Effective Sampling: Fast Segmentation Using Robust Geometric Model Fitting

Ruwan Tennakoon, Alireza Sadri, Reza Hoseinnezhad, and Alireza Bab-Hadiashar, *Senior Member, IEEE*

***Abstract*—Identifying the underlying models in a set of data points that is contaminated by noise and outliers leads to a highly complex multi-model fitting problem. This problem can be posed as a clustering problem by the projection of higher-order affinities between data points into a graph, which can be clustered using spectral clustering. Calculating all possible higher-order affinities is computationally expensive. Hence, in most cases, only a subset is used. In this paper, we propose an effective sampling method for obtaining a highly accurate approximation of the full graph, which is required to solve multi-structural model fitting problems in computer vision. The proposed method is based on the observation that the usefulness of a graph for segmentation improves as the distribution of the hypotheses that are used to build the graph approaches the distribution of the actual parameters for the given data. In this paper, we approximate this actual parameter distribution by using a kth-order statistics-based cost function, and the samples are generated using a greedy algorithm that is coupled with a data sub-sampling strategy. The experimental analysis shows that the proposed method is both accurate and computationally efficient compared to the state-of-the-art robust multi-model fitting techniques. The implementation of the method is publicly available from https://github.com/RuwanT/model-fitting-cbs.**

***Index Terms*—Model-fitting, Spectral clustering, Data segmentation, motion segmentation, Hyper-graph**

I. Introduction

The robust fitting of geometric models to data that are contaminated with both noise and outliers is a well-studied problem with many applications in computer vision [1]–[4]. Visual data often contain multiple underlying structures, and there are both pseudo-outliers (measurements that represent structures other than the structure of interest [5]) and gross outliers (which are produced by errors in the data generation process). Fitting models to this combination of data involves solving a highly complex multi-model fitting problem. This multi-model fitting problem can be viewed as a combination of two sub-problems: *data labelling* and *model estimation*. Although solving one of the sub-problems when the solution to the other is given is straightforward, solving both problems simultaneously remains challenging.

Traditional approaches to multi-model fitting were based on the fit-and-remove strategy: apply a high-breakdown robust estimator (e.g., RANSAC [1] or a least k-th-order residual) to generate a model estimate and remove its inliers to prevent the estimator from converging to the same structure again. This approach is not optimal because any errors made in the initial stages tend to make the subsequent steps unreliable (e.g., small structures can be absorbed by models that are created by accidental alignment of outliers with several structures) [6]. To address this issue, two types of solutions are proposed. The first group is based on using energy minimization techniques in which a cost function consisting of a combination of data fidelity and model complexity (number of model instances) terms is optimized. The second approach is to use clustering methods on the parameters of putative solutions for the whole problem. These two approaches are explained here.

In the energy minimization approach, a cost function in terms of a compromise between the goodness and parsimony of the solution [7] is optimized to simultaneously recover the number of structures and their data association. Commonly, such cost functions are optimized using discrete optimization methods (metric labeling [2]). They start from a large number of proposed hypotheses and gradually converge to the true models. The outcome of those methods depends on the balance between the terms in the cost function (controlled by an input parameter) and the quality of the initial hypotheses. Sensitivity to the parameters included in the sum of terms with different dimensions is also an issue associated with the application of several other subspace learning and clustering methods. For instance, Robust-PCA [8] splits the data matrix into a low-rank matrix and a sparse error matrix. The aim is to minimize the cost function (which is a norm of the error matrix) while it is regularized by the rank of the representation matrix. In factorization methods such as [9], the low-rank representation is obtained by learning a dictionary and coefficients for each data point. The effect of regularization is included using a parameter. These parameters often depend on the noise scales, the complexity of structures and even the number of underlying structures and their data points. As such, these variables vary among datasets, which limits the application of these methods.

Another approach to multi-model fitting is to pose the problem as a clustering problem [10] [11]. In this approach, the idea is that a pure sample (members of the same structure) of the observed data from a cluster can be represented by a linear combination of other data points from the same cluster. Then, the relations of all points to each sampled subset can encode the relations among data points. For example, Sparse Subspace Clustering (SSC) [3] tries to find a sparse block-diagonal matrix that relates data points in each cluster. The optimization task in this work is to minimize the error and the L_1-norm of this latent sparse matrix. In contrast, the regularization term in LRR [12] uses the nuclear norm of this sparse matrix. Recently, [13] presented a deterministic analysis of LRR and suggested that the regularization parameter can be estimated based on the number of data points. Although this improves the speed and accuracy of those methods, it

R.B. Tennakoon, A. Sadri, R. Hoseinnezhad and A. Bab-Hadiashar are with the School of Engineering, RMIT University, Melbourne, Australia.
E-mail: {ruwan.tennakoon, s3391149, reza.hoseinnezhad, abh}@rmit.edu.au

remains unclear what would happen when the number of data points is very large (similar to the databases studied in this work). We should note that methods such as LRSR [14] and CLUSTEN [15], which have more constraints for the regularization and, therefore, more parameters, have also been proposed. A similar strategy is also used to solve the problem of Global Dimension Minimization in [16], which is used to estimate the fundamental matrix for the problem of two-view motion segmentation. The method is somewhat more accurate than LRR and SSC but is computationally expensive. The abovementioned clustering-based methods generally adopt different norms for describing noise, which is equivalent to assuming that the data are corrupted by specific types of noise. In [17], the authors used a mixture of Gaussians to model noise that is more complex. In this approach, the search is initialized with a few Gaussians and the parameters of the mixture are obtained through Expectation-Maximization steps. When the number of Gaussians is too low, the noise cannot be characterized properly and structures may be missed. Increasing the number of Gaussians is computationally expensive for the EM part.

A widely used clustering method is Spectral Clustering [18]. Spectral clustering, which is based on eigen-analysis of a pairwise similarity graph, finds a partitioning of the similarity graph such that the data points in different clusters have very low similarities and the data points within a cluster have high similarities. A simple measure of similarity between a pair of points that lie on a vector field is the Euclidean distance. However, measures that are based on only two points will not work when the problem is to identify data points that are explained by a known structure with multiple degrees of freedom. For instance, in a 2D line fitting problem, any two points will perfectly fit a line, regardless of their underlying structure. Hence, a similarity cannot be derived by using only two points. In such cases, an effective similarity measure can be devised using higher-order affinities (for example, in the 2D line fitting problem, the least-square error among three or more points will provide a suitable affinity measure [10]).

There are several methods for representing higher-order affinities using either a hyper-graph or a higher-order tensor. Since spectral clustering cannot be applied directly to those higher-order representations, they are commonly projected to a graph (which is discussed further in Section II). Moreover, the number of elements in a higher-order affinity tensor (or number of edges in a hyper-graph) will increase exponentially with the order of the affinities (h), which is directly related to the complexity of the model (p). Hence, for complex models, it would not be computationally feasible (in terms of memory utilization or computation time) to generate the full affinity tensor (or hyper-graph), even for a moderately sized dataset. The most commonly used method for overcoming this problem is to use a sampled version of the full tensor (or hyper-graph) that is obtained by using random sampling [11], [10]. The information content of the projected graph heavily depends on the quality of the samples that are used [19], [20], [21] and we analyze this behavior in Section II.

In this paper, we propose an efficient sampling method, called cost-based sampling (CBS), for obtaining a highly accurate approximation of the full graph that is required to solve multi-structural model fitting problems in computer vision. The proposed method is based on the observation that the usefulness of a graph for segmentation improves as the distribution of hypotheses (used to build the graph) approaches the actual parameter distribution for the given data.

This basic approach can be implemented with different choices of cost functions and optimization methods. The choice of optimization method mostly determines the speed and the choice of cost function affects the accuracy. For example, LBF [22] attempts to improve the generated samples of the cost function (chosen to be the β-number of the residuals of a model) by guiding the samples and increasing their size. Its optimization method is slower than our proposed method and the chosen cost function is very steep around the structures, which makes the initialization of the method very difficult and can lead to missing structures. The recipe for overcoming these shortcomings is based on using an extra constraint, such as spatial contiguity, to ensure the purity of the samples before increasing their sizes. In this paper, we approximate this actual parameter distribution using the kth-order cost function, which, in turn, enables us to generate samples using a greedy algorithm that incorporates a faster optimization method. The advantage of the proposed method is that it only uses information in the data with respect to a putative model and does not require any additional assumptions such as spatial smoothness.

The rest of this paper is organized as follows: Section II discusses the use of clustering techniques for robust model fitting and the need for better sampling methods. Section III describes the proposed method in detail and Section IV presents the results of experiments on real data and comparisons with state-of-the-art model-fitting techniques. Additional discussion regarding the merits and shortcomings of the method is presented in Section V, followed by a conclusion in Section VI.

II. Background

Consider the problem of clustering data points $X = [\mathbf{x}_i]_{i=1}^{N}; \mathbf{x}_i \in \mathbb{R}^d$, assuming that there are underlying models (structures) $\Theta = \left[\theta^{(j)}\right]_{j=1}^{m}; \theta^{(j)} \in \mathbb{R}^p$ that relate some of those points. Here, N is the number of data points, and m is the number of structures in the dataset, with the zeroth structure assigned for outliers. Clustering a dataset in such a way that elements of the same group have higher similarity than elements in different groups is a well-studied problem with attractive solutions such as spectral clustering. Spectral clustering operates on a pairwise undirected graph with an affinity matrix, which is denoted as $\mathbf{A}$, that contains affinities between pairs of points in the dataset. As explained earlier, for model fitting applications, only higher-than-pairwise-order affinities provide a useful similarity measure and spectral clustering cannot be directly applied to higher-order affinities.

Agrawal *et al.* [10] introduced an algorithm in which the higher-order affinities (in multi-structural multi-model fitting problems) were represented as a hyper-graph. They proposed a two-step approach for partitioning a hyper-graph with $h = p + 1$ (p is the number of parameters of the model) affinities.

In the first step, the hyper-graph was approximated with a weighted graph using the clique averaging technique. Then, the resulting graph was segmented using spectral clustering. Constructing a hyper-graph with all possible $p+1$ edges is very expensive. As such, they used a sampled version of the hyper-graph that was constructed by random sampling.

Govindu [11] posed the same problem in a tensor-theoretic approach in which the higher-order affinities were represented as an h-dimensional tensor, which was denoted as $\mathcal{P}$. Using the relationship between the higher-order SVD (HOSVD) of the h-mode representation and the eigenvalue decomposition, [11] showed that the super-symmetric tensor $\mathcal{P}$ (in which the similarity does not depend on the ordering of points in the h-tuple) can be decomposed into a pairwise affinity matrix using $\mathbf{A} = \mathbf{P}\mathbf{P}^\top$. Here, $\mathbf{P}$ is the flattened matrix representation[1] of $\mathcal{P}$ along any dimension. The size of the matrix $\mathbf{P}$ is still very large. For example, the size of $\mathbf{P}$ for a similarity tensor that is constructed using h-tuples from a dataset that contains N data points is $N \times N^{h-1}$. As with the hyper-graphs, to make the computation tractable, Govindu [11] suggested using a sampled version of the flattened matrix ($\mathbf{H} \approx \mathbf{P}$). Each column of $\mathbf{H}$ was obtained using the residuals of a model (θ) that was estimated using $h-1$ randomly selected data points. In the remainder of the text, we adopt this tensor-theoretic approach.

The sampling strategy used to construct the sample matrix $\mathbf{H}$ critically affects the clustering and, thus, the overall performance of the model fitting solution.

A. *Why is the distribution of samples important?*

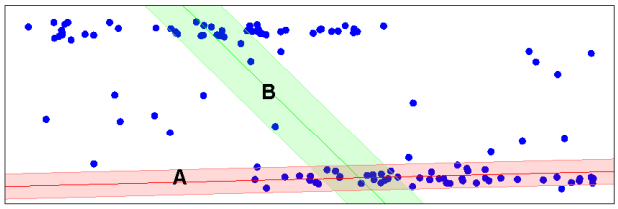
In the tensor-theoretic approach, pairwise affinity matrix $\mathbf{A}$ is constructed by multiplying the matrix $\mathbf{H}$ with its transpose, where $h_{i,l} = e^{-r^2_{\theta_l}(i)/2\sigma^2}$, $r^2_{\theta_l}(i)$ is the squared residual of point i for model θ_l (obtained by fitting to a tuple τ_l) and σ is a normalization constant.

$$\mathbf{A}_{[N\times N]} = \mathbf{H}\mathbf{H}^\top = \sum_{l=1}^{n_H} \underbrace{\left[\mathbf{H}^{(l)}\mathbf{H}^{(l)\top}\right]}_{\mathbf{A}^{(l)}_{[N\times N]}} \tag{1}$$

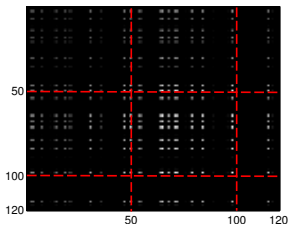
where $\mathbf{H}^{(l)}$ is the l^{th} column of $\mathbf{H}$, which corresponds to the hypothesis θ_l; $\mathbf{A}^{(l)}$ is the contribution of hypothesis θ_l to the overall affinity matrix ($\mathbf{A}$); and n_H is the total number of hypotheses.

When a model hypothesis θ_l is close to an underlying structure in the data (Hypothesis A in Figure 1a), the inlier points of that structure have relatively small residuals, and the resulting $\mathbf{A}^{(l)}$ (Figure 1b) has high affinity values between the inliers and low affinity values for all other point pairs (outlier-outlier and outlier-inlier). In contrast, when a model hypothesis θ_l is far (in parameter space) from any underlying structure, the presumption is that the resulting residual is large, thus leading to $\mathbf{A}^{(l)} \approx \mathbf{0}_{[N\times N]}$. However, as shown in Figure 1a (for Hypothesis B), this is not always the case in model fitting. It is highly likely that some data points yield small residuals even for such hypotheses (far from any underlying model), thus leading to high $h_{i,l}$ values. The resulting $\mathbf{A}^{(l)}$ (Figure 1c) has high affinities between unrelated points that can be viewed as noise in the overall graph. The effect of these bad hypotheses can be amplified because the normalization factor, σ, is often overestimated (using robust statistical methods) when the hypothesis θ_l is far (in parameter space) from any underlying structure. It is important to note that if none of the hypotheses (used in constructing the graph) are close to an underlying structure, then the overall graph does not have higher affinities between the data points in that structure and the clustering methods will not be able to segment that structure.

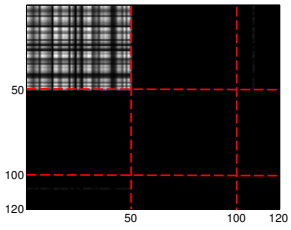
[1]The flattened matrix (P_d) along dimension d is a matrix in which each column is obtained by varying the index along dimension d while holding all other dimensions fixed.



(a) Data



(b) $\mathbf{A}^{(A)}$



(c) $\mathbf{A}^{(B)}$

Fig. 1. Example line fitting scenario on a synthetic dataset containing two lines and multiple outliers. Lines A and B show two model hypotheses, and the shaded areas around the lines indicate the corresponding σ values. (b) and (c) show the contributions of hypotheses A and B, respectively, to the overall graph. The data points are sorted according to their model affiliation, where the first 50 data points belong to line one followed by line two (50 points) and the outliers (20 points). The dashed lines indicate the cluster boundaries.

The above example shows that the sampling process influences the level of noise in the graph. While spectral clustering can tolerate some level of noise, it has been proved that this noise level is related to the size of the smallest cluster that we want to recover (the tolerable noise level increases rapidly with the size of the smallest cluster) [23]. Because model fitting often involves recovering small structures, it is highly important to limit the noise level in the affinity matrix.

For any two data points x_i, x_j, we can write

$$a_{i,j} = \frac{1}{n_H}\sum_{l=1}^{n_H} \underbrace{e^{-\frac{\left(r^2_{\theta_l}(i)+r^2_{\theta_l}(j)\right)}{2\sigma^2}}}_{a^{(\theta_l)}_{i,j}} \xrightarrow[n_H\uparrow]{as} \int P_\theta \cdot a^{(\theta_l)}_{i,j}\, d\theta \tag{2}$$

For any model fitting problem with $p > 2$, there exist an infinite number of models θ_l, where $a^{(\theta_l)}_{i,j} \to 1$. This implies that for any two points, $a_{i,j}$ (according to Equation 2) can be maximized or minimized by choosing P_θ accordingly.

For a graph to have a block diagonal structure that is suitable for clustering, $a_{i,j}$ must be large if both data points x_i and x_j are from the same structure θ_t and small otherwise. If hypotheses are selected from a Gaussian mixture distribution

with sharp peaks around the underlying model parameters, with low density in other places, and with θ_t representing the true underlying structures, then we have

$$P_\theta = \sum_{t=1}^{m} \phi_t \, \mathcal{N}(\theta_t, \Sigma_t). \tag{3}$$

The edge weights approach the following values when $\Sigma_t \rightarrow \mathbf{0}$:

$$a_{i,j} \rightarrow \begin{cases} \phi_t & i \wedge j \in \theta_t \\ 0 & i \wedge j \notin \theta_t \end{cases} \tag{4}$$

A results in a graph that has a block diagonal structure that is suitable for clustering. However, generating sample hypotheses from this distribution is not possible because it is unknown until the problem has been solved.

This point is further illustrated by a simple model fitting experiment that employs a synthetic dataset that contains four lines. Each line contain 100 data points with additive Gaussian noise $\mathcal{N}(0, 0.02^2)$ and 50 gross outliers were also added to those lines. First, 500 hypotheses were generated using uniform sampling, random sampling (using 5-tuples), and the sampling scheme that was proposed in this paper (CBS). Then, these hypotheses were used to generate the three graphs that are shown in Figure 2. As the data are arranged based on the structures' membership, a properly constructed graph should show a block diagonal structure with high similarities between points in the same structure and low similarities for data from different structures. The figure shows that while the CBS method has resulted in a favorable graph for clustering, the other two sampling strategies have produced graphs with little information. The corresponding hypothesis distributions (Figure 2 (e-f)) show that only CBS has generated many hypotheses that are close to the underlying structure.

Govindu [11] used $h-1$ (for affinities of order h) randomly sampled data points and calculated a column of **H** by computing the affinity between those and each point in the dataset. The probability of obtaining a clean sample, which leads to a hypothesis that is close to a true structure in the data, decreases exponentially with the size of the tuple [10]. Hence, it becomes increasingly unlikely to obtain a good graph for models with a large number of parameters using random sampling.

There are several techniques in the literature that try to tackle the clustering problem by tapping into available information regarding the likelihood distribution of good hypotheses. For instance, spectral curvature clustering [19], which is an algorithm that was designed for affine subspace clustering, employs an iterative sampling mechanism that increases the chance of finding good hypotheses. In this scheme, a randomly chosen **H** is used to build a graph, which is partitioned using spectral clustering to generate an initial segmentation of the dataset. Then, data points within each segment of this clustering are used to generate a new set of columns of **H**. This process is repeated several times to improve the final clustering results. In such an iterative scheme, the errors that are made in the initial random stage can bias the overall solution. In contrast, our method does not rely on the previous iterations of the graph in building **H**.

Similarly, Ochs and Brox [20] used higher-order affinities in a hyper-graph setting for motion segmentation of video sequences. In their method, the affinity matrix is obtained using a sampling strategy that is partly random and partly deterministic. The higher-order affinities are based on 3-tuples that are generated by choosing two points randomly. Then, the third points are chosen as a mixture of 12 nearest-neighbor points and 30 random 3rd points.

The previous guided sampling approaches generate the columns of **H** using tuples of minimal size. Purkait *et al.* [21] advocated the use of larger tuples and showed that if those tuples are selected correctly, the hypothesis distribution will be closer to the true model parameters compared to smaller tuples. However, selecting larger all-inlier (correct) tuples using random sampling is highly unlikely. Purkait *et al.* [21] suggested the use of Random Cluster Models (RCM) [24] to improve the sampling efficiency. RCM is based on selecting the tuples iteratively such that at every iteration, the samples are selected using the segmentation results that are obtained by enforcing the spatial smoothness on the results of the previous iteration. This approach is particularly advantageous if the application satisfies the spatial smoothness requirements. Our proposed approach for constructing the affinity matrix without relying on the existence of spatial smoothness is explained in the next section.

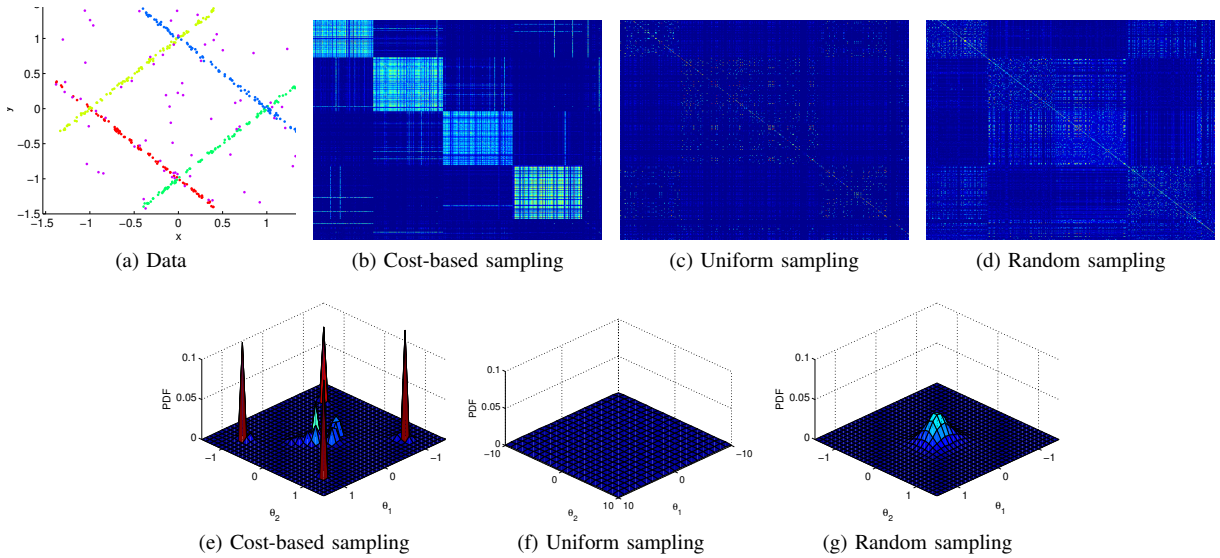
III. Proposed Method

This section describes a new approach for multi-structural model fitting problems. Similar to [10], [11], we approach multi-structural fitting as a clustering problem with the intention of applying spectral clustering. In this approach, the pairwise affinity matrix **A** for spectral clustering is obtained by projecting the higher-order affinity tensor ($\mathcal{P}$) via multiplying an approximated flattened matrix **H** with its transpose. For affinities of order h, each column of **H** is obtained by sampling $h-1$ data points and calculating the affinity of each point to those sampled points. The affinity of a data point i to an $h-1$ tuple is calculated as $e^{-r^2_{\theta_l}(i)/(2\sigma^2)}$, where θ_l is the vector of fitted model parameters to the $h-1$ tuple and σ is the normalization factor. For clarity, in the remainder of this text, an $h-1$ tuple (τ_l) that is used to generate a column of **H** is referred to as an edge, while its corresponding model (θ_l) is called a hypothesis.

As discussed in Section II, the way in which we sample the edges affects the information content of the resulting graph and our ultimate goal is to sample edges in such a way that the distribution of their associated hypotheses resembles the true distribution of the model parameters. While the true distribution of the model parameters for a given dataset $p(\theta \mid X)$ is unknown until the problem is solved, using Bayes' theorem it can be written as follows:

$$p(\theta|X) \propto p(X|\theta)p(\theta) \tag{5}$$

where $p(X|\theta)$ is the likelihood of observing data X under the model θ and $p(\theta)$ is the prior distribution of θ. Given that the prior is uninformative (i.e., any parameter vector is equally likely), the posterior is largely determined by the data



(a) Data (b) Cost-based sampling (c) Uniform sampling (d) Random sampling

(e) Cost-based sampling (f) Uniform sampling (g) Random sampling

Fig. 2. A synthetic dataset containing four line structures is shown in (a), and the graphs produced by the cost-based sampling, random and uniform sampling (-10,10) methods are shown in (b-d), respectively. The respective hypothesis distributions are shown in (e-f). While the CBS method has resulted in a favorable graph for clustering, the other two sampling strategies have produced graphs with little information.

(the posterior is data-driven) and can be approximated by $p(\theta|X) \propto p(X|\theta)$.

A robust objective function is often used in multi-structural model fitting applications to quantify the likelihood of the existence of a structure in the data [5]. On that basis, we argue that it can be a good approximation of the model parameter likelihood. For example, the sample consensus objective function, as employed in RANSAC, is expected to have a peak in places where a true structure is present (in the parameter space) and low values where there are no structures. It should be noted that when there are structures of different size, the sample consensus function associates higher values with larger structures (hence, it is biased toward large structures). In this work, we select the cost function of the least k-th-order statistics (LkOS) estimator as the objective function, as it has been shown to perform stably and with a high breakdown point [25] in various applications and is not biased toward large structures (LkOS is biased toward structures with low variance, which is a desirable property). A modified version of the LkOS cost function used in [26] is as follows:

$$C(\theta) = \sum_{j=k-h+1}^{k} r^2_{i[j|\theta]}(\theta) \tag{6}$$

where $r_i^2(\theta)$ is the i-th sorted squared residual with respect to model θ and $i[j \mid \theta]$ is the index of the j-th sorted squared residual with respect to model θ. Here, k refers to the minimum acceptable size of a structure in a given application, and its value should be significantly larger than the dimension of the parameter space ($k \gg p$). Because the above cost function is designed to have minima around the underlying structures, the model parameter likelihood function can be expressed as

$$P_\theta \propto p(X|\theta) \approx \frac{1}{Z} e^{-C(\theta)}. \tag{7}$$

The above function is highly non-linear, and its evaluation over the entire parameter space, which is required for calculating the normalizing constant Z, is not feasible. The most common approach for sampling from a distribution that can only be evaluated up to a proportional constant on specified points is to use the Markov Chain Monte Carlo (MCMC) method (e.g., by using Metropolis-Hasting algorithm). However, such algorithms require a good update distribution to be effective, and simple update distributions such as random walk would be inefficient and may not traverse the full parameter space [27]. In particular, setting up random walk distributions requires information regarding the span of the model parameters, which is unknown until the problem is solved.

A. Sampling edges using the robust cost function

Using derivatives of the order statistics function in (6), a greedy iterative sampling strategy was proposed in [26] that is intentionally biased toward generating data samples from a structure in the data. Then, this sampling strategy was used to generate putative model hypotheses for tuples of different sizes in conjunction with the fit-and-remove strategy to recover multiple structures in the data [28], [26]. Because the fit-and-remove strategy is susceptible to errors in the initial stages, the sampling had to be reinitialized (randomly) several times to reduce the probability of error propagation in the sequential fit-and-remove stages.

In this paper, we propose a modified version of this iterative update procedure (recalled in Algorithm 1) to generate model estimates (edges) that are close to the peaks of the true parameter density function $p(\theta|X)$. Each edge that is used in constructing the **H** matrix of the proposed method is obtained as follows: Initially, an h-tuple ($h = p + 2$) is selected according to the inclusion weights **w** (as explained later). Using this tuple as the starting point, the following update is run until convergence. A model hypothesis is generated using

the selected tuple and the residuals from each data point to this hypothesis are calculated. Then, these residuals are sorted, and the h points around the kth sorted index are selected as the updated tuple for the next iteration.

In practice, the above update step has the following property: If the current h-tuple is a clean sample (all inliers) from a structure in the data, there is a high probability that the next sample will also be from the same structure because there should be at least k points that agree with each true structure. In contrast, if the current hypothesis is not supported by k points (not a structure in the data), the next hypothesis will be at a distance in the parameter space from the current hypothesis. At worst, this can be thought of as changing the hypothesis randomly. However, it is shown that the residuals of a data structure with respect to an arbitrary hypothesis have a high probability of clustering together in the sorted residual space [29], [4]. As the next h-tuple is selected from the sorted residual space, the probability of selecting points that are associated with a structure at this stage is higher than that of selecting it randomly. The advantage of this approach over the random and guided sampling methods is that those methods do not converge locally and, hence, walk away from a good hypothesis (continue sampling), even if a good sample is found early. Since the proposed update converges locally, it has the ability to stop quickly when a good hypothesis is found.

Following [28], we use the following criterion to decide whether the update procedure has converged to a structure in the data:

$$F_{stop} = \left(r^2_{i[k|\theta_l]}(\theta_l) < \frac{1}{h} \sum_{j=k-h+1}^{k} \underbrace{r^2_{i\left[j|\theta_{(l-1)}\right]}(\theta_l)}_{\text{(a)}} \right) \wedge \left(r^2_{i[k|\theta_l]}(\theta_l) < \frac{1}{h} \sum_{j=k-h+1}^{k} \underbrace{r^2_{i\left[j|\theta_{(l-2)}\right]}(\theta_l)}_{\text{(b)}} \right). \tag{8}$$

Here, (a) and (b) are the squared residuals of the edge points in iterations $l-1$ and $l-2$ with respect to the current parameters θ_l. This criterion checks the data points that are associated with the two previous samples to determine whether the average residuals of those points (with respect to the current parameters) are still lower than the inclusion threshold that is associated with having k points (assuming that a structure has at least k points implies that data points with residuals of less than $r^2_{i[k|\theta_l]}(\theta_l)$ are all inliers). This indicates that the samples selected in the previous three iterations are likely to be from the same structure; hence, the algorithm has converged.

B. Sub-sampling data

Although the above update procedure has a high probability of generating an edge that results in a hypothesis that is close to a peak in $p(\theta|X)$, there is no guarantee that all of the structures in the data will be visited since the update step is reinitialized from random locations. If some of the structures are not visited by the sampling procedure, the resulting graph will not contain the information that is required to identify those structures.

Algorithm 1 Step-by-step algorithm of sample generation ($runCBS\_SG$)

Inputs: Data Points ($X \in [x_i]_{i=1}^{N}$), minimum cluster size (k), T, inclusion weights (**w**)
Output: Final data indices I_l, Scale σ

1: $l_{max} \leftarrow 50$, $h \leftarrow p+2$, $l \leftarrow 0$
2: Select an h-tuple (I_0) from the data points according to weights **w**.
3: Generate model hypothesis θ_0 using the h-tuple I_0.
4: **repeat**
5: $\quad \left[r^2(\theta_l), i[\cdot \mid \theta_l]\right]$ =SortedRes(X, θ_l).
6: $\quad I_{l+1} \leftarrow \left[x_{i[j|\theta_l]}\right]_{j=k-h+1}^{k}$
7: $\quad \theta_{l+1} \leftarrow$ LeastSquareFit(I_{l+1})
8: $\quad$ Evaluate the stopping criterion (F_{stop})
9: $\quad$ **if** F_{stop} **then** break **end if**
10: **until** $(l{+}{+} > l_{max})$
11: $\sigma \leftarrow$ MSSE(X, θ_l, k, T)

To ensure that the algorithm visits all structures in the data, we propose using a data sub-sampling strategy. Each run of the update procedure in Algorithm 1 is executed only on a subset of the data that is selected based on an inclusion weight (**w**). The inclusion weight, which is initialized to one, is designed such that at every iteration, it will give higher importance to data points that are not modeled by the hypothesis used in the previous iterations. This will progressively increase the chance of unmodeled data being included in the sampling process. This idea is similar to the Bagging predictors [30] with boosting [31], [32] in machine learning. In Bagging predictors multiple subsets of data, which are formed by bootstrap replicates of the dataset, are used to estimate the models, which are aggregated to obtain the final model. Boosting improves the bagging process by giving importance to unclassified data points in successive classifiers.

The complete edge generation procedure is as follows: A subset of size N_s is sampled from the data using the inclusion weights **w** without replacement (**w** is normalized in the $sampleData(\cdot)$ function). Then, this sub-sample is used in the update procedure in algorithm 1, which produces an edge. Next, the inclusion weights **w** of the inliers to the above hypothesis are decreased, while the inclusion weights of the remaining points are increased. This process is repeated for a fixed number of iterations. The complete steps of the proposed method (CBS) are listed in Algorithm 2.

The scale of the noise plays a crucial role in the success of segmentation methods. In spectral-clustering-based model fitting methods, the scale is used to convert the residuals to an affinity measure. While most competing algorithms require this as an input parameter [21], [33], the proposed method estimates the scale of the noise from the given data. In this implementation, we selected the MSSE [34] for estimating the scale of the noise. The MSSE algorithm requires a constant threshold T as an input. This threshold defines the inclusion percentage of inliers. Assuming a normal distribution for

Algorithm 2 Step-by-step algorithm of the proposed model-fitting method

Inputs: Data Points ($\mathbf{X} \in [x_i]_{i=1}^{N}$), minimum size of structure (k), number of structures (n_c), number of hypotheses (n_H), $T \leftarrow [2.0 \sim 3.5]$

1: $\mathbf{w} \leftarrow [\frac{1}{N} \dots \frac{1}{N}]_{1\times N}$; $N_s \leftarrow N/n_c$; $w^* \leftarrow \frac{20}{N}$
2: **repeat**
3: Sample N_s data points from X based on inclusion weights $\mathbf{w}$; $[X_s, \mathbf{w}_s] \leftarrow sampleData(X, \mathbf{w})$.
4: $[I_s, \sigma] \leftarrow runCBS\_SG(\mathbf{X}_s, k, T, \mathbf{w}_s)$
5: Calculate residuals ($r^2_{I_s}$) of all data points from the h-tuple I_s.
6: $h_{:,i} \leftarrow exp(-r^2_{I_s}/2\sigma_i^2)$
7: Calculate inliers C_{inl} using r_{I_s}, σ_i.
8: $\mathbf{w} \leftarrow \mathbf{w} \times 2$
9: $\mathbf{w}(C_{inl}) \leftarrow \mathbf{w}(C_{inl}) \div 4$
10: $\mathbf{w}(\mathbf{w} > w^*) \leftarrow 1/N$
11: $\mathbf{w} \leftarrow \mathbf{w}/sum(\mathbf{w})$
12: **until** $i{+}{+} > n_H$
13: $\mathbf{A} \leftarrow \mathbf{H}\mathbf{H}^\top$
14: $[labels] \leftarrow spectralClustering(\mathbf{A}, n_c)$

noise, it is usually set to 2.5; i.e., $T = 2.5$ will include 99% of the normally distributed inliers. Desirable properties of this estimator for dealing with small structures were discussed in [35].

IV. Experimental Results

We have evaluated the performance of the proposed method for multi-object motion segmentation on several well-known datasets. Then, the results of the proposed cost-based sampling (CBS) method were compared with those obtained using state-of-the-art robust multi-model fitting methods. The selected methods use higher-order affinities, namely, Spectral Curvature Clustering (SCC [19], HOSC [21] and OB [20]), or are based on energy minimization (RCMSA [33], PEARL [7] and QP-MF [36]).

The accuracies of all methods were evaluated using the commonly used clustering error (CE) measure [21]:

$$CE = \min_{\Gamma} \frac{\sum_{i=1}^{N} \delta\left(L^*(i) \neq L_r^{\Gamma}(i)\right)}{N} \times 100 \qquad (9)$$

where $L^*(i)$ is the true label of point i, $L_r(i)$ is the label obtained via the method under evaluation and Γ is a permutation of labels. The function $\delta(\cdot)$ returns one when the input condition is true and zero otherwise.

The proposed CBS algorithm was coded in MATLAB (the code is publicly available: https://github.com/RuwanT/model-fitting-cbs) and the results for competing methods were generated using the codes provided by the authors of those works. The experiments were run on a Dell Precision M3800 laptop with an Intel i7-4712HQ processor.

A. Analysis of the proposed method

In this section, we investigate the significance of each part of the proposed algorithm and the effect of its parameters on its accuracy. This analysis was conducted using a two-view motion segmentation problem (see Section IV-B for more details).

We used the "*posters-checkerboard*" sequence from the RAS dataset [37] to evaluate the significance of the main components of the CBS method. This sequence contains three rigid moving objects with 100, 99, and 81 point matches and 99 outlier points. In the first experiment, the matrix $\mathbf{H}$ was generated with edges obtained by pure random sampling (RDM); by the CBS method without the sub-sampling strategy, i.e., with lines 3 and 8-11 removed from Algorithm 2 (CBS-nSS); and by the complete proposed method (CBS). For each sampling method, the number of hypotheses (n_H) was varied, and the mean clustering error and run time were recorded (averaged over 100 runs per value of n_H). Figure 3e shows the variation of the mean clustering error with the sampling time (computing time). The results show that for this problem, accurate identification of models could not be achieved with pure random sampling even when many edges were sampled. It also shows that the sub-sampling strategy of the proposed CBS method significantly contributes toward the accurate and efficient identification of the underlying models in the data. In addition, the inclusion of the sub-sampling step significantly reduces the number of points that must be sorted (step 5 Algorithm 1), which is a bottleneck of the proposed method. The time reduction in sampling 50 edges with CBS compared to CBS-nSS, which is shown in Figure 3e, is mainly due to the low number of sorted data points.

Next, we use the same image sequence to study the variations in accuracy of the proposed method with the value of parameter k. This parameter defines the minimal acceptable size for a structure (in terms of the number of points) in a given application. Here, we vary the value of k from 10 to 80 (CBS use edges of size 10 and the smallest structure in this sequence has only 81 points; hence, any value outside this range is not realistic). The number of hypotheses was set to 100 for both sampling methods. The results that are plotted in Figure 3f show that for CBS-nSS and CBS, the clustering error decreases steeply up to approximately $k = 20$. In CBS-nSS, the CE remains relatively unchanged after that, while in CBS the clustering error starts to increase when k exceeds 40. This behavior can be explained as follows: The CBS method estimates the scale of the noise from the data and the analysis of [35] showed that the estimation of the noise scale from the data requires *at least* 20 data points to limit the effects of finite-sample bias. As such, the CBS method does not have high accuracy when $k < 20$. In addition, the data sub-sampling in CBS reduces the number of points that are available for each run of the sample generator. Hence, the clustering error is increased for large k values. Using large values for k is also not desirable because structures with smaller sizes would be ignored.

Next, we used the same image sequence to show that the proposed local update step (Algorithm 1) can converge to a true structure in the data even when it is initialized with a hypothesis that is far (in parameter space) from any structure. Here, we repeated algorithm 1 10000 times, with each initial h-tuple (in step 2 in Algorithm 1) being selected randomly

from gross outliers that were derived using ground-truth labels. The stopping criterion in Algorithm 1(steps 8 and 9) was omitted in this evaluation. The mean and standard deviation of the cost function value at each iteration across the 1000 random runs are shown in Figure 4. The results were compared with those that were obtained by running the above experiment with the initial h-tuple being selected randomly from true structures. The results show that on average, the convergence of the local update step does not depend on the initialization.

Next, we compared the proposed hypothesis generation process against several well-known sampling methods for robust model fitting (e.g., MultiGS [38] and Lo-RANSAC [39]). These methods are designed to bias the sampling process toward selecting points from a structure in the data. For completeness, we have also included pure spatial sampling (to generate a hypothesis using points that are closer in space and selected via a KDtree) and random sampling. Similar to the proposed method, the hypotheses from these sampling methods were used to generate a graph, which is cut to perform the clustering. Figure 3f shows that the CBS method is capable of generating highly accurate clusterings faster than other sampling methods.

While we have only presented the results for one two-view motion segmentation case, similar trends were observed across all other problems tested in this paper.

B. Two-view motion segmentation

Two-view motion segmentation is the task of identifying the point correspondences of each object in two views of a dynamic scene that contains multiple independently moving objects. If the point matches between the two views are given as $[X_1, X_2]$, where $X_i = (x, y, 1)^\top$ is a coordinate of a point in view i, each motion can be modeled using the fundamental matrix $F \in \mathcal{R}^{3\times3}$ as [40]:

$$X_1^\top F X_2 = 0 \tag{10}$$

The distance from a given model to a point pair can be measured using the Sampson distance [41].

We tested the performance of the CBS method on the Adelaide-RMF dataset [42], which contains key-point matches (obtained using SIFT) of dynamic scenes, together with the ground-truth clustering. The clustering error and the computation time of the CBS method on each sequence, together with those of the competing methods (PEARL, FLOSS, RCMSA and QP-MF)[2], are given in Table I. The results show that in comparison to the competing methods, the proposed method has achieved comparable or better accuracy over all sequences. Moreover, on average, the computation time of the proposed method is approximately 4 times less than that of QP-MF and twice that of RCMSA when its computational bottlenecks are implemented using C (MATLAB MEX), whereas our method is implemented using a simple MATLAB script. One expects implementation in the C language to yield significant improvements in terms of speed.

[2]SSC and SCC are not used here as they are not robust to outliers (especially when the percentage of outliers is not known).

In these experiments, the parameter k of the proposed method was set to $k = min(0.1 \times N, 20)$. The number of samples in QP-MF was set to 200 (determined through trial and error; no significant improvement in accuracy was observed when the number of samples was increased beyond 200 for a test sequence).

C. 3D motion segmentation of rigid bodies

The objective of 3D motion segmentation is to identify multiple moving objects using point trajectories through a video sequence. If the projections (to the image plane) of N points that are tracked through F frames are available, denoted $[x_{f\alpha}]_{\alpha=1\ldots N}^{f=1\ldots F} : x_{f\alpha} \in \mathcal{R}^2$, then [44] has shown that the point trajectories $P_\alpha = [x_{1\alpha}, y_{1\alpha}, x_{2\alpha}, \ldots x_{F\alpha}, y_{F\alpha}]^\top \in \mathcal{R}^{2F}$ that belong to a single rigid moving object are contained within a subspace of $rank \leq 4$ under the affine camera projection model. Hence, the problem of 3D motion segmentation can be reduced to a subspace clustering problem.

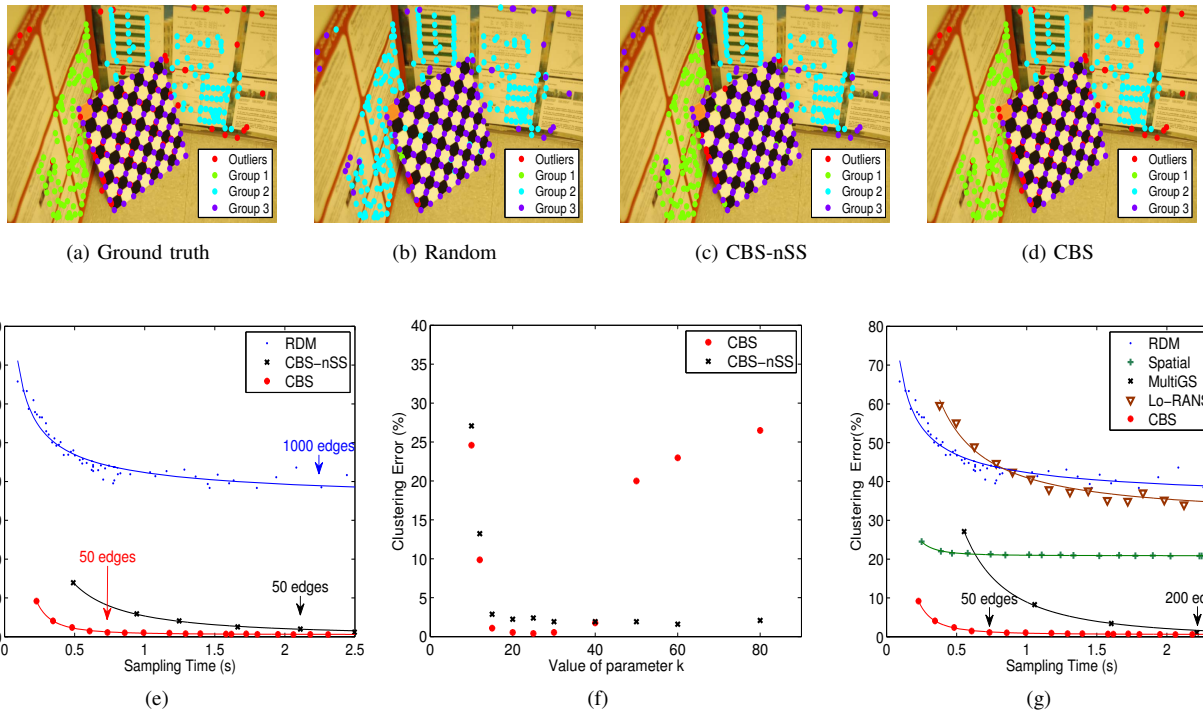
One of the characteristics of subspace segmentation is that the dimension of the subspaces may vary between two and four, depending on the nature of the motions. This means that the model that we are estimating is not fixed. The proposed method, which was not specifically developed to solve this problem (unlike some competing techniques [3]), is not capable of identifying the number of dimensions of a given motion and requires this information as an input. In our implementation, we have used the eigenvalues of the sampled data point to select a dimension d of the model such that $2 \leq d \leq 4$.

We utilized the commonly used "checkerboard" image sequence in the Hopkins 155 dataset [45] to evaluate the CBS algorithm. This dataset contains trajectory information of 104 video sequences that are categorized into two main groups, depending on the number of motions in each sequence (two or three motions).

The clustering error (mean and median) and the computation time for CBS and competing higher-order affinity-based methods are shown in Table II. The results show that CBS has achieved comparable clustering accuracies to those achieved by competing methods while being significantly faster than those methods (specially on 3-motion sequences). For completeness, we have also included the results for methods that are based on energy minimization (PEARL [7], QP-MF [36]) and fit & remove (RANSAC, HMSS [28]), as reported in [36]. To gain a better understanding of the methods (that have high accuracy) across all sequences, we have plotted the cumulative distributions of the errors per sequence in Figure 5a (two-motion sequences) and Figure 5b (three-motion sequences). For algorithms with random elements, the mean error across 100 runs is used.

To provide a qualitative measure of the performance, the final segmentation results of several sequences in the Hopkins 155 dataset, where CBS was both successful and unsuccessful, are shown in Figure 6.

The sequences in the Hopkins 155 dataset are outlier-free. To test the robustness to outliers, we added synthetically generated outlier trajectories to each three-motion sequence of



(a) Ground truth (b) Random (c) CBS-nSS (d) CBS

(e) (f) (g)

Fig. 3. Results for the "*posters-checkerboard*" sequence. 3a shows the ground-truth clustering, and 3b - 3d shows the clusterings that were obtained with RDM, CBS-nSS and CBS at 1 s. 3e and 3f show the variation of the clustering error with time and the value of parameter k, respectively, and 3g shows the variation in the clustering error with the value of parameter k (best viewed in color).

TABLE I
TWO-VIEW MOTION SEGMENTATION RESULTS ON THE ADELAIDE-RMF DATASET. THE MEDIAN CE VALUES OF PEARL AND FLOSS [43] REPORTED IN [33] ARE USED HERE.

	PEARL	FLOSS	*QP-MF*		*RCMSA*		*CBS*	
	Median CE	*Median CE*	*Median CE*	*Time*	*Median CE*	*Time*	*Median CE*	*Time*
biscuitbookbox	8.11	11.58	5.02	4.78	7.72	0.56	**0.00**	0.95
boardgame	16.85	17.92	17.38	4.49	12.09	0.50	**11.28**	0.99
breadcartoychips	12.24	15.82	8.65	4.52	9.97	0.64	**5.63**	0.93
breadcubechips	9.57	11.74	3.04	4.47	9.78	0.54	**0.87**	0.85
breadtoycar	10.24	11.75	6.33	4.20	8.73	0.44	**3.96**	0.75
carchipscube	10.30	16.97	17.27	3.59	4.85	0.42	**2.44**	0.65
cubebreadtoychips	9.02	11.31	2.14	5.07	8.87	0.71	**1.91**	1.13
dinobooks	19.17	20.28	17.92	5.20	17.50	0.73	**12.98**	1.25
toycubecar	12.00	13.75	14.50	3.71	**11.00**	0.38	19.19	0.70

the Hopkins 155 dataset[3]. The clustering results of the CBS method and those that were obtained by the best-performing method (SCC) are plotted in Figure 5c. The results show that CBS was able to achieve high accuracy in the presence of outliers on more sequences. The SSC algorithm is not designed to handle outliers and, therefore, was not included in this analysis.

D. Long-term analysis of moving objects in video

The point trajectories of the "Hopkins155" dataset, which was used in the above analysis, are hand-tuned (i.e., the point trajectories of each sequence are cleaned by a human such that they do not contain gross outliers or incomplete trajectories). Recently, the more realistic "Berkeley Motion Segmentation Dataset" (BMS-26) was introduced by [47], [48] for long-term analysis of moving objects in video. This dataset consists of point trajectories that were obtained by running a state-of-the-art feature point tracker (the large-displacement optical flow [49]) on 26 videos directly without any further post-processing. Thus, those feature trajectories contain noise and outliers and, most importantly, some are incomplete. Incomplete trajectories are trajectories that do not run for the whole duration of the video. They can appear in any frame of the video and disappear on or before the last frame. These incomplete trajectories are mainly caused by occlusion and disocclusion.

The traditional approach of using two views to segment

[3]The code available at http://www.vision.jhu.edu/data/hopkins155/ was used to generate outlier trajectories. In their code, a randomly selected trajectory in a given sequence is modified by, selecting a random point in that trajectory and moving the point to a new location by the same displacement of another randomly selected point in a different trajectory.

TABLE II
COMPARATIVE PERFORMANCES IN TERMS OF ACCURACY AND SPEED USING THE HOPKINS 155 CHECKERBOARD SEQUENCE.

	RANSAC	PEARL	QP-MF	HMSS	SSC*	SCC	HOSC	CBS
	Two-Motion Sequences							
Mean	6.52	5.28	9.98	3.98	2.23	**1.40**	5.28	1.60
Median	1.75	1.83	1.38	0.00	0.00	**0.04**	0.02	0.10
Time	-	-	-	-	0.65	0.66	1.27	**0.48**
	Three-Motion Sequences							
Mean	25.78	21.38	15.61	11.06	5.77	5.74	7.38	**4.98**
Median	26.01	21.14	8.82	1.20	**0.95**	1.48	1.53	1.04
Time	-	-	-	-	1.47	1.29	2.00	**0.55**

*The results for SSC are generated using the faster ADMM [3] implementation that is provided at http://vision.jhu.edu/ without any modifications. The SSC CSX implementation [46] is more accurate but has significantly higher computational cost.

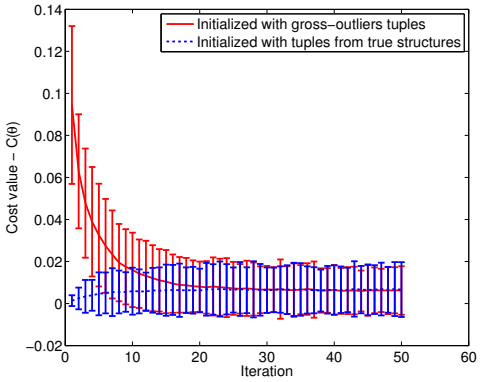


Fig. 4. Mean and standard deviation of the cost function value at each iteration across 10000 random runs of Algorithm 1, initialized with tuples from either complete outliers or true structures. The plots show that on average, the convergence of the local update step does not depend on the initialization values.

objects is susceptible to short-term variations (e.g., a human who is standing for a short time can be merged with the background). Hence, Brox and Malik [48] proposed a long-term video analysis in which a similarity between two points trajectories was used to build a graph that was segmented using spectral clustering. Such pairwise affinities only model translations and do not account for scaling and rotation. Ochs and Brox [20] used affinities that are defined on higher-order tuples, which results in a hyper-graph. Using a nonlinear projection, this hyper-graph was converted to an ordinary graph, which was segmented using spectral clustering.

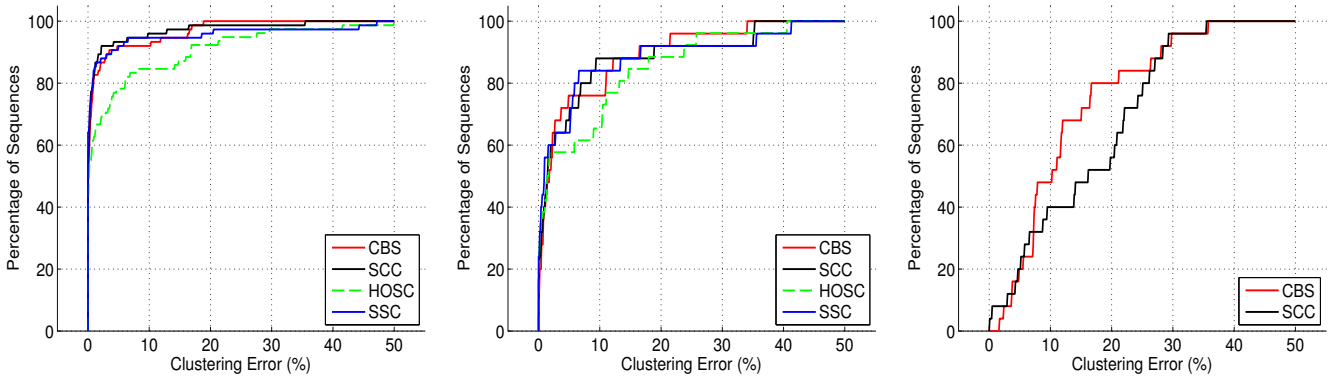
In this analysis, we use the approach that was proposed by Ochs and Brox [20] in which a motion of an object is modelled using a special similarity transformation $\mathcal{T} \in \text{SSim}(2)$, with parameters for scaling (s), rotation (α) and translation (v). The distance from a trajectory $(c_i(t) \rightarrow c_i(t'))$ to the model $\mathcal{T}_t$ is calculated using the L_2-distance: $d_{\mathcal{T}_t,i} = \|\mathcal{T}_t c_i(t) - c_i(t')\|$. A motion hypothesis $\mathcal{T}_t$ at time t can be obtained using two or more point trajectories that exist in the interval $[t, t']$. In our implementation, we used edges of size $h = p + 2 = 4$ to generate hypotheses. It should be noted here that the distance measure is only valid if the trajectories that are used to generate the hypothesis and the trajectory for which the distance is calculated all coexist in time. Hence, a distance of infinity is assigned to all points that do not exist in the time interval $[t, t']$. This behavior causes complications in the weight update of the proposed method as now some trajectories can be identified as outliers even though they belong to the same object. To overcome this, we uniformly sample small windows (of size 7 frames) and limit the weight updates to each window.

Another important feature of this dataset is that most sequences have many frames and data points (e.g., sequence "tennis", even with down-scaling by 8 times [20], includes more than 450 frames and 40,000 data points). Storing a graph of that size is challenging, especially on a PC. Hence, in cases in which the number of frames is large, we divide the video into a few large windows (e.g., 100 frames) and solve the problem in each large window independently. Next, we calculated the mutual distance between each structure in different windows and clustered them using k-means to obtain the desired number of structures. The number of clusters is a parameter that is selected such that it yields reasonable accuracy with minimal over-segmentation.

Once the clusters were obtained, they were evaluated using the method that was provided with the dataset (man-made masks on specific frames of the videos). We compare our results with those of [20], [21], which are based on higher-order affinities. The results that are given in Table III show that our method has achieved similar accuracies with significant improvements in computation time. The computation time is related to the number of hyper-edges that are used. OB used $N^2 \times (30 + 12)$ hyper edges in their implementation, whereas HOSC used $2N/5 + N$. Our method uses fewer hyper-edges ($N/10$), which are selected using the k-th-order cost function. The results show that if the edges are selected appropriately, similar accuracies can be achieved and using fewer edges results in a lower computational time. Moreover, while the two competing methods [20], [21] use spatial contiguity in selecting the edges for constructing the affinity graph, the proposed method does not use any such additional information.

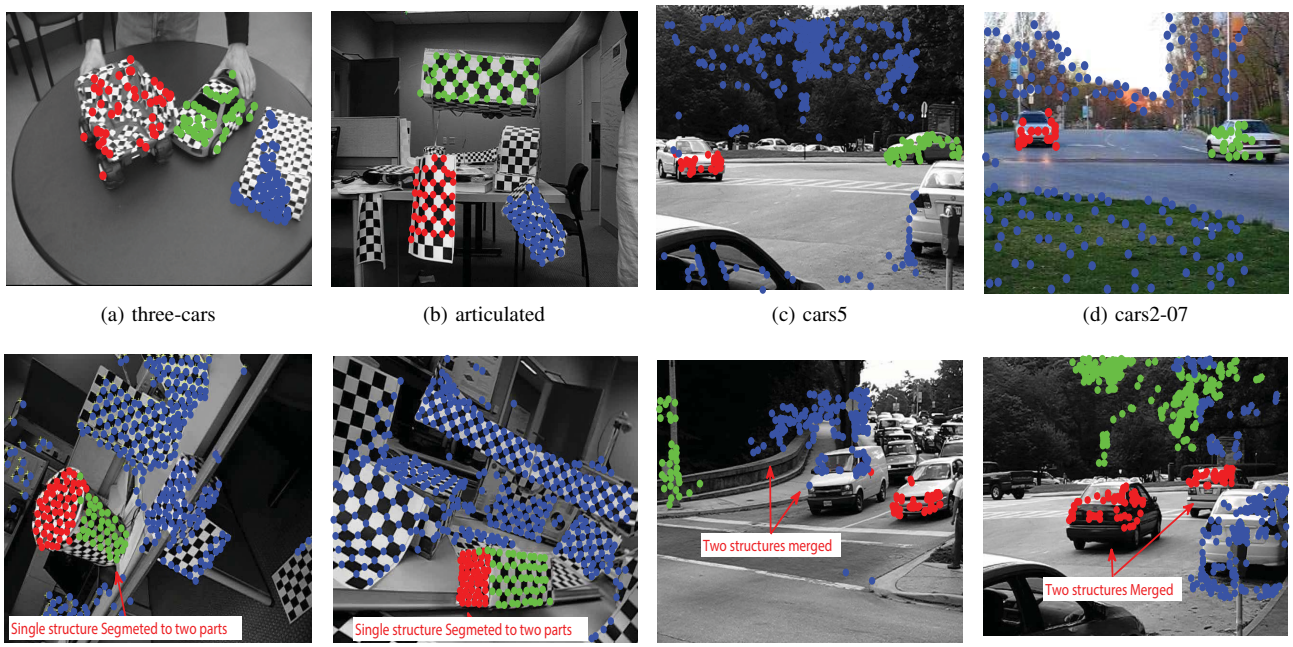
V. DISCUSSION

The proposed method requires the value of k, which defines the minimal acceptable size for a structure in a given application, as input. Any robust model fitting method must establish the minimal acceptable structure size (either explicitly or implicitly); otherwise, it may yield a trivial solution. For example, if we are given a set of 2D points and asked to identify



(a) Two-motion sequences (b) Three-motion sequences (c) Three-motion sequences with outliers

Fig. 5. Cumulative distributions of the clustering errors (CE) per sequence of the Hopkins dataset. Figure 5a Two-motion sequences, Figure 5b three-motion sequences and Figure 5c three-motion sequences with added synthetic outliers.



(a) three-cars (b) articulated (c) cars5 (d) cars2-07

(e) 1RT2RTCRT-B (f) 2RT3RCR (g) cars9 (h) cars2B

Fig. 6. Clustering results that were obtained using the proposed method on several example sequences from the Hopkins dataset. The top row show cases in which the proposed method was successful, whereas the bottom row show cases in which the proposed method failed to identify all the clusters correctly (best viewed in color).

lines in the data without any additional constraints, there would be no basis for excluding the trivial solution because any two points will result in a perfect line. Hence, to find a meaningful solution, there must be additional constraints, such as the minimal acceptable size for a structure. The proposed method estimates the scale of the noise from the data and the analysis of [35] showed that estimation of the noise scale from the data requires at least approximately 20 data points to limit the effects of finite-sample bias. This leads to a lower bound on k of approximately 20.

Similar to competing clustering-based methods (e.g., SCC [19], SSC [3]), the proposed method requires prior knowledge on the number of clusters. This is one of the main limitations of the proposed method. The problem of identifying simultaneously the number of structures and the scale of the noise remains a highly researched area. Remaining outliers can always be viewed as members of a model with large noise values. Zelnik-Manor and Perona [50] proposed a method for automatically estimating the number of clusters in a graph using eigenvector analysis. However, using such methods to identify the number of clusters in a graph requires smoothing parameters or thresholds that are similar to those that are used in energy-based methods. Since our focus in this paper is on efficiently generating the graph (not on how to cluster it), we have not included clustering in the evaluations. Various model fitting methods that are based on energy minimization [7] have been devised for estimating the number of structures given the scale of the noise. They achieve this by adding a model

TABLE III
MOTION SEGMENTATION RESULTS ON THE BERKELEY MOTION SEGMENTATION DATASET (BMS-26).

	Density	Overall error	Average error	Over-segmentation rate	Extracted objects	Total Time (s)
OB	1.03%	5.68%	24.74%	1.48	30	434545
HOSC	1.03%	8.05%	27.84%	2.1	22	11966
CBS	1.03%	7.80%	22.60%	2.08	22	7875

complexity term to the cost function that penalizes additional structures in a given solution. However, these methods require an additional parameter that balances the data fidelity cost with the model complexity (the number of structures in [21]). Our experiments on [21] showed that the output of these methods was heavily dependent on this parameter and required hand-tuning on each image (of Table I) to generate reliable results.

The proposed method uses a data-sub-sampling strategy that is based on a set of inclusion weights to bias the algorithm to produce edges from different structures. These inclusion weights are iteratively calculated using the inlier/outlier dichotomy for each edge. However, in cases in which there is additional information about the problem, such as spatial contiguity, one can use this approach to improve the sub-sampling. For example, in two-view motion segmentation, the Euclidean distance between points can be used to construct a KDtree, which can be used to perform the sampling directly (i.e., select the initial point randomly and include the N_s points that are closest to that point as the data sub-sample). However, our approach does not facilitate the integration of higher-order potentials such as those that were introduced in [51]. In the performance evaluations of this paper, we have not used any such additional information.

VI. CONCLUSION

In this paper, we proposed an efficient sampling method for obtaining a highly accurate approximation of the full graph that is required for solving multi-structural model fitting problems in computer vision. The proposed method is based on the observation that the usefulness of a graph for segmentation improves as the distribution of the hypotheses that are used to build the graph approaches the actual parameter distribution for the given data. In this paper, we approximate this actual parameter distribution using the kth-order statistical cost function, and the samples are generated using a greedy algorithm coupled with a data sub-sampling strategy.

The performance of the algorithm in terms of accuracy and computational efficiency was evaluated on several instances of multi-object motion segmentation problems and was compared with the performances of state-of-the-art model fitting techniques. The comparisons show that the proposed method is both highly accurate and computationally efficient.

ACKNOWLEDGMENT

This research was partly supported under Australian Research Council (ARC) Linkage Projects funding scheme.

REFERENCES

[1] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[2] A. Delong, L. Gorelick, O. Veksler, and Y. Boykov, "Minimizing energies with hierarchical costs," *International Journal of Computer Vision*, vol. 100, no. 1, pp. 38–58, 2012.

[3] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[4] H. Chen and P. Meer, "Robust regression with projection based m-estimators," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, 2003, pp. 878–885.

[5] C. V. Stewart, "Bias in robust estimation caused by discontinuities and multiple structures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 8, pp. 818–833, 1997.

[6] M. Zuliani, C. S. Kenney, and B. S. Manjunath, "The multiransac algorithm and its application to detect planar homographies," in *Proceedings - International Conference on Image Processing, ICIP*, vol. 3, 2005, pp. 153–156.

[7] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[8] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 11, 2011.

[9] R. Cabral, F. D. L. Torre, J. P. Costeira, and A. Bernardino, "Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2488–2495.

[10] S. Agarwal, J. Lim, L. Zelnik-Manor, P. Perona, D. Kriegman, and S. Belongie, "Beyond pairwise clustering," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2005, pp. 838–845.

[11] V. M. Govindu, "A tensor decomposition for geometric grouping and segmentation," in *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, vol. I, 2005, pp. 1150–1157.

[12] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.

[13] G. Liu, H. Xu, J. Tang, Q. Liu, and S. Yan, "A deterministic analysis for lrr," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 3, pp. 417–430, 2016.

[14] J. Wang, D. Shi, D. Cheng, Y. Zhang, and J. Gao, "Lrsr: Low-rank-sparse representation for subspace clustering," *Neurocomputing*, vol. 214, pp. 1026–1037, 2016.

[15] E. Kim, M. Lee, and S. Oh, "Robust elastic-net subspace representation," *IEEE Transactions on Image Processing*, vol. 25, no. 9, pp. 4245–4259, 2016.

[16] B. Poling and G. Lerman, "A new approach to two-view motion segmentation using global dimension minimization," *International Journal of Computer Vision*, vol. 108, no. 3, pp. 165–185, 2014.

[17] B. Li, Y. Zhang, Z. Lin, and H. Lu, "Subspace clustering by mixture of gaussian regression," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June-2015, 2015, pp. 2094–2102.

[18] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*, 2002.

[19] G. Chen and G. Lerman, "Spectral curvature clustering (scc)," *International Journal of Computer Vision*, vol. 81, no. 3, pp. 317–330, 2009.

[20] P. Ochs and T. Brox, "Higher order motion models and spectral clustering," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2012, pp. 614–621.

[21] P. Purkait, T. . Chin, H. Ackermann, and D. Suter, *Clustering with hypergraphs: The case for large hyperedges*, ser. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2014, vol. 8692 LNCS, no. PART 4.

[22] T. Zhang, A. Szlam, Y. Wang, and G. Lerman, "Hybrid linear modeling via local best-fit flats," *International Journal of Computer Vision*, vol. 100, no. 3, pp. 217–240, 2012.

[23] S. Balakrishnan, M. Xu, A. Krishnamurthy, and A. Singh, "Noise thresholds for spectral clustering," in *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011, NIPS 2011*, 2011.

[24] R. H. Swendsen and J. . Wang, "Nonuniversal critical dynamics in monte carlo simulations," *Physical Review Letters*, vol. 58, no. 2, pp. 86–88, 1987.

[25] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*. John Wiley & Sons, 2005, vol. 589.

[26] A. Bab-Hadiashar and R. Hoseinnezhad, "Bridging parameter and data spaces for fast robust estimation in computer vision," in *Proceedings - Digital Image Computing: Techniques and Applications, DICTA 2008*, 2008, pp. 1–8.

[27] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan, "An introduction to mcmc for machine learning," *Machine Learning*, vol. 50, no. 1-2, pp. 5–43, 2003.

[28] R. B. Tennakoon, A. Bab-Hadiashar, Z. Cao, R. Hoseinnezhad, and D. Suter, "Robust model fitting using higher than minimal subset sampling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 350–362, 2016.

[29] R. Toldo and A. Fusiello, *Robust multiple structures estimation with J-linkage*, ser. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2008, vol. 5302 LNCS, no. PART 1, cited By :165.

[30] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[31] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *ICML Vol. 96, pp. 148-156*, 1996, Conference Proceedings.

[32] ——, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[33] T. T. Pham, T. Chin, J. Yu, and D. Suter, "The random cluster model for robust geometric fitting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1658–1671, 2014.

[34] A. Bab-Hadiashar and D. Suter, "Robust segmentation of visual data using ranked unbiased scale estimate," *Robotica*, vol. 17, no. 6, pp. 649–660, 1999.

[35] R. Hoseinnezhad, A. Bab-Hadiashar, and D. Suter, "Finite sample bias of robust estimators in segmentation of closely spaced structures: A comparative study," *Journal of Mathematical Imaging and Vision*, vol. 37, no. 1, pp. 66–84, 2010.

[36] J. Yu, T. Chin, and D. Suter, "A global optimization approach to robust multi-model fitting," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011, pp. 2041–2048.

[37] S. R. Rao, A. Y. Yang, S. S. Sastry, and Y. Ma, "Robust algebraic segmentation of mixed rigid-body and planar motions from two views," *International Journal of Computer Vision*, vol. 88, no. 3, pp. 425–446, 2010.

[38] T. Chin, J. Yu, and D. Suter, "Accelerated hypothesis generation for multistructure data via preference analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 625–638, 2012.

[39] O. Chum, J. Matas, and J. Kittler, "Locally optimized ransac," ser. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2003, vol. 2781, pp. 236–243.

[40] P. H. S. Torr and D. W. Murray, "The development and comparison of robust methods for estimating the fundamental matrix," *International Journal of Computer Vision*, vol. 24, no. 3, pp. 271–300, 1997.

[41] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[42] H. S. Wong, T. Chin, J. Yu, and D. Suter, "Dynamic and hierarchical multi-structure geometric model fitting," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 1044–1051.

[43] N. Lazic, I. Givoni, B. Frey, and P. Aarabi, "Floss: Facility location for subspace segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2009, pp. 825–832.

[44] Y. Sugaya and K. Kanatani, *Geometric structure of degeneracy for multi-body motion segmentation*, ser. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2004, vol. 3247, pp. 13–25.

[45] R. Tron and R. Vidal, "A benchmark for the comparison of 3-d motion segmentation algorithms," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

[46] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, 2009, pp. 2790–2797.

[47] P. Ochs, J. Malik, and T. Brox, "Segmentation of moving objects by long term video analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 6, pp. 1187–1200, 2014.

[48] T. Brox and J. Malik, *Object segmentation by long term analysis of point trajectories*, ser. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2010, vol. 6315 LNCS, no. PART 5.

[49] ——, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 500–513, 2011.

[50] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Advances in Neural Information Processing Systems*, 2005.

[51] T. T. Pham, T. Chin, K. Schindler, and D. Suter, "Interacting geometric priors for robust multimodel fitting," *IEEE Transactions on Image Processing*, vol. 23, no. 10, p. 1, 2014.