evaluate the performance of classification, our approach on classifying motion matrices can achieve as high as 97%. And shows good performance on both segmentation and tracking tasks.

## 1 Introduction

Motion is an important and fundamental cue for the human vision system. Motion information is also considered to be fundamental to solve the tasks such as moving object segmentation and tracking. Moving object segmentation and tracking is a key step for computer vision applications such as autonomous driving vehicles. Another one of the most common applications of moving object segmentation is extracting vehicle information from traffic monitoring cameras. To best utilize the motion cues, previous approaches usually describe motion by motion vectors which have been widely used in motion segmentation. However, motions vector based approach is not as reliable for videos captured in freely moving cameras. The shape of moving objects can change greatly when moving along different aspects to the camera. In this situation, motion vectors and optical flow methods can not be a good choice to describe motions in the 3D space, no need to mention those motions with distortion or rotation.

To better describe motions for each object, we proposed to use transformation matrices to describe and analyze motion information of objects. Transformation matrix is commonly used in panoramic synthesis to describe the 3D transformation between two images. However, instead of capturing transformation between two images, we use it to describe each small sub-region of video frames. Based on this insight, a motion analyze framework is proposed and can be used in moving objects segmentation and tracking. The framework can be divided into two parts. First, the local features

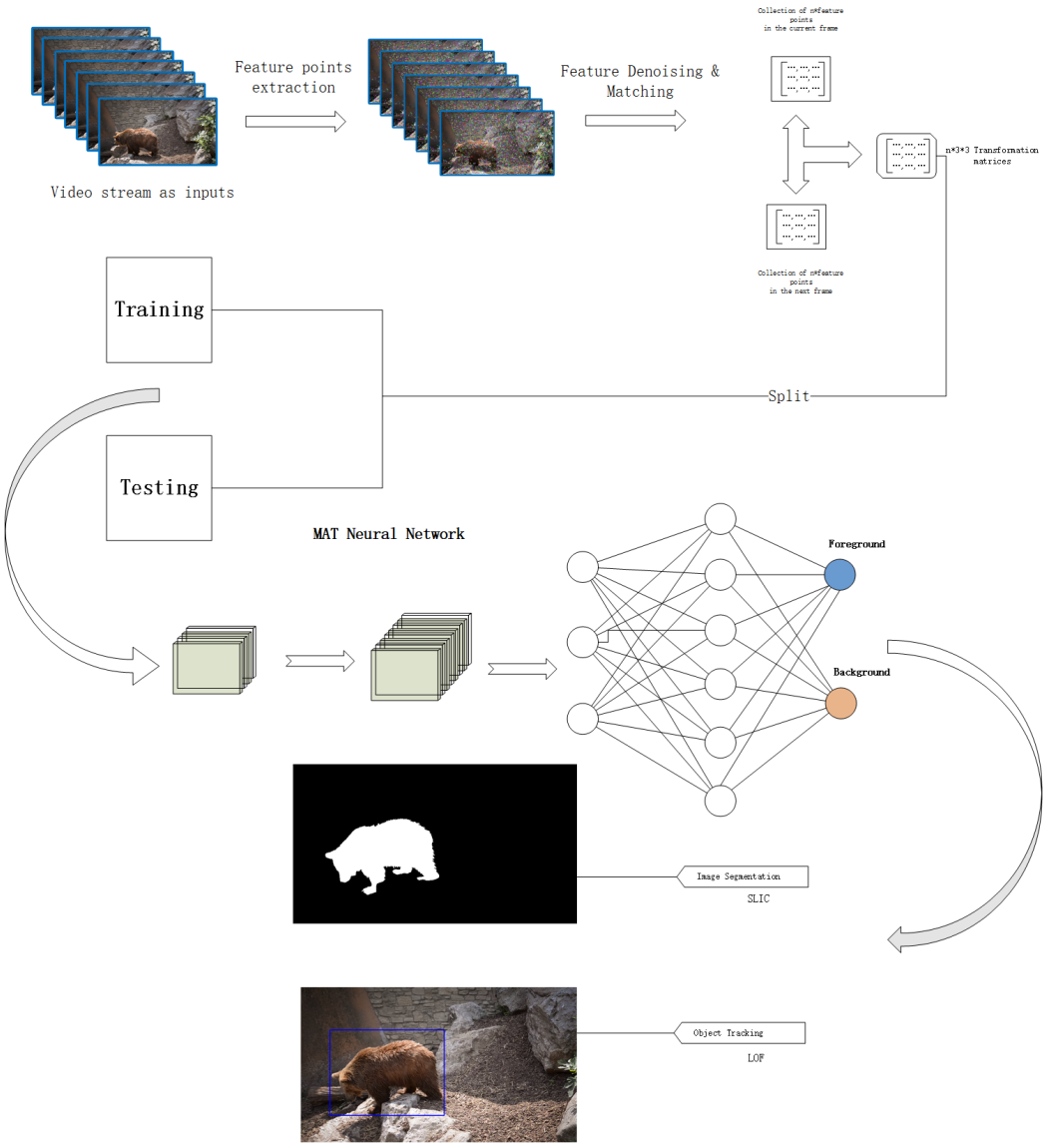


Figure 1: This flow chart shows the main process to compute and classify motion matrix, and how it can be used for moving object segmentation and tracking.

are obtained by feature detection algorithms such as SIFT, SURF, and ORB. The transformation matrices can be captured between the local descriptors from two frames. Each feature descriptor can associate with a transformation matrix that is computed by this feature descriptor and its nearest few descriptors. Since these transformation matrices contain complete information about the motion for classification. A classifier is then proposed to classify these matrices. In particular, a deep learning network designed specifically for homography matrices is employed to train this classifier.

After getting the prediction from the classifier, we are able to predict which location of the local descriptor contains foreground motions and which are not. Based on this information, we are able to use the SLIC superpixel algorithm to utilize semantic information of each frame to do moving object segmentation. As for tracking, the LOF algorithm is used for outlier detection to get accurate bounding boxes prediction. We conduct extensive experiments for both segmenting and tracking on DAVIS-16 datasets to evaluate the performance of classification. We proved that this method is valid under different scenarios, and even for the condition that the classifier was trained and tested on totally different scenes. Our work shows a brand new way to describe and utilize motion information with high robustness with complicated motions that traditional methods can fail.

## 2 Related Works

Moving object segmentation is a challenging task that has been widely used in many different areas of computer vision, such as motion estimation, semantic segmentation and scene modelling. In recent years, there is a significant performance improvement for moving object segmentation and a large number of methods have been proposed. We classified these methods into the following three different categories, and a brief discussion is presented.

### 2.1 Traditional Methods

Traditional methods often involve motion estimation with the utilization of optical flow or geometric reconstruction. For example, Bideau et al. [2] proposed to train a motion segmentation network in a self-supervised manner. It is a combination of traditional optical flow and neural network based approaches. In partciular, motion segmentation task is broken down into two parts: (1) modifying the flow field to remove the observer's rotation and (2) segmenting the rotation-compensated flow into static environment and independently moving objects. Besides optical flow based approaches, Lin et al. [3] proposed a graph-based segmentation method which adopts both local and global motion information encoded by the tracked dense point trajectories, and achieves high accuracy on trajectory clustering. However, when the appraoch is applied into videos including fast motion, motion blur and occlusions pose, the accuracy of optical flow limits the performance of their method. To address it, Hu et al. [4] proposed a novel saliency estimation technique as well as a novel neighbourhood graph, based on optical flow and edge cues. However, most approaches performs not as good as on small moving target or target at distance.

### 2.2 Deep Learning Based Approaches

In recent years, deep learning network has become popular for moving objects segmentation. The tracking branch mainly leverages the cue of appearance similarity. In addition, scholars reduce noises on images to improve accuracy on appearance cues. For instance,Lee et al. [5] adds a novel spatial attention-guided mask (SAG-Mask) branch to anchor-free one stage object detector in the same vein with Mask R-CNN. They Plugged into the FCOS object detector, the SAG-Mask branch predicts a segmentation mask on each detected box with the spatial attention map that helps to focus on informative pixels and suppress noise. Moreover,to save human labor, a lot of unsupervised learning or semi-unsupervised(use the first frame only) are proposed [6–8]. More specifically, Lu et al. [9] propose a unified unsupervised/weakly supervised learning framework, called MuG, that comprehensively captures intrinsic properties of VOS at multiple granularities. This approach can help advance understanding of visual patterns in VOS and significantly reduce annotation burden. However, since a lot of details will be ignored during learning, an end-to-end network has been proposed. Patil et al. [10] proposed a multi-frame multi-scale encoder-decoder network for MOS(Moving object segmentation). The proposed network takes video frames and optical flow as inputs to learn the inherent correlation between multi-scale encoder features of three successive frames. In contrast, time-costing comes with accuracy, therefore, Deep NN architecture method can not avoid long training time and exhausted labeling if we want to prevent accuracy.

### 2.3 Clustering Based Approaches

Since classification plays a big role in NN and our approach relies very much on clustering, we need to find something promising and visible instead of doing training in a Blacbox, even though clustering Via Neural net work is popular nowadays [11–13]. Normally, clustering algorithms can be broadly applied to distance classification, However, that does not suit our proposed algorithm because we can hardly define distance of matrices. Reconsider the clustering task from its definition to develop Deep Self-Evolution Clustering to jointly learn representations and cluster data. For this purpose, the clustering task is recast as a binary pairwise-classification problem to estimate whether pairwise patterns are similar [14].

## 3 Homography Matrices Classification Approach

The details of our proposed method will be explained in this section. Components of our algorithm are feature points detection, computation of Transformation matrices, training classification on matrices with our neural network model, respectively.

### 3.1 Find appropriate feature points to form a coordinate matrix.

In pictures, every single frame is remaining rich information. And feature points technology is widely applied because they contain information about the content of an image. In addition, a stream can be splitting into a few frames, which means we can forward apply feature points in videos. Since we can detect feature points on every single images, tracing feature points reveal sufficient information changes within frame difference. Hence, in order to find motion patterns, we will try to match the same feature points among frames. There are many ways to generate feature points, such as Harris, Scale-invariant feature transform (SIFT). In this paper, we extract feature points on Images by Speeded Up Robust Features (SURF) (1)/ Oriented FAST and Rotated BRIEF (ORB). During our points extraction of experiments, SURF provides robustness, while ORB serves as efficiency boosters. Once we find all feature points initially, a rough filtering processing are imported. We will eliminate significant drifting points, that means we discord points if their matches are obvious wrong within two framesits best match are almost the same to it second best match. After gathering robust points, we set a feature point and its neighbors feature points to form a coordinate matrix. Moreover, as shown in fig1, we can dynamically filter out some ambiguous matching among feature points before forming matrices. That means we will retrieve multiple-matches of each point since, then delete vague points if distance of their best match and the second match are similar enough.

$$H = \begin{pmatrix} L_{XX} & L_{XY} \\ L_{XY} & L_{YY} \end{pmatrix} \Rightarrow \det H = L_{XX}L_{YY} - 0.9L_{XY}^2 \tag{1}$$

### 3.2 Retrieve accurate Transformation matrices which contains motion information among two frames.

Motion can be perfectly represented in matrix. As showing in Equations (2), we can use a transformation matrix which contains scalar rotation and translation to describe a point' motion behavior. Hence, we can find multiple holography matrices within two frames since we clearly know coordinate of each feature points. As demonstrated in Equations (3-5), to get a homograph matrix, at least the coordinate of three points is required. However, at the meantime, we have a N-by-N coordinate matrix, which increases robustness and accuracy significantly rather than a solvable answer. Moreover, our algorithm is flexible. When a matrix (formed in A) cannot find the Homograph matrix among two frames, we dynamically add N number feature points to expand this coordinate matrix to re-compute the transformation matrix. Until we retrieve a transformation matrix, we stop enlarging the current coordinate matrix and start another retrieve for the next coordinate matrix.

$$H = s \begin{bmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \sim \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{3}$$

$$\begin{aligned} x' &= \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \\ y' &= \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}} \end{aligned} \tag{4}$$

$$\begin{aligned} h_{11}x + h_{12}y + h_{13} - h_{31}xx' - h_{32}yx' - h_{33}x' = 0 \\ h_{21}x + h_{22}y + h_{23} - h_{31}xy' - h_{32}yy' - h_{33}y' = 0 \end{aligned} \tag{5}$$

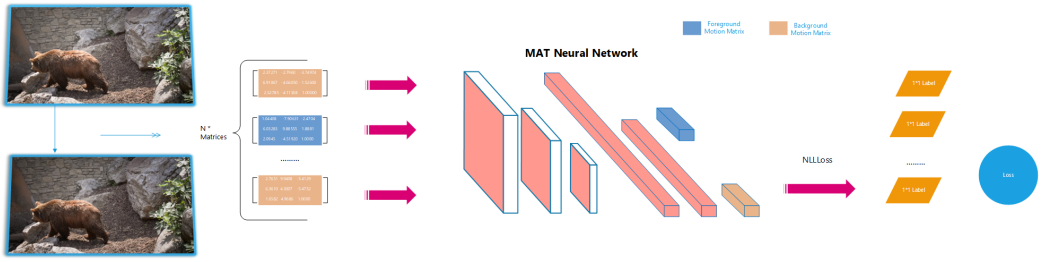


Figure 2: Illustration of our Algorithm. Motion matrices within two frames are generated from feature points detection and match within every two frames. Then each motion matrix will be predicted whether it belongs to the foreground or is most likely background. Thus each feature point will be assigned a label (each matrix corresponds to each feature point).

### 3.3 Segmentation and Tracking & Promising Results

Since these transformation matrices contain enough information for Classification, a classifier is then proposed to classify these matrices. There are sufficient feature points in each frame, which means we got plenty of data of transformation matrices. Our neural network model only takes 5% percentage of total data as input. That means, we take motion matrices as inputs per 20 frames; all data will be splitted into the training set and validation set, which takes parts of 5% and 95%, respectively. And our results have proven that determining motions via Matrices Classification in a super shallow model is competent enough. As in fig1, we designed shallow neural networks performs well, which contains merely two Convolutional layers and three fully connected layers. We creatively use motion matrices to separate foreground and background motions, which is rotation invariant and more robust than traditional vectors because Homographic matrices contain much more information than other descriptors. So far, we have proved we can apply our algorithm not only works in moving object segmentation, but also to any CV issues that need motions, such as object tracking. We can predict any unseen videos with our trained model. In our experiments, we apply it in image segmentation and moving object tracking with additional technologies. **Moving Object Segmentation**: In order to get binary Mask, we import the SLIC (Simple Linear Iterative Clustering) to split each frame into patches. Once we have enough foreground feature points predicted through our trained model in a patch, we draw this patch (here we introduce a hyper parameter that is a threshold for drawing). Otherwise, the program will continue looping and checking to the next patch until it retrieves all patches. **Object Tracking**: Before getting the bounding box, we will filter out a few mispredicted foreground points, or out-liners. The LOF algorithm provides quick deletion on out-liners. Once we have promising foreground feature points, we set coordinate by MIN(x1,y1){(x1,y1) $\in$ foreground points} and MAX(x2,y2){(x2,y2) $\in$ foreground points}. Finally, draw a bounding box in each frame. In addition, our method is robust, as shown in fig. experimental results (3). We tested our algorithm on the whole DAVIS data set. We can easily handle many edge cases. For examples, in video mallard-water, disturbing of water around the foreground are mainly ignored, as same as video kite-walk. Besides that, since feature points detection does not depend on brightness, yet it is Scale-invariant, we can ensure our motion matrices are within a high-level confidence interval. Moreover, we can generalize a mixed model to predict an unseen video, which means we train our model with multiple mixed motion matrices to make sure the Neural network fully understands differences between foreground motions and background motions. For each single Video model, our Testset Accuracy can reach at around 87-95% on average. However, we can mix all motion matrix from different video together, we can get a mixed model that can even predict an unseen video.

Table 1: Details of our network architecture, which consists of 2 convolutional layers, 3 linear layers and a Log softmax operator.

| Layer type | Parameters | Layer size | Output Shape |
|---|---|---|---|
| 2D Convolution | 5,120 | $3 \times 3 \times 512$ | $1 \times 1 \times 512$ |
| 2D Convolution | 525,312 | $1 \times 1 \times 2014$ | $1 \times 1 \times 1024$ |
| LeakyReLU | 0 | | $1 \times 1 \times 1024$ |
| Linear | 524,800 | 512 | 512 |
| Linear | 131,328 | 256 | 256 |
| Linear | 514 | 2 | 2 |
| LogSoftmax | 0 | | 2 |

# 4 Experiments

Experiments on different videos of Davis-16 Dataset [1].

We implement our movement tracking task approach for different proportions labeled/unlabeled data(use Pre-trained model to predict a total unseen video). The Davis dataset consists of 50 videos at approximately 70-80 frames each (version of 1080P), which the moving camera films all videos. Among them, there are a few edge cases, such as Surfing/Auto racing. Because of waves, the former factor will cause some feature points to be accidentally detected, which may mislead the H matrix we find. The latter is due to motion blurring, feature points cannot be extracted well in this case, but deblurring on the data preprocessing can solve this problem. Overall, our algorithm work on car-sets the best since their feature points are balanced and sparsed spread on foreground.

## 4.1 Further Potential Improvement

Other than motion itself, we can expand our H matrix into 3 by 5 matrix, which $4^{th}$ colum means the coordinates of the current feature point and the $5^{th}$ row is the coordinates of the mapped point corresponding to it. By adding this linear transformation information, the accuracy of model prediction can increase about 3% for each video.

Moreover, there is another clue often used in either transitional method or Deep Learning, which is the RGB value, this is not the best option for our case. Even though adding information like RGB could improve single video performance again(another 5% increasing), RGB pattern could severe impact unseen video leads to bias of Generalized Predictability.


Figure 3: Experimental Results. The third and fourth row are predicted through our trained model. The mixed model performs good in most cases, as well as water/smoke disturbed videos

## 5 Conclusion

In this paper, we proposed motion matrices classification to split foreground and background motions. The proposed algorithm gives a brand new thought to the CV Fields. Since we have proved the unseen video can be predicted, it may be imported in further applications if we feed the model enough motion matrices. Specifically, when we apply the proposed algorithm into a video that has plenty amount of feature points, the accuracy is obviously higher than the others. In contrast to traditional methods such as motion vector clustering, motion matrices classification has additional information with higher dimension expression. Thus, if we can retrieve all information from matrices analysis, the CV field would have significant improvements with no doubt.

## References


[1] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016.

[2] Pia Bideau, Rakesh R Menon, and Erik Learned-Miller. Moa-net: self-supervised motion segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[3] L. Chen, J. Shen, W. Wang, and B. Ni. Video object segmentation via dense trajectories. *IEEE Transactions on Multimedia*, 17(12):2225–2234, 2015.

[4] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G Schwing. Unsupervised video object segmentation using motion saliency-guided spatio-temporal propagation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 786–802, 2018.

[5] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13906–13915, 2020.

[6] Wenguan Wang, Jianbing Shen, Fatih Porikli, and Ruigang Yang. Semi-supervised video object segmentation with super-trajectories. *IEEE transactions on pattern analysis and machine intelligence*, 41(4):985–998, 2018.

[7] Fei Pan, Inkyu Shin, Francois Rameau, Seokju Lee, and In So Kweon. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3764–3773, 2020.

[8] Peng Tang, Xinggang Wang, Song Bai, Wei Shen, Xiang Bai, Wenyu Liu, and Alan Yuille. Pcl: Proposal cluster learning for weakly supervised object detection. *IEEE transactions on pattern analysis and machine intelligence*, 42(1):176–191, 2018.

[9] Xiankai Lu, Wenguan Wang, Jianbing Shen, Yu-Wing Tai, David J Crandall, and Steven CH Hoi. Learning video object segmentation from unlabeled videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8960–8970, 2020.

[10] Prashant W Patil, Kuldeep M Biradar, Akshay Dudhane, and Subrahmanyam Murala. An end-to-end edge aggregation network for moving object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8149–8158, 2020.

[11] Jiahui Huang, Sheng Yang, Tai-Jiang Mu, and Shi-Min Hu. Clustervo: Clustering moving instances and estimating visual odometry for self and surroundings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2168–2177, 2020.

[12] Andrés Hoyos-Idrobo, Gaël Varoquaux, Jonas Kahn, and Bertrand Thirion. Recursive nearest agglomeration (rena): fast clustering for approximation of structured signals. *IEEE transactions on pattern analysis and machine intelligence*, 41(3):669–681, 2018.

[13] Jufeng Yang, Jie Liang, Kai Wang, Paul L Rosin, and Ming-Hsuan Yang. Subspace clustering via good neighbors. *IEEE transactions on pattern analysis and machine intelligence*, 42(6):1537–1544, 2019.

[14] Yinan Yu, Weiqiang Ren, Yongzhen Huang, Kaiqi Huang, and Tieniu Tan. Clumoc: Multiple motion estimation by cluster motion consensus. In *2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance*, pages 172–177. IEEE, 2012.