Video Scene Understanding Using Multi-scale Analysis

Yang Yang
Computer Vision Lab
University of Central Florida
yyang@cs.ucf.edu

Jingen Liu
Computer Vision Lab
University of Central Florida
liujg@cs.ucf.edu

Mubarak Shah
Computer Vision Lab
University of Central Florida
shah@cs.ucf.edu

Abstract

We propose a novel method for automatically discovering key motion patterns happening in a scene by observing the scene for an extended period. Our method does not rely on object detection and tracking, and uses low level features, the direction of pixel wise optical flow. We first divide the video into clips and estimate a sequence of flow-fields. Each moving pixel is quantized based on its location and motion direction. This is essentially a bag of words representation of clips. Once a bag of words representation is obtained, we proceed to the screening stage, using a measure called the 'conditional entropy'. After obtaining useful words we apply Diffusion maps. Diffusion maps framework embeds the manifold points into a lower dimensional space while preserving the intrinsic local geometric structure. Finally, these useful words in lower dimensional space are clustered to discover key motion patterns. Diffusion map embedding involves diffusion time parameter which gives us ability to detect key motion patterns at different scales using multi-scale analysis. In addition, clips which are represented in terms of frequency of motion patterns can also be clustered to determine multiple dominant motion patterns which occur simultaneously, providing us further understanding of the scene. We have tested our approach on two challenging datasets and obtained interesting and promising results.

1. Introduction

The standard approach for analysis of video sequences involves the detection of objects of interest (which are mainly moving objects); classification of objects into different categories (e.g. a car, a person); tracking of such objects from frame to frame; and recognition of behavior or activities performed by the objects. There has been a lot of progress made in each of the modules in the above pipeline, and complete end to end systems have even been developed for automatic video surveillance and monitoring (e.g. KNIGHT [12] for fixed camera surveillance and monitoring, and COCOA [13] for UAV video analysis).

Several attempts have been made to model and learn a scene. In general, scene understanding may involve, understanding the scene *structure* (e.g. pedestrian sidewalks, east-west roads, north-south roads, intersections, exits and eateries), scene *status* (e.g. traffic light status, traffic jam), scene *motion patterns* (e.g. vehicles making u-turns, east-west traffic and north-south traffic), etc. With the knowledge of scene structure, activities and motion patterns, low-level tracking and abnormal activity detection (anomalous motion detection) can be improved. High-level activity analysis and video retrieval can be accomplished. Most of the previous work [1, 5, 14, 15] used object tracks to model the scene. In [5], scene activities are modeled and learned by observing the trajectories of objects observed by a static camera over extended periods of time. The motion patterns of the objects in the scene are modeled as a multivariate non-parametric probability density function of spatio-temporal variables. Kernel density estimation is used to learn this model in a completely unsupervised fashion. Wang *et al*. [1] used location, velocity, and size to classify activities. The activities are classified using a B-tree based approach called Numeric Iterative Hierarchical Clustering method and the co-occurrence statistics in the quantized feature space. In some of the other related works [15, 2, 3] multiple features of observed tracks are used for clustering tracks into the main paths of the scene.

The performance of all these methods is heavily dependent on the ability to detect, track, and classify moving objects. However, in a complicated or crowded scene, it is not possible to detect individual objects since the size of objects decreases with the density of crowd. Also, it is difficult to obtain reliable tracks, since tracks may be broken due to short and long-term occlusions, and contain errors due to clutter.

There are very few works have been done based on non-trajectories video scene understanding. Work done in [7] represented activities as bags of event n-grams that capture the global structure of an activity using its local event statistics. Xiang *et al.* [6] used DPNs to model complex activities of multiple objects in cluttered scenes. [8]

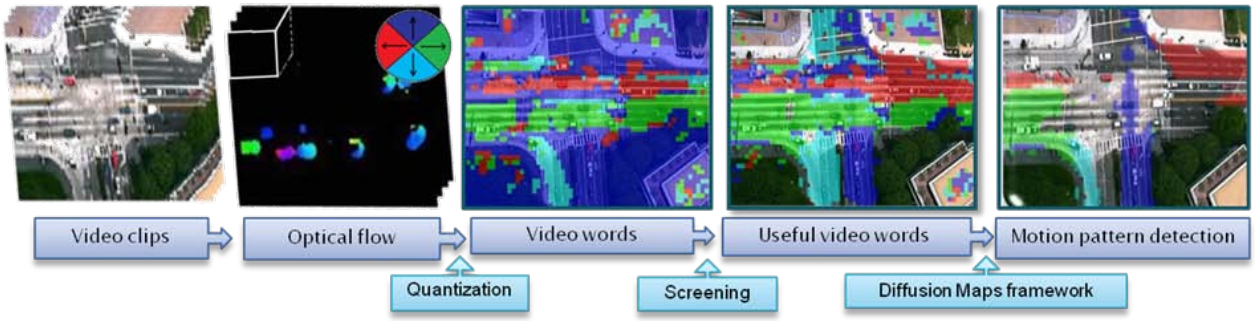


Figure 1: Flowchart of our approach. To extract key motion patterns, we first extract low-level motion features through computing optical flow. These motion features are then quantized into video words based on their direction and location. Next, some video words are screened out based on the entropy over all clips for a given word. Key motion patterns are discovered automatically using diffusion maps embedding and clustering.

improved LDA and HDP models to capture the co-occurrence of words and documents.

In this paper, we propose a novel framework which provides a principle way to automatically detect multi-scale key motion patterns in complex scenes. Key motion patterns essentially group pixels with similar motion into spatiotemporal regions, and can provide a coarse abstraction of a scene, which can be very useful for later processing such as tracking of an individual in a crowded scene, detection of an unusual behavior, and unsupervised learning and recognition of a particular activity or action. We follow a bottom-up approach, which starts with dividing a video into smaller clips, and compute pixel-wise optical flow between consecutive frames of each clip as our low level features. Flow vectors with small magnitude, which may be due to some noise, are removed and optical flow directions are quantized into four directions: North, South, East and West. A clip is divided into small special cuboids. The local histogram of four motion directions is then computed for each cuboid in the clip, where each local histogram has an associated implicit spatial location of the cuboid. These local optical flow direction histograms are concatenated into one long vector to represent a clip. Each quantized direction of a given cuboid is the smallest possible unit of representation of a clip, which is similar to a word in a document. The long vector representation of a clip is essentially a bag of those words (motion directions observed at specific locations in the scene). Since these vectors reside in a high dimensional space, we propose to use the diffusion map embedding [4, 9, 10] to embed them into more discriminative and compact manifolds, and to perform clustering in order to obtain motion patterns in the scene.

Diffusion maps (DM) framework embeds the manifold points into a lower dimensional space while preserving the intrinsic local geometric structure. The diffusion process begins by organizing the data points into a weighted graph, which is a meaningful way to represent the complex relationships between the feature points, where the weight between two feature points is the feature similarity. Once we normalize the weight matrix which is symmetric and positive, we can further interpret the pairwise similarities as edge flows in a Markov random walk on the graph. In this case, the similarity is analogous to the transition probability on the edge. Then utilizing the spectral analysis on the Markov matrix of the graph, we can find the dominant α eigenvectors as the coordinates of the embedding space and map the feature points to the low dimensional space while preserving their local geometric structures.

One of the advantages of DM is that the definition of the weight is totally application driven. It can measure the similarities between low-level features based on the combination of different semantic information, whereas [8] only capture the co-occurrence similarity. In addition, by adjusting the diffusion time t of the Markov chain, DM can be also used to employ multi-scale analysis on the scene data. If we consider the embedding process as clustering, DM embeds the semantically similar features into the same cluster. The size of the cluster is determined by the diffusion time. A larger diffusion time corresponds to a bigger cluster, which means a larger group of correlated motion patterns. For instance, in our case, motion pattern "vehicle coming to the intersection and stopping from east to west" can be described as "vehicle moving along east-west road" at a larger scale, or as "east-west traffic" at a even larger scale. With the multi-scale data analysis, we can easily analyze the motion patterns in a scene at different scales. We believe DM is more suitable for scene understanding since the low-level features seem to have better manifold structure. Our results in this paper confirm this point. We show that the motion patterns can be automatically discovered at different scales from videos of the scenes.

2. Our Proposed Framework

In this section, we describe each step of our method in detail. Given a long video, our goal is to automatically detect and learn key motion patterns occurring there-in, and apply the learned model to clip categorization. Our framework is illustrated in Figure 1.

2.1. Low-level feature quantization

We first divide the video into clips and estimate a sequence of flow-fields. The optical flow between two neighboring frames is computed with each pixel denoted as $p = (x, y, v, \theta)$, where (x, y) is its spatial location, and (v, θ) is the optical flow magnitude and direction. A threshold on the magnitude of the pixels is used to remove pixels due to slight camera motion and variations in illumination. Each clip is split into spatiotemporal cuboids (3D patches of dimension NxNxL, where N by N patches at a given location in L frames of clips are used), and the motion of a moving pixel is quantized in four directions – North, South, East and West. For each cuboid in a clip, a 4 - bin histogram is computed. Each bin in a histogram of a given cuboid corresponds to one of the four motion directions at the location of the cuboid, and can be considered as a video word representing the clip. These local histograms for all cuboids in a clip are then concatenated into one long vector denoted by X. If the size of each image in a clip is (m x N) x (n x N), then the size of the vector X is m x n x 4 (m x n *is the number of cuboids)*, where *4* represents the directions of motion of the pixels. Each bin of X is a codeword, resulting in a codebook of size m x n x 4. This is essentially a bag of words representation of clips. Each pixel is assigned a word from the codebook, which specifies a rough location and motion direction. Once a bag of words representation is obtained, we proceed to the screening stage, using '*conditional entropy'*, which is described next.

2.2. Obtaining useful words

The frequency of each video word in different clips is summarized in a 2-D matrix (see figure 2). This matrix is normalized to obtain probabilities. The entropy over all clips for a given word is used as a measure to determine which words (elements) in the bag (vector X) are useful for motion pattern detection. The conditional entropy is defined as

$$H_{C|w} = -\sum_{c} p(c \mid w) \log p(c \mid w), \tag{1}$$

where

$$p(c \mid w) = \frac{n_{c,w}}{\sum_{\varepsilon} n_{\varepsilon,w}}, \tag{2}$$

and $n_{c,w}$ is the number of times word w appears in clip C.

Words with low entropy correspond to abnormal events such as a pedestrian crossing a street in a direction perpendicular to the flow of traffic in a no-crossing zone, and are not good descriptors of a normal scene. High entropy words can be equally uninformative since they are indicators of a static scene or noise in the optical flow, which were not discarded by the threshold used. Normal dynamic parts of the scene, such as roads, are therefore represented by words with intermediate entropy. We use this measure to our advantage in two ways. First, the words with intermediate entropy are retained and used to discover key motion patterns, while the rest are discarded. Second, the words with low entropy are helpful to detect abnormal behaviors.

Video words corresponding to the same kind of motion pattern often co-exist in video clips. To detect these motion patterns, we cluster video words based on their co-occurrence in the clips using the diffusion map embedding.

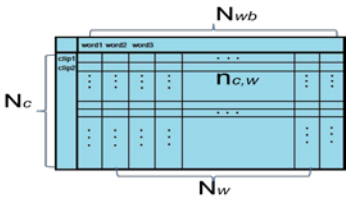


Figure 2 the 2-D matrix captures the frequency of each word for a given clip. $n_{c,w}$ is the number of times word w appears in clip c, N_c is the number of clips, N_{wb} represents the number of words before screening, N_w is the number of useful words.

2.3. Diffusion maps embedding

Clustering is a common technique for statistical data analysis, which is used in many fields, including machine learning, pattern recognition, and image analysis. Partitioning a large set of data points into homogeneous clusters is a fundamental operation in machine learning. Although some algorithms like k-means are well suited for clustering, the quality of the clustering is sensitive to the length of feature vectors and long feature vectors contain redundant information. Also, clustering criterion is typically based on appearance similarity, and, hence, k-means is unable to capture the co-occurrence relation between features. Besides, the affinity between feature vectors is measured using the Euclidean distance while most of the long feature vectors lie in manifold-space instead of a linear space. Therefore, we propose to use Diffusion maps embedding to handle high-dimensional data points contained in a non-linear manifold.

We first compute the Point-wise Mutual Information (PMI) between the clips c and words w using

$$m_{c,w} = \log\left(\frac{f_{c,w}}{\sum_{\varepsilon} f_{\varepsilon,w} \sum_{w} f_{c,w}}\right), \tag{3}$$

where $f_{c,w} = n_{c,w}/N_w$, $n_{c,w}$ is the number of times word w appears in clip c, and N_w is the number of useful words. We can then represent each word in terms of an N_c dimensional feature vector as

$$x_i = [m_{1,i}, m_{2,i}, \ldots m_{N_c,i}]'. \tag{4}$$

To find a low dimensional embedding, we first construct a graph **G** (**Ω**, **W**) with n (where n is number of words) nodes in set **Ω**, where **W**= $\{w_{ij}(x_i, x_j)\}$ is its weighted adjacency matrix that is symmetric and positive. The definition of **W** is totally application-driven, but it needs to represent the degree of similarity or affinity of two data points (words). We weight the distance between two nodes with a Gaussian kernel function, leading to a matrix with entries

$$w_{ij}(x_i, x_j) = e^{\frac{-\|x_i - x_j\|^2}{2\sigma^2}}, \quad (5)$$

where σ^2 indicates the variance of the Gaussian. This graph **G** (**Ω**, **W**) represents our knowledge of the local geometric relationships between the nodes of the graph. We then define a Markov random walk on the graph **G**, by treating the normalized edge weight as the transition probability between them. As a result, we form matrix $\mathbf{P}^{(1)} = \{p_{ij}^{(1)}\}$ by normalizing the matrix **W** such that its rows add up to 1.

$$p_{ij}^{(1)} = \frac{w_{ij}}{\sum_k w_{ik}}. \quad (6)$$

In other words, the quantity $\mathbf{P}^{(t)}$ reflects the intrinsic geometry of the dataset defined via the connectivity of the graph in a diffusion process and the time t of the diffusion plays the role of a scale parameter in the analysis.

We define the diffusion distance D using the random walk forward probabilities $p_{ij}^{(t)}$ to relate the spectral properties of a Markov chain (its matrix and its eigenvalues and eigenvectors) to the geometry of the data.

$$[D^{(t)}(x_i, x_j)]^2 = \sum_{q\in\Omega} \frac{(p_{iq}^{(t)} - p_{jq}^{(t)})^2}{\varphi(x_q)^{(0)}}, \quad (7)$$

where $\varphi(x_q)^{(0)}$ is the unique stationary distribution which measures the density of the data points. It is defined by $\varphi(x_q)^{(0)} = \frac{d_q}{\sum_j d_j}$, where d_q is the degree of node x_q defined by $d_q = \sum_j p_{qj}$.

The diffusion distance can be represented by the first α right eigenvectors (v_s) and eigenvalues ((λ_s^t) of matrix $\mathbf{P}^{(t)}$; we only need a few terms in the above sum for certain accuracy because of the decay of the eigenvalues:

$$[D^{(t)}(x_i, x_j)]^2 \simeq \sum_{s=1}^{\alpha} (\lambda_s^t)^2 (v_s(x_i) - v_s(x_j))^2,$$

where $\lambda_\alpha^t > \delta\lambda_1^t$.

Hence, we introduce diffusion map embedding and the low-dimensional representation is given by

$$\Pi_t : x_i \mapsto \{\lambda_1^t v_1(x_i) \;\; \lambda_2^t v_2(x_i) \ldots \lambda_\alpha^t v_\alpha(x_i)\}^T. \quad (8)$$

The link between diffusion maps and distances can be summarized by the spectral identity

$$\|\Pi_t(x_i) - \Pi_t(x_j)\|^2 = [D^{(t)}(x_i, x_j)]^2, \quad (9)$$

which means that the diffusion map embeds the data into a Euclidean space in which the Euclidean distance is equal to the diffusion distance in the original space.

2.4. Motion patterns detection at different scale

After embedding, the data points reside in much lower dimensional and meaningful manifold space and can be clustered to obtain relevant motion patterns. From a data analysis point of view, the reason for studying this Markov chain is that the matrix P contains geometric information about the data set **Ω**. Indeed, the transitions that it defines directly reflect the local geometry defined by the immediate neighbors of each node in the graph of the data. One of the main ideas of the diffusion framework is that running the chain forward in time, or equivalently, taking larger powers of P, will allow us to integrate the local geometry and therefore will reveal relevant geometric structures of **Ω** at different scales.

So, different values of σ and t can be used to perform different embeddings and motion patterns at different scales can be discovered. Therefore, in contrast to traditional methods, our proposed approach is capable of detecting motion patterns at different scales, through multi-scale analysis. The proposed multi-scale analysis is reminiscent of multi scale edge detection using different values of σ in Gaussian smoothing. Multi scale analysis is prevalent in many areas. For instance, in NLP (Natural Language Processing), "*sport*" is on a larger scale than "*baseball*" and "*football*", and "*baseball*" is on a larger scale than "*team*".

We employ and validate the multi-scale analysis both on our synthetic dataset and on real scene datasets. Results of real scenes is shown and illustrated later in the experiments section. The results of the synthetic dataset are shown in figure3.

We generated a swiss roll of 1000 points in 2-D. From this set, we built a graph using equation 5, where $\sigma = 2$, and formed the corresponding Markov matrix P. In the figure we plot results obtained using two powers of the matrix P, namely $t = 2$ and $t = 9$. For $t = 2$, the set appears to be made of four distinct clusters. For $t = 9$, the two closest clusters have merged, and the dataset appears to be made of only two clusters. From a random walk point of view, the key idea in this example is that a cluster is a region for which the probability of escaping this region is low. The higher the value of t, the higher is the probability that a data point can be diffused with other points which are further away from it.

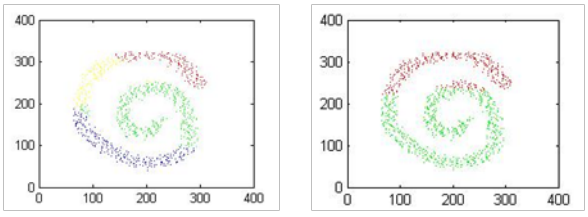


Figure3: Multi-scale analysis on our synthetic dataset.
Left: when t=2, σ=2, there are 4 clusters; Right: increasing t to 9, four clusters merge into 2 clusters.

3. Experimental results and discussion

We use two challenging traffic datasets: the NGSIM dataset [11] and the far-field traffic scene dataset from [8]. Both of them contain multiple motion patterns and also include illumination changes, occlusions, a variety of object types, and different environmental effects.

3.1. Experiments on NGSIM dataset

We first employ our method on the NGSIM traffic scene dataset, which is almost 19 minutes long. The video is divided into 224 clips. Each clip is 5 seconds long with image size of 420x600, and contains 50 frames. Each clip is split into 10x10x50 cuboids. The optical flow between consecutive frames is computed and quantized in to 4 directions. Therefore, our codebook size is 42x60x4.

Considering a real traffic scene, it is unlikely that a vehicle will drive from south to north on an east-west road. In other words, a video word whose position is on an east-west road, and whose motion direction is south rarely appears in the video. We can consider these words corresponding to rarely happening motion as abnormal words. According to information theory, these kinds of words are distinct and have high information content, which can be used for abnormality detection. Similarly, if a word always occurs in most of clips, it brings us little information to distinguish one clip from others. Therefore, both of the words mentioned above are useless for us to capture the normal meaningful motion patterns in the scene. To screen out useless words, we compute entropy over all clips for a given word and obtain 2280 useful words with entropy between $2<H<5$. Figure 4 shows the useful words after we discard most of the useless words applying the threshold on the entropy. The results are meaningful and coincide with our common sense understanding.

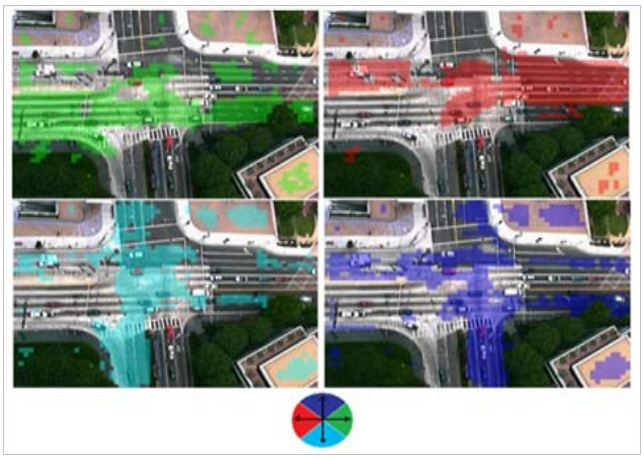
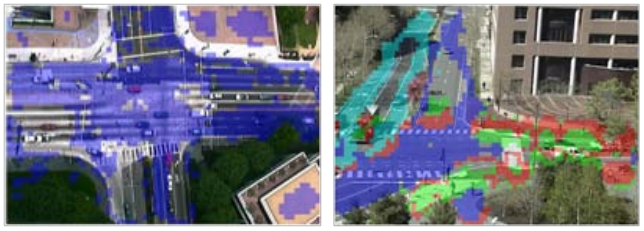


Figure 4: We show useful video words with 4 colors corresponding to four directions. Words with green, red, cyan and blue color correspond to vehicle movement from roughly west to east, east to west, north to south and south to north. Through this step, we discard most of the words which bring little information for key motion patterns detection in the video scene. Judging from the results, the screening method is meaningful and effective: Words in green and red (east, west direction) appear mostly on the east-west road; words in cyan and blue (north, south direction) appear mostly on the north-south road. However, there are still some words which are due to noise generated in optical flow that we cannot discard, due to the low resolution and low frame rate, and probably due to illumination changes in the NGSIM dataset.

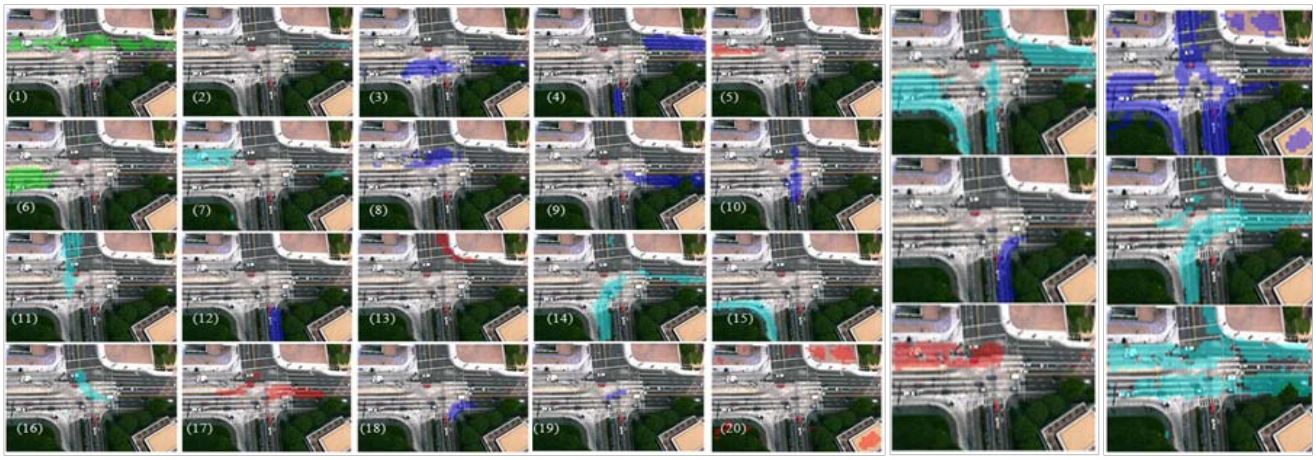


(a) (b)

Figure 5: motion patterns detected by using k-means clustering without embedding. We show one of the motion patterns with largest number of video words. Most of the video words are clustered into one group without capturing any semantic meaning. (a) NGSIM, k = 10. (b) Crowd scene, k = 15.

After screening we obtain 2280 words to represent each clip. Next, we cluster these words using the method described in section 2. Note that each word is represented by a 224 dimensional vector, where 224 is the number of clips in the video. As we described in the previous section, we put useful words into diffusion maps framework using $t = 2$ and $\sigma = 8$, and detected 30 motion patterns shown in figure 6(a). As illustrated, key motion patterns at a low level are detected, such as: Vehicle moving along west-east road unhindered, coming to the intersection and stopping, making left or right turn. These key motion patterns capture general movement and location of objects. E.g. 'vehicles moving from east to west' is split into 'vehicles crossing the intersection unhindered', 'vehicles coming to the intersection and stopping due to red light' and 'vehicles crossing the intersection but stopping because of the red light of next intersection'. Further, to verify that diffusion maps embedding does help to cluster words which may lie in a high dimensional space, we compare the motion pattern detected by using proposed diffusion maps approach and standard k-means directly. From figure 5 we can see that k-means cannot capture the meaningful motion patterns of video scenes, especially when the data lie in a high dimensional space.

In the next experiment, we detected the motion patterns at a larger scale using $t = 4$, and $\sigma = 6$. We obtain 10 motion patterns. Some motion patterns are merged together, e.g.: "vehicles moving from east to west unhindered" and "vehicles coming to the intersection and stopping" merge into "traffic from east to west"; the motion patterns corresponding to "vehicle on east-west road coming and stopping" and "vehicles moving along north-south road" are detected as a single motion pattern indicating "north-south



(a) (b) (c)

Figure 6: Automatic discovery of motion patterns at three different scales. (a) 20 motion patterns when t=2, σ=8. (1) — (9) are "horizontal" motion patterns. (1) and (3) are "vehicle moving through west-east road unhindered". (4) and (6) correspond to "vehicle moving along east-west road to intersection and stopping". (8) and (9) correspond to "vehicle proceeding after it stopped at the red traffic light". (5) and (7) correspond to "vehicle stopping to wait for traffic light after crossing the intersection along east-west". (10) and (11) correspond to "vehicle moving along north-south". (12)— (19) correspond to "vehicle making left or right turn". From the video, we find that the east-west road is crowded, whereas traffic flow of north-south road is small. This is consistent with our results since there are more east-west motion patterns compared to north-south. Also (17) shows that when vehicles from north-south make a right turn, there are always vehicles from south-north making a right turn at the same time. Figure (b) shows 3 of 10 patterns obtained by our method when t = 4, σ = 6, which are significantly different from the previous twenty. The pattern shown by cyan color ("vehicles going up north and vehicles in motion east-west, stopping to wait for light change, and some vehicles making right turn") is essentially combination of (4), (6), (10), (13), (15) from (a). The second pattern is combination of (12) and (18), and the third is the combination of (7) and (8) in (a). (c) Three motion patterns are detected by our method when t = 8, and σ = 5. The first corresponds to major traffic which is in the north-south direction. The second corresponds to vehicle approaching from north and making a left turn. We noticed that this motion pattern is quite frequent in the video. The third motion pattern corresponds to east-west traffic. Through this multi-scale analysis, we verify that it is possible to embed high dimensional data into different levels by using diffusion time t.

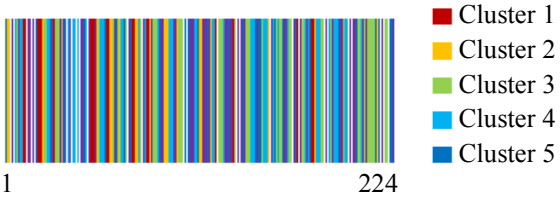


Figure 7: clip clustering result on NGSIM dataset. Each cluster is shown by different color.

traffic". This high level (or larger scale) motion pattern analysis makes sense because we are also able to cluster words by considering larger neighborhood due to large scale. If we increase t to 8 and σ to 5, we finally get 3 major motion patterns. Figure 6(b)(c) shows our results using multiple scales analysis.

Furthermore, we can also cluster clips and identify dominating motion patterns in each cluster. Note that since each clip was originally represented as a histogram of words, we can now represent a clip as a histogram of motion patterns (group of words). Results are shown in figure 7; our clustering accuracy is 90.1% according to the ground truth. From the clustering results, we can determine the dominant motion patterns of each cluster of clips as shown in figure 8.

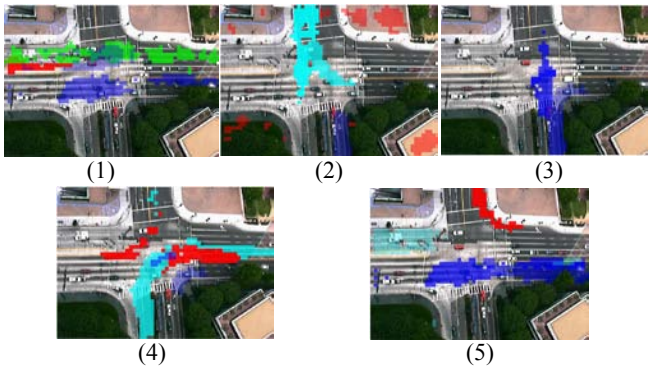


Figure 8: The main motion patterns in 5 clusters of clips in NGSIM dataset. (1) crowded traffic in the east-west direction; (2) vehicle moving from north to south and the same time vehicle making right-turn form south to north; (3) vehicle moving from south to north, some making right-turn; (4) vehicle making left-turn; (5) light traffic in the east-west direction.

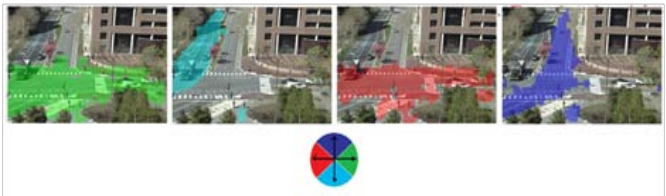


Figure 9: useful words in four directions.

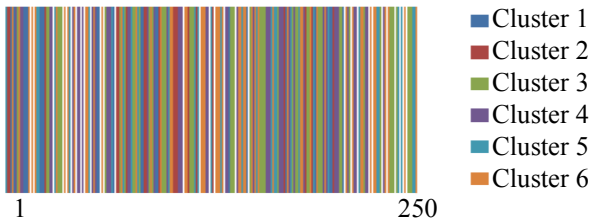


Figure 10: clip clustering result on crowd scene dataset. We show the first 250 clips of 549 clips. Each cluster is shown by different color.

3.2. Experiments on crowd scene dataset

We also tested our approach on the crowd traffic scene dataset. The dataset consists of 1.5-hours of video with 30 frames per second and a variable number of motion patterns. We divide the video into 549 clips of 10 seconds each, and quantize optical flow in the same way as the NGSIM dataset. The image size is 480 by 720, so our codebook size is 48*72*4. After the screening stage, we obtain useful words shown in figure 9. Figure 12 shows motion patterns detected at two different scales. Also, we cluster clips based on the key motion patterns. We show the results in figure 10, and the dominating motion patterns corresponding to each cluster of clips are shown in figure 11. Compared to our ground truth, we got accuracy of 86%.

4. Conclusion

In this paper, we propose a novel unsupervised approach for video scene understanding. We first quantize low-level features into words, then screen out useful words for motion pattern detection. We use diffusion maps framework to detect motion patterns at different scales. Based on clustering of clips, we can also identify dominant motion patterns occurring in each clip cluster. We tested our approach on two complicated video traffic datasets and obtained meaningful results.

References

[1] X.Wang, K. Tieu, and E. Grimson. Learning semantic scene models by trajectory analysis. ECCV, 2006.

[2] D.Makris and T. Ellis. Learning semantic scene models from observing activity in visual surveillance. Systems, Man and Cybernetics, Part B, IEEE Transactions on, 2005.

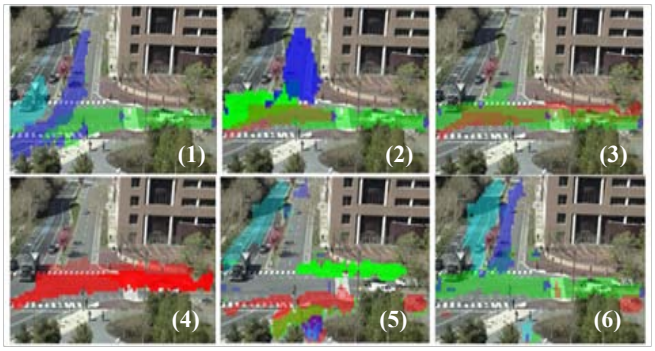


Figure 11: The main motion patterns in 6 clusters of clips. (1) vehicle moving on the north-south roads and at the same time some vehicles making right-turn form south to north; (2) vehicle moving on the east-west roads and some vehicles making left-turn form west to east; (3) crowded traffic in the east-west direction; (4) Vehicle and pedestrian moving from east to west; (5) pedestrian crossing road along east-west, meanwhile, vehicle from north-south road coming to the intersection and stopping; (6) vehicle moving from west to east, some of them making left-turn. Vehicle from north-south road coming and stoping.

[3] W. Hu, X. Xiao, Z. Fu, D. Xie, and S. Maybank. A system for learning statistical motion patterns. TPAMI, 2006.

[4] R.R. Coifman and S. Lafon, Diffusion maps, in Applied and Computational harmonic Analysis, 21:5-23, 2006.

[5] I. Saleemi, K. Shafique, M.Shah. Probabilistic Modeling of Scene Dynamics for Applications in Visual Surveillance. IEEE TPAMI 2008.

[6] T. Xiang and S. Gong. Beyond Tracking: Modelling Activity and Understanding Behaviour. Int. Journal of Computer Vision, vol. 67:1, pp. 21-51, Feb. 2006.

[7] R. Hamid, A. Johnson, S. Batta, A. Bobick, C. Isbell, and G. Coleman. Detection and explanation of anomalous activities: representing activities as bags of event n-grams. CVPR, 2005.

[8] X. Wang, X. Ma, and E. Grimson. Unsupervised activity perception by hierarchical bayesian models. CVPR, 2007.

[9] S. Lafon and A. B. Lee, Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parameterization, PAMI, 28:1393-1430, 2006.

[10] J. Liu, Y. Yang and M. Shah, Learning Semantic Visual Vocabularies Using Diffusion Distance, CVPR, 2009.

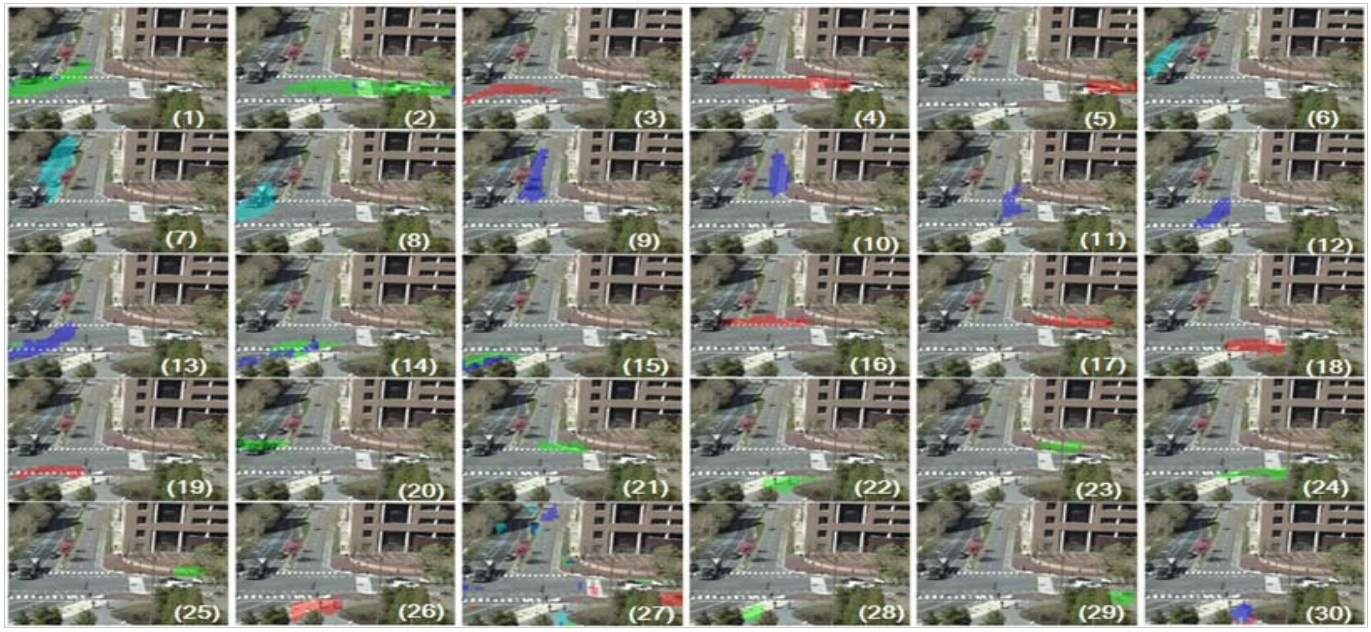
[11] NGSIM dataset, available from: http://www.ngsim.fhwa.dot.gov/.

[12] O. Javed, K. Shafique and M. Shah, "Automated Surveillance in Realistic Scenarios", IEEE Multi Media, January/March 2007.

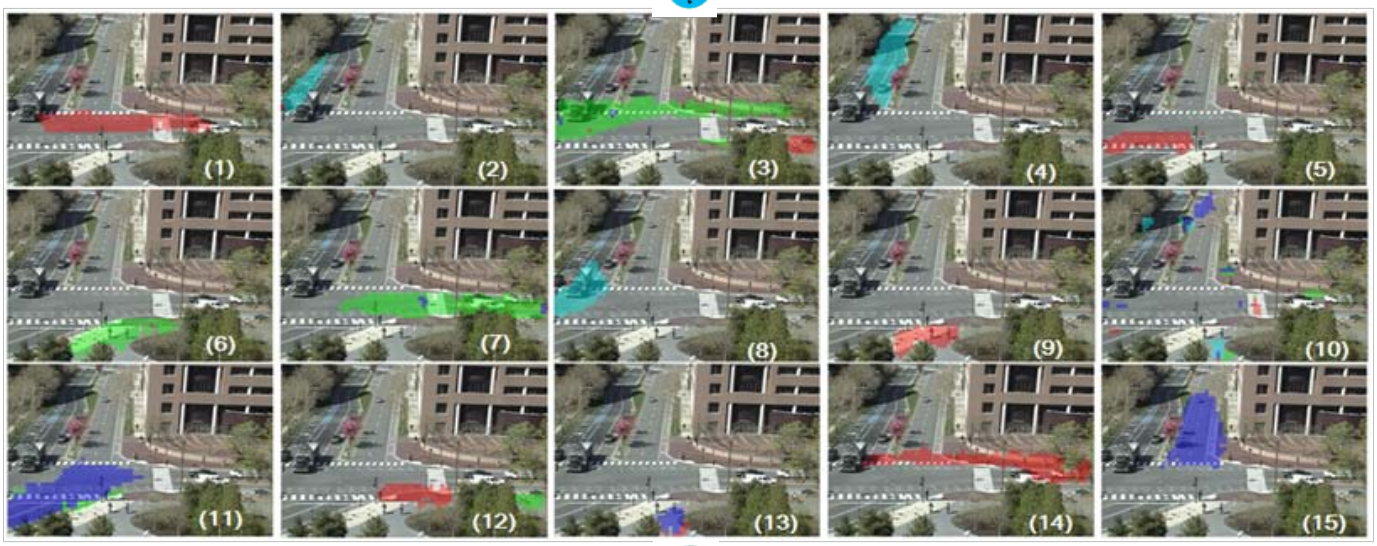
[13] S. Ali and M. Shah, (2006), COCOA - Tracking in Aerial Imagery, SPIE Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications, Orlando, Florida.

[14] C. Stauffer and W.E.L. Grimson, Learning Patterns of Activity Using Real-Time Tracking. IEEE TPAMI, 2000.

[15] A. Basharat, A. Gritai and M. Shah, Learning Object Motion Patterns for Anomaly Detection and Improved Object Detection. CVPR, 2008.



(a)



(b)

Figure 12: (a) 30 motion patterns detected by our algorithm using t = 2, σ =10. (1)-(15) correspond to vehicle movement. (2) and (4) correspond to "vehicle moving along east-west road", (5) corresponds to "vehicle moving along east-west road to intersection and stopping". (1) and (3) correspond to "vehicle moving along west-east road and making left-turn". (6-10) correspond to "vehicle moving along north-south road at different lanes" and (11-13) correspond to "vehicle crossing intersection from south to north". (14-15) correspond to "vehicle making right turn from south to north". (16)-(30) correspond to pedestrian movement. (16-19) correspond to "pedestrian crossing street on crosswalk from east to west". (20)(21)(23)(24) correspond to "pedestrian crossing street on crosswalk from west to east". (22)(25)(26-30) correspond to "pedestrian moving on pavement".
(b) 15 motion patterns detected using t=5, σ=10. Some multiple motion patterns in (a) merge into a single motion pattern in (b). (1) corresponds to "vehicle moving from east to west". (7) corresponds to "vehicle moving from west to east". (2)(4)(15) correspond to "vehicle moving along south-north road". (8)(11) correspond to "vehicle crossing the intersection along north-south road". (3) is the combination of (20)(21)(23)(25) in Figure 12(a), corresponds to "pedestrian crossing street from west to east on crosswalk". (14) is the combination of (16)(17) in Figure 12(a), corresponds to "pedestrian crossing street from east to west on crosswalk". (5)(6)(9)(10)(12)(13) correspond to "pedestrian movement".