Multi-Structure Model Selection via Kernel Optimisation*

Tat-Jun Chin, David Suter and Hanzi Wang
School of Computer Science, The University of Adelaide, South Australia
{tjchin, dsuter, hwang}@cs.adelaide.edu.au

Abstract

Our goal is to fit the multiple instances (or structures) of a generic model existing in data. Here we propose a novel model selection scheme to estimate the number of genuine structures present. In contrast to conventional model selection approaches, our method is driven by kernel-based learning. The input data is first clustered based on their potential to have emerged from the same structure. However the number of clusters is deliberately overestimated to obtain a set of initial model fits onto the data. We then resolve the oversegmentation via a series of kernel optimisation conducted through multiple kernel learning, and the concept of kernel-target alignment is used as a model selection criterion. Experiments on synthetic and real data show that our method outperforms previous model selection schemes. We also focus on the application of multi-body motion segmentation. In particular we demonstrate success on estimating the number of motions on sequences with more than 3 unique motions.

1. Introduction

Many computer vision problems involve data with multiple structures. A "structure" is defined as an instance of a generic model in the data [18]. For example in homography detection a scene often contains multiple planar surfaces, each giving rise to a set of multi-view point correspondences that can be related by a specific homography. Most applications would be interested to recover all of the genuine structures present and to fit (i.e. estimate parameter values for) the generic model onto these structures.

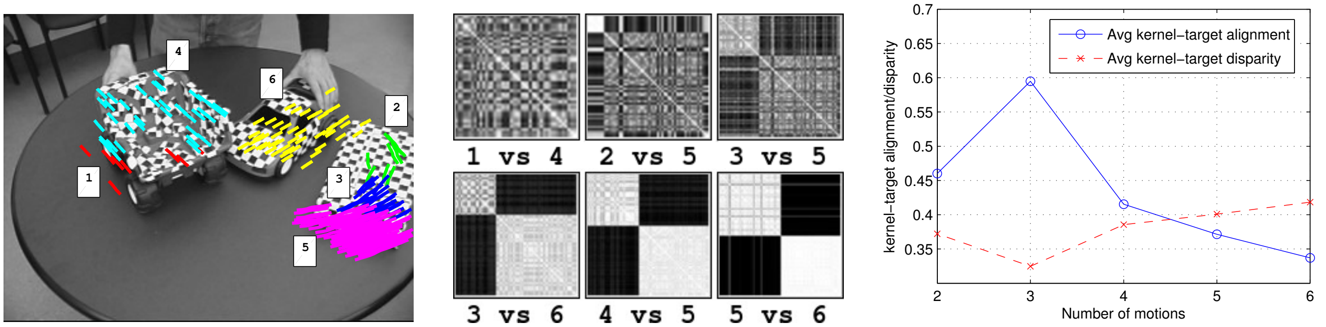
Many recent efforts on multi-structure recovery are directed towards Multi-body Structure-and-Motion (MSaM) problems. A particularly active area is multi-body motion segmentation, i.e. given multi-view point correspondences or trajectories, recover the multiple distinct motions contained in the data. However many such works (including very recent ones [8, 14, 7, 2]) solve the segmentation part of the problem only, i.e. separate the points based on the motion they belong to given the number of motions is known *a priori*. The problem of estimating the number of motions has received less attention, although in practical settings the number of motions is usually unknown beforehand.

Indeed, it has been observed [5] that estimating the number of motions is actually *more* challenging than segmenting the motions. Straightforward solutions like counting the number of zeros in matrices computed from motion data [5] almost always fail in practice because of noise in the data. One has to resort to some form of model selection, where a "model" here implies a specific number of motions and their fit onto the data. The idea is to strike a balance between the goodness of fit of a model and the model complexity (which in the multi-structure case is proportional to the number of structures). Previous work on model selection in motion segmentation can roughly be categorised into three groups: Rank detection methods [13], combinatorial methods [15, 19] and cluster detection methods [11, 4, 3].

Under the first group, Kanatani and Matsunaga [13] propose to detect the *effective rank* of the observation (trajectory) matrix using the Geometric AIC and Geometric MDL model selection criteria. Under the affine camera model where each motion occupies a distinct subspace [5], the rank of the observation matrix is proportional to the number of motions. The method performs well on synthetic motion sequences. However recent tests [3] on the Hopkins 155 dataset [21] show that it is not very accurate in practice. Moreover it is prone to breakdown [3] due to a lack of an explicit mechanism to deal with outliers in the data.

The second group of methods [15, 19] advocate a hypothesise-then-select approach. A set of candidate motions are first generated from the data by random sampling in the manner of RANSAC [9]. The likeliest subset of motions are then selected via combinatorial optimisation with the GRIC [20] model selection cost as the objective function. One problem is that the global minimum can only be found through exhaustive search [15]. This is usually intractable since one typically generates a large number of hypothesis motions. In [15] approximate solutions are sought using Taboo search which is a set of heuristics to greedily

*This work is partly supported by the ARC grant DP0878801.



(a) The "three-cars" sequence from Hopkins 155 with 3 unique motions, each from a distinct object. Initial cluster detection reveals 6 motions.

(b) Actual optimised kernels for pairs of motions. Observe that kernels for dissimilar motions (second row) are more block diagonal.

(c) Model selection based on kernel optimisation. The maximum (minimum) average kernel-target alignment (disparity) is achieved at 3 motions.

Figure 1. Applying the proposed model selection scheme on motion data. (a) The input data is first oversegmented using the approach of [4, 3]. (b) A kernel is then optimised for each pair of clusters, where the target kernel is an ideal block diagonal matrix. (c) The alignment between the learnt and the target kernel is used to drive a structure merging model selection scheme, where a pair of structures are more likely to be merged if their kernel alignment is low. The average alignment of a model serves as an accurate model selection criterion.

move towards the local minimum by adding one motion at every step. A more recent work [19] proposes a branch-and-bound strategy to prune the search space.

The third group of methods [11, 4, 3] attempt to directly detect the number of clusters in the data. In [11], Dirichlet Process Mixture Model (DPMM) learning is used to simultaneously infer the number of clusters and the cluster membership of each point. However the authors propose to repeat the clustering several times and to validate the "fitness" of each clustering. This suggests that the results are likely to be ambiguous and that model selection is still required. More recently [4, 3] introduce a novel Mercer kernel to cluster data based on the potential of two points to have emerged from the same structure. Recognizing that clean clusters are unattainable in real data they deliberately oversegment the data. Extraneous clusters are then removed with a sequential structure-removal operation which is essentially a model selection step. However their merging criterion was developed based on simple 2D geometrical structures and might not be optimal for more complex structures. Nonetheless evaluations [3] on the Hopkins 155 dataset show that this approach is the most accurate in model selection for affine camera motion segmentation.

In this paper we propose an unconventional but highly effective model selection scheme based on kernel learning. Like [4, 3] we first oversegment the data to obtain a candidate set of structures and attempt to merge the structures. To this end we treat each pair of structures as samples for a binary classification problem. A kernel is then optimised via Multiple Kernel Learning (MKL) [1, 17] to separate the classes. The key insight is that if two structures are indeed separate instances of the generic model the optimised kernel will have a high alignment [6] (or as defined later in Sec. 3) with the target kernel. On the other hand a low alignment suggests that the two structures should be merged. We propose a structure merging operation that is driven by kernel-target alignment, and show that the maximum overall kernel-target alignment is achieved when the correct number of structures are fitted onto the data. Fig. 1 illustrates the idea. Experiments (see Sec. 4) show that our approach outperforms previous methods. In particular we show superior results over [13, 4, 3] in model selection for motion segmentation using the Hopkins 155 dataset [21]. We also demonstrate success on model selection for sequences with more than 3 unique motions.

The rest of the paper is organised as follows: In Sec. 2 we first examine the cluster detection method of [4, 3] which we apply to oversegment the data. In Sec. 3 we describe the proposed model selection scheme. We present experimental results in Sec. 4 and draw conclusions in Sec. 5.

2. Cluster Detection

In this section we describe the kernel-based multi-structure robust fitting approach of [4, 3]. In particular we examine the Mercer kernel proposed in [4, 3] which is used to cluster data based on the potential of two points to have emerged from the same structure. We also show how it can be used to obtain an initial set of model fits onto the data.

2.1. The Ordered Residual Kernel

Let the model to be fitted be defined by p parameters, e.g. $p = 2$ for lines, $p = 3$ for circles. Given input data $\mathcal{X} = \{x_i\}_{i=1,\ldots,D}$ of D points we first randomly sample a set of M model hypotheses $\{\theta_j\}_{j=1,\ldots,M}$, where each hypothesis θ_j is fitted from a minimal subset of p points. For each data point x_i, we compute its absolute residual set $\mathbf{r} = \{r_1, \ldots, r_M\}$ as measured to the M hypotheses, e.g.

for lines r_j is the orthogonal distance of x_i to the line θ_j.

We sort the elements in $\mathbf{r}$ to obtain the sorted residual set $\tilde{\mathbf{r}} = \{r_{\lambda_1}, \ldots, r_{\lambda_M}\}$, where the permutation $\{\lambda_1, \ldots, \lambda_M\}$ is obtained such that $r_{\lambda_1} \leq \cdots \leq r_{\lambda_M}$. The sorted hypothesis set of point x_i is then defined as

$$\boldsymbol{\lambda}_i = \{\lambda_1, \ldots, \lambda_M\}. \qquad (1)$$

Intuitively $\boldsymbol{\lambda}_i$ depicts the *preference* of x_i to the M hypotheses. As proposed in [4] the Ordered Residual Kernel (ORK) between two data points x_i and x_j is defined as

$$k_{\tilde{r}}(x_i, x_j) = \frac{1}{T} \sum_{t=1}^{M/h} \frac{1}{t} k_{\cap}^{t}(\boldsymbol{\lambda}_i, \boldsymbol{\lambda}_j), \qquad (2)$$

where $T = \sum_{t=1}^{M/h} 1/t$ is a normalisation constant. Component $k_{\cap}^{t}$ is the Difference of Intersection Kernel (DOIK)

$$k_{\cap}^{t}(\boldsymbol{\lambda}_i, \boldsymbol{\lambda}_j) = \frac{1}{h}(|\boldsymbol{\lambda}_i^{1:\alpha_t} \cap \boldsymbol{\lambda}_j^{1:\alpha_t}| - |\boldsymbol{\lambda}_i^{1:\alpha_{t-1}} \cap \boldsymbol{\lambda}_j^{1:\alpha_{t-1}}|) \quad (3)$$

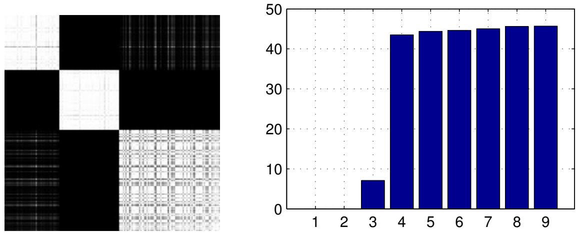
where $\alpha_t = t \cdot h$ and $\alpha_{t-1} = (t-1)h$. Symbol $\boldsymbol{\lambda}_i^{a:b}$ indicates the set formed by the a-th to the b-th elements of $\boldsymbol{\lambda}_i$. Note that h is a stepsize that is determined based on M [4].

Given two input points, $k_{\tilde{r}}$ evaluates the rate of increase of the hypotheses they mutually prefer as a fictitious inlier threshold rises (in steps of h) from 0 to ∞ [4]. Thus given two points from the same structure, $k_{\tilde{r}}$ will return a high value, while a low value is obtained when the points are from different structures. Since $k_{\tilde{r}}$ is also provably a valid Mercer kernel [16] (see [4] for proofs) it induces a mapping which maps the input data to a high dimensional feature space whereby the inner product is simply the kernel $k_{\tilde{r}}$. Therefore by construction, $k_{\tilde{r}}$ maps points to clusters in the feature space, where each cluster corresponds to a structure in the data. For more details of ORK refer to [4, 3].

2.2. How Many Clusters?

To detect the number of clusters, [4, 3] propose to apply spectral clustering (e.g. see [10]). The data is first projected onto their principal components in the ORK-induced feature space. The affinity matrix for spectral clustering is constructed from the reduced dimension data, and the eigenspectrum of the Laplacian matrix is examined for the number of zeros (where each zero indicates one cluster). A separate k-means step (with k set to the number of detected clusters) is then used to cluster the reduced dimension data.

As an example we apply ORK and spectral clustering on the "three-cars" sequence in Fig. 1 with $M = 400$ and $h = 50$. Since each motion under the affine camera model occupies a 4D subspace (assuming independent and non-degenerate motions), we sample 4D subspaces from the observation matrix. Fig. 2 shows the resulting affinity matrix and the first-few eigenvalues of the Laplacian matrix.



(a) Affinity matrix. (b) First-9 eigenvalues of Laplacian.

Figure 2. Detecting the number of clusters (where each cluster is a unique motion) for the "three-cars" sequence.

The results in Fig. 2 illustrate typical difficulties in cluster detection for real data. Making an unambiguous conclusion that three clusters exist from the eigenspectrum of the Laplacian matrix is non-trivial. Firstly due to limits on computational precision the first-two *seemingly* zero eigenvalues are not exactly zero. This implies thresholding is required (c.f. the zero thresholding method in [5]). However, setting a threshold which is too low (e.g. $\approx 1.0e^{-3}$) causes the third cluster to be ignored— due to noise the third eigenvalue is significantly larger than the first two. One has to apply a much higher threshold (e.g. ≈ 20) to include all genuine clusters. Almost certainly this ad-hoc threshold does not generalise well to other sequences where a different eigenspectrum of the Laplacian matrix may exist.

In view of the difficulty of cluster detection, we propose to overcluster the data and attempt to merge the clusters later via a model selection scheme. We use a consistent thresholding rule across all data based on normalised cumulative eigenvalues. Let $\mathbf{W}$, $\mathbf{D}$ and $\mathbf{L}$ respectively be the affinity, degree and Laplacian matrix of the data, where

$$\mathbf{D}_{p,p} = \sum_i \mathbf{W}_{i,p} \;\; \text{and} \;\; \mathbf{D}_{p,q} = 0 \;\forall\; p \neq q \qquad (4)$$

and $\mathbf{L} = \mathbf{D} - \mathbf{W}$. Let the eigendecomposition of $\mathbf{L}$ be $\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$. The normalised cumulative eigenvalue vector is defined as $\bar{\boldsymbol{\sigma}} = [\; \bar{\sigma}_1 \; \bar{\sigma}_2 \; \ldots \;]$, where each value $\bar{\sigma}_p$ is

$$\bar{\sigma}_p = \sum_{i=1}^{p} \boldsymbol{\Sigma}_{i,i} / \sum_j \boldsymbol{\Sigma}_{j,j}. \qquad (5)$$

Using a fixed threshold (e.g. 0.1 for 10% energy) we count the number of values in $\bar{\boldsymbol{\sigma}}$ which are below this threshold as the number of clusters. Fig. 1(a) illustrates the result on the "three-cars" sequence, where 6 clusters are detected.

3. Model Selection via Kernel Optimisation

Statistical model selection theory and its geometric extensions [12, 20] suggest to balance goodness-of-fit and model complexity. In the case of multi-structure fitting we wish to determine the correct *number* of instances of a fixed geometric model existing in the data. Hence, a "model" here implies a specific number of instances of the geometric model and their fit onto the data. The complexity of a model is thus proportional to the number of structures. Having a small number of structures ensures lower complexity but risks obtaining a poor fit (i.e. high fitting error) onto the data. Previous works on model selection (see Sec. 1) attempt to minimise a cost function which includes two components: fitting error and model complexity measure.

Here we propose a novel structure merging scheme driven by kernel optimisation for model selection. A pair of structures are treated as different groups for which a Support Vector Machine (SVM) classifier is to be trained. A kernel matrix is then optimised for the SVM using Multiple Kernel Learning (MKL). The measure of success of optimising this kernel serves as the model selection criterion.

3.1. Multiple Kernel Learning

Let $\mathcal{S} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ be a set of training data for binary classification, with feature vectors x_i and target labels $y_i \in \{+1, -1\}$. An important ingredient in SVM is to specify a kernel function $k(\cdot, \cdot|\theta)$ that is suitable for the classification problem at hand. Symbol θ indicates the type (i.e. parametric form) of the kernel as well as its parameter values, e.g. Gaussian kernel with a specific bandwidth value, polynomial kernel with a specific degree. Training an SVM produces a classifier of the form

$$f(x) = sign\left(\sum_{i=1}^{N} \alpha_i y_i k(x, x_i|\theta) + b\right) \tag{6}$$

where b is a constant bias, and the α_i's are coefficients determined from SVM training. Vectors x_i for which the coefficient α_i is nonzero are called "support vectors". Unsurprisingly choosing the correct kernel (and its parameter values) plays a crucial role in the performance of SVMs.

Realising the difficulty in crafting appropriate kernels or setting parameter values, the idea of MKL is proposed (e.g. see [1, 17]). Instead of a pre-determined kernel, MKL requires a set of *base kernels* $\{k(\cdot, \cdot|\theta_k)\}_{k=1,\ldots,K}$. The goal is to produce a convex combination of the base kernels to obtain a strong overall kernel

$$\hat{k}(\cdot, \cdot) = \sum_{k=1}^{K} \beta_k k(\cdot, \cdot|\theta_k). \tag{7}$$

This is then plugged into the SVM classifier

$$f(x) = sign\left(\sum_{i=1}^{N} \alpha_i y_i \sum_{k=1}^{K} \beta_k k(x, x_i|\theta_k) + b\right). \tag{8}$$

Efficient algorithms have been proposed [1, 17] to *simultaneously* optimise the coefficients α_i and β_k. Intuitively base kernels with higher β_k values are deemed more useful for the problem at hand. If the base kernels are of the same type but of different parameter values, MKL is effectively optimising the best parameters for this type of kernel.

3.2. Base Kernels for Model Selection

We proceed from having segmented the input data $\mathcal{X}$ into P clusters $\{\mathcal{S}_p\}_{p=1\ldots P}$, where $\mathcal{S}_p \cap \mathcal{S}_q = \emptyset$ for all $p \neq q$ and $\mathcal{X} = \bigcup_{p=1}^{P} \mathcal{S}_p$. Our goal is to determine if a pair of clusters contain points that emerged from the same instance of a generic model in the data. Our idea is to treat this as a binary classification problem. Given two distinct clusters $\mathcal{S}_p$ and $\mathcal{S}_q$ we aggregate their points into $\{x_1, \ldots, x_N\} = \mathcal{S}_p \cup \mathcal{S}_q$. We give each x_i a binary class label y_i, where

$$y_i = \begin{cases} +1 & \text{if } x_i \in \mathcal{S}_p \\ -1 & \text{if } x_i \in \mathcal{S}_q \end{cases}, \tag{9}$$

and try to optimise a kernel function for an SVM classifier. The level of difficulty in producing such a kernel function then serves as a measure of how similar the two clusters are.

We apply MKL to optimise the kernel function. The trick is to choose a set of base kernels that is suitable for the problem. To this end we observe that the ORK in Eq. (2) is basically a smoothed version of the intersection kernel [16]

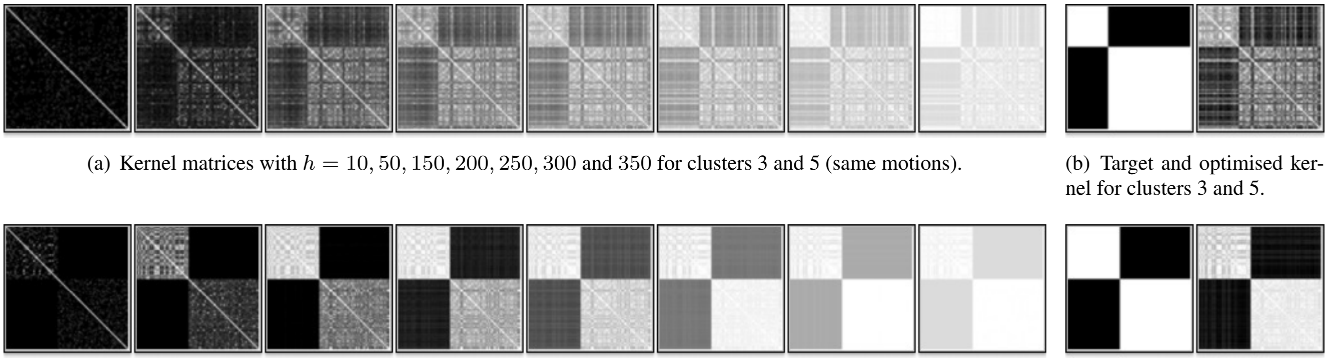
$$k_\cap(x_i, x_j|h) = \frac{1}{h}|\boldsymbol{\lambda}_i^{1:h} \cup \boldsymbol{\lambda}_j^{1:h}|, \quad 1 \leq h \leq M. \tag{10}$$

The smoothing compensates for the uncertainty in selecting window size h which is a kernel parameter. The intersection kernel counts the number of hypotheses x_i and x_j mutually prefer among their h "most preferred" hypotheses $\boldsymbol{\lambda}_i^{1:h}$ and $\boldsymbol{\lambda}_j^{1:h}$. If both points are from the same structure then $k_\cap(x_i, x_j|h)$ will be high (and vice versa). Parameter h effectively controls the discriminative power of the intersection kernel since the size of h affects the probability of sharing preferred hypotheses by two points.

Figs. 3(a) and 3(c) illustrate a few kernel matrices computed using the intersection kernel on clusters 3 and 5 (same motions) and clusters 4 and 5 (different motions) from the segmented "three-cars" sequence in Fig. 1(a). Observe that when h is low, the kernel is highly discriminative and both intra- and inter-class responses are attenuated. When h is high the kernel loses its discriminative power and most entries contain a high value. Also the block diagonal structure is more apparent in kernel matrices from clusters with different motions thus indicating they are easier to separate.

We use intersection kernels of different window sizes as the base kernels. Given a pair of clusters $\mathcal{S}_p$ and $\mathcal{S}_q$ and the binary class labels of the points they contain, we apply MKL to seek the optimal combination kernel

$$\hat{k}_{p,q}(\cdot, \cdot) = \sum_{h=0}^{M} \beta_h k_\cap(\cdot, \cdot|h) \tag{11}$$



(a) Kernel matrices with $h = 10, 50, 150, 200, 250, 300$ and 350 for clusters 3 and 5 (same motions).

(b) Target and optimised kernel for clusters 3 and 5.

(c) Kernel matrices with $h = 10, 50, 150, 200, 250, 300$ and 350 for clusters 4 and 5 (different motions).

(d) Target and optimised kernel for clusters 4 and 5.

Figure 3. Optimising SVM kernels for the "three-cars" sequence in Fig. 1. (a) and (c): Kernel matrices with different h values for the intersection kernel, i.e. Eq. (10). The number of hypotheses M is set to 400. (b) and (d): Target and learnt kernel from MKL.

to separate the points. The idea is that if $\mathcal{S}_p$ and $\mathcal{S}_q$ are indeed from different structures many of the base kernels are useful for classification, thus it would be relatively easy to optimise a highly discriminative $\hat{k}_{p,q}$.

3.3. Model Selection Algorithm

The level of difficulty in optimising $\hat{k}_{p,q}$ serves as a criterion for structure merging in our model selection algorithm. Define the *target kernel* for $\mathcal{S}_p$ and $\mathcal{S}_q$ as

$$k^{tar}_{p,q}(x_i, x_j) = 0.5|y_i + y_j|, \tag{12}$$

i.e. the perfect kernel for discriminating $\mathcal{S}_p$ and $\mathcal{S}_q$. To measure the level of difficulty in learning $\hat{k}_{p,q}$ we compute the *kernel-target alignment* [6] between $\hat{k}_{p,q}$ and $k^{tar}_{p,q}$:

$$A(\hat{k}_{p,q}, k^{tar}_{p,q}) = \frac{\langle \hat{\mathbf{K}}_{p,q}, \mathbf{K}^{tar}_{p,q} \rangle_F}{\sqrt{\langle \hat{\mathbf{K}}_{p,q}, \hat{\mathbf{K}}_{p,q} \rangle_F \langle \mathbf{K}^{tar}_{p,q}, \mathbf{K}^{tar}_{p,q} \rangle_F}}. \tag{13}$$

Symbols $\hat{\mathbf{K}}_{p,q}$ and $\mathbf{K}^{tar}_{p,q}$ respectively indicate the kernel matrix of the optimised kernel and the target kernel, and $\langle \cdot, \cdot \rangle_F$ refers to the matrix dot product [6]. Note that $\mathbf{K}^{tar}_{p,q}$ is perfectly block diagonal and can be computed as

$$\mathbf{K}^{tar}_{p,q} = \mathbf{y}^T \mathbf{y} \tag{14}$$

where $\mathbf{y} = [y_1 \dots y_N]$. Eq. (13) then reduces to

$$A(\hat{k}_{p,q}, k^{tar}_{p,q}) = \frac{\langle \hat{\mathbf{K}}_{p,q}, \mathbf{y}^T \mathbf{y} \rangle_F}{N \sqrt{\langle \hat{\mathbf{K}}_{p,q}, \hat{\mathbf{K}}_{p,q} \rangle_F}}. \tag{15}$$

Note that $0 \leq A(\hat{k}_{p,q}, k^{tar}_{p,q}) \leq 1$ and the alignment approaches 1 if the two kernel matrices are exactly the same.

Figs. 3(b) and 3(d) illustrate $\hat{\mathbf{K}}_{p,q}$ and $\mathbf{K}^{tar}_{p,q}$ for clusters 3 vs 5 and clusters 4 vs 5 respectively. The alignment value is significantly higher for clusters 4 and 5, indicating that they correspond to different structures.

We propose a model selection algorithm based on greedy structure merging. Given a set of clusters $\{\mathcal{S}_p\}_{p=1\dots P}$ we compute the alignment between all unique pairs of clusters. The algorithm then chooses the pair with the *lowest* alignment to merge, and the process repeats on the remaining $P-1$ clusters. Each configuration of clusters represents a model that explains the data, and the average pair-wise kernel-target alignment for a model is used as a basis for comparing models. The maximum average kernel-target alignment is achieved when all the clusters in a particular model are sufficiently distinguished from each other according to the proposed measure in Eq. (15). This model selection criterion implicitly compares the goodness of fit and complexity of proposal models. Algorithm 1 summarises the proposed model selection scheme.

Algorithm 1 Model selection based on kernel optimisation

```
Require: Set of clusters $\{\mathcal{S}_p\}_{p=1\dots P}$.
  for $i = 1, \dots, (P-1)$ do
    $\mathcal{M}_i \longleftarrow$ Current set of clusters.
    Compute all pairwise kernel-target alignment.
    $\bar{A}_i \longleftarrow$ Average kernel-target alignment.
    Merge cluster pair with the lowest alignment.
  end for
  return $\mathcal{M}_{i^*}$ where $i^* = \arg\max_i \bar{A}_i$.
```

Note that Algorithm 1 is unable to return a model with one structure since computing the proposed kernel-target alignment requires at least a pair of clusters. For data where the maximum average alignment occurs at two structures, we impose a threshold where the remaining structures are merged if their alignment value does not surpass the threshold. Since the kernel-target alignment is normalised ($0 \leq A \leq 1$) a consistent threshold can be easily determined. Moreover this only affects selection between models of 1 or 2 structures which is acceptable to many vision applications (e.g. in motion segmentation there are usually at least 2 motions, one from the moving object and one from the background). Finally an inverse algorithm can be constructed by using the disparity between the optimised and target kernel as the model selection criterion:

$$D(\hat{k}_{p,q}, k^{tar}_{p,q}) = \frac{1}{N^2}\|\hat{\mathbf{K}}_{p,q} - \mathbf{K}^{tar}_{p,q}\|_F. \qquad (16)$$

Fig. 1(c) illustrates applying the proposed model selection algorithm to estimate the number of motions for the "three-cars" sequence. It can be seen that the maximum (minimum) average kernel-target alignment (disparity) is correctly achieved when 3 motions remain.

4. Results

We test the proposed model selection approach on synthetic and real data and compare it to previous techniques. We use the MKL implementation of [1][1] for Algorithm 1.

How many lines? We first test the proposed approach on synthetic data. Points in 2D are generated within $[\ 0\ 1\ 0\ 1\ ]$ which form lines in a specific configuration. Each line contains 50 points which are distributed normally with variance σ^2 around the line. A number of Q outliers which do not correspond to any lines are also randomly added. Five types of patterns are used in this experiment as illustrated in Fig. 4. The task is to apply model selection to estimate the number of lines present in the data.

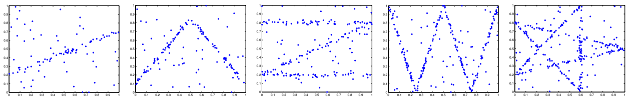


Figure 4. Five patterns of lines with 1 to 5 lines per data set.

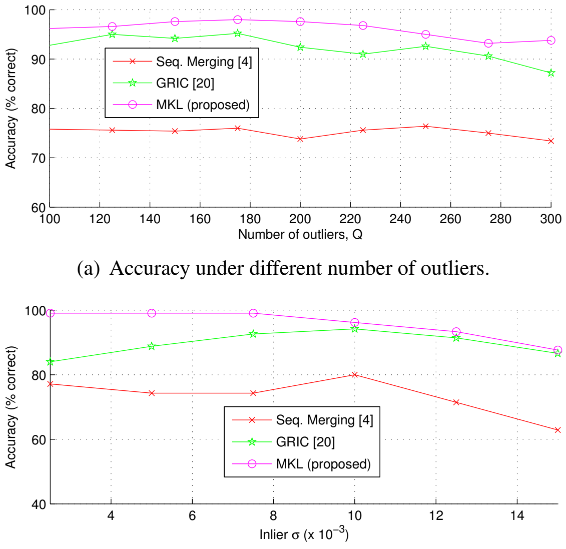
Given a set of data we preprocess it by the outlier removal scheme proposed in [4]. Our technique is then applied to estimate the number of lines existing in the data. We first oversegment the data using ORK by sampling $M = 5000$ putative hypotheses, where each hypothesis is a line estimated from a randomly selected pair of points. Intersection kernels with window size $h = 100, 200, \ldots 5000$ are used as the base kernels for MKL. Parameter h is incremented by 100 (instead of using all possible window sizes within $1 \leq h \leq 5000$) to reduce the learning time of MKL. Fig. 5 illustrates a typical result of the proposed approach.

We examine the performance of the approach under various outlier rates (i.e. $Q = 100, 125, \ldots, 300$) and inlier scales (i.e. $\sigma = 0.0025, 0.005, \ldots, 0.015$). When varying Q we fix σ at 0.01 while Q is maintained at 200 when varying σ. For each setting of σ and Q we generate 100 instances of each configuration in Fig. 4. We then apply the proposed approach on each data instance. By stochastically generating increasingly difficult data we examine the stability of the proposed approach under different conditions. For each σ and Q we obtain the accuracy (over all instances of the 5 patterns) of the proposed approach in model selection.

We compare our model selection criterion to Geometric AIC [12], Geometric MDL [12] and GRIC [20]. These can be summarised in the following form

$$Criterion(\mathcal{M}) = \kappa_1 J(\mathcal{M}) + \kappa_2 K(\mathcal{M}), \qquad (17)$$

where $\mathcal{M}$ indicates a particular model (i.e. a specific number of structures and their fit onto the data), $J(\mathcal{M})$ is the fitting error of $\mathcal{M}$ and $K(\mathcal{M})$ is the complexity measure of $\mathcal{M}$ (refer to [12, 20] for their specific algebraic form). Positive constants κ_1 and κ_2 encode the relative importance of $J(\mathcal{M})$ and $K(\mathcal{M})$ and we tune these to the best of our efforts for the experiments. We adapt Algorithm 1 to these criteria by evaluating $\bar{A}_i \longleftarrow Criterion(\mathcal{M})$ and choosing clusters to merge by exhaustive search, i.e. find the pair that causes the largest decrease in $Criterion(\mathcal{M})$ after merging. Finally we also compare our approach to the sequential structure-removal model selection scheme of [4].



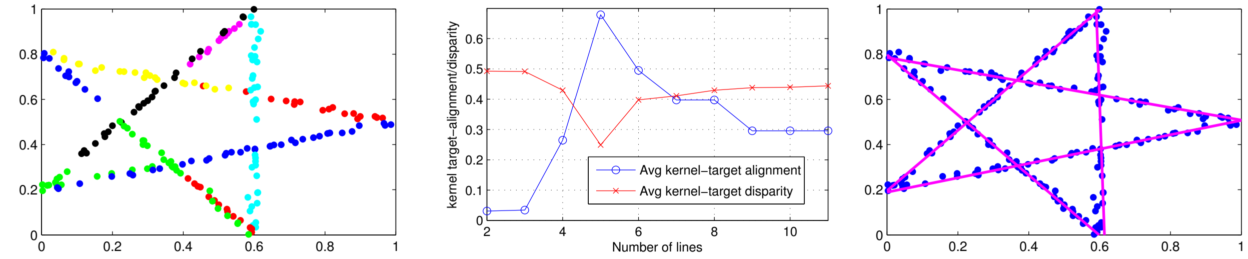
(a) Accuracy under different number of outliers.

(b) Accuracy under varying inlier scale.

Figure 6. Comparing accuracy in estimating the number of lines.

Fig. 6 displays the obtained results. Since our outcomes show that all 3 criteria from [12, 20] perform similarly (with GRIC being slightly better) only GRIC is shown in Fig. 6. It can be seen that all methods perform consistently across different outlier rates, while the accuracies decrease with an increase in σ (which causes lines to be less well-defined). The sequential structure-removal approach [4] is the weakest method with the lowest average accuracy. Our method

[1] Available at http://www.di.ens.fr/˜fbach/path/.



(a) Initial clustering reveals 11 structures. (b) Max. alignment achieved at 5 structures. (c) Results after refining model fits.

Figure 5. Applying MKL-based model selection for 2D line fitting (best viewed in color). The original data is shown in Fig. 4. Outliers are first removed using the method in [4]. The remaining points are oversegmented using ORK resulting in (a) (note that the colours repeat). After the optimal model is obtained using Algorithm 1, a line is fitted onto each cluster in the model via least squares, yielding (c).

is on average more accurate that GRIC in Fig. 6(a), while in Fig. 6(b) our method substantially outperforms GRIC when σ is lower but is matched by GRIC as σ increases. Note that as σ increases the outlier removal scheme of [4] deteriorates (e.g. incorrectly removing inliers and retaining outliers) thus impacting all compared methods negatively.

How many motions? Our second experiment relates to model selection in motion segmentation under the affine camera model. We first run the proposed approach on sequences from the Hopkins 155 benchmark dataset [21] to estimate the number of motions. This dataset contains 155 sequences with tracked feature points. In terms of visual content the dataset can be divided into 3 categories: checkerboard sequences (104), traffic sequences (38) and articulated/non-rigid motions (13). Only 2- and 3-motion sequences are available in the dataset, where 120 sequences contain 2 motions. An example sequence is shown in Fig. 1.

We compare the proposed approach to the rank detection method of [12] (which is based on Geometric AIC) and the clustering-based method of [3]. We also adapt Algorithm 1 to using GRIC [20] as the model selection and structure merging criterion. Table 1 summarises the obtained results. It can be seen that the proposed method with MKL as the criterion is the most accurate for 2-motion sequences (99 correct), while Algorithm 1 adapted to using GRIC is the most successful for 3-motion sequences (23 correct). In terms of overall accuracy the proposed method with MKL is higher than the other methods.

	# correct	# correct	% correct
	2 motions	3 motions	overall
Method	120 seqs.	35 seqs.	155 seqs.
Rank detect. [12]	97	5	65.80
Clustering [3]	80	17	62.58
Algorithm 1	**99**	17	**74.84**
Algo. 1 + GRIC	88	**23**	71.61

Table 1. Model selection performance on the Hopkins 155 dataset.

Testing on sequences with 2 or 3 motions only does not reveal the generalisation capability of the model selection approach to sequences with more than 3 motions. To solve this problem we concatenate motion trajectories to produce sequences with more than 3 motions. A trajectory matrix with n tracked feature points across F frames is defined as

$$\mathbf{T} = \begin{bmatrix} \mathbf{x}_{11} & \dots & \mathbf{x}_{n1} \\ \vdots & \ddots & \vdots \\ \mathbf{x}_{1F} & \dots & \mathbf{x}_{nF} \end{bmatrix} \in \mathbb{R}^{2F \times n}, \qquad (18)$$

where each $\mathbf{x}_{if} = [\ x_{if}\ y_{if}\ ]^T$ is the coordinate of the i-th feature point in f-th frame. Given two trajectory matrices $\mathbf{T}_1 \in \mathbf{R}^{2F_1 \times n_1}$ and $\mathbf{T}_2 \in \mathbb{R}^{2F_2 \times n_2}$ we combine them to create a new trajectory matrix $\mathbf{T}_3$ as follows:

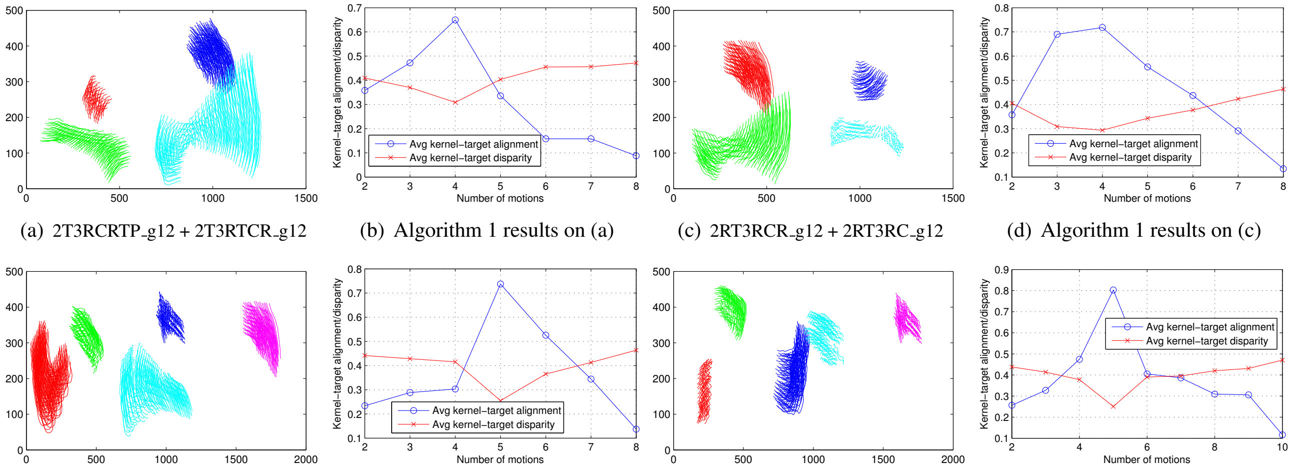
$$\mathbf{T}_3 = [\ \mathbf{T}'_1\ \mathbf{T}'_2\ ] \in \mathbb{R}^{2F_3 \times n_3} \qquad (19)$$

where $F_3 = \min(F_1, F_2)$ and $n_3 = n_1 + n_2$. If $F_j = F_3$ then $\mathbf{T}'_j = \mathbf{T}_j$, else $\mathbf{T}'_j$ is taken as rows 1 to $2F_3$ of $\mathbf{T}'_j$. Further, we add to the x_{if}'s in $\mathbf{T}'_2$ the width of the video frame of the sequence corresponding to $\mathbf{T}_1$.

We combine sequences from the Hopkins 155 dataset in the manner of Eq. (19) to create new sequences with 4 and 5 motions. Fig. 7 illustrates the new sequences and results from applying Algorithm 1. It can be seen that the maximum (minimum) alignment (disparity) is achieved at the correct number of motions. Table 2 depicts the results for all the compared approaches. Only the proposed approach with MKL as the model selection criterion correctly estimated the number of motions for all the sequences in Fig. 7.

	No. of estimated motions			
Method	7(a)	7(c)	7(e)	7(g)
Rank detect. [12]	2	2	3	3
Clustering [3]	4	4	6	5
Algorithm 1	**4**	**4**	**5**	**5**
Algo. 1 + GRIC	5	4	5	6

Table 2. Number of estimated motions for sequences in Fig. 7.



(a) 2T3RCRTP_g12 + 2T3RTCR_g12 (b) Algorithm 1 results on (a) (c) 2RT3RCR_g12 + 2RT3RC_g12 (d) Algorithm 1 results on (c)

(e) 1RT2TCRT_A_g12 + 2R3RTCRT_g12 + 2RT3RCR_g12 (motion 1 only) (f) Algorithm 1 results on (e) (g) 1R2TCRT_g12 + 1RT2RCRT_g12 + 2R3RTCRT_g12 (motion 1 only) (h) Algorithm 1 results on (g)

Figure 7. Model selection on sequences with 4 and 5 motions (best viewed in color). (a), (c), (e) and (g) show the trajectory of (actual) tracked points in the combined sequences, where each color corresponds to a unique motion, and the captions indicate the original sequences from Hopkins 155 which make up the combined sequences. Ground truth: (a) and (c) 4 motions, (e) and (g) 5 motions.

5. Conclusions

In this paper we propose a novel approach for model selection based on kernel optimisation. Our method first over-segments the input data to arrive at an initial set of structures. We then carry out a series of kernel optimisation to determine if each pair of structures should be merged. The algorithm then iteratively merges pairs of structures, and the model selection criterion is simply the average kernel-target alignment of a particular model. Experiments show that the proposed approach is highly accurate in determining the number of instances of a generic model in the input data. We also demonstrate its successful application on model selection for affine camera multi-body motion segmentation. Future work will be directed towards improving the efficiency of MKL in our model selection approach.

References

[1] F. Bach, R. Thibaux, and M. I. Jordan. Computing regularization paths for learning multiple kernels. In *NIPS*, 2004.

[2] A. M. Cheriyadat and R. J. Radke. Non-negative matrix factorization of partial track data for motion segmentation. In *ICCV*, 2009.

[3] T.-J. Chin, H. Wang, and D. Suter. The ordered residual kernel for robust motion subspace clustering. In *NIPS*, 2009.

[4] T.-J. Chin, H. Wang, and D. Suter. Robust fitting of multiple structures: The statistical learning approach. In *ICCV*, 2009.

[5] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *IJCV*, 29(3), 1998.

[6] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola. On kernel-target alignment. In *NIPS*, 2001.

[7] N. da Silva and J. Costeira. The normalized subspace inclusion: Robust clustering of motion subspaces. In *ICCV*, 2009.

[8] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *CVPR*, 2009.

[9] M. A. Fischler and R. C. Bolles. Random sample concensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24:381–395, 1981.

[10] D. Forsyth and J. Ponce. *Computer vision: a modern approach*. Prentice Hall, 2002.

[11] Y.-D. Jian and C.-S. Chen. Two-view motion segmentation by mixtures of Dirichlet process with model selection and outlier removal. In *ICCV*, 2007.

[12] K. Kanatani. Geometric information criterion for model selection. *IJCV*, 26(3):171–189, 1998.

[13] K. Kanatani and C. Matsunaga. Estimating the number of independent motions for multibody segmentation. In *ACCV*, 2002.

[14] F. Lauer and C. Schnörr. Spectral clustering for linear subspaces for motion segmentation. In *ICCV*, 2009.

[15] K. Schindler and D. Suter. Two-view multibody structure-and-motion with outliers through model selection. *IEEE TPAMI*, 28(6):1–13, 2006.

[16] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge University Press, 2004.

[17] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *JLMR*, 7, 2006.

[18] C. V. Stewart. Robust parameter estimation in computer vision. *SIAM Review*, 41(3):513–537, 1999.

[19] N. Thakoor and J. Gao. Branch-and-bound hypothesis selection for two-view multiple structure and motion segmentation. In *CVPR*, 2008.

[20] P. Torr. Geometric motion segmentation and model selection. *Phil. Trans. Royal Society of London A*, 356(1740):1321–1340, 1998.

[21] R. Tron and R. Vidal. A benchmark for the comparison of 3-D motion segmentation algorithms. In *CVPR*, 2007.