

Graph Spectral Motion Segmentation Based on Motion Vanishing Point Analysis

Tian, D.; Kao, J.-Y.; Mansour, H.; Vetro, A.

TR2015-122 October 2015

Abstract

Motion segmentation relies on identifying coherent relationships between image pixels that are associated with motion vectors. However, perspective differences can often deteriorate the performance of conventional techniques. In this paper, we develop a motion segmentation scheme that utilizes the motion map of a single frame to identify motion representations based on motion vanishing points. Segmentation is achieved using graph spectral clustering where a novel graph is constructed using the motion representation distances in the motion vanishing point image associated with the image pixels. Experimental results show that the proposed graph spectral motion segmentation algorithm outperforms state-of-the-art methods for dense segmentation on image sequences with strong perspective effects using motion vectors between only two images.

2015 IEEE International Workshop on Multimedia Signal Processing (MMSP)

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Graph Spectral Motion Segmentation Based on Motion Vanishing Point Analysis

Dong Tian ^{#1}, Jiun-Yu Kao ^{#*2}, Hassan Mansour ^{#3}, Anthony Vetro ^{#4}

*# Mitsubishi Electronic Research Labs (MERL)
201 Broadway, Cambridge, MA 02139, USA*

¹tian@merl.com

³mansour@merl.com

⁴avetro@merl.com

** Department of Electrical Engineering, University of Southern California
3740 McClintock Ave. Los Angeles, CA 90089, USA*

²jiunyuka@usc.edu

Abstract—Motion segmentation relies on identifying coherent relationships between image pixels that are associated with motion vectors. However, perspective differences can often deteriorate the performance of conventional techniques. In this paper, we develop a motion segmentation scheme that utilizes the motion map of a single frame to identify motion representations based on motion vanishing points. Segmentation is achieved using graph spectral clustering where a novel graph is constructed using the motion representation distances in the motion vanishing point image associated with the image pixels. Experimental results show that the proposed graph spectral motion segmentation algorithm outperforms state-of-the-art methods for dense segmentation on image sequences with strong perspective effects using motion vectors between only two images.

I. INTRODUCTION

Motion is an important cue for image segmentation tasks considering the fact that parts of a rigid object often exhibit similar motions over time. In particular, it is often desirable to segment out objects having different motions in a video scene. Motion segmentation has received considerable attention from both academic and industrial researchers over recent decades.

Epipolar plane image (EPI) analysis assumes that an image is composed of homogeneous regions bounded by straight lines, no matter what shape, texture or intensity changes are contained in the image. Such observations have been utilized to build 3D geometric description of a static scene from a video sequence [1]. Epipolar geometry is an intrinsic projective geometry between two views (or images) which has been used for motion segmentation across views [2]. One limitation of using two views is that the motion within the epipolar plane cannot be detected. To overcome this limitation, the epipolar constraints were recently extended to three views. For example, a three-view epipolar constraint called “parallax-based multiplanar constraint” was proposed by Xu et al. [3] to classify each image pixel as either belonging to the background or to moving objects. In this paper, we limited the number of input images to 2 so as to minimize the buffering memory required.

Another approach for motion segmentation uses dynamic textures analysis based on a spatio-temporal generative model for video, which represents video sequences as observations from a linear dynamical system. Chan et al. [4] proposed to use mixtures of dynamic textures as a representation for both appearance and dynamics of a variety of visual processes. However, the approach suffers in the presence of strong perspective effects since it does not account for the epipolar geometry of the scene.

Sparse subspace clustering (SSC) has recently been used as a robust algorithm for motion segmentation using graph spectral clustering techniques [5]. In SSC, feature point trajectories are extracted from several video frames. Sparse optimization is then used to find trajectory associations by estimating each feature trajectory using a sparse linear combination of other feature trajectories. The sparse weights are then used to construct a graph that relates the features and spectral clustering is used to segment the features that occupy the same subspace. The limitation is its reliance on computing trajectories across multiple images. Moreover, the computation complexity of the sparse optimization problem quickly grows with the density of feature points. Liu et al. [6] proposed to adopt graph spectral clustering based on feature descriptors of superpixels assuming that an object appears in multiple images and it may involve high computation using multiple images as inputs. In another related approach [7], a “hypergraph” is built rather than a traditional graph where the hypergraph is constructed based on similarities defined on higher order tuples rather than pair of nodes.

Yet another approach for motion segmentation relies on a variation of robust principal component analysis (RPCA) where a moving background is separated from moving foreground objects [8]. Motion vectors were utilized to align images to the same perspective before applying RPCA to extract a low-rank background from sparse moving foreground objects. One striking limitation of this scheme is that the background alignment assumes that objects lie in the same depth plane which may not necessarily hold true. Another limitation is that the technique requires multiple images to produce a

clean segmentation. In [9], depth was later introduced in the scene motion modeling to overcome the perspective effects in addition to enforcing group sparsity to take advantages from the spatial correlation within sparse components.

In summary, a common limitation observed with conventional methods is their inability to cope well with complex motion, especially with strong perspective effects appearing in the scene. Or some extra inputs has to be introduced as in [9]. In this paper, we consider the situation where only motion vectors (MVs) from a video encoder are available as input and motion segmentation is performed using only the motion between every two video frames. Therefore, our approach does not rely on detecting and tracking features points or extracting pixel trajectories over more than two images. We discuss in Section II the concept of motion vanishing points that finds associations between motion vectors in the scene. In Section III, we propose to construct a graph defined on the image pixels with weights computed using the distances between the motion vanishing points. We then use graph spectral clustering to perform the motion segmentation. We present our experimental results in Section IV followed by a discussion about the performance comparison with several benchmark methods. Finally, we draw our conclusions in Section V.

II. MOTION VANISHING POINT IN IMAGE PLANE

A. Motion Vanishing Point

Geometric vanishing point is a well-known concept used for many tasks, e.g. to infer information of 3D structures. Typically, there are three geometric vanishing points along three orthogonal directions of the 3D world.

In this work, we are interested in *motion vanishing point*. Unlike geometric vanishing points that are derived from the lines appearing on objects in a scene, motion vanishing points are incurred at the intersections between motion vectors.

Fig. 1 illustrates the concept of a *motion vanishing point* v . In the 3D world coordinate system, consider a point is moving from P_1 to P'_1 with a corresponding motion vector $P_1 \rightarrow P'_1$. Similarly, the motion of another point on the same object is represented by $P_2 \rightarrow P'_2$. Because of the perspective effect, $P_1 \rightarrow P'_1$ and $P_2 \rightarrow P'_2$ would intersect at point \mathcal{V} at ∞ distance in the 3D world.

Look into instead the projection of these motions onto the image plane of a camera with camera center at C . The corresponding motion vectors are $p_1 \rightarrow p'_1$ and $p_2 \rightarrow p'_2$ in the image plane. They intersect at a motion vanishing point v . It can be seen that the motion of all points on the same object will share the same *motion vanishing point* v in the image plane just like they share the same motion vanishing point \mathcal{V} in the 3D world.

Based on this observation, the distances between motion vanishing points may serve well to distinguish different moving objects and group pixels from one object together even when parts of the object have motions at different directions due to perspective.

As a side note, the motion vanishing point may co-locate with the geometric vanishing point along z -axis in some cases.

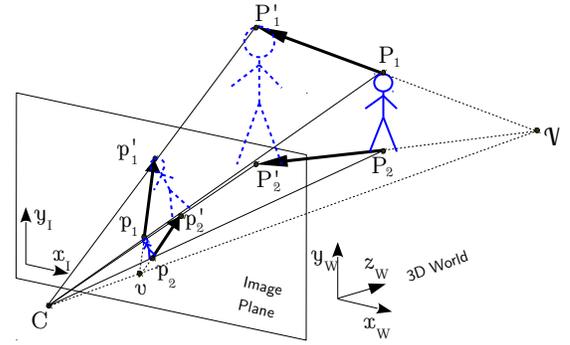


Fig. 1. Illustration of motion vanishing point in 3D world coordinate system and image coordinate system

For example, when the camera strictly moves forward along z -axis and there is no motion in the scene except the global motion, the motion vectors of all pixels would share the same motion vanishing point, and it is exactly the location of geometric vanishing point in z direction. In such cases, it may be beneficial to utilize the motion vanishing point to detect the geometric vanishing point, especially if there are no strong lines or the lines in the scene have random directions.

Finally, to complete the concept of motion vanishing point, we also consider the case when the motion of an object in 3D world is parallel to the image plane. In such a case, the projected motion onto the image plane will be parallel to each other as well and the *motion vanishing point* is thus defined at a motion direction θ relative to x -axis and at an ∞ distance from the origin of the image plane.

B. Motion Vanishing Point Image

In our problem setup, we assume dense motion vectors MV_i available as inputs, that are extracted from a video bitstream coded by an H.265/HEVC [10] encoder or similar. We want to segment the image based on the dense motion vector map.

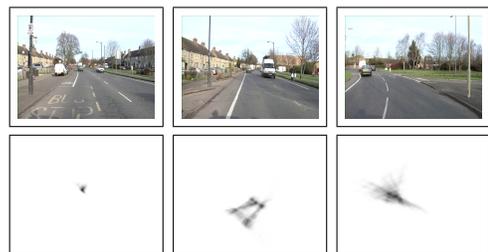


Fig. 2. Example motion vanishing point images. **Left:** Moves forward smoothly. **Center:** Uneven road condition and an oncoming vehicle. **Right:** Turning left. Darker value means more motion vanishing points occurs.

By recording the motion vanishing point v_{ij} between MV_i and MV_j for $\forall i, j$ and $i \neq j$, we can plot a motion vanishing point image \mathbf{V} as shown in Fig. 2 for a car driving sequence in [11] captured by a camera mounted on a car.

The pixel value at a location (x, y) in \mathbf{V} represents the number of motion pairs (i, j) sharing the same vanishing point at (x, y) . A darker color at (x, y) indicates more vanishing points falling there. When the car moves forward smoothly

and there are no other moving objects in the video (left in Fig. 2), the vanishing points concentrate around a single motion center. This is also an example when a geometric vanishing point co-locates with the motion vanishing point. When other moving objects appear in the scene, such as an upcoming vehicle (center in Fig. 2), the vanishing point image would show multiple convergence points. Similarly, when the car turns left (right in Fig. 2), the vanishing point image exhibits a sharper round boundary. It would be interesting to study the relationship between car moving status and the vanishing point image in a separate work. In this paper, we will present a way to use the vanishing point image to assist in building a graph for further graph spectral segmentation.

III. GRAPH SPECTRAL MOTION SEGMENTATION

A. Graph Spectral Clustering

In general graph signal processing [12], an undirected graph $G = (V, E)$ consists of a collection of nodes $V = \{1, 2, \dots, N\}$ connected by a set of links $E = \{(i, j, w_{ij})\}, i, j \in V$ where (i, j, w_{ij}) denotes the link between nodes i and j having weights w_{ij} . For image processing applications, a pixel may be treated as a node in a graph. The adjacency matrix \mathbf{W} of the graph is an $N \times N$ matrix with weights w_{ij} as its entries, and the degree d_i of a node i is the sum of link weights connected to node i . The degree matrix is the diagonal matrix $\mathbf{D} := \text{diag}\{d_1, d_2, \dots, d_N\}$, and the combinatorial Laplacian matrix is $\mathcal{L} := \mathbf{D} - \mathbf{W}$.

Furthermore, the normalized Laplacian matrix is defined as $\mathbf{L} := \mathbf{D}^{-1/2} \mathcal{L} \mathbf{D}^{-1/2}$, which is a symmetric positive semi-definite matrix. Hence, it admits an eigendecomposition $\mathbf{L} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^t$, where $\mathbf{U} = \{\mathbf{u}_1, \dots, \mathbf{u}_N\}$ is an orthogonal set of eigenvectors and $\mathbf{\Lambda} = \text{diag}\{\lambda_1, \dots, \lambda_N\}$ is its corresponding eigenvalue matrix. The eigenvectors and eigenvalues of the Laplacian matrix provide a spectral interpretation of the graph signals. Note that eigenvalues $\{\lambda_1, \dots, \lambda_N\}$ can be treated as graph spectral frequencies and are always situated in the interval $[0, 2]$ on the real line. We assume the eigenvalues are sorted in a non-decreasing order, i.e. $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N$.

The principle eigenvectors, e.g. those eigenvectors among the first K eigenvectors $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_K\}$ after removing the eigenvectors corresponding to zero eigenvalues, could be used to cluster the pixels into k groups [13]. We use k -means algorithm [14] to cluster the rows of the principal eigenvectors.

We adopt this graph spectral clustering approach for performing motion segmentation after constructing the graph in the next section.

B. Graph Construction for Motion Segmentation

In this section, we propose to construct the graph based on the motion vanishing points as described in Section II. We consider a sparse graph connectivity where every MV associated to a pixel is connected to its 4 spatial neighboring MVs. This topology is selected since it can capture the local changes of the motion field in the graph structure.

1) *Representation Point of a MV*: Unlike conventional approaches for image processing where the graph weights are often computed from the pixel intensities, we construct the graph weights using the locations of the vanishing points associated with the MVs. For every MV, we propose to select a *representation point*, which corresponds to the location along the ray of the motion vector that has the maximum pixel value in the motion vanishing point image \mathbf{V} .

Fig. 3 illustrates the process of selecting a representation point R_i for a motion vector MV_i . For zero MVs, their representation point R_i will co-locate at its spatial location in the image, since its length is zero.

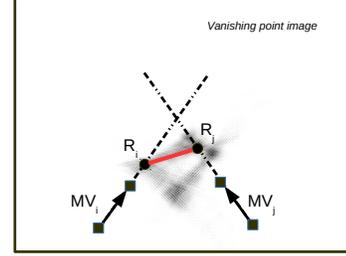


Fig. 3. Representation point of a MV and distance between a MV pair

2) *Perspective Distances Between a MV Pair*: We define the distance between a pair of motion vectors as the distance between their corresponding representation points. In Fig. 3, the distance between MV_i and MV_j is given as,

$$d_{p,ij} = \|R_i - R_j\| = \sqrt{(R_{xi} - R_{xj})^2 + (R_{yi} - R_{yj})^2}, \quad (1)$$

where (R_{xi}, R_{yi}) is the location of the *representation point* of MV_i . This distance is called the *perspective distance*.

Under certain circumstances, it may be desirable to solely use the above distance to define graph weights to achieve perspective invariant motion segmentation. In this paper, we set the distance to be a weighted sum of the perspective distance $d_{p,ij}$ and the Euclidean distance $d_{m,ij}$ between the motion vectors,

$$d_{ij} = w_p d_{p,ij} + (1 - w_p) d_{m,ij}. \quad (2)$$

Finally, the graph adjacency matrix \mathbf{W} is computed using,

$$w_{ij} = e^{-\beta d_{ij}} + \epsilon. \quad (3)$$

where $\beta = 25$ and $\epsilon = 0$ are constants in this work.

3) *Perspective Distances with Parallel MVs*: When the perspective effect is not very strong, one may observe many MVs from one object to be nearly parallel to each other. In such cases, the representation point R_i may fall outside the image range, i.e. at (θ, ∞) with motion direction being θ and distance from the origin of the image plane being ∞ .

If both the motion representation points of a pair of motion vectors MV_i and MV_j fall outside the image range, their distance is evaluated as the difference in their motion directions,

$$d_p = |\theta_i - \theta_j|. \quad (4)$$

If only one of the MVs under evaluation has a representation point that falls outside the image range, we set their distance to a maximum value, for example,

$$d_p = \sqrt{\text{width}^2 + \text{height}^2}, \quad (5)$$

where *width* and *height* stand for the image resolution, because they belong to different objects.

4) Perspective Distances from Pixels with Motion Absent:

In a practical system, there may exist some pixels without motion information available if they are coded in Intra prediction modes rather than Inter prediction modes by an H.265/HEVC encoder. We propose to keep the graph topology construction described at the beginning of this section as untouched, no matter whether the motion is available or not. In addition, we propose to use a conventional in-painting method to fill up those isolated holes in the motion field. For larger holes that are typically caused by disocclusions from motion and occur at object boundaries, we propose the following procedure. Suppose pixel *i* has MV available while pixel *j* does not, as shown in Fig. 4. When calculating the distance between *i* and *j*, we propose first to search along the extended graph edge to check if there exists a pixel *k* that has motion vector accessible. If *k* exists, the *representation point* R_j is replaced by R_k and the distance between *i* and *j* is calculated using R_i and R_k . Otherwise (no *k* found), e.g. if the search for *k* ends at the image boundary, we set the distance between *i* and *j* to be the maximum distance as in Eqn. (5). Finally, if both pixels *i* and *j* have no MV available, their distance is set to be equal to the minimum distance 0. In this way, we could bring together the motion changes on each side of a Intra coded block and have them reflected in the graph structure.

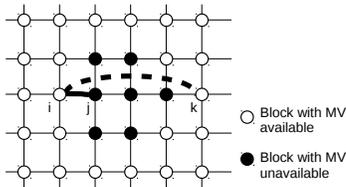


Fig. 4. How to calculate distance from a block *j* without motion available

IV. EXPERIMENTS AND DISCUSSIONS

A. Experimental Setup and Benchmarks

Our framework is applied to the Hopkins 155 dataset, which is available at <http://www.vision.jhu.edu> [15]. We target the 10 outdoor traffic sequences, *cars1-cars10*, in the dataset. The images are captured by a hand-held moving camera consisting of vehicles moving on the street. Five of them have two motions and another five have three motions.

Since we are targeting a dense segmentation, the feature points and their trajectories provided in the dataset are not used. Instead, we encode the raw video sequences using the

H.265/HEVC test model [16] with default encoding settings. Motion vectors with quarter pixel accuracy are extracted afterwards where each motion vector corresponds to a 4×4 pixel block regardless the prediction unit (PU) partitioning. Since we do not consider raw pixel information in our scheme, the segmentation accuracy is limited to 4×4 blocks.

For our proposed graph spectral motion segmentation, we first compute the *motion vanishing image*, then construct the graph and cluster the motion vectors as described in Section II and III. For those sequences with two motions, the number of clusters is set to $k = 4$ while for those with three motions, the number of clusters is set to $k = 5$. The number of clusters is set larger than the number of true motions in the scene in order to tolerate errors in the MVs from H.265/HEVC encoding, since the video encoder is optimized to favor a rate-distortion cost instead of exact block matching [16].

In the Hopkins 155 dataset, the ground truth for motion segmentation is provided on the extracted feature points as described in [15]. However, since we target a dense segmentation in this paper, the ground truth of every pixel in the sequences is not available. In order to judge the performance of the proposed method, we generate 3 benchmarks with the following state-of-the-art algorithms.

SSC: Sparse subspace clustering [5] uses graph spectral clustering with the graph constructed from feature trajectories. The graph weights are computed using constrained sparse optimization and correspond to the coefficients by which a feature trajectory is approximated using the remaining trajectories. In order to match this scheme to our setup, we compute the corresponding positions of blocks in the previous image for each block in the current image using its associated motion vector. Then cascading the positions in the current and previous images for one block will result in its trajectory between two images.

RPCA: To fully employ the potentials from RPCA, the benchmark segmentation is conducted in the pixel domain instead of in the motion field. Since the camera is moving, the motion field is used to perform global motion estimation and compensation as described in [8]. Since the RPCA method models the image into two parts, one is background (corresponding to the low rank component) and the other is moving foreground (corresponding to the sparse component), only two segments are extracted using this approach. Additionally, a thresholding on the sparse component output is needed to generate the foreground mask. In the experiments, we mark those pixels as foreground if their values in the sparse component are larger than a hard threshold 2.

Simple Graph Spectral Clustering: The simple graph spectral clustering benchmark uses the same procedure to construct the graph except that the distance between two MVs only considers their magnitude differences. That is, we set $w_p = 0$ in Eqn. (2). After the graph is constructed, the same graph spectral clustering approach described in Section III is applied to cluster the motion vectors and generate the segmentation results.

Due to the high complexity involved with SSC for the dense

motion field, we subsample the input images from 640×480 to 160×120 for all methods except for the RPCA benchmark.

B. Experimental Results and Discussions

Fig. 5 compares the motion segmentation results from our proposed method versus the three benchmarks described above on both two-motion and three-motion sequences.

Overall, it shows that our proposed graph method using motion vanishing points is representative enough for segmenting different motions in the video, which outperforms the other three benchmarks under the setting of dense segmentation.

Moreover, one attractive feature of our proposed scheme is its ability to perform dense segmentation using the output of video encoding, i.e., the compressed domain motion vectors. This is despite the fact that compressed domain motion vectors can be very noisy. Our scheme avoids additional preprocessing steps, such as feature detection and tracking. Hence, a real-time motion segmentation could be achieved along with the video encoding.

Although SSC enjoys a good performance with sparse and long trajectories [5], it fails with dense and short trajectories due to the high coherence between trajectories in different subspaces. By checking the connectivity of the graph constructed from the constrained optimization problem in SSC, all graph nodes are connected to a very few number of other nodes, and thus it leads to a stripe-like segmentation from the left-top corner. On the contrast, the 4-connected graph topology in our proposed method would avoid such over-biased connections. It is interesting to notice that with SSC, the sparse constraint being enforced favors a sparse solution with fewer number of motion vectors to represent the current motion vector, while the *representation point* for a MV in our method is the one with votes from most possible number of motion vectors.

Secondly, when compared to the RPCA method, the moving objects segmented by our method are more complete and could better outline the full moving object, while RPCA generally outputs scattered pixels and may only classify areas with edges as moving target. RPCA also suffers from taking some background pixels as foregrounds when there are strong perspective effects. However, a graph spectral clustering method (including the proposed method) by itself cannot give any semantic meaning for each segment, while it is an advantage for RPCA to tell the foreground from the background. When jointly inspecting the results of RPCA and our proposed method, we would expect to segment a cleaner moving foreground and aggregate the background segments together.

Thirdly, the comparison between the Simple Spectral Clustering and the proposed graph spectral clustering shows that introducing the motion vanishing points as part of the motion descriptor can improve the segmentation with motion vectors as input. It is worthwhile to note that the selection of w_p in Eqn. (2) depends on how much perspective effect is shown in the images. A general guideline is that stronger perspective effect would require a larger w_p . For experiments with the Hopkins 155 dataset, we fix $w_p = 0.25$ for the proposed method.

In the end, we would like to point out that some extensions may enhance the proposed method. On the one hand, the graph for our experiments is constructed by connecting a node with its 4 spatial neighbors as mentioned in Section III-B. However, we noticed that for some sequences, connecting the nodes with their 8 neighbors may improve the segmentation results. A fully connected graph or a k -nearest-neighborhood graph might be considered as an option, especially for those sequences where the spatial distances are poorly correlated to the cluster distances, e.g. for the car driving sequences in [11].

On the other hand, while the usage of motion vanishing points and a single representation point are fairly powerful for those motions with strong perspective effects, it may not be straightforward to accommodate some special categories of motions. For example, the vanishing point image of circular motions would be evenly distributed over all directions. Motivated by this, it would be interesting to explore and use the relationship between the distribution of motion vanishing points and motion distributions instead of using a single representation point during the graph construction.

V. CONCLUSIONS

In this paper, we studied the dense motion segmentation problem with motion vectors available from an H.265/HEVC codec. A novel graph spectral clustering method was proposed to especially to deal with image sequences with strong perspective effects. We first explored the motion vanishing point in 3D world and its mapping in the image plane and obtained a motion vanishing point image. Then we proposed to identify a representation point for each motion vector as part of the motion descriptor. Finally, a novel graph was constructed based on the distances measured between the representation points in the motion vanishing point image, and motion segmentation was realized via a graph spectral clustering method. Compared to state-of-art approaches, experimental results showed that the proposed graph spectral motion segmentation algorithm works better especially with image sequences having perspective effects. In addition, we believe the method has a good potential for further improvements from several extensions or combinations, which is subject to our future work.

REFERENCES

- [1] R. C. Bolles and H. H. Baker, "Epipolar-plane image analysis: A technique for analyzing motion sequences," *Readings in Computer Vision: Issues, Problem, Principles, and Paradigms*, p. 26, 2014.
- [2] B. Matusik and T. Pajdla, "Estimation of omnidirectional camera model from epipolar geometry," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 1. IEEE, 2003, pp. 1–485.
- [3] F. Xu, G. Gu, K. Ren, and W. Qian, "Motion segmentation by new three-view constraint from a moving camera," *Mathematical Problems in Engineering*, 2015.
- [4] A. B. Chan and N. Vasconcelos, "Modeling, clustering, and segmenting video with mixtures of dynamic textures," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 5, pp. 909–926, 2008.
- [5] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 11, pp. 2765–2781, 2013.

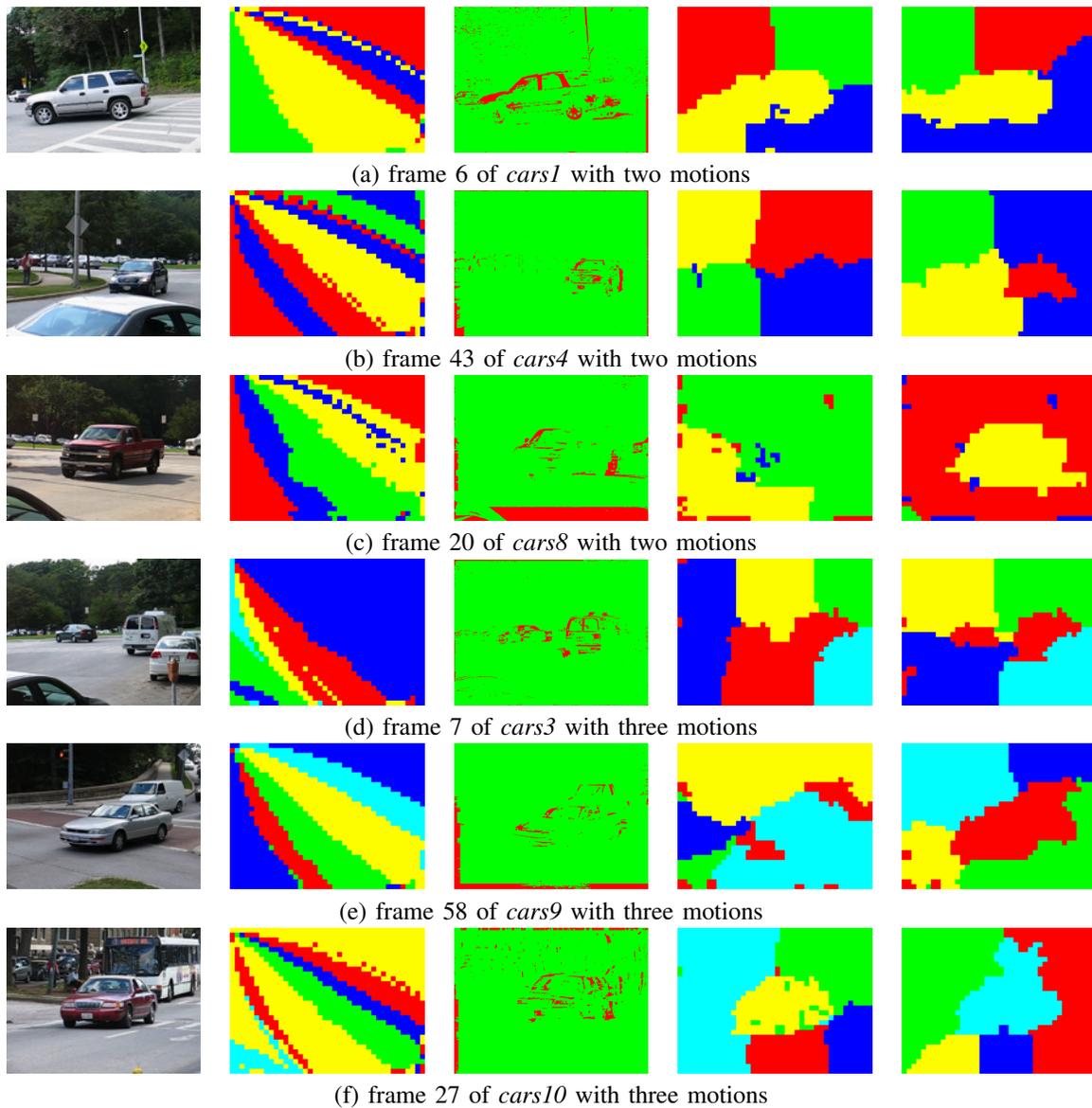


Fig. 5. **From top to bottom:** Results on sequences obtained with a spatial subsampling of 4, except for RPCA. **From left to right:** Original image, benchmark with SSC [5], RPCA [8], simple graph spectral clustering, and the proposed graph spectral clustering based on motion vanishing points.

- [6] Y. Liu, J. Liu, Z. Li, J. Tang, and H. Lu, "Weakly-supervised dual clustering for image semantic segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 2075–2082.
- [7] P. Ochs and T. Brox, "Higher order motion models and spectral clustering," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 614–621.
- [8] H. Mansour and A. Vetro, "Video background subtraction using semi-supervised robust matrix completion," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6528–6532.
- [9] D. Tian, H. Mansour, and A. Vetro, "Depth-weighted group-wise principal component analysis for video foreground/background separation," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, p. Accepted.
- [10] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1649–1668, 2012.
- [11] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *Computer Vision–ECCV 2008*. Springer, 2008, pp. 44–57.
- [12] D. Shuman, S. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *Signal Processing Magazine, IEEE*, vol. 30, no. 3, pp. 83–98, 2013.
- [13] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*. MIT Press, 2001, pp. 849–856.
- [14] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979. [Online]. Available: <http://www.jstor.org/stable/2346830>
- [15] R. Tron and R. Vidal, "A benchmark for the comparison of 3-d motion segmentation algorithms," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, June 2007, pp. 1–8.
- [16] I.-K. Kim, K. McCann, K. Sugimoto, B. Bross, W. Han, and G. Sullivan, "High efficiency video coding (hevc) test model 10 (hm10) encoder description," *JCTVC Document: JCTVC-L1002-v3, Joint Collaborative Team on Video Coding (JCT-VC)*, 2013.