Sparse motion segmentation using Multiple Six-Point Consistencies

Vasileios Zografos, Klas Nordberg and Liam Ellis

Computer Vision Laboratory, Linköping University, Sweden
{zografos,klas,liam}@isy.liu.se

Abstract. We present a method for segmenting an arbitrary number of moving objects in image sequences using the geometry of 6 points in 2D to infer motion consistency. The method has been evaluated on the Hopkins 155 database and surpasses current state-of-the-art methods such as SSC, both in terms of overall performance on two and three motions but also in terms of maximum errors. The method works by finding initial clusters in the spatial domain, and then classifying each remaining point as belonging to the cluster that minimizes a motion consistency score. In contrast to most other motion segmentation methods that are based on an affine camera model, the proposed method is fully projective.

1 Introduction

Motion segmentation can be defined as the task of separating a sequence of images into different regions, each corresponding to a distinct rigid motion. There are several strategies for solving the motion segmentation problem, some of which are based on first producing a dense motion field, using optical flow techniques, and then analyzing this field. Examples of this approach are [1] where the optic flow is given as a parametric model and the parameters are determined for each distinct object, or the normalised graph cuts by [2].

Other approaches are instead applied to a sparse set of points, typically interest points that are tracked over time, and their trajectories analysed in the image. A common simplifying assumption is that only small depth variations occur and an affine camera model may be used. The problem can then be solved using the factorization method by [3]. This approach has attracted a large interest in recent literature, with the two current state-of-the-art methods, relative to standard datasets such as Hopkins 155 [4], being Sparse Subspace Clustering (SSC) [5] and Spectral Clustering of linear subspaces (SC) [6].

Other common methods in the literature are based on Spectral Curvature Clustering (SCC) [7], penalised MAP estimation of mixtures of subspaces using linear programming (LP) [8], Normalised Subspace Inclusion (NSI) [9], Non-negative Matrix Factorisation (NNMF) [10], Multi-Stage unsupervised Learning (MSL) [11], Local Subspace Affinity (LSA), Connected Component Search (CCS) [12], unsupervised manifold clustering using LLE (LLMC) [13], Agglomerative Lossy Compression (ALC) [14], Generalised Principal Component Analysis (GPCA) [15], or on RANdom SAmple Consensus (RANSAC) [4].

In this paper we describe a motion segmentation method for sparse point trajectories, which is based on the previous work on six point consistency (SPC) [16], but with the additional novelties and improvements: (i) an alternative method for estimating the vector $\mathbf{s}$ (Sec. 2.2), (ii) a new matching score (Sec. 2.3), and (iii) a modified classification algorithm (Sec. 3).

2 Mathematical background

Our proposed method uses the consistent motion in the image plane generated by 6 points located on a rigid 3D object. The mathematical foundation of this theory was formulated by Quan [17] and later extended by other authors [18–20]. A similar idea was presented in [21], and later used for motion segmentation in [16]. [21] shows that the consistency test can be formulated as a constraint directly on the image coordinates of the 6 points and that, similarly to epipolar lines emerging from the epipolar constraint, this 6-point constraint generates 6 lines that each must intersect its corresponding point.

More formally, we consider a set of six 3D points, with homogeneous coordinates $\mathbf{x}_k$, projected onto an image according to the pinhole camera model:

$$\mathbf{y}_k \sim \mathbf{C}\,\mathbf{T}\,\mathbf{x}_k, \quad k = 1, \ldots, 6, \tag{1}$$

where $\mathbf{y}_k$ are the corresponding homogeneous image coordinates, $\mathbf{C}$ is the $3{\times}4$ camera matrix, and $\sim$ denotes equality up to a scalar multiplication. $\mathbf{T}$ is a 4×4 time dependent transformation matrix that rotates and translates the set of 3D points from some reference configuration to the specific observation that produces $\mathbf{y}_k$. This implies that also $\mathbf{y}_k$ is time dependent. The problem addressed here is how we can determine if an observed set of image points $\mathbf{y}_k$ really is given by (1) for a particular set of 3D points $\mathbf{x}_k$ but with $\mathbf{C}$ and $\mathbf{T}$ unknown.

In general, the homogeneous coordinates of the 3D points can be transformed by a suitable 3D homography $\mathbf{H_x}$ to *canonical* homogeneous 3D coordinates $\mathbf{x}'{=}\mathbf{H_x}\,\mathbf{x}$, and similarly, for a particular observation of the image points we can transform them to canonical homogeneous 2D coordinates $\mathbf{y}'_k{=}\mathbf{H_y}\,\mathbf{y}_k$. The canonical coordinates are given by:

$$(\mathbf{x}'_1\,\mathbf{x}'_2\,\mathbf{x}'_3\,\mathbf{x}'_4\,\mathbf{x}'_5\,\mathbf{x}'_6) \sim\sim \begin{pmatrix} 1&0&0&0&1&X \\ 0&1&0&0&1&Y \\ 0&0&1&0&1&Z \\ 0&0&0&1&1&T \end{pmatrix}, (\mathbf{y}'_1\,\mathbf{y}'_2\,\mathbf{y}'_3\,\mathbf{y}'_4\,\mathbf{y}'_5\,\mathbf{y}'_6) \sim\sim \begin{pmatrix} 1&0&0&1&u_5&u_6 \\ 0&1&0&1&v_5&v_6 \\ 0&0&1&1&w_5&w_6 \end{pmatrix}.$$

Here $\sim\sim$ denotes equality up to an individual scalar multiplication on each column. $\mathbf{H_x}$ and $\mathbf{H_y}$ depend on the 3D points $\mathbf{x}_1$,...,$\mathbf{x}_5$ and on the image points $\mathbf{y}_1$,...,$\mathbf{y}_4$, respectively, and after these transformation are made the relation between 3D points and image points is given by $\mathbf{y}'_k{\sim}\mathbf{H_y}\,\mathbf{C}\,\mathbf{T}\,\mathbf{H}_\mathbf{x}^{-1}\,\mathbf{x}'_k$. One of the main results in [17] is that from these transformed coordinates we can compute a set of five *relative invariants* of the image points, denoted i_k, and of the 3D

points, denoted $\tilde{I}_k$, according to:

$$\mathbf{z} = \begin{pmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \end{pmatrix} = \begin{pmatrix} w_6(u_5 - v_5) \\ v_6(w_5 - u_5) \\ u_5(v_6 - w_6) \\ u_6(v_5 - w_5) \\ v_5(w_6 - u_6) \end{pmatrix}, \qquad \mathbf{s} = \begin{pmatrix} \tilde{I}_1 \\ \tilde{I}_2 \\ \tilde{I}_3 \\ \tilde{I}_4 \\ \tilde{I}_5 \end{pmatrix} = \begin{pmatrix} XY - ZT \\ XZ - ZT \\ XT - ZT \\ YZ - ZT \\ YT - ZT \end{pmatrix} \tag{2}$$

such that they satisfy the constraint $\mathbf{z} \cdot \mathbf{s} = i_1 \tilde{I}_1 + i_2 \tilde{I}_2 + i_3 \tilde{I}_3 + i_4 \tilde{I}_4 + i_5 \tilde{I}_5 = 0$.

To realize what this means, we notice that this constraint includes scalars derived from the reference 3D coordinates $\mathbf{x}_k$ (before they are transformed) and observed image points $\mathbf{y}_k$ (after the transformation $\mathbf{T}$ is made), but neither $\mathbf{C}$ nor $\mathbf{T}$ are explicitly included. Therefore, the constraint is satisfied regardless of how we transform the 3D points (or move the camera), as long as they are all transformed by the same $\mathbf{T}$. As long as the observed image coordinates are consistent with (1), the corresponding relative image invariants $\mathbf{z}$ must satisfy the constraint for a fixed $\mathbf{s}$ computed from the 3D reference points. The canonical transformations $\mathbf{H_x}$ and $\mathbf{H_y}$ can conveniently be included into the unknowns $\mathbf{C}$ and $\mathbf{T}$. In short, the above constraint is necessary but not sufficient for the matching between the observed image points and the 3D reference points.

2.1 The 6-point matching constraint

The matching constraint is expressed in terms of the relative invariants $\mathbf{z}$ and $\mathbf{s}$ that have been derived by transforming image and 3D coordinates. In particular, this means that it cannot be applied directly onto the image coordinates, similar to the epipolar constraint. The transformation $\mathbf{H_y}$ is *not* a linear transformation on the homogeneous image coordinates since it also depends on these coordinates (see the Appendix of [17]). If however, we make an explicit derivation of how $\mathbf{z}$ depends on the 6 image points, it turns out that it has a relatively simply and also useful form:

$$\mathbf{z} = \alpha \begin{pmatrix} D_{126}D_{354} \\ D_{136}D_{245} \\ D_{146}D_{253} \\ D_{145}D_{263} \\ D_{135}D_{246} \end{pmatrix}, \qquad \begin{aligned} \alpha &= \frac{D_{123}}{D_{124}D_{234}D_{314}}, \\ D_{ijk} &= (\mathbf{y}_i \times \mathbf{y}_j) \cdot \mathbf{y}_k = \det\left(\mathbf{y}_i\ \mathbf{y}_j\ \mathbf{y}_k\right). \end{aligned} \tag{3}$$

Since $\mathbf{z}$ can be represented as a projective element, the scalar α can be omitted in the computation of $\mathbf{z}$. An important feature of this formulation is that each element of $\mathbf{z}$ is computed as a multi-linear expression in the 6 image coordinates. This can be seen from the fact that each point appears exactly once in the computations of the two determinants in each element of $\mathbf{z}$.

This formulation of $\mathbf{z}$ allows us to rewrite the constraint as $\mathbf{z} \cdot \mathbf{s} = \mathbf{l}_1 \cdot \mathbf{y}_1 = 0$ with

$$\mathbf{l}_1 = \mathbf{l}_{26}D_{354}\tilde{I}_1 + \mathbf{l}_{36}D_{245}\tilde{I}_2 + \mathbf{l}_{46}D_{253}\tilde{I}_3 + \mathbf{l}_{45}D_{263}\tilde{I}_4 + \mathbf{l}_{35}D_{246}\tilde{I}_5 \tag{4}$$

where $\mathbf{l}_{ij}=\mathbf{y}_i\times\mathbf{y}_j$. $\mathbf{l}_1$ depends on the five image points $\mathbf{y}_2$,...,$\mathbf{y}_6$ and on the elements of $\mathbf{s}$. A similar exercise can be made for the other five image points and in general we can write the matching constraint as $\mathbf{z}\cdot\mathbf{s}=\mathbf{l}_k\cdot\mathbf{y}_k=0$ where $\mathbf{l}_k$ depends on $\mathbf{s}$ and five image points: $\{\mathbf{y}_i, i\neq k\}$. With this description of the matching constraint it makes sense to interpret $\mathbf{l}_k$ as the dual homogeneous coordinates of a line in the image plane. To each of the 6 image points, $\mathbf{y}_k$, there is a corresponding line, $\mathbf{l}_k$, and the constraint is satisfied if any of the 6 lines intersects its corresponding image point. The existence of the lines allows us to quantify the matching constraint in terms of the Euclidean distance in the image between a point and its corresponding line. Assuming that $\mathbf{y}_k$ and $\mathbf{l}_k$ have been suitably normalized, their distance is given simply as

$$d_k = |\mathbf{y}_k \cdot \mathbf{l}_k| \tag{5}$$

2.2 Estimation of s

$\mathbf{s}$ can be computed from (2), given that 3D positions are available, but it can also be estimated from observations of the 6 image points based on the constraint. For example, from only three observations of the 5-dimensional vector $\mathbf{z}$, $\mathbf{s}$ can be restricted to a 2-dimensional subspace of $\mathbb{R}^5$. From this subspace, $\mathbf{s}$ can be determined using the internal constraint [17]. This gives in general three solutions for $\mathbf{s}$, that satisfy the internal constraint and are unique except for degenerate cases. This approach was used in [16].

Alternatively, for $B \geq 4$ observations of $\mathbf{z}$ a simple linear method finds $\mathbf{s}$ as a total least squares solution of minimizing $\|\mathbf{Z}\,\mathbf{s}\|$ for $\|\mathbf{s}\|=1$, where $\mathbf{Z}$ is a $B\times 5$ matrix consisting of the observed vectors $\mathbf{z}$ in its rows. $\mathbf{z}$ is then given by the right singular vector of $\mathbf{Z}$ corresponding to the smallest singular value. This approach has the advantage of producing a single solution for $\mathbf{s}$ which, on the other hand, may not satisfy the internal constraint. However, this can be compensated for by including a large number of observations, B, in the estimation of $\mathbf{s}$. This is the estimation strategy we use in this paper and it works well, provided that there are enough images in each sequence.

2.3 Matching score

In the case of motion segmentation we want to be able to consider a set of 6 points, estimate $\mathbf{s}$, and then see how well this $\mathbf{s}$ matches to the their trajectories. The matching between $\mathbf{s}$ and observations of the 6 points over time is measured as follows. For each observation (at time t) of the 6 points $\mathbf{y}_1(t)$,...,$\mathbf{y}_6(t)$ we use $\mathbf{s}$ to compute the 6 corresponding lines, $\mathbf{l}_1(t)$,...,$\mathbf{l}_6(t)$, and then compute the distances d_k from (5). Finally, we compute a matching score $\tilde{E}$ of the 6 point trajectories:

$$\tilde{E}(P_1,\ldots,P_6) = \operatorname*{median}_t \left[d_1^2(t) + \ldots + d_6^2(t)\right]^{1/2}, \tag{6}$$

where P_k denotes image point k, but without reference to a particular image position in a particular frame. The median operation is used here in order to effectively reduce the influence of possible outliers.

```
Create spatial clusters using k-means
foreach point P_k do
    foreach cluster C_j do
        Select 6 points {P_k, P_2^j, ..., P_6^j}.
        Calculate score E(P_k, C_j) from (6).
    end
    Assign P_i to cluster with min(E(P_k, C_j)).
end
Reject inconsistent clusters.
Initial NBC merging.
Final refinement merging.
```

Algorithm 1: Motion segmentation pseudocode.

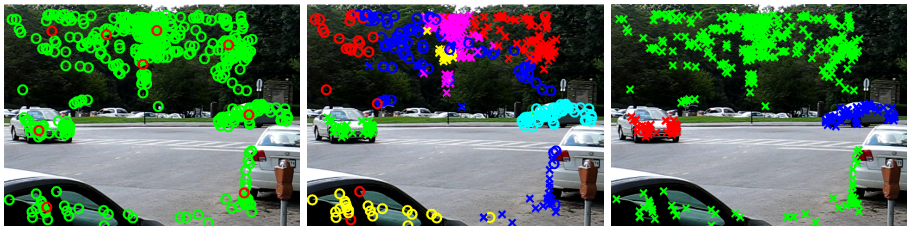

Fig. 1. A K-means initialisation example on the left. On the centre the classification result before the merging, and the final merged results on the left.

3 A motion segmentation algorithm

In this section we describe a simple yet effective algorithm that can be used for the segmentation of multiple moving rigid 3D objects in a scene. The input data is the number of motion segments and a set of N point trajectories over a set of images in an image sequence. Our approach includes: a *spatial initialisation* step for establishing the initial motion hypotheses (or *seed* clusters), from which the segmentation will evolve; a *classification* stage, whereby each tracked point P_k, is assigned to the appropriate motion cluster; and a *merging* step, that combines clusters based on their similarity, to form the final number of moving objects in the scene.

Initialisation: The first step is the generation of initial 6-point clusters, each representing a 3D motion hypothesis. For this we use spatial K-means clustering in the image domain (see Fig. 1). The initial clustering is carried out in an arbitrary frame from each sequence (usually the first or the last). We define a seed cluster C_j=$\{P_1^j$,...,$P_I^j\}$ as the I points at minimum distance to each K-means center. From the subsequent computations it is required that $I \geq 5$, and we use I=6.

Point classification: Following the initialisation step, we assign the remaining points to the appropriate seed cluster. For each of the unclassified points P_k and for each seed cluster C_j, we estimate $\mathbf{s}$ according to Sec. 2.2 and compute a point-to-cluster score E from (6) as $E(P_k, C_j)$=$\tilde{E}(P_k, P_2^j$,...,$P_6^j)$. This gives $M(N\text{-}6M)$ score calculations in total, and produces an $M \times (N\text{-}6M)$ matrix $\mathbf{A}$=$[a_{ik}]$, with column k referring to particular point P_k and element a_{ik} as the

index of the cluster that has the i-th smallest score relative to P_k. We employ a "winner takes all" approach with P_k assigned to the cluster that produces the lowest score, i.e., to the cluster index a_{1k}. This implies that the clusters will grow during the classification step, however, it should be noted that the scores for a particular point are always computed relative to the seed clusters. Note also that there is no threshold associated with the actual classification stage. A typical classification result can be seen in Fig. 1. The growth of the clusters is independent of the order that the points are classified, so the latter may be considered in parallel, leading to a very efficient and fast implementation.

Cluster merging and rejection This is the final stage of our method, and results in the generation of motion consistent clusters each associated with a unique moving object in the scene. This stage consists of a quick *cluster rejection* step; an *initial merging* step using redundant classification information; and a final merging or *refinement* step where intermediate clusters are combined using agglomerative clustering based on some similarity measure.

-*Cluster rejection:* Any clusters that contain very few points (e.g. $\leq$7) are indicative of seed initialisation between motion boundaries, and represent unique and erroneous motion hypotheses. Therefore, any such clusters are promptly removed and their points re-classified with the remaining clusters.

-*Initial merging:* A direct result of the classification in Sec. 3 is the matrix $\mathbf{A}$, where so far we have only used the top row in order to classify points. However, $\mathbf{A}$ provides also information on cluster similarity, which we can exploit to infer initial merge pairings. We call this "Next-Best Classification" (NBC) merging and we now look at the cluster with the second best score for each point, since it contains enough discriminative power to accurately merge clusters. NBC merging involves generating the zero-diagonal sparse symmetric $M\times M$ matrix $\mathbf{L}=[l_{ij}]$ that contains the merging similarity between the clusters. Its elements are defined as:

$$l_{ij} = \sum_{k=1}^{N-6M} \left[\frac{1(k,i,j)}{E(P_k, C_j)} + \frac{1(k,j,i)}{E(P_k, C_i)} \right], \tag{7}$$

where the summation is made over the $N-6M$ points not included in the seed clusters. $1(k,i,j)$ is an indicator function that takes the value 1 when $a_{1k}=i$ and $a_{2k}=j$ and 0 otherwise. In other words, this function is =1 iff P_k is assigned to cluster i and has cluster j as second best option.

The matrix $\mathbf{L}$ describes all the consistent pairings inferred by the NBC merging. However, since usually inconsistent clusters will generate non-zero entries in $\mathbf{L}$ we need to threshold out low response entries due to noise. Using a threshold τ we obtain the sparser *adjacency matrix* $\mathbf{L}^*$. From $\mathbf{L}^\star$ we can then construct an undirected graph G which contains the intermediate clusters as disconnected sub-graphs. If $\mathbf{L}^*$ is insufficient to provide the final motion clusters, due to for example noisy data, then a final refinement step may be required. The result of the cluster rejection and initial merging steps is a set of $\tilde{M}\leq M$ clusters $\tilde{C}_1$,...,$\tilde{C}_{\tilde{M}}$.

-*Refinement merging:* The last step involves the merging of the intermediate clusters, (resulting from the NBC merging), into the final clusters each representing a distinct motion hypothesis. This is achieved by pairwise agglomerative

	GPCA	LSA	RANSAC	MSL	ALC	SSC	SCC	SPC	SC	LP	NNMF	NSI	LLMC	CCS	**MSPC**
Checkerboard: 78 sequences															
Mean:	6.09	2.57	6.52	4.46	1.55	1.12	1.77	4.49	0.85	3.21	-	3.75	4.37	16.37	**0.41**
Median:	1.03	0.27	1.75	0.00	0.29	0.00	0.00	3.69	0.00	0.11	-	-	0.00	10.62	0.00
Traffic: 31 sequences															
Mean:	1.41	5.43	2.55	2.23	1.59	**0.02**	0.63	0.22	0.90	0.33	0.1-	1.69	0.84	5.27	0.09
Median:	0.00	1.48	0.21	0.00	1.17	0.00	0.14	0.00	0.00	0.00	0.–	-	0.00	0.00	0.00
Articulated: 11 sequences															
Mean:	2.88	4.10	7.25	7.23	10.70	**0.62**	4.02	2.18	1.71	4.06	10.–	8.05	6.16	17.58	0.95
Median:	0.00	1.22	2.64	0.00	0.95	0.00	2.13	0.00	0.00	0.00	2.6-	-	1.37	7.07	0.00
All: 120 sequences															
Mean:	4.59	3.45	5.56	4.14	2.40	0.82	1.68	3.18	0.94	2.20	-	-	3.62	12.16	**0.37**
Median:	0.38	0.59	1.18	0.00	0.43	0.00	0.07	1.08	0.00	0.00	-	-	0.00	0.00	0.00

Table 1. 2 motion results

clustering and a maximum similarity measure between clusters. Assume that we wish to merge two clusters, say $\tilde{C}_1$ and $\tilde{C}_2$. We can generate K 6-point mixture clusters $\tilde{C}'$ by randomly selecting 3 points each from $\tilde{C}_1$ and $\tilde{C}_2$. If $\tilde{C}_1$ and $\tilde{C}_2$ belong to the same motion-consistent object and there is little noise present, we expect the scores $\tilde{E}$ calculated for each selection of C' to be grouped near zero, with little variation and few outliers. Conversely, if $\tilde{C}_1$ and $\tilde{C}_2$ come from different objects, $\tilde{E}$ should exhibit a larger dispersion and be grouped further away from zero. Instead of defining the similarity based on location and dispersion of sample statistics, we fit a parametric model to the sample data (using Maximum Likelihood Estimation) and compute the statistics from the model parameters. This allows for a much smaller number of samples and a more accurate estimate than what can be obtained from sample statistics (e.g. mean and variance). Given therefore that the scores in (6) should generally group around a median value with a few extremal outliers and assuming that the distances d_k in (5) are i.i.d., then the score distribution may be well approximated by a Generalised Extreme Value (GEV) distribution [22]. A robust indication of average location in a data sample with outliers is the mode, which for the GEV model can be computed by:

$$\widetilde{m} = \mu + \sigma \left[(1+\xi)^{-\xi} - 1\right] / \xi \quad \text{for} \;\; \xi \neq 0, \tag{8}$$

where μ, σ and ξ are the location, scale and shape parameters respectively recovered by the MLE. Using this as a similarity metric we can merge two clusters when (8) is small or reject them when it is large. The clustering proceeds until we reach the pre-defined number of motions in the scene. The overall method is included in pseudocode in Algorithm 1.

4 Experimental results

We have carried out experiments on real image sequences from the Hopkins 155 database [4]. It includes motion sequences of 2 and 3 objects, of various degrees

	GPCA	LSA	RANSAC	MSL	ALC	SSC	SCC	SPC	SC	LP	NNMF	NSI	LLMC	CCS	**MSPC**
Checkerboard: 26 sequences															
Mean:	31.95	5.80	25.78	10.38	5.20	2.97	6.23	10.71	2.15	8.34	-	2.92	10.70	28.63	**1.43**
Median:	32.93	1.77	26.01	4.61	0.67	0.27	1.70	9.61	0.47	5.35	-	-	9.21	33.21	1.25
Traffic: 7 sequences															
Mean:	19.83	25.07	12.83	1.80	7.75	0.58	1.11	0.73	1.35	2.34	**0.1-**	1.67	2.91	3.02	0.71
Median:	19.55	23.79	11.45	0.00	0.49	0.00	1.40	0.73	0.19	0.19	0.–	-	0.00	0.18	0.36
Articulated: 2 sequences															
Mean:	16.85	7.25	21.38	2.71	21.08	**1.42**	5.41	6.91	4.26	8.51	15.–	6.38	5.60	44.89	2.13
Median:	28.66	7.25	21.38	2.71	21.08	0.00	5.41	6.91	4.26	8.51	15.–	-	5.60	44.89	2.13
All: 35 sequences															
Mean:	28.66	9.73	22.94	8.23	6.69	2.45	5.16	8.49	2.11	7.66	-	-	8.85	26.18	**1.32**
Median:	28.26	2.33	22.03	1.76	0.67	0.20	1.58	8.36	0.37	5.60	-	-	3.19	31.74	1.17

Table 2. 3 motion results

of classification difficulty and is corrupted by tracking noise, but without any missing entries or outliers. Typicall parameter settings for these experiments were: M=10-40 K-means clusters at the first or last frame of the sequence, reject clusters of $\leq$7 points, and K=50-100 mixture samples for the final merge (where necessary). Our results for 2 and 3 motions and the whole database are presented and compared with other state-of-the-art and baseline methods in Tables 1–3.

Our approach (Multiple Six Point Consistency - MSPC) outperforms every other method in the literature overall, in 2 and 3 motions and for all sequences combined. We achieve an overall classification error of 0.37% for two motions, less than 1/2 than the best reported result (SSC); an overall error of 1.32% for three motions, about 2/3 of the best reported result (SC); and an overall error of 0.59% for the whole database, less than 1/2 than the best reported result (SC). We also come first for the checkerboard sequences constituting the majority of the data, with almost 1/2 the classification errors reported by the SC method. For the articulated and traffic sequences (which are problematic for most methods) we perform well, coming a very close second to the best performing SSC or NNMF.

From the cumulative distributions in Fig. 2 we see that our method outperforms all others (where available) with only the SSC being slightly better (between 0.5-1% error) for 20-30% of the sequences. However, SSC soon degrades quite rapidly for the remaining 5-20% of the data with an error differential between 15-35% relative to MSPC. Furthermore, our method degrades gracefully from 2 to 3 motions as we do not have misclassification errors greater than 5% for any of the sequences, unlike SSC which produces a few errors between 10-20% and 40-50%. This is better illustrated in the histograms in Fig. 3.

5 Conclusion

We have presented a method for segmenting moving objects using the geometry of 6 points to infer motion consistency. Our evaluations on the Hopkins 155

	GPCA	LSA	RANSAC	MSL	ALC	SSC	SCC	SPC	SC	LP	NNMF	NSI	LLMC	CCS	**MSPC**
Checkerboard: 104 sequences															
Mean:	*12.55*	*3.37*	*11.33*	*5.94*	2.47	*1.58*	*2.88*	6.05	1.17	*4.49*	-	*3.54*	*5.95*	*19.43*	**0.66**
Median:	-	-	-	-	0.31	-	-	5.27	0.00	-	-	-	-	-	0.25
Traffic: 38 sequences															
Mean:	*4.80*	*9.04*	*4.44*	*2.15*	2.77	***0.12***	*0.71*	0.31	0.98	*0.70*	***0.1-***	*1.68*	*1.22*	*4.85*	0.20
Median:	-	-	-	-	1.10	-	-	0.00	0.00	-	-	-	-	-	0.00
Articulated: 13 sequences															
Mean:	*5.02*	*4.58*	*9.42*	*6.53*	13.71	***0.74***	*4.23*	2.91	2.10	*4.74*	*10.76*	*7.79*	*6.07*	*21.78*	1.13
Median:	-	-	-	-	3.46	-	-	0.00	0.00	-	-	-	-	-	0.00
All: 155 sequences															
Mean:	10.34	4.94	9.76	5.03	3.56	1.24	*2.46*	4.38	1.20	*3.43*	-	-	*4.8*	*15.32*	**0.59**
Median:	2.54	0.90	3.21	0.00	0.50	0.00	-	1.95	0.00	-	-	-	-	-	0.00

Table 3. All motion results (italics are approximated from Tables 1 and 2)

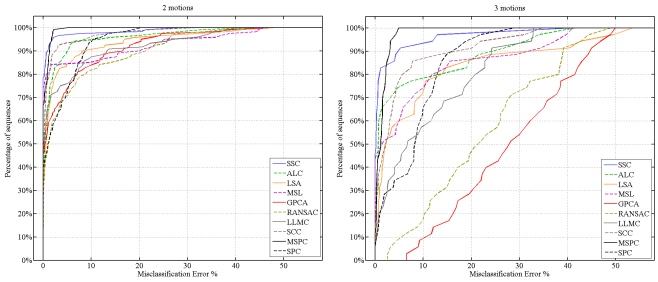

Fig. 2. Cumulative distributions of the errors per sequence for two and three motions.

database have shown superior results than current state-of-the-art methods, both in terms of overall performance and in terms of maximum errors. The method finds initial cluster seeds in the spatial domain, and then classifies points as belonging to the cluster that minimizes a motion consistency score. The score is based on a geometric matching error measured in the image, implicitly describing how consistent the motion trajectories of 6 points are relative to a rigid 3D motion. Finally, the resulting clusters are merged by agglomerative clustering using a similarity criterion.

References

1. Black, M.J., Jepson, A.D.: Estimating Optical Flow in Segmentated Images Using Variable-Order Parametric Models With Local Deformations. PAMI **18** (1996) 972–986
2. Shi, J., Malik, J.: Normalized cuts and image segmentation. PAMI **22** (2000) 888–905
3. Tomasi, C., Kanade, T.: Shape from motion from image streams under orthography: A factorization method. IJCV **9** (1992) 137–154

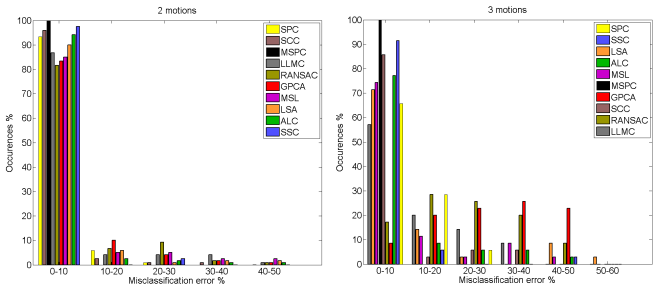

Fig. 3. Histograms of the errors per sequence for two and three motions.

4. Tron, P., Vidal, R.: A Benchmark for the Comparison of 3-D Motion Segmentation Algorithms. In: CVPR. (2007)
5. Elhamifar, E., Vidal, R.: Sparse Subspace Clustering. In: CVPR. (2009)
6. Lauer, F., Schnorr, C.: Spectral clustering of linear subspaces for motion segmentation. In: ICCV. (2009)
7. Chen, G., Lerman, G.: Motion Segmentation by SCC on the Hopkins 155 Database. In: ICCV. (2009)
8. Hu, H., Gu, Q., Deng, L., Zhou, J.: Multiframe motion segmentation via penalized map estimation and linear programming. In: BMVC. (2009)
9. da Silva, N.M.P., Costeira, J.: The normalized subspace inclusion: Robust clustering of motion subspaces. In: ICCV. (2009)
10. Cheriyadat, A.M., Radke, R.J.: Non-negative matrix factorization of partial track data for motion segmentation. In: ICCV. (2009)
11. Sugaya, Y., Kanatani, K.: Geometric Structure of Degeneracy for Multi-body Motion Segentation. In: SMVC. (2004)
12. Roweis, S., Saul, L.: Think globally, fit locally: unsupervised learning of low dimensional manifolds. J. Mach. Learn. Res. **4** (2003) 119–155
13. Goh, A., Vidal, R.: Segmenting motions of different types by unsupervised manifold clustering. CVPR (2007) 1–6
14. Rao, S.R., Tron, R., Vidal, E., Ma, Y.: Motion Segmentation via Robust Subspace Separation in the Presence of Outlying, Incomplete, or Corrupted Trajectories. In: CVPR. (2008)
15. Vidal, R., Tron, R., Hartley, R.: Multiframe Motion Segmentation with Missing Data Using PowerFactorization and GPCA. IJCV **79** (2008) 85–105
16. Nordberg, K., Zografos, V.: Multibody motion segmentation using the geometry of 6 points in 2d images. In: ICPR. (2010)
17. Quan, L.: Invariants of Six Points and Projective Reconstruction From Three Uncalibrated Images. PAMI **17** (1996) 34–46
18. Carlsson, S.: Duality of Reconstruction and Positioning from Projective Views. In: Workshop on Representations of Visual Scenes. (1995)
19. Weinshall, D., Werman, M., Shashua, A.: Duality of multi-point and multi-frame Geometry: Fundamental Shape Matrices and Tensors. In: ECCV. (1996)
20. Torr, P.H.S., Zisserman, A.: Robust parameterization and computation of the trifocal tensor. IVC **15** (1997) 591–605
21. Nordberg, K.: Single-view matching constraints. In: ISVC. (2007)
22. Leadbetter, M.R., Lindgreen, G., Rootzn, H.: Extremes and related properties of random sequences and processes. New York (1983)