Enhancing motion segmentation by combination of complementary affinities

Vasileios Zografos

Original Publication:

Vasileios Zografos , Enhancing motion segmentation by combination of complementary affinities, 2012, Proceedings of the 21st Internationa Conference on Pattern Recognition, 2198-2201. ISBN: 978-1-4673-2216-4

21st International Conference on Pattern Recognition (ICPR 2012), 11-15 November 2012, Tsukuba, Japan

21st International Conference on Pattern Recognition (ICPR 2012)
November 11-15, 2012. Tsukuba, Japan

Enhancing motion segmentation by combination of complementary affinities

Vasileios Zografos
Department of Electrical Engineering
Linköping University, SWEDEN
zografos@isy.liu.se

Abstract

Complementary information, when combined in the right way, is capable of improving clustering and segmentation problems. In this paper, we show how it is possible to enhance motion segmentation accuracy with a very simple and inexpensive combination of complementary information, which comes from the column and row spaces of the same measurement matrix. We test our approach on the Hopkins155 dataset where it outperforms all other state-of-the-art methods.

1. Introduction

Motion segmentation is the task of partitioning every image from a sequence F into separate regions, each associated with a distinct 3D motion. Here we focus on methods that are applied to a sparse set of points N, typically interest points, that are consistently tracked over time, and their trajectories analysed in the images.

The most common way of solving these types of problems, is to define some type of motion-based similarity between the points (pairwise or higher-order), calculate and collect all such similarities into a symmetric, $N \times N$ affinity matrix $\mathbf{A}$, and then carry out segmentation by clustering in some appropriate lower-dimensional space. Since we are only considering a sparse set of feature points, working with and manipulating $\mathbf{A}$ is not a difficult or expensive task, as could be the case for fully dense motion segmentation approaches.

The segmentation problem may be simplified further by assuming that we generally only encounter small depth variations in the imaged scene, and so we may utilise an affine camera model. Under this assumption, points on different moving objects will lie on a union of low-dimensional linear subspaces and the segmentation problem is abstracted to one of subspace clustering.

This idea of enhancing segmentation through combination of additional, complementary information is the main focus of this paper. Here we show how it is possible to obtain a very accurate motion segmentation result by simple combination of affinities, that come from the same point trajectory data, but from different parts of it. Our experiments clearly demonstrate that our scheme produces segmentation results beyond the state-of-the-art methods in literature, while keeping the computational cost low. In addition, we have observed that using a simple combination approach (Hadamard product), outperforms more complicated approaches from machine learning research. We evaluate our scheme on the extended Hopkins155 dataset[11], which is commonly used in sparse motion segmentation research.

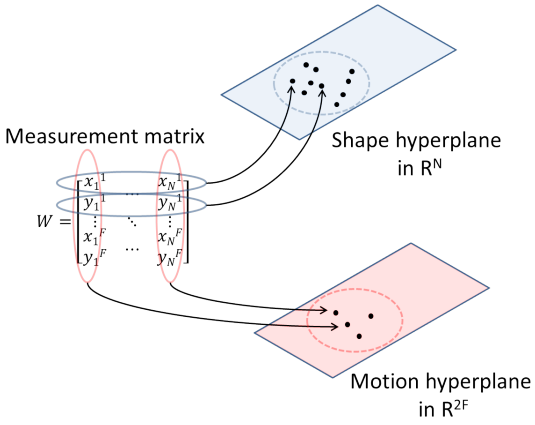

Figure 1. An illustration of the induced subspaces from the matrix W.

2. Background

In this section we introduce some important background concepts, together with a brief review of the state-of-the-art motion segmentation approaches as well as different combination schemes for clustering.

We begin with the notion of an *affinity matrix* $\mathbf{A}$.

Given N feature points and an appropriate similarity measure, an affinity matrix is the $N \times N$ symmetric matrix that contains all pairwise similarities between the points in the scene. Once we have an affinity matrix, we can obtain the segmentation by standard clustering approaches, such as spectral clustering [8].

The next concept is that of a $2F \times N$ *measurement* matrix $\mathbf{W}$, under an affine camera model. The rows of $\mathbf{W}$ contain the coordinates of all the N feature points in *each view*, whereas its columns contain the x,y coordinates of each point across *all views* F:

$$\underbrace{\begin{bmatrix} x_1^1 & \dots & x_N^1 \\ y_1^1 & \dots & y_N^1 \\ \vdots & \ddots & \vdots \\ x_1^F & \dots & x_N^F \\ y_1^F & \dots & y_N^F \end{bmatrix}}_{W_{[2F \times N]}} = \underbrace{\begin{bmatrix} \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_F \end{bmatrix}}_{M_{[2F \times 4]}} \underbrace{\begin{bmatrix} \mathbf{X}_1 & \dots & \mathbf{X}_N \\ \mathbf{Y}_1 & \dots & \mathbf{Y}_N \\ \mathbf{Z}_1 & \dots & \mathbf{Z}_N \\ 1 & \dots & 1 \end{bmatrix}}_{S_{[4 \times N]}}. \tag{1}$$

As we can see $\mathbf{W}$ factors into a *motion* matrix $\mathbf{M}$ containing all the 2×4 projection matrices, and a 3D *shape* matrix $\mathbf{S}$. Observe that rank($\mathbf{W}$)$\leq$4.

Assuming for simplicity a single object in the scene, then each column of $\mathbf{W}$, defines a point on a 4D linear subspace (hyperplane) embedded in $\mathbb{R}^{2F}$ (see Fig. 1). This hyperplane, contains the motion trajectories of all the points on the 3D rigid object. Furthermore, we may obtain every point on the hyperplane as a linear combination of 4 basis columns from $\mathbf{W}$. If we do the same for the rows of $\mathbf{W}$ we obtain a different 4D hyperplane, this time in $\mathbb{R}^N$. The points contained therein, each represent the coordinates of the 3D shape in a single view. Again, we may synthesise the shape of the object in *any* novel view by a linear combination of 4 rows of $\mathbf{W}$ (see [13]). Obviously these two spaces are related and provide complementary information that may be utilised to improve motion segmentation. This hypothesis as well as whether the combination can be carried out in a simple way, are the two questions we will try to examine.

Motion segmentation. The column based method was initially used by [10] for motion segmentation and 3D reconstruction. They used SVD to factor $\mathbf{W}=\mathbf{U}\Sigma\mathbf{V}^T \Rightarrow \mathbf{M}=\mathbf{U}\Sigma^{\frac{1}{2}}, \mathbf{S}=\Sigma^{\frac{1}{2}}\mathbf{V}^T$, in a similar fashion to (1), and obtain a pairwise affinity via $\mathbf{Q}=\mathbf{V}\mathbf{V}^T$.

The majority of the motion segmentation methods in literature, use the same approach and cluster directly on the induced hyperplane(s). Where they differ, is on the way they define their motion affinity. So for example (SSC) by [3] exploits the sparsity of representation between points on the same subspace as a form of affinity. (SCC) [1] iteratively clusters points based on the polar curvature between the trajectories. The (SC) method by [7] generates a pointwise affinity matrix directly from a reduced column $\mathbf{V}$. The affinity is then defined as

$$\mathbf{A}(i,j) = \left(\frac{v_i^T v_j}{\|v_i\| \, \|v_j\|} \right)^2, \tag{2}$$

where v_i,v_j are columns of $\mathbf{V}$. (2) is simply the first principal angle between the subspaces inside a cosine kernel. More recently, [12] have introduced a method (SLBF) that iteratively fits linear subspaces from a local neighbourhood of points, the size of which is determined automatically.

The row based approach for motion segmentation was identified by [13] (LCV), where they synthesised probable motion trajectories of feature points using 4 basis rows from $\mathbf{W}$, and compared them with the actual trajectories to define a motion affinity for segmentation.

Combination schemes. Combination of information in order to improve segmentation or clustering results has been examined in machine learning literature. We may briefly categorise combination methods into two types: ones that perform the fusion on the affinity level and those that combine the final labelling.

In the first category, we have the simple addition or componentwise product (Hadamard) between the matrices. This can be further extended by a weighted combination and multi-view expansion of the affinity matrices [2]. More advanced methods such as the co-regularisation by [6] that enforce clustering agreement or the co-training method by the same authors [5] where the graph structure of one affinity matrix is modified by the clustering results of the other, are also of interest.

In the second category, multiple clusterings are performed separately and their results are combined in the end. Here we test the clustering ensemble method by [9] that uses a set of consensus functions to determine the correct clustering, and the graph approach by [4].

3. Method

As mentioned previously, we exploit the fact that there is a clear relationship between the two linear subspaces induced by the matrix $\mathbf{W}$. Although each subspace contains different information (shape and motion) they are related since they originate from the same data.

Our combination is done on the affinity level by generating an affinity matrix from the shape information (column-based) and one from the motion trajectories (row-based). We combine the two matrices with a simple Hadamard product. Once the combined affinity matrices have been generated, we perform the segmentation using spectral clustering. For the column-based

	SC [7]	LCV [13]	SCC [1]	SLBF [12]	SSC [3]	HAff	HEigen
Mean error % / Var.	2.10 / 0.0	2.34 / 0.07	4.68 / 0.25	1.71 / 0.0	1.47 / 0.06	1.63 / 0.04	**1.14** / 0.01
Average time (sec)	4.87	**1.47**	1.61	9.98	111.78	6.34	5.28

Table 1. Comparison between motion segmentation methods in literature and our combination scheme, on the Hopkins155 dataset.

	Mean error %
Co-regularisation [6]	4.17
MCLA [9]	2.36
HBGF [4]	2.09

Table 2. Advanced information combination approaches.

affinity we use a reformulation of (2) and for the row-based, the approach by [13]. An overview of the method shown in Algorithm 1.

Algorithm 1 Algorithm of proposed method

Generate n-way affinity matrix $\mathbf{E}$ using LCV [13]
q_1=0, q_2=0;
for $\alpha = 1 \rightarrow 10$ **do**
  $\mathbf{A}_R = (\mathbf{E}^2 + \alpha^2)^{-0.5}$
  quality $\leftarrow$ SpectralClustering($\mathbf{A}_R$)
  if quality$>q_1$ then $\mathbf{A}_{R\text{opt}} = \mathbf{A}_R$, q_1=quality
end for
$[\mathbf{U}, \Sigma, \mathbf{V}]$=SVD($\mathbf{W}$)
for $\beta = 1 \rightarrow 10$ **do**
  $\mathbf{X} = \mathbf{V}(:, 1:\beta)$
  $\mathbf{X}_0 = \text{diag}(1/\sqrt{\sum_{\text{col.}} \mathbf{X}^2})\mathbf{X}$
  $\mathbf{A}_C = \left|\mathbf{X}_0\mathbf{X}_0^T\right|^4$
  quality $\leftarrow$ SpectralClustering($\mathbf{A}_C$)
  if quality$>q_2$ then $\mathbf{A}_{C\text{opt}} = \mathbf{A}_C$, q_2=quality
end for
$\mathbf{A} = \mathbf{A}_{R\text{opt}} * \mathbf{A}_{C\text{opt}}$
Labels= SpectralClustering($\mathbf{A}$)

Note here that since we have two affinity matrices we need to optimise two kernel parameters α, β. In this 2D space, it would be prohibitively expensive to do spectral clustering (i.e. eigen-decomposition) for every choice of parameters. The simple way around this is to generate the affinity matrices individually, thus solving the optimisation problem as independent 1D slices, and combine the results in the end. Although this might not necessarily be the global optimum in α, β-space, it gives very good results at a fraction of the cost.

4. Experiments

In this section we test our approach against other methods in literature. All the tests have been carried out on the Hopkins155 dataset [11], which contains 159 sequences of 2-5 motions with ground truth trajectory data. We have carried out 100 test runs for each method and quote the mean segmentation error and run-times over the whole dataset, averaged over all the test runs.

In Table 1, we compare the individual motion segmentation methods. SSC and SLBF are amongst the best but also amongst the slowest. Our affinity combination approach (HAff) is shown in the 7th column. We see that this is a much improved result than the individual SC and LCV methods and comparable to the state-of-the-art.

In addition, we have tested the combination in the eigen-space instead of in the affinity space. That is, during spectral clustering, we obtain the row-normalised $N \times k$ matrix $\mathbf{Y}$ on which we run the clustering step. $\mathbf{Y}$ contains the k-largest eigen-vectors of the affinity matrix $\mathbf{A}$, where k is the number of objects in the scene. We then create the $N \times N$ matrix $\mathbf{K}=\mathbf{Y}\mathbf{Y}^T$ and we use that for the information fusion. Thus in the same way as in Algorithm 1, $\mathbf{K}=\mathbf{K}_{R\text{opt}} * \mathbf{K}_{C\text{opt}}$, and the final steps of the spectral clustering are done on the $\mathbf{K}$ matrix instead, without the need for an additional eigen-decomposition. The results, shown in the last column (HEigen) of Table 1. We see that this type of Hadamard combination has a dramatic effect in the overall miss-classification error, and is also slightly faster than the affinity-level combination.

We now look at the more complicated combination schemes such as co-regularisation [6] and the clustering ensemble methods [9, 4]. For the latter shown in Table 2. We see that only the HBGF method slightly improves the segmentation over the individual results, but it is still far away from the affinity-based and eigen-based Hadamard products.

To illustrate exactly how important it is to combine complementary information, we also show in Table 4 the Hadamard products for the 10 row-based only and 10 column-based only affinity matrices from Algorithm 1, both unweighted and weighted by their clustering quality. That is, affinity matrices with poor clustering scores, will be suppressed as $\mathbf{A}=\prod \mathbf{A}_i^{1-q_i}$, where q_i is the normalised distortion error. We see that none of the scores in Table 4 are anywhere near those in Table 1. These results suggest that the improvement does not come from the Hadamard product alone but also from

	LCV	SC	HEigen
Distortion <	0.123	0.121	**0.052**
Silhouette >	0.854	0.864	**0.945**
Davies-Bouldin <	0.180	0.190	**0.06**
Dunn >	8.88	6.92	**21.34**
Calinski-Harabasz >	4.39E3	4.84E3	**1.45E4**
Hartigan >	14.15	**16.05**	5.69
Krzanowski-Lai >	29.44	**33.40**	11.85

Table 3. Comparison of clustering measures on the 2RT3RC sequence.

	Mean error %	Std
H_{Row}	2.9	0.0
Weighted H_{Row}	7.24	0.0
H_{Col}	6.44	0.14
Weighted H_{Col}	3.86	0.08

Table 4. Hadamard combination of non-complementary information.

the existence of complementary information.

In Table 3 we isolate a sequence from the Hopkins155 and show the improvement in common clustering quality measures, between the row-based and column-based methods and their Hadamard product. These criteria measure properties such as distortion, distance from the cluster centroid, cluster dispersion and so on. The symbol next to each measure indicates whether this measure should be maximum or minimum for a good clustering result. We see that the Hadamard product improves most of the criteria. Such a behaviour is typical for most of the sequences in the Hopkins155.

Finally, we show a cdf error plot in Fig. 2 for the original methods (LCV, SC and SLBF) and the overall improvement introduced by the Hadamard product combination. We see that not only has the new approach improved on the mean error but also reduced the occurrence of very large classification errors, something which is a problem with the competing methods.

5. Conclusion

We have presented a simple combination scheme for improving the accuracy of motion segmentation. This scheme exploits the related but complementary information that exists in the column and row spaces of the measurement matrix $\mathbf{W}$. We show that our method outperforms any other method in literature while still remaining computationally attractive. In addition, the simple Hadamard product performs much better than other combination schemes from machine learning research. We wish to extend this method to incorporate non-motion cues and use their complementary information to determine other important parameters such as the number of moving objects.

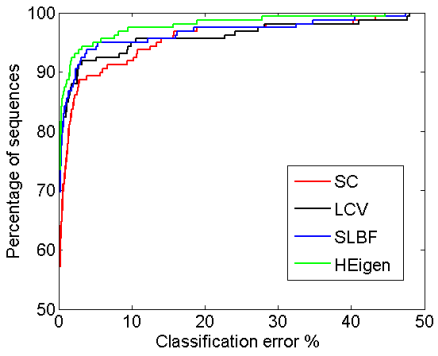

Figure 2. Error distributions for different segmentation methods.

References

[1] G. Chen and G. Lerman. Motion Segmentation by SCC on the Hopkins 155 Database. In *ICCV*, 2009.

[2] V. R. de Sa. Spectral clustering with two views. In *ICML*, 2005.

[3] E. Elhamifar and R. Vidal. Sparse Subspace Clustering. In *CVPR*, 2009.

[4] X. Z. Fern and C. E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *ICML*, 2004.

[5] A. Kumar and H. Da. A co-training approach for multi-view spectral clustering. In *ICML*, 2011.

[6] A. Kumar, P. Rai, and H. D. III. Co-regularized spectral clustering with multiple kernels. In *NIPS*, 2010.

[7] F. Lauer and C. Schnorr. Spectral clustering of linear subspaces for motion segmentation. In *ICCV*, 2009.

[8] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2001.

[9] A. Strehl and J. Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *JMLR*, 3:583–617, 2002.

[10] C. Tomasi and T. Kanade. Shape from motion from image streams under orthography: A factorization method. *IJCV*, 9(2):137–154, 1992.

[11] P. Tron and R. Vidal. A Benchmark for the Comparison of 3-D Motion Segmentation Algorithms. In *CVPR*, 2007.

[12] T. Zhang, A. Szlam, Y. Wang, and G. Lerman. Randomized hybrid linear modeling by local best-fit flats. In *CVPR*, pages 1927–1934, 2010.

[13] V. Zografos and K. Nordberg. Fast and accurate motion segmentation using linear combination of views. In *BMVC*, 2011.