Fast and accurate motion segmentation using Linear Combination of Views

Vasileios Zografos
zografos@isy.liu.se
Klas Nordberg
klas@isy.liu.se

Computer Vision Laboratory
Linköping University
SE-581 83 Linköping, Sweden

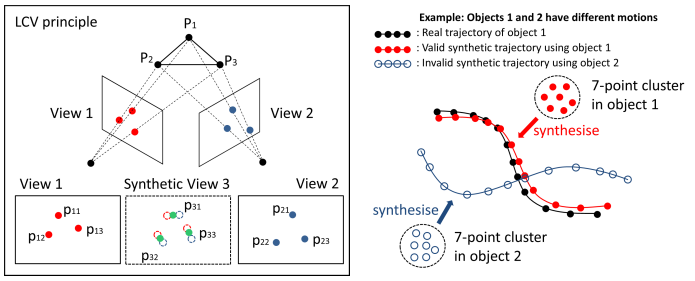

Figure 1: The main principle of LCV synthesis is illustrated on the left. On the right, we show how this idea can be used for generating motion affinities, leading to segmentation. Here the synthetic trajectory (red) that uses 7-points from object 1 is similar to the real trajectory (black) of object 1. The opposite holds for the trajectory (blue) synthesised from object 2.

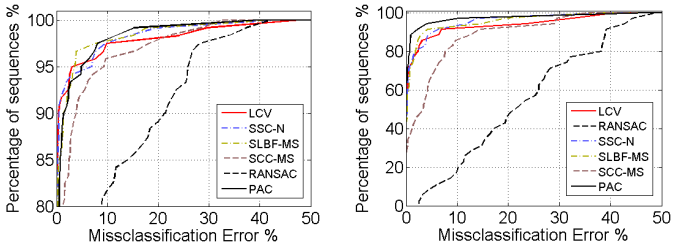

Figure 2: The misclassification error CDFs for 2 motions (left) and 3 motions (right) for the different methods on the Hopkins155

Method	RANSAC	SCC[1]	SLBF[7]	SSC[2]	PAC[6]	LCV
Average time (sec)	0.387	1.264	10.83	165	952.25	0.93
Total time (sec)	60	196	1680	25620	147600	145
Overall error (%)	9.48	2.70	1.35	1.36	1.24	1.86

Table 1: The average and total runtime for the different methods on the full Hokpins155 dataset (155 sequences), together with their overall misclassification errors (2+3 motions). All implementations in Matlab.

Motion segmentation, is the problem of segmenting an image sequence into regions, each corresponding to a distinct 3d rigid motion. This task may be useful in the domains of video surveillance and tracking, robotic navigation, or as a pre-processing step for high-level scene understanding. We introduce a simple and efficient procedure for rigid motion segmentation, using the theory of "linear combination of views" (LCV) by [5]. LCV states that, under an affine camera model, the set of views depicting a 3d rigid object, can be expressed as a linear combination of 2 views of that object, via the equations:

$$\begin{aligned} x^3 &= a_0 + a_1 x^1 + a_2 y^1 + a_3 x^2 + a_4 y^2 \\ y^3 &= b_0 + b_1 x^1 + b_2 y^1 + b_3 x^2 + b_4 y^2 \end{aligned} \quad , \tag{1}$$

where (x^3, y^3) are the coordinates of the novel third view and (x^1, y^1), (x^2, y^2) are the coordinates of the two existing views. (a, b) are the LCV coefficients which depend on the camera parameters only and are independent of the 3d shape and the feature points chosen. They can be recovered by solving:

$$\begin{bmatrix} x_1^3, \ldots, x_r^3 \\ y_1^3, \ldots, y_r^3 \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 \\ b_0 & b_1 & b_2 & b_3 & b_4 \end{bmatrix} \begin{bmatrix} x_1^1, \ldots, x_r^1 \\ y_1^1, \ldots, y_r^1 \\ x_1^2, \ldots, x_r^2 \\ x_1^2, \ldots, x_r^2 \end{bmatrix} , \tag{2}$$

for $r \geq 5$. This idea is illustrated in Fig 1, where we see that the coordinates of the third, synthetic view, are linearly interpolated from the two views.

We may use the LCV theory for motion segmentation in the following simple way: given a known feature trajectory t of a point p over F frames, we can attempt to synthesise that trajectory $\hat{t}$, using (1) and a small number (for example 7) of corresponding points c in two views (tracked correspondences are provided by standard methods such as [3]). If all c and p lie on the same 3d rigid object, then the synthesis will be accurate and the difference E between t and $\hat{t}$ will be small. Conversely, the synthesis will be invalid and the difference will be large (see Fig. 1). The similarity measure

$$E(p,c) = \left(\|t - \hat{t}\|_H / F + \sigma^2 \right)^{-1/2} , \tag{3}$$

which is a kernelised geometric difference in image space, determines the motion consistency between two trajectories and is used to cluster motion trajectories together. A simplified algorithm is shown in Algorithm 1.

We have experimented with real data from the Hopkins155 public dataset [4]. This dataset contains 155 rigid (general, articulated and degenerate) motion sequences of 2 and 3 objects, with manually refined, extracted feature trajectories. For all our tests, the only parameter assumed known is the number of motions k. The rest are the number of 7-point clusters, here chosen as $C = 100 \cdot k$, and the kernel width σ in (3), which is automatically determined. We have tested our method (LCV) by performing 500 runs on the whole database, and compared against the top performing methods in literature.

The misclassification error CDFs for 2 and 3 motions are presented in Fig. 2. We can see that the LCV approach performs very well, close to the state-of-the-art methods. However, where LCV really excels, is in the segmentation speed, due to the fact that the algorithm mostly contains low-rank matrix inversions eq. (2) and simple summations eq. (1) and (3). We show the execution time (total and per sequence average), in seconds, for a single run of the Hopkins155 dataset in Table 1. Our method has amongst the best combined speed and accuracy values, meaning that it is well suited for real-time SfM, navigation and tracking applications. Possible future extensions, as with any motion segmentation approach, would be to cope with feature points with noisy or missing trajectories and the existence of outliers, that is trajectories that do not correspond to 3d rigid motions.

Algorithm 1 Motion segmentation (Simplified)

Select C random 7-point samples in any one frame
For each cluster $c \in C$ {
  Pre-calculate (a, b) using (2) at each frame }
For each scene point $p \in N$ {
  For each cluster $c \in C$ {
    Synthesise trajectory $\hat{t}$ of p using (1) at each frame
    Calculate similarity $E(p, c)$ using (3)
Spectral clustering on $A \approx EE^T$

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

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

[3] B. D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *IJCAI*, 1981.

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

[5] S. Ullman and R. Basri. Recognition by linear combinations of models. *IEEE PAMI*, 13(10):992 –1006, 1991.

[6] L. Zappella, E. Provenzi, X. Lladó, and J. Salvi. Adaptive motion segmentation algorithm based on the principal angles configuration. In *ACCV*, volume 6494, pages 15–26, 2011.

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