

Segmentation by Continuous Latent Semantic Analysis for Multi-structure Model Fitting

Guobao Xiao · Hanzi Wang · Jiayi Ma · David Suter

Received: date / Accepted: date

Abstract In this paper, we propose a novel continuous latent semantic analysis fitting method, to efficiently and effectively estimate the parameters of model instances in data, based on latent semantic analysis and continuous preference analysis. Specifically, we construct a new latent semantic space (LSS): where inliers of different model instances are mapped into several independent directions, while gross outliers are distributed close to the origin of LSS. After that, we analyze the data distribution to effectively remove gross outliers in LSS, and propose an improved clustering algorithm to segment the remaining data points. On the one hand, the proposed fitting method is able to achieve excellent fitting results; due to the effective continuous preference analysis in LSS. On the other hand, the proposed method can efficiently obtain final fitting results due to the dimensionality reduction in LSS. Experimental results on both synthetic data and real images demonstrate that the proposed method achieves significant superiority over several state-of-the-art model fitting

methods on both fitting accuracy and computational speed.

Keywords Latent semantic analysis · preference analysis · geometric model fitting · multi-structure data.

1 Introduction

Geometric model fitting is one of the most challenging tasks in computer vision, and it has been applied to several types of computer vision tasks, such as, homography/fundamental matrix estimation [40], vanishing point detection [16], 3D reconstruction [43], and motion segmentation [42]. One of the main challenges in model fitting is to robustly estimate the model parameters for data involving a large number of outliers and multiple structures.

1.1 Background

1.1.1 Several Primary Definitions

We firstly provide some definitions in model fitting to address the model fitting problem.

A *geometric model* is usually specified before model fitting. Several popular geometric models have been used, such as, line, circle, homography/fundamental matrix, etc.. Once a type of geometric model is specified, one can explain data with the model.

A *model instance* (also called a structure) is an instance of the specified type of model in the data. Multi-structural data contain multiple model instances.

A *minimal subset* is the minimum number of data points that are required generating a model hypothesis,

✉ Hanzi Wang
E-mail: hanzi.wang@xmu.edu.cn

Guobao Xiao
Fujian Key Laboratory of Sensing and Computing for Smart City, School of Informatics, Xiamen University, China; Fujian Provincial Key Laboratory of Information Processing and Intelligent Control, College of Computer and Control Engineering, Minjiang University, China

Hanzi Wang
Fujian Key Laboratory of Sensing and Computing for Smart City, School of Informatics, Xiamen University, China

Jiayi Ma
Electronic Information School, Wuhan University, China

David Suter
School of Science, Edith Cowan University, Australia

e.g., 2 data points for line fitting, 4 data points for homography fitting.

A *model hypothesis* is generated based on each minimal subset. The subset is sampled using different types of sampling strategies. For data with outliers, many model hypotheses are usually required being generated to estimate the model instances in data.

Based on the above definitions, the model fitting problem can be described as: Given a set of input data points and a specified type of model, the goal is to estimate the parameters of model instances that best explain the input data [38, 15, 25, 17].

Generally, there are two steps included in the model fitting procedure: 1) Sample a number of minimal subsets to generate a set of model hypotheses; 2) Estimate the parameters of model instances according to the generated model hypotheses by model selection.

Some methods have been proposed to improve either the quality of sampling (e.g., Proximity [14], PROSAC [7], MultiGS [6] and MAGSAC [3, 4]) or the performance of model selection (e.g., HMSS [31] and RansaCov [22]).

In this paper, we mainly focus on both the effectiveness and computational efficiency of model selection.

1.1.2 Prior Work

Existing fitting methods can be roughly classified into two categories, i.e., the consensus analysis based and the preference analysis based fitting methods. The consensus analysis based fitting methods (e.g., RANSAC [11], RansaCov [22], RHT [41], GPbM [26], AKSWH [35] and MSHF [36]) directly select model instances from a number of generated model hypotheses according to some evaluation criterions (such as, the number of inliers or the weighting scores of model hypotheses), and then segment data points according to the user-specified or estimated inlier noise scale.

Among these methods, RANSAC is one of the most popular fitting methods due to its simplicity and robustness to outliers. The main steps of RANSAC are described as follows:

- Randomly sample a number of minimal subsets of data points and then generate the corresponding model hypotheses.
- Select the model hypothesis that includes the maximum number of inliers as the estimated model instance.

RANSAC is simple yet robust. However, it suffers from several limitations. For example, it requires randomly sampling a large number of minimal subsets for model hypothesis generation to “hit”¹ a model instance in

¹ That is, among the generated model hypotheses, there is at least one model hypothesis (generated by a minimal subset) corresponding to a true model instance in data.

data with high proportions of outliers. However, most of the sampled minimal subsets are not significant (i.e., most of the sampled minimal subsets include outliers). This will cause high computational complexity, especially for high dimensional models. Moreover, RANSAC sequentially selects the best model hypothesis as the estimated model instance for multi-structural data via a “fit-and-remove” framework, which is not very effective and computationally efficient [35].

To improve the fitting performance, RansaCov formulates the model fitting problem as a problem of set coverage and it searches for the best sets that include the maximum number of inliers. RHT [41] assigns each generated model hypothesis a score, and then selects the model hypothesis with the maximum score as the estimated model instance. GPbM [26] formulates the model fitting problem as an optimization problem. AKSWH [35] directly clusters the generated model hypotheses, and then selects the model hypothesis with the maximum weighting score from each cluster as an estimated model instance. MSHF [36] combines hypergraphs and mode-seeking to directly select model hypotheses as the estimated model instances according to their weight scores and a similarity matrix. Although these fitting methods are not very sensitive to data distribution (i.e., their performance is relatively less affected by unbalanced data distribution), their fitting performance largely depends on the quality of the generated model hypotheses.

The preference analysis based fitting methods (e.g., SWS [28], HF [39], RPA [23], J-linkage [32], T-linkage [21] and MCT [24]) estimate model instances by labeling data points based on the generated model hypotheses. Specifically, SWS [28] adopts large hyperedges to represent the relationship between model hypotheses and data points for model fitting. HF [39] introduces a novel hypergraph model to formulate the model fitting problem as a hypergraph partition problem. RPA [23] combines principal component analysis and non negative matrix factorization to deal with multi-structure model fitting. J-linkage [32]/T-linkage [21] directly cluster data points according to their preference sets. MCT [24] extends T-linkage to handle different nested classes of models.

Moreover, there are also some energy-based model fitting methods, e.g., PEaRL [12], RCMSA [27] and CORAL [1]. We can see that, most of the currently existing fitting methods estimate the fitting results from the residual matrix, which is derived from model hypotheses and data points. However, to cover all model instances in data, they often require sampling a large number of model hypotheses. Then, the fitting methods may suffer from high computational complexity when

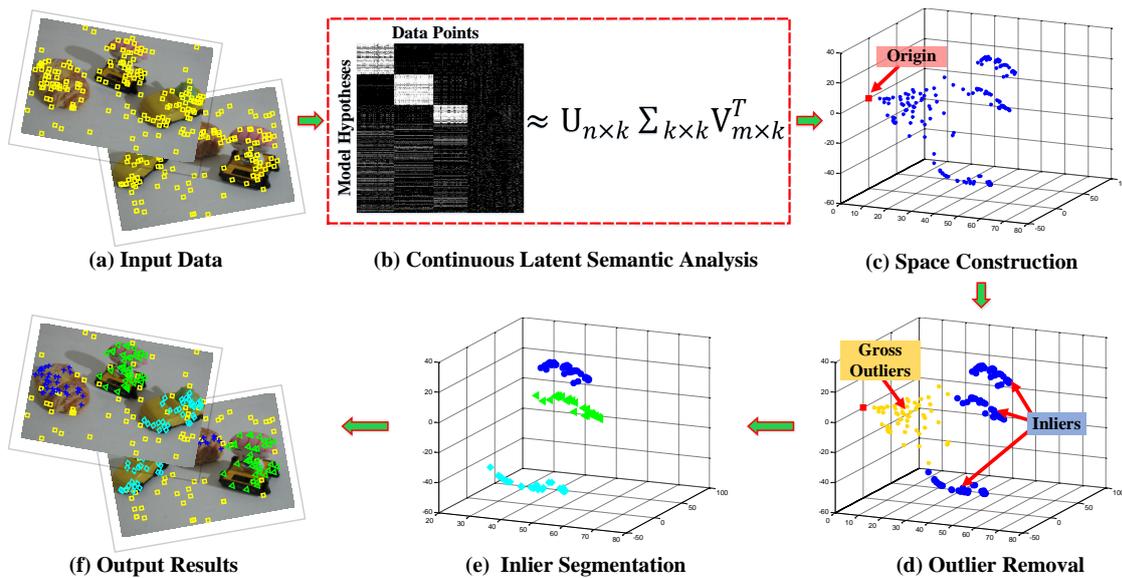


Fig. 1 Overview of the proposed method for two-view based motion segmentation.

they compute the similarities between all data points, especially for large-size data.

1.2 Motivations and Contributions

In this paper, we propose a continuous latent semantic analysis method (called as CLSA) for robust multi-structure fitting. We aim to exploit the advantage of preference analysis (that is able to achieve good fitting accuracy) and yet reduce the high computational complexity of multi-structural model fitting by applying latent semantic analysis (LSA) [8] on data points (involving a large number of outliers). Note that LSA is a valuable analysis tool, which was originally used to construct a low-dimensional subspace by analyzing the relationships between a set of documents and terms for word processing, to effectively reduce the dimensions of word-document count vectors. Inspired by that work, we construct a new low-dimensional latent semantic space (LSS), to analyze the relationships among data points, by both LSA and continuous preference analysis (CPA)² implemented by T-linkage [21], for solving the model fitting problem.

It is worth pointing out that the combination of LSA and CPA is an important improvement over either LSA or CPA. That is, for LSA, CPA can capture more accurate relationships between model hypotheses and data points (note that LSA has no notion of these). For CPA, LSA can be used to reduce the high dimension of a preference matrix generated by CPA while keep-

ing its principal preference information. Thus, both LSA and CPA jointly lead to better fitting performance and higher computational efficiency. To the best of our knowledge, we are the first to exploit the relationships between data points and model hypotheses via both LSA and CPA, for robust multi-structure model fitting.

Moreover, we analyze the constructed LSS to remove gross outliers, and cluster the remaining data points into independent directions in LSS. Note that there usually are two kinds of outliers, i.e., gross outliers and pseudo outliers in data points. Gross outliers are the data points that do not belong to the inliers of any model instance in data; while pseudo outliers are the data points that belong to the inliers of one model instance, but they are the outliers to the other model instances in data. In our work, we aim to effectively remove gross outliers and cluster/segment inliers (belonging to different model instances) in LSS. Firstly, we adaptively set a threshold based on an information theoretic approach [10] to remove gross outliers in LSS. Then, by analyzing the distribution of the remaining data points in LSS, we cluster these points into different directions via an improved clustering algorithm. Note that, in LSS, the high-order model fitting problem (e.g., homography estimation and two-view based motion segmentation) is formulated as a simple clustering problem, which is much easier to deal with.

We show the overview of the proposed CLSA method applied to two-view based motion segmentation in Fig. 1. In Fig. 1(b), a preference matrix, whose entries are the preference values of data points to model hypotheses, and the key step of continuous latent semantic analysis are shown. From Fig. 1(c), we can see that, each

² CPA is a soft generalization of the binary preference analysis.

of the input feature matches is mapped to a point in LSS by the proposed method. In LSS, the points corresponding to the inliers and gross outliers are distributed differently. That is, the points corresponding to the gross outliers are distributed close to the origin of LSS, while the points corresponding to the inliers of different model instances are distributed in their respective directions and they are far from the origin of LSS (see Fig. 1(d)). After removing the gross outliers from the input data, the complex model fitting problem becomes a much easier clustering problem, which can be solved in an effective and efficient manner (see Fig. 1(e)).

In this paper we are primarily concerned with proposing a model fitting method to segment data. In model fitting, one tries to jointly determine both the parameters of the underlying models, and the segmentation induced by the assignment of data to the models they fit. Since our main aim is to segment the data, the accuracy of the estimated model instances is only a means to an end. In contrast, for model extraction/estimation, the accuracy of concern is that of the model parameter estimates, not the induced segmentation per se. Despite the focus of this work on segmentation, the ideas could also be used in areas where the model parameters are more of interest but that is beyond the scope of this paper.

The key contributions of this work are summarized as follows:

- We propose a way to analyze the complex relationships between data points and model hypotheses by exploiting the advantage of both latent semantic analysis and continuous preference analysis, to improve the performance on both segmentation accuracy and computational efficiency for robust model fitting.
- We propose an adaptive gross outlier removal strategy in the new latent semantic space, where inliers and gross outliers are separately distributed. Then we formulate the complex model fitting problem as a simple clustering problem on the remaining data without gross outliers.
- We improve a traditional clustering method [20] by introducing a preference function to effectively seed the initial centers, which achieves more stable clustering results for the task of model fitting.

Our experimental evaluations and comparisons demonstrate the significant superiority of the proposed method over several state-of-the-art geometric model fitting methods for different model fitting tasks.

The rest of the paper is organized as follows: We describe the components of the proposed fitting method

in Sec. 2. We summarize the complete method in Sec. 3. We present the experimental results on both synthetic and real data in Sec. 4. We discuss the proposed method in Sec. 5 and draw conclusions in Sec. 6.

2 The Methodology

In this section, we describe the details of the proposed continuous latent semantic analysis fitting method (called CLSA). We first describe the problem formulation in Sec. 2.1, and then review the background of latent semantic analysis (LSA) [8] (Sec. 2.2). We then construct a novel latent semantic space by LSA and the continuous preference analysis (CPA) in Sec. 2.3. After that, we analyze the data distribution in the constructed latent semantic space, and then propose an adaptive gross outlier removal strategy (Sec. 2.4). Then we propose an improved clustering algorithm to segment the remaining data points (Sec. 2.5).

2.1 Problem Formulation

Given a set of data points $X = \{x_1, \dots, x_n\}$ and a set of model hypotheses $\Theta = \{\theta_1, \dots, \theta_n\}$ generated by sampling, we define the *preference function* [21], which represents the preference of a data point x_i to a model hypothesis θ_j , as follows:

$$f(x_i, \theta_j) = \exp\left\{-\frac{d(x_i, \theta_j)}{\psi}\right\}, \quad (1)$$

where $d(x_i, \theta_j)$ denotes the residual value measured with the Sampson distance [33] from a data point x_i to a model hypothesis θ_j . ψ is a threshold (and the influence of its value on the proposed CLSA will be discussed with more details in Sec. 4.1).

By using Eq. (1), we compute the preference matrix, which is used to estimate the parameters of model instances in the data. Then, we also segment input data points belonging to different model instances into different groups. The cost function is defined as [22,26]:

$$SE = \frac{\# \text{ mislabeled data points}}{\# \text{ data points}}, \quad (2)$$

where $\# \text{ mislabeled data points}$ is the number of mislabeled data points, and $\# \text{ data points}$ is the number of input data points. Note that, if the estimated model instance is close to the true model instance, the value of SE will be small; Otherwise, the value of SE will be large.

2.2 Latent Semantic Analysis (LSA)

Latent semantic analysis (LSA) was proposed to analyze a binary matrix of documents/terms, via singular value decomposition (SVD), for document representation.

Specifically, given documents $Do = \{do_1, \dots, do_n\}$ with terms $Te = \{te_1, \dots, te_m\}$, generate a binary matrix \mathbf{TD} , which denotes the relationship between documents and terms (i.e., $td(i, j) = 1$ if the i -th document belongs to j -th term; Otherwise, $td(i, j) = 0$). The key idea of LSA is to map documents to a latent space, where documents are close if they share meaningful association, based on \mathbf{TD} by SVD.

LSA firstly decomposes \mathbf{TD} by SVD as follows:

$$\mathbf{TD}_{n \times m} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \quad (3)$$

where \mathbf{U} and \mathbf{V} are the orthogonal matrices of left and right singular vectors $\mathbf{U}\mathbf{U}^T = \mathbf{V}\mathbf{V}^T = \mathbf{I}$ (here \mathbf{I} is an identity matrix), respectively. $\mathbf{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_r)$ is a diagonal matrix with singular values on the diagonal and the singular values are listed in non-increasing order, i.e., $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$, $r = \text{rank}(\mathbf{TD})$.

Then \mathbf{TD} is approximately computed by the top k singular values in $\mathbf{\Sigma}$, i.e.,

$$\mathbf{TD} = \tilde{\mathbf{TD}} \approx \mathbf{U}_{n \times r_k} \mathbf{\Sigma}_{r_k \times r_k} \mathbf{V}_{m \times r_k}^T, \quad (4)$$

where r_k is the number of considering $\text{rank}(\mathbf{TD})$. After that, LSA uses the rows of $\mathbf{U}_{n \times r_k} \mathbf{\Sigma}_{r_k \times r_k}$ as the coordinates for documents in the latent space. This is because the document-to-document inner products are given by:

$$\tilde{\mathbf{TD}}\tilde{\mathbf{TD}}^T = \mathbf{U}_{n \times r_k} \mathbf{\Sigma}_{r_k \times r_k} \underbrace{\mathbf{V}_{m \times r_k}^T \mathbf{V}_{m \times r_k}}_{\mathbf{I}} \mathbf{U}_{n \times r_k}^T \mathbf{\Sigma}_{r_k \times r_k}. \quad (5)$$

Based on the latent space, LSA is able to effectively analyze the relationship between documents.

2.3 Latent Semantic Space Construction

Inspired by LSA, we propose to analyze an inlier indexing matrix (generated by considering both data points and model hypotheses) via SVD for the model fitting problem. A binary inlier indexing matrix is used to represent the relationship between data points and model hypotheses. Here, we use a continuous preference matrix [21] (derived from a preference function) instead of a binary one to improve its fitting performance.

For n input data points and m generated model hypotheses, a singular value decomposition of a continuous preference matrix $\mathbf{F}_{n \times m}$ (each entry is derived from

Eq. (1)) has the following form:

$$\mathbf{F}_{n \times m} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \quad (6)$$

i.e.

$$\begin{aligned} & \begin{bmatrix} f(x_1, \theta_1) & \cdots & f(x_1, \theta_m) \\ \vdots & \ddots & \vdots \\ f(x_n, \theta_1) & \cdots & f(x_n, \theta_m) \end{bmatrix} \\ &= \left[\begin{bmatrix} u_1 \\ \vdots \\ u_r \end{bmatrix} \cdots \begin{bmatrix} u_r \\ \vdots \\ u_r \end{bmatrix} \right] \cdot \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_r \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ \vdots \\ v_r \end{bmatrix} \end{aligned} \quad (7)$$

Recall that, a preference matrix denotes the preference between data points and model hypotheses. Specifically, if the value of $f(x_i, \theta_j)$ is large, it means that the data point x_i prefers the model hypothesis θ_j . Thus, after decomposition by SVD, a left singular vector u_i denotes the preference of all data points to the i -th topic model of model hypotheses. A topic model is an estimated model instance for model fitting. Accordingly, a right singular vector v_j denotes the preference of all model hypotheses to the j -th topic model of data points.

Each singular value σ_i denotes the strength of correlation between a data point and a model hypothesis. That is, if the σ_i value is large, then it is a high possibility that the i -th topic model of data points and the i -th topic model of model hypotheses correspond to the same model instance in data; Otherwise, the i -th topic model of data points and the i -th topic model of model hypotheses are usually insignificant and thus they can be ignored. This is because an inlier and the corresponding model instance have much stronger correlation than the other cases. Therefore, we only need to analyze the top r_k data points and model hypotheses instead of analyzing all data points and model hypotheses. Here, r_k is usually set to a small value, i.e., $r_k \ll r$ (we also further discuss its influence in Sec. 4.1). Thus, Eq. (6) can be rewritten as:

$$\mathbf{F}_{n \times m} = \tilde{\mathbf{F}}_{n \times m} \approx \mathbf{U}_{n \times r_k} \mathbf{\Sigma}_{r_k \times r_k} \mathbf{V}_{m \times r_k}^T. \quad (8)$$

Eq. (8) is also consistent with the property of SVD that the top 10% (even 1%) of singular values account for over 99% of all singular values.

Then, to simplify the complex model fitting problems, we construct a latent space where the data points belonging to the same model instance (these data points share similar preference to the same model hypothesis) are close to each other. That is, the value of a data point x_i in the constructed space only relates to the left singular vectors and has nothing with the origin value of x_i . Here we call the constructed space the ‘‘latent semantic space (LSS)’’, because the space involves the latent

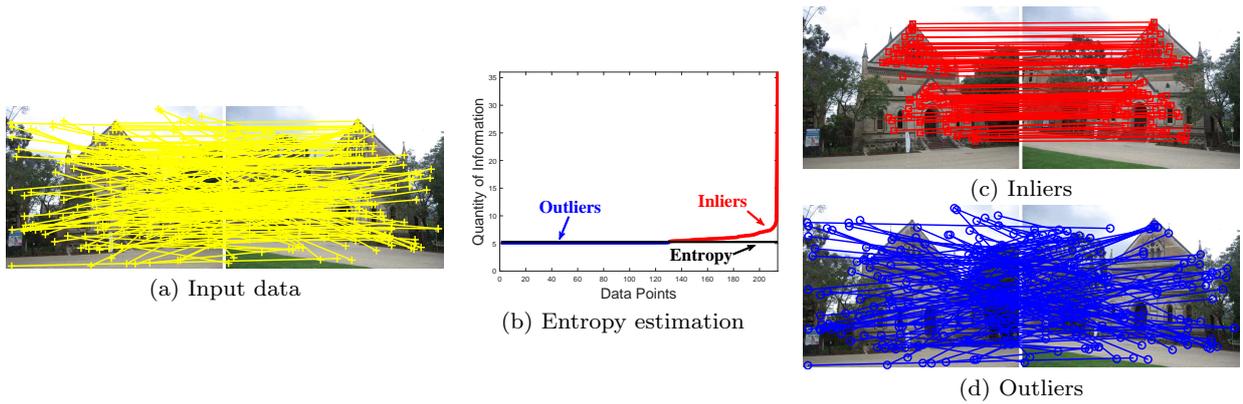


Fig. 2 An example of outlier removal based on the information theoretic approach for homography estimation. (a) The input “Elderhalla” image pair. (b) The proposed outlier removal strategy. (c) The estimated inliers. (d) The removed gross outliers. The gross outliers are marked in blue, and the inliers of different model instances are marked in red.

semantic information (the inliers belonging to the same model instance in the data usually correspond to the mapped points in one direction of LSS).

Note that the points-to-points inner products based on Eq. (8) are given by:

$$\begin{aligned} \tilde{\mathbf{F}}_{n \times n} &= \tilde{\mathbf{F}}_{n \times m} \tilde{\mathbf{F}}_{n \times m}^T \\ &= \mathbf{U}_{n \times r_k} \Sigma_{r_k \times r_k}^2 \mathbf{U}_{n \times r_k}^T. \end{aligned} \quad (9)$$

Now, we have a $n \times n$ matrix $\tilde{\mathbf{F}}$ to determine singular values Σ and left singular vectors \mathbf{U} . Thus, we use the rows of $\mathbf{U}_{n \times r_k} \Sigma_{r_k \times r_k}$ as defining coordinates for data points in LSS.

After that, we no longer analyze the preference matrix but LSS, and the dimension is reduced from a large m value of a preference matrix to a much smaller r_k value of LSS. It is also worth pointing out that, the value of each data point in LSS has its specific physical interpretation. That is, the value of a data point x_i in the j -th dimension of LSS corresponds to the preference of x_i to the j -th topic model of model hypotheses. Thus, the complex model fitting problem is reduced to a much easier LSS analysis problem.

2.4 Adaptive Outlier Removal

Based on the constructed LSS, we propose a novel simple but effective outlier removal strategy in this subsection.

In LSS, a data point x_i corresponding to an inlier is assigned a large value in a direction (which means the data point is distributed far from the origin of LSS), while x_i corresponding to a gross outlier is assigned a small value. The reason behind this is that, the value of x_i in the j -th dimension of LSS corresponds to the preference of x_i to the j -th topic model of model

hypotheses. Obviously, a topic model of model hypotheses prefers x_i if it is an inlier while not any topic model of model hypotheses prefers x_i if it is a gross outlier. Thus, in LSS, the points corresponding to inliers of model instances are distributed in several independent directions while the points corresponding to gross outliers are distributed close the origin (\mathbf{O}) of LSS. That is, the points corresponding to inliers are far from the origin \mathbf{O} while the points corresponding to gross outliers are close to \mathbf{O} . Thus, we remove the points corresponding to gross outliers by using a threshold, according to the distance between the mapped points and \mathbf{O} in LSS. However, it is difficult to manually set a proper threshold value for different input data.

In this paper, we propose to use information theory [10] to adaptively set the threshold value for outlier removal. Firstly, given a set of mapped points $\hat{\mathbf{X}} = \{\hat{x}_i\}_{i=1, \dots, n}$ in LSS, we compute the Euclidean distance $\hat{d}_{\hat{x}_i}^o$ between each mapped point \hat{x}_i and \mathbf{O} in LSS. Then, we compute the gap g_i between $\hat{d}_{\hat{x}_i}^o$ and the maximum distance of the mapped points to \mathbf{O} as:

$$g_i = \max\{\mathbf{D}^o\} - \hat{d}_{\hat{x}_i}^o, \quad (10)$$

where $\mathbf{D}^o = \{\hat{d}_{\hat{x}_i}^o\}_{i=1, \dots, n}$. After that, we measure the quantity of information provided by each mapped point \hat{x}_i :

$$Q(\hat{x}_i) = -\log p(\hat{x}_i), \quad (11)$$

where $p(\hat{x}_i)$ is the prior probability of each mapped point \hat{x}_i to correspond to an outlier, and it is computed by normalizing the gap:

$$p(\hat{x}_i) = g_i / \sum_{i=1}^n g_i. \quad (12)$$

The corresponding entropy is computed as:

$$L = - \sum_{i=1}^n p(\hat{x}_i) \log p(\hat{x}_i). \quad (13)$$

We remove the mapped points with quantity of information lower than the estimated entropy value L and retain the points with quantity of information higher than the entropy value L :

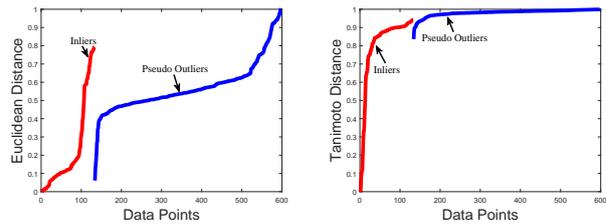
$$\hat{\mathbf{X}}_I = \{x_i | Q(\hat{x}_i) > L\}. \quad (14)$$

To illustrate the outlier removal strategy, in Fig. 2, we show an example of outlier removal (based on our information theoretic approach) for homography estimation on the ‘‘Elderhalla’’ image pair. We can see that, inliers are assigned a large quantity of information while gross outliers are assigned a smaller quantity of information (see Fig. 2(b)). The proposed outlier removal strategy successfully distinguishes the mapped points in LSS (see Fig. 2(c) and (d)).

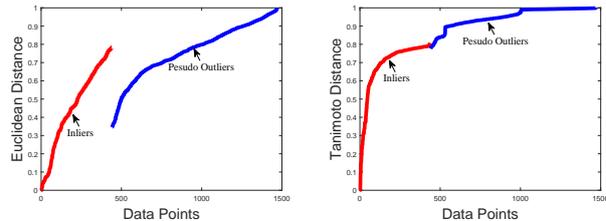
It is worth pointing out that, although [35, 36] and the proposed outlier removal strategy in this paper, use information theory similar to that in [10] to improve the fitting performance, they are significantly different in that: 1) They deal with different objects, i.e., the proposed strategy in this paper uses information theory to remove gross outliers from all data points while [35, 36] use it to remove insignificant model hypotheses from all generated model hypotheses; 2) They work in different spaces, i.e., the proposed strategy works in LSS while [35, 36] work in the parameter space.

2.5 Inlier Segmentation

The constructed latent semantic space is well approximated by multiple low-dimensional spaces. Thus, after removing the points corresponding to gross outliers (those with high probabilities in LSS), we propose to cluster the remaining mapped points, where each cluster corresponds to a model instance and the corresponding data points are the inliers of the model instance. From the distribution of the mapped points in LSS (see Fig. 1(d)), we can see that the points belonging to different clusters have obvious discrimination. The reason behind this is that, the inliers belonging to the same model instance, i.e., the same topic model of model hypotheses, will prefer the same directions in LSS. Thus, we can use a clustering method (e.g., K-means [20], DBSCAN [9] and Clusterdp [29]) to cluster the remaining points in LSS. However, these clustering methods have some limitations, e.g., K-means is sensitive to the selection of the initial centers; for DBSCAN and Clusterdp,



(a) Line fitting based segmentation



(b) Homography based segmentation

Fig. 3 Two examples of the distance distribution between an inlier and the other data points (i.e. the other inliers and the pseudo outliers) based on the Euclidean distance (Left) and the Tanimoto distance (Right) for two model fitting tasks.

it is difficult to set their parameters for different input data.

In this paper, we follow the idea of K-means++ [2], which improves K-means by employing a novel technique for seeding the initial centers. Specifically, we propose a new way of seeding the initial centers for K-means by introducing the preference function (see Eq. (1)). We measure the similarity between two points (\hat{x}_i, \hat{x}_j) in LSS based on their Tanimoto distance [30]:

$$s(\hat{x}_i, \hat{x}_j) = 1 - \frac{\langle \mathbf{PS}_i, \mathbf{PS}_j \rangle}{\|\mathbf{PS}_i\|^2 + \|\mathbf{PS}_j\|^2 - \langle \mathbf{PS}_i, \mathbf{PS}_j \rangle}, \quad (15)$$

where \mathbf{PS}_i and \mathbf{PS}_j are the preference sets of the data points \hat{x}_i and \hat{x}_j for all model hypotheses³. $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ denote the standard inner product and the corresponding induced norm, respectively.

Compared with the Euclidean distance, the Tanimoto distance is much more robust to the distribution of data points, for our case. Note that, if two data points share similar preference to model hypotheses, then their Tanimoto distance will be a small value; Otherwise, their Tanimoto distance will be a large value. In Fig. 3, we show two examples of the distance distribution between an inlier and the other data points (including the other inliers and the pseudo outliers) based on the Euclidean distance and the Tanimoto distance on the ‘‘6 lines’’ data and the ‘‘Unihouse’’ data, respectively.

³ The elements of the preference set \mathbf{PS}_i are computed according to the preference function $f(\cdot)$ in Eq. (1), based on the corresponding data point x_i and all the generated model hypotheses. Similarly, the elements of \mathbf{PS}_j can be computed.

Algorithm 1 The K-means*++ Clustering Algorithm

Input: Data points \hat{X} , the number of model instances k

- 1: Select one center c_1 randomly from the input data points in LSS and add it to the center pool.
- 2: Compute the Tanimoto distance (referred to as T-distance) between all data points and the centers in the center pool by Eq. (15).
- 3: Select the data point with the maximum T-distance as a new center and add it to the center pool.
- 4: Repeat Steps 2 to 3 until k centers are selected.
- 5: Proceed with the standard K-means algorithm [20] to obtain the labels of data points.

Output: The labels of data points \hat{X} .

We can see that, many pseudo outliers have the same distance as inliers, when the Euclidean distance is used; while only a few pseudo outliers share the same distance with inliers, when the Tanimoto distance is used. Thus, the Tanimoto distance is able to more effectively capture the distance between inliers from the same model instance than the Euclidean distance, and we employ the Tanimoto distance, rather than the Euclidean distance, in the proposed fitting method.

Then, based on the similarity, we seed the centers from data points, according to their Tanimoto distance to the closest center that is chosen. We summarize the proposed clustering algorithm (called K-means*++) in Algorithm 1. The proposed K-means*++ is an improvement over K-means++ in terms of initial center selection. Compared with the standard K-means [20], both K-means++ and K-means*++ are able to achieve more stable clustering results, due to their effectiveness for seeding the initial centers. However, it is worth pointing out that K-means*++ is able to achieve better clustering results than K-means++ for model fitting. This is because that K-means*++ uses the preference information to measure the similarity between data points, which is more effective than the Euclidean distance used by K-means++, for robust model fitting.

3 The Complete Method and Complexity Analysis

In this section, based on the components described in the previous sections, we summarize the complete Continuous Latent Semantic Analysis (CLSA) fitting method in Algorithm 2.

The proposed CLSA mainly includes three parts: the latent semantic space (LSS) construction, outlier removal, and inlier segmentation. CLSA firstly constructs LSS by both latent semantic analysis and continuous preference analysis, to map inliers of different model instances to several independent directions while gross outliers are mapped to the region close to \mathbf{O} in LSS.

Algorithm 2 The CLSA fitting method

Input: Data points X , the threshold value of ψ and the dimension r_k .

- 1: Sample a number of minimal subsets to generate model hypotheses.
- 2: Compute the preference matrix by Eq. (1).
- 3: Construct the latent semantic space (LSS) and map all data points to LSS according to Eq. (9).
- 4: Remove the points corresponding to gross outliers in LSS by Eq. (14).
- 5: Label the remaining points in LSS via the proposed K-means*++ (see Algorithm 1).
- 6: Estimate the parameters of model instances in data according to the labels of the points in LSS.

Output: The parameters of model instances and the labels of data points.

Then, CLSA analyzes the distribution of the mapped points in LSS and adaptively removes gross outliers by using information theory, which makes the model fitting problem easier to deal with. After that, CLSA labels the remaining points via an improved K-means*++ clustering algorithm, by which the parameters and inliers of model instances are estimated and segmented, respectively. As the experimental results show in Sec. 4, CLSA is able to efficiently and effectively deal with the multi-structure fitting problem.

For the computational complexity of the proposed CLSA, we mainly focus on improving both the effectiveness and computational efficiency of model selection for model fitting in this paper, and thus we do not consider the time for model hypothesis generation (i.e. Step 1). The preference matrix can be directly derived from the residual values between data points and model hypotheses (i.e. Step 2), thus its computational complexity is $O(n)$, where n is the number of data points. For constructing the latent semantic space (i.e. Step 3), the computational complexity is approximately $O(nr_k)$, where r_k is the dimension of LSS, respectively. For outlier removal (i.e. Step 4), the computational complexity is $O(n)$. For Algorithm 1, the computational complexity is $O(\hat{n}m)$, where \hat{n} is the number of estimated inliers. Thus the total complexity of CLSA approximately amounts to $O(nr_k + \hat{n}m)$.

4 Experiments

In this section, we investigate the performance of the proposed CLSA fitting method, and compare it with several state-of-the-art model fitting methods⁴, including the consensus analysis based fitting methods (i.e.,

⁴ For MSHF, we use the code provided by the authors. For RansaCov, T-linkage and RPA, we download the codes from the following website: <http://www.diegm.uniud.it/fusiello/demo>.

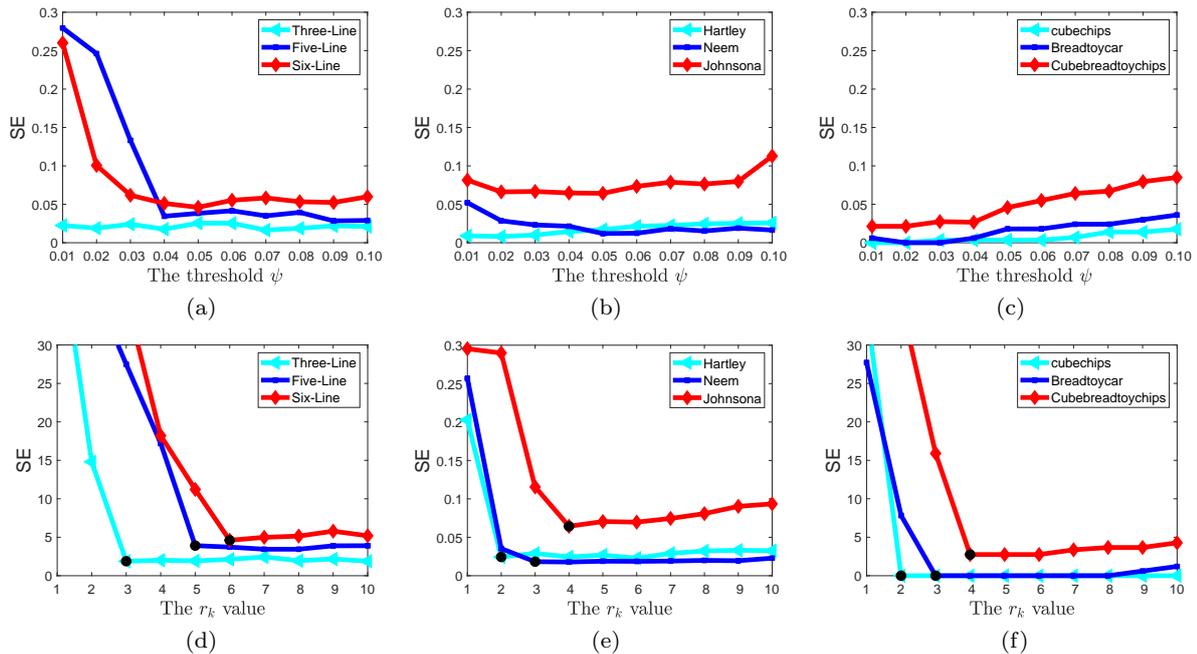


Fig. 4 The segmentation errors obtained by the proposed CLSA2 fitting method with different parameter values for line fitting based segmentation ((a) and (d)), homography based segmentation ((b) and (e)), and two-view based motion segmentation ((c) and (f)). The horizontal coordinate values of the black points in (d), (e) and (f) denote the groundtruth number of model instances in data.

MSHF [36] and RansaCov [22]) and the preference analysis based fitting methods (i.e., T-linkage [21] and R-PA [23]), on both synthetic data and real images. Moreover, to show the effectiveness of the proposed clustering algorithm, we test two versions of CLSA: CLSA1, which uses the traditional K-means algorithm [20] to cluster points in LSS, and CLSA2, which uses the proposed K-means*++ algorithm to cluster points in LSS. We also run RCMSA [27] as a baseline for high-order fitting tasks (i.e., homography based segmentation and two-view based motion segmentation) due to its effectiveness.

All experiments are run on MS Windows 10 with Intel Core i7-3770 CPU 3.4GHz and 16GB RAM. The segmentation error (SE) is measured by the cost function in Eq. (2).

4.1 Parameter Analysis and Settings

In this subsection, we analyze the influence of the parameter (i.e., the threshold ψ in Eq. (1)) of the proposed CLSA2 fitting method on its performance. We test different values of the parameters for line fitting based segmentation, homography based segmentation and two-view based motion segmentation (for each task in Sec. 4.4, Sec. 4.5 and Sec. 4.6, we test three datasets, respectively), and show the segmentation errors ob-

tained by CLSA2 with different parameter values in Fig. 4 (a)-(c).

We can see that, CLSA2 is able to achieve low segmentation errors in the three fitting tasks for most cases. Specifically, when the value of the threshold ψ (Fig. 4(a)~(c)) is respectively set to 0.04 ~ 0.10 (for line fitting based segmentation), 0.02 ~ 0.09 (for homography based segmentation) and 0.01 ~ 0.04 (for two-view based motion segmentation), CLSA2 is able to achieve relatively stable results. Thus, we set the value of ψ to 0.04 for all the following experiments.

We have also tested the proposed CLSA2 fitting method with different r_k values in Eq (9) for line fitting based segmentation, homography based segmentation and two-view based motion segmentation. We show the segmentation errors obtained by CLSA2 with different r_k values in Fig. 4 (d)-(f).

We can see that, when the r_k value is less than the ground truth number of model instances in data, CLSA2 fails to fit model instances for all three fitting tasks. In contrast, when the r_k value is equal to or larger than the ground truth number of model instances, CLSA2 is able to achieve low segmentation errors. This is because some inliers of different model instances are wrongly mapped into one direction, if the r_k value is less than the ground truth number of model instances, which causes K-means*++ (used by CLSA2) to fail to correctly label these points. Thus, we set the r_k value to

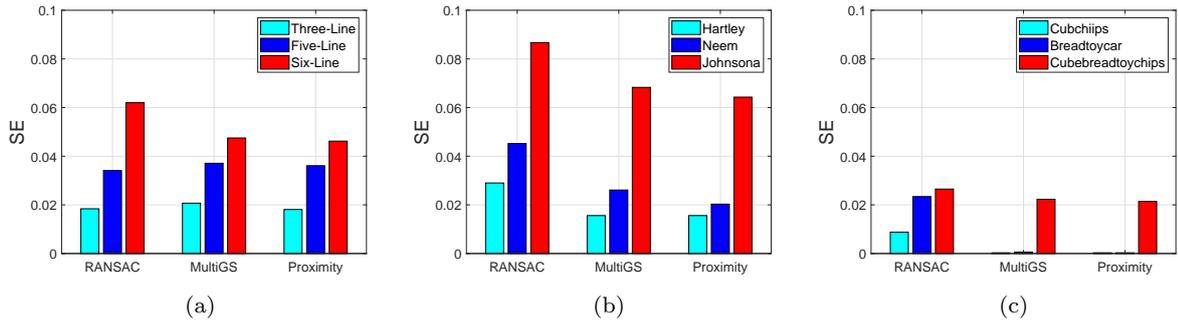


Fig. 5 The segmentation errors obtained by the proposed CLSA2 fitting method with different sampling algorithms (i.e., RANSAC, MultiGS and Proximity) for the tasks of (a) line fitting based segmentation, (b) homography based segmentation and (c) two-view based motion segmentation.

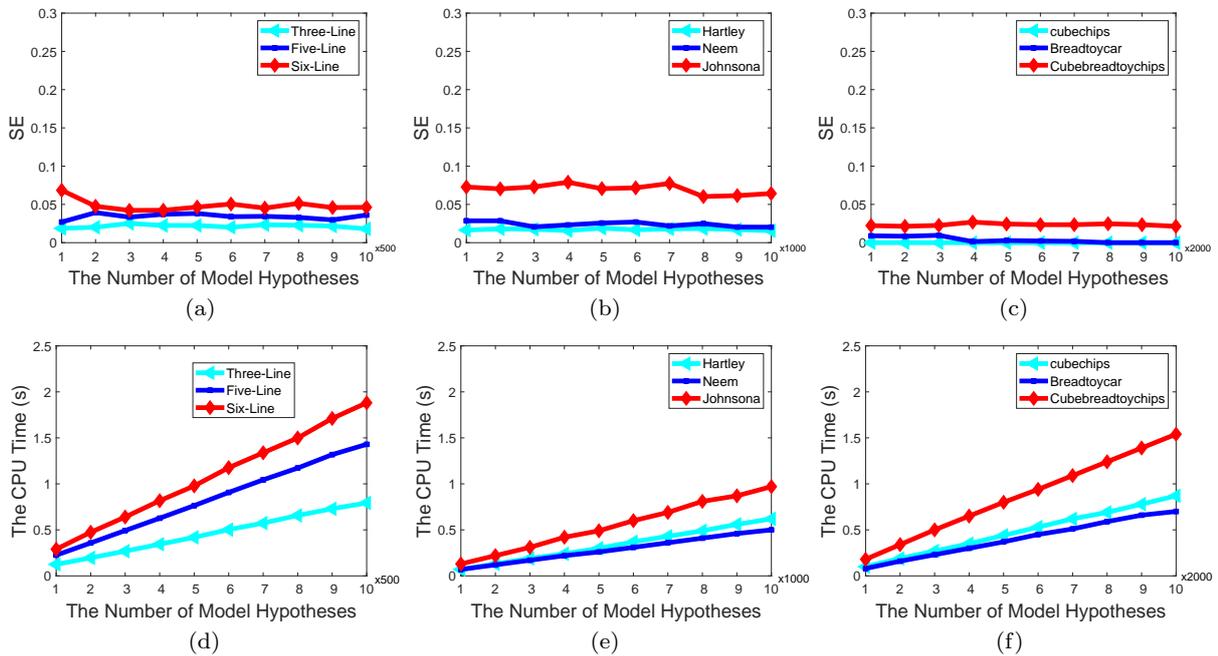


Fig. 6 The segmentation errors obtained by and the CPU time used by the proposed CLSA2 fitting method with different numbers of model hypotheses for the tasks of line fitting based segmentation ((a) and (d)), homography based segmentation ((b) and (e)) and two-view based motion segmentation ((c) and (f)).

a large value, which is obviously larger than the ground truth number of model instances for Eq. (9).

In this paper, we mainly focus on model selection. Thus, we provide the number of model instances to the proposed method as the competing methods (i.e., RansCov, RPA and T-linkage) do. That is, r_k is equal to k in Algorithm 1.

4.2 The Influence of Sampling Algorithms

We also analyze the performance of CLSA2 with three popular sampling algorithms (i.e., RANSAC [11], MultiGS [6] and Proximity [14]) for line fitting based segmentation, homography based segmentation and two-view

based motion segmentation, respectively. For each task in Sec. 4.4, Sec. 4.5 and Sec. 4.6, we test three datasets, respectively, and we also show the segmentation errors obtained by CLSA2 in Fig. 5.

We can see that, for line fitting based segmentation, three versions of CLSA2 using all three sampling algorithms are able to achieve low segmentation errors (see Fig. 5 (a)) compared with the other competing fitting methods. However, when using the random sampling algorithm (which is used in RANSAC), CLSA2 generates a high proportion of bad model hypotheses for high-order fitting tasks. For this case, CLSA2 achieves relatively higher segmentation errors than those achieved by CLSA2 using the guided sampling algorithms (see Fig. 5 (b) and (c)). Among the three sampling algo-

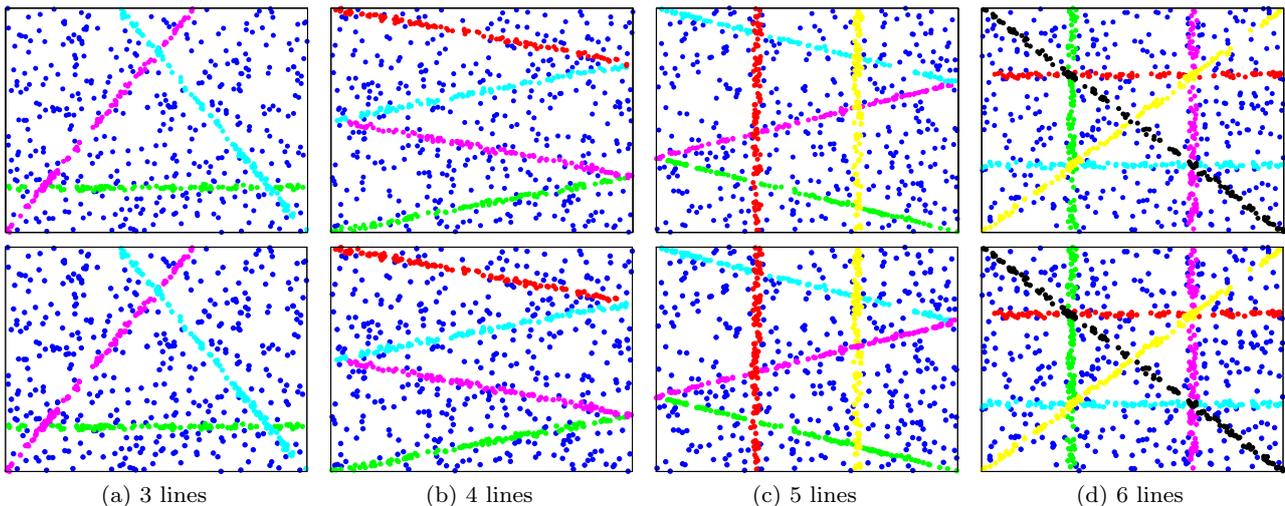


Fig. 7 The results obtained by the proposed methods (CLSA2) for line fitting based segmentation. The first row of each sub-figure shows the ground truth segmentation results and the second row shows the fitting and segmentation results obtained by CLSA2. The inlier noise scale is 1.5 and each line has 100 inliers. The outliers are labeled in blue, and the inliers of each estimated model instance are labeled in different colors (the following figures have the same settings).

gorithms, Proximity generates a similar quality of model hypotheses as MultiGS does, but it is much faster than MultiGS. Therefore, we select Proximity as the sampling algorithm for the proposed method and all the other competing methods in the following experiments.

4.3 The Influence of the Number of Model Hypotheses

In this subsection, we analyze the performance of CLSA2 with different numbers of model hypotheses for line fitting based segmentation, homography based segmentation and two-view based motion segmentation. For each task, we also test the three datasets, and we show the segmentation errors obtained by CLSA2 and the CPU time used by CLSA2 in Fig. 6.

We can see that, CLSA2 is able to achieve low segmentation errors even when we generate a small number of model hypotheses for all three fitting tasks (see Fig. 6 (a)-(c)), and it also achieves stable fitting results (across different numbers of model hypotheses), which further shows the stable properties of CLSA2. For the CPU time, CLSA2 spends more time with increasing number of model hypotheses for the three fitting tasks (see Fig. 6 (d)-(f)). This is because that the preference analysis becomes more complex when more model hypotheses are generated. Therefore, for CLSA2, we can generate a small number of model hypotheses for the three fitting tasks. However, to fairly compare with the other competing methods, we use the same setting of the number of model hypotheses, as that used in [35] and [36]. That is, we generate 5,000, 10,000 and

Table 1 Quantitative comparison results obtained by the six competing methods for line fitting based segmentation on four synthetic data. The best results of SE and CPU time are respectively labeled in magenta and cyan.

Data		MSHF	RansaCov	T-linkage	RPA	CLSA1	CLSA2
3 lines	SE	0.05±0.05	0.03± 0.00	0.03±0.01	0.05±0.01	0.03±0.04	0.02±0.01
	Time	1.96	2.88	88.94	175.31	0.65	0.80
4 lines	SE	0.06±0.05	0.03±0.01	0.10±0.08	0.03±0.01	0.08±0.14	0.02±0.00
	Time	2.12	3.46	119.12	222.11	0.83	1.12
5 lines	SE	0.03±0.01	0.03± 0.00	0.11±0.11	0.03±0.02	0.08±0.11	0.02±0.01
	Time	2.11	3.51	154.61	360.55	0.99	1.47
6 lines	SE	0.11±0.04	0.08± 0.02	0.35±0.10	0.41±0.09	0.10±0.11	0.05±0.02
	Time	2.16	4.37	193.74	430.03	1.24	1.87

20,000 model hypotheses for line fitting based segmentation, homography based segmentation and two-view based motion segmentation, respectively. Moreover, all the competing methods select model instances based on the same model hypotheses.

4.4 Line Fitting based Segmentation

We evaluate the performance of the six fitting methods for line fitting based segmentation on four challenging synthetic data. We repeat each experiment 50 times. We report the average segmentation errors (in the format of *average errors*±*standard deviation*) obtained by the six fitting methods, and the average CPU time⁵ (in seconds) used by the competing methods in Table 1. We also show the fitting results obtained by the proposed method in Fig. 7.

⁵ For the CPU time, we exclude the time used for sampling minimal subsets and generating model hypotheses for all the competing fitting methods in all the following experiments.

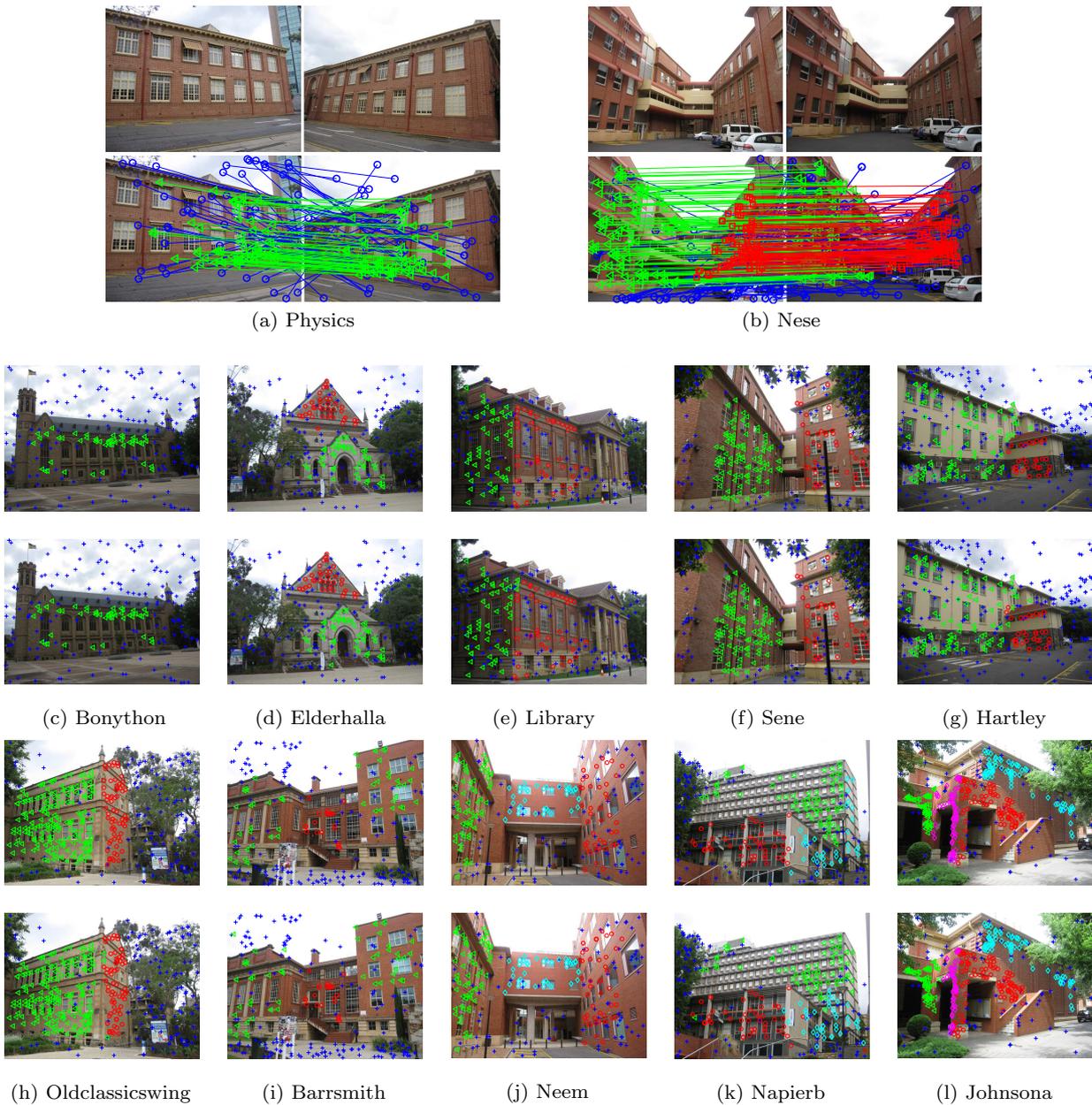


Fig. 8 The results obtained by CLSA2 for homography based segmentation on some of the AdelaideRMF dataset (two views are shown for each case in (a)-(b), and only one of the two views is shown for each case in (c)-(l)). The first row in (a)-(b) shows the original images. The first row in (c)-(l) shows the ground truth segmentation results, and the second row in (a)-(l) shows the fitting results obtained by CLSA2.

From Fig. 7 and Table 1, we can see that the proposed CLSA2 fitting method achieves significantly better results than the other five competing fitting methods, with regard to the average segmentation errors. For the CPU time, CLSA2 achieves similar performance to CLSA1 but it is much faster than the other four fitting methods: CLSA2 is about 1.10 to 229.96 times faster than the other four fitting methods. CLSA1/CLSA2 are faster than the other four competing methods since they effectively reduce the high dimension of the prefer-

ence matrix, by which they are able to greatly improve the computational efficiency. Moreover, they formulate the line fitting problem as a simpler clustering problem, which further improves the computational speed.

For the segmentation accuracy, MSHF, RansaCov and CLSA1 achieve similar low segmentation errors for all the four data. T-linkage and RPA achieve low segmentation errors for the “3 lines”, “4 lines” and “5 lines” data, but both may fail to fit the “6 lines” data among the 50 repetitions because they deal with the

Table 2 Quantitative comparison results obtained by the seven competing methods for homography based segmentation on all 19 image pairs from the AdelaideRMF dataset. “ k ” is the actual number of model instances in each data and “%Outlier” is the percentage of gross outliers. The best results of SE and CPU time are respectively labeled in magenta and cyan.

Data	k	%Outlier		MSHF	RansaCov	T-linkage	RPA	RCMSA	CLSA1	CLSA2
Unionhouse	1	18.78	<i>SE</i>	0.00±0.00	0.00±0.00	0.32±0.25	0.02±0.00	0.04±0.00	0.00±0.00	0.00±0.00
			<i>Time</i>	1.09	7.49	49.54	60.57	20.18	0.21	0.21
Bonython	1	75.13	<i>SE</i>	0.00±0.00	0.00±0.00	0.06±0.09	0.03±0.01	0.05±0.08	0.00±0.00	0.00±0.00
			<i>Time</i>	1.66	12.00	16.57	25.01	1.15	0.28	0.28
Physics	1	46.60	<i>SE</i>	0.00±0.00	0.03±0.00	0.32±0.12	0.02±0.01	0.28±0.08	0.00±0.00	0.00±0.00
			<i>Time</i>	1.65	6.85	3.36	11.19	0.89	0.16	0.16
Elderhalla	2	60.75	<i>SE</i>	0.01±0.00	0.01±0.00	0.02±0.01	0.02±0.01	0.10±0.11	0.01±0.00	0.01±0.00
			<i>Time</i>	1.27	6.97	18.06	44.73	1.15	0.41	0.46
Ladysymon	2	33.48	<i>SE</i>	0.03±0.01	0.04±0.01	0.07±0.02	0.23±0.01	0.06±0.02	0.09±0.03	0.05±0.04
			<i>Time</i>	1.93	11.41	24.69	55.04	1.29	0.14	0.17
Library	2	56.13	<i>SE</i>	0.02±0.01	0.06±0.01	0.06±0.01	0.06±0.03	0.03±0.00	0.06±0.01	0.05±0.02
			<i>Time</i>	1.25	5.47	18.37	52.65	1.13	0.41	0.48
Nese	2	30.29	<i>SE</i>	0.00±0.00	0.04±0.00	0.02±0.00	0.04±0.00	0.01±0.00	0.00±0.00	0.00±0.00
			<i>Time</i>	1.93	11.19	28.71	51.59	1.12	0.35	0.35
Sene	2	44.49	<i>SE</i>	0.01±0.00	0.02±0.00	0.00±0.00	0.00±0.01	0.01±0.00	0.00±0.00	0.00±0.00
			<i>Time</i>	1.56	9.19	27.67	55.59	1.14	0.47	0.54
Napiera	2	64.73	<i>SE</i>	0.28±0.03	0.15±0.01	0.07±0.01	0.33±0.04	0.22±0.06	0.25±0.04	0.14±0.03
			<i>Time</i>	1.19	13.98	42.32	91.66	1.73	0.53	0.62
Hartley	2	62.22	<i>SE</i>	0.02±0.00	0.03±0.00	0.03±0.01	0.12±0.00	0.06±0.11	0.02±0.00	0.02±0.00
			<i>Time</i>	1.22	7.97	46.33	94.73	1.60	0.59	0.62
Oldclassicswing	2	32.23	<i>SE</i>	0.01±0.00	0.05±0.05	0.02±0.01	0.20±0.00	0.01±0.00	0.02±0.01	0.01±0.01
			<i>Time</i>	2.30	7.78	67.18	98.73	1.55	0.67	0.67
Barrsmith	2	69.79	<i>SE</i>	0.24±0.10	0.10±0.02	0.22±0.04	0.12±0.00	0.16±0.13	0.02±0.00	0.01±0.00
			<i>Time</i>	1.44	14.43	29.16	66.02	1.72	0.56	0.65
Neem	3	37.83	<i>SE</i>	0.02±0.00	0.11±0.02	0.15±0.09	0.22±0.12	0.06±0.02	0.03±0.04	0.02±0.01
			<i>Time</i>	1.74	9.29	24.98	82.02	1.13	0.45	0.56
Elderhallb	3	49.80	<i>SE</i>	0.03±0.01	0.12±0.02	0.13±0.02	0.18±0.02	0.18±0.11	0.09±0.07	0.08±0.06
			<i>Time</i>	1.51	13.28	29.23	97.94	1.24	0.71	0.86
Napierb	3	37.13	<i>SE</i>	0.13±0.04	0.16±0.03	0.18±0.01	0.19±0.01	0.19±0.04	0.17±0.02	0.11±0.01
			<i>Time</i>	1.65	16.24	26.58	77.72	1.08	0.47	0.57
Johnsona	4	21.25	<i>SE</i>	0.03±0.01	0.06±0.01	0.06±0.02	0.16±0.00	0.04±0.01	0.15±0.09	0.06±0.00
			<i>Time</i>	2.22	16.48	65.09	152.66	1.68	0.64	0.91
Unihouse	5	18.78	<i>SE</i>	0.09±0.00	0.07±0.03	0.14±0.00	0.05±0.02	0.04±0.00	0.19±0.06	0.18±0.01
			<i>Time</i>	5.70	62.14	1,833.87	3,133.69	20.24	3.47	4.29
Bonhall	6	6.43	<i>SE</i>	0.32±0.09	0.21±0.02	0.33±0.07	0.39±0.08	0.10±0.04	0.37±0.06	0.32±0.03
			<i>Time</i>	3.55	45.96	650.09	1,111.15	6.72	1.52	1.91
Johnsonb	7	12.02	<i>SE</i>	0.17±0.05	0.13±0.03	0.19±0.06	0.25±0.10	0.10±0.01	0.40±0.15	0.26±0.06
			<i>Time</i>	2.73	26.38	206.05	539.91	3.37	1.71	1.76
<i>Mean</i>				0.074	0.080	0.131	0.139	0.091	0.099	0.070
<i>Median</i>				0.024	0.060	0.074	0.124	0.058	0.032	0.020

crossing model instances ineffectively (note that the “6 lines” data includes many crossing model instances). In contrast to CLSA2, CLSA1 is not very stable. This is because that K-means used by CLSA1 is sensitive to the initialization. Thus, CLSA1 is not able to achieve low standard deviations of error. In contrast, CLSA2 succeeds in fitting all the four data, and achieves the lowest average segmentation errors among the six competing fitting methods, for all the four data. Compared with CLSA1, CLSA2 is also very stable and achieves relatively lower standard deviations due to the effectiveness of the proposed K-means*++ (used by CLSA2).

4.5 Homography based Segmentation

We evaluate the competing methods for homography based segmentation on the 19 real image pairs from the AdelaideRMF dataset [37]⁶ (the dataset contains 19 image pairs for homography based segmentation, and 19 image pairs for two-view based motion segmentation, which will be evaluated in Sec. 4.6). We repeat each experiment 50 times, and report the average results obtained by the seven competing methods in Table 2. We also show some fitting results obtained by CLSA2 for homography based segmentation in Fig. 8.

⁶ <http://cs.adelaide.edu.au/~hwong/doku.php?id=data>

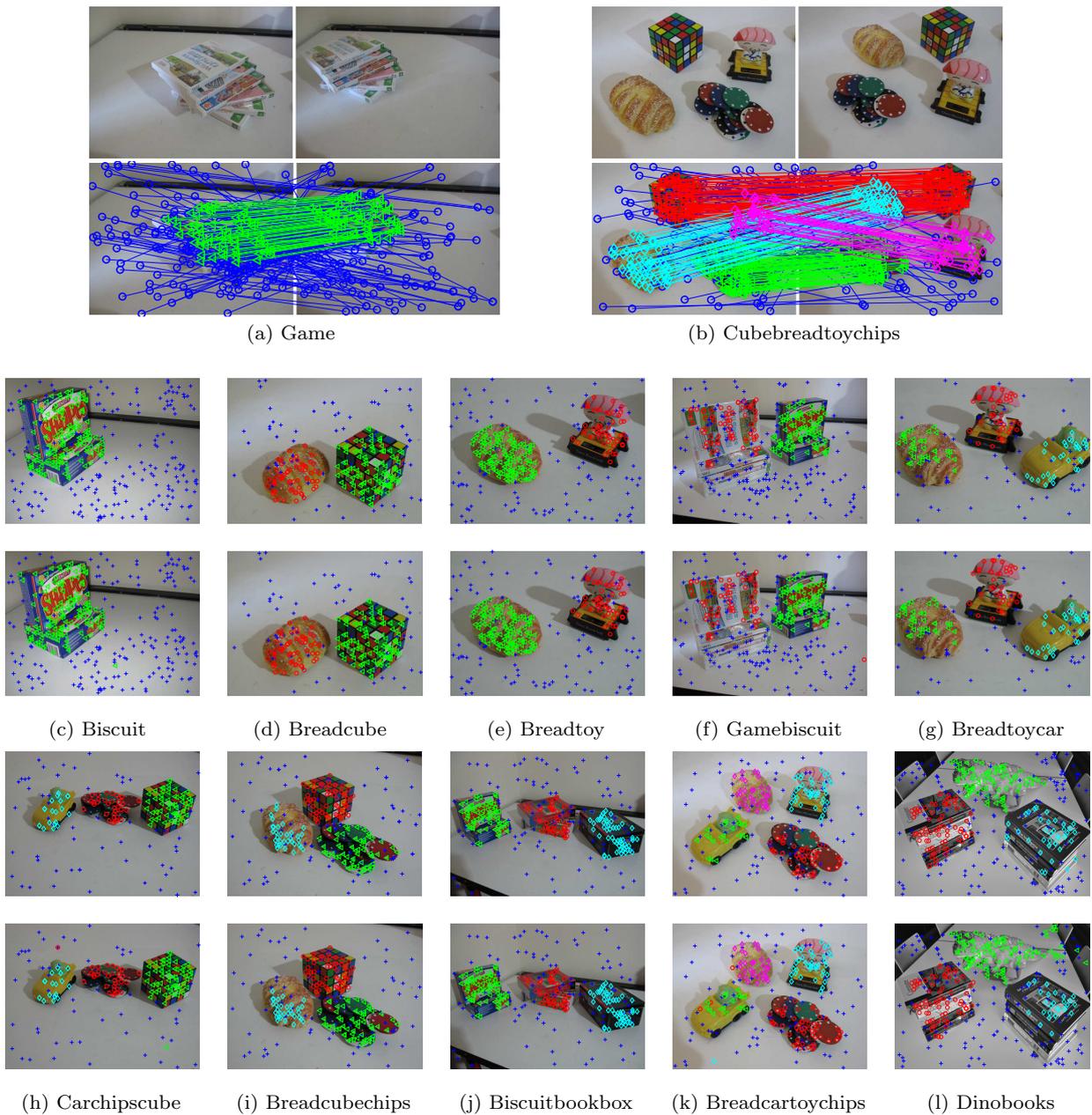


Fig. 9 Some results obtained by CLSA2 for two-view based motion segmentation on the AdelaideRMF dataset (two views are shown for each case in (a)-(b), and only one of the two views is shown for each case in (c)-(l)). The first row in (a)-(b) shows the original images. The first row in (c)-(l) shows the ground truth segmentation results, and the second row in (a)-(l) shows the fitting results obtained by CLSA2.

From Fig. 8 and Table 2, we can see that CLSA2 is able to achieve good results, obtaining the lowest average segmentation errors for 11 out of the 19 image pairs and obtaining the top three lowest segmentation errors for 7 out of the other 8 image pairs among all the seven competing fitting methods. Considering all the 19 image pairs, CLSA2 also achieves the lowest mean and median values of overall segmentation errors. It is worth pointing out that CLSA1/CLSA2 im-

plement the process of model selection within one second for 16 out of the 19 image pairs, which is much less than the time used by the other five competing methods. In contrast, MSHF only succeeds in fitting 13 out of the 19 image pairs. RansaCov also achieves low segmentation errors for 11 out of the 19 image pairs. However, RansaCov is slower than MSHF for all image pairs and it produces a higher median value of overall segmentation errors than MSHF. This is because

Table 3 Quantitative comparison results obtained by the seven competing methods for two-view based motion segmentation on 19 image pairs from the AdelaideRMF dataset.

Data	k	%Outlier		MSHF	RansaCov	T-linkage	RPA	RCMSA	CLSA1	CLSA2	
Cube	1	69.49	<i>SE</i>	0.02±0.01	0.04±0.01	0.06±0.03	0.05±0.02	0.06±0.04	0.03±0.00	0.03±0.00	
			<i>Time</i>	5.00	9.14	75.73	42.61	3.48	0.81	0.81	
Book	1	44.32	<i>SE</i>	0.01±0.00	0.06±0.01	0.13±0.08	0.05±0.02	0.02±0.02	0.00±0.00	0.00±0.00	
			<i>Time</i>	4.34	10.14	24.59	16.78	2.56	0.53	0.53	
Biscuit	1	57.68	<i>SE</i>	0.01±0.01	0.06±0.01	0.05±0.04	0.09±0.03	0.00±0.00	0.00±0.00	0.00±0.00	
			<i>Time</i>	5.08	12.72	95.21	51.14	3.93	0.88	0.88	
Game	1	73.48	<i>SE</i>	0.02±0.01	0.07±0.02	0.05±0.03	0.09±0.02	0.04±0.02	0.00±0.00	0.00±0.00	
			<i>Time</i>	4.74	4.07	40.62	24.98	3.10	0.41	0.41	
Biscuitbook	2	47.51	<i>SE</i>	0.03±0.02	0.05±0.01	0.02±0.01	0.06±0.02	0.03±0.01	0.00±0.00	0.00±0.00	
			<i>Time</i>	4.99	13.60	100.96	64.28	2.96	0.91	1.12	
Breadcube	2	32.19	<i>SE</i>	0.02±0.01	0.05±0.01	0.03±0.01	0.06±0.02	0.15±0.10	0.00±0.00	0.00±0.00	
			<i>Time</i>	4.53	10.45	49.61	33.99	2.42	0.68	0.82	
Breadtoy	2	37.41	<i>SE</i>	0.19±0.08	0.33±0.02	0.04±0.01	0.12±0.06	0.05±0.06	0.00±0.00	0.00±0.00	
			<i>Time</i>	5.01	14.86	74.39	48.91	2.60	0.76	0.86	
Cubechips	2	51.62	<i>SE</i>	0.04±0.01	0.07±0.01	0.01±0.01	0.08±0.02	0.04±0.01	0.00±0.00	0.00±0.00	
			<i>Time</i>	4.66	10.63	67.19	45.72	2.64	0.76	0.87	
Cubetoy	2	41.42	<i>SE</i>	0.02±0.01	0.08±0.01	0.03±0.02	0.04±0.03	0.06±0.04	0.00±0.00	0.00±0.00	
			<i>Time</i>	4.38	9.14	52.17	36.01	2.36	0.70	0.83	
Gamebiscuit	2	51.54	<i>SE</i>	0.02±0.01	0.08±0.04	0.03±0.01	0.08±0.01	0.05±0.03	0.01±0.00	0.01±0.00	
			<i>Time</i>	4.97	11.20	91.71	59.04	2.91	0.88	1.05	
Breadtoyocar	3	34.15	<i>SE</i>	0.08±0.02	0.23±0.01	0.08±0.01	0.14±0.02	0.14±0.08	0.14±0.15	0.00±0.00	
			<i>Time</i>	4.12	3.41	19.03	25.65	1.91	0.48	0.59	
Carchipscube	3	36.59	<i>SE</i>	0.26±0.08	0.24±0.10	0.16±0.04	0.32±0.07	0.07±0.06	0.10±0.11	0.03±0.00	
			<i>Time</i>	3.95	6.14	18.61	23.78	1.81	0.60	0.77	
Toycubecar	3	36.36	<i>SE</i>	0.14±0.07	0.35±0.09	0.10±0.02	0.14±0.03	0.13±0.02	0.11±0.02	0.08±0.00	
			<i>Time</i>	4.18	8.20	29.75	30.60	2.03	0.85	0.93	
Breadcubechips	3	35.22	<i>SE</i>	0.04±0.02	0.13±0.08	0.04±0.01	0.10±0.01	0.09±0.07	0.21±0.10	0.01±0.00	
			<i>Time</i>	4.55	6.81	42.99	38.97	2.44	0.83	1.03	
Biscuitbookbox	3	37.21	<i>SE</i>	0.03±0.01	0.23±0.02	0.03±0.02	0.04±0.02	0.03±0.01	0.11±0.13	0.01±0.00	
			<i>Time</i>	4.76	16.20	57.73	47.22	2.47	0.89	1.11	
Cubebreadtoy chips	4	28.03	<i>SE</i>	0.13±0.07	0.24±0.05	0.04±0.12	0.07±0.02	0.11±0.05	0.11±0.08	0.02±0.00	
			<i>Time</i>	4.83	9.57	93.86	83.49	3.12	1.09	1.53	
Breadcartoychips	4	35.20	<i>SE</i>	0.25±0.06	0.16±0.07	0.06±0.09	0.10±0.02	0.09±0.03	0.11±0.06	0.05±0.00	
			<i>Time</i>	4.36	5.70	43.37	47.24	2.60	0.84	1.16	
Dinobooks	3	44.54	<i>SE</i>	0.18±0.02	0.31±0.02	0.23±0.01	0.16±0.02	0.20±0.06	0.16±0.06	0.10±0.00	
			<i>Time</i>	4.95	12.46	114.17	91.99	3.43	1.43	1.87	
Boardgame	3	42.48	<i>SE</i>	0.22±0.08	0.22±0.03	0.24±0.06	0.14±0.03	0.16±0.04	0.21±0.02	0.17±0.00	
			<i>Time</i>	4.56	9.45	65.25	52.16	2.56	0.87	1.48	
				Mean	0.102	0.158	0.076	0.100	0.079	0.069	0.027
				Median	0.050	0.126	0.048	0.085	0.063	0.030	0.006

MSHF removes most of the ineffective model hypotheses before performing the process of model selection, while RansaCov considers all generated model hypotheses during the model selection procedure. Although T-linkage and RPA are preference analysis based fitting methods (like CLSA2), both are not able to achieve low segmentation errors (compared to CLSA2) for most of the image pairs. This is because that CLSA2 only keeps the principal preference information while removing the redundant information by latent semantic analysis. Moreover, CLSA2 uses an effective clustering algorithm (i.e., K-means*++) to label data points. Note that, CLSA1 achieves the same average segmentation errors as CLSA2 for 7 out of the 19 image pairs, but it produces higher average segmentation errors than CLSA2 for the other 12 image pairs, which further shows the ef-

fectiveness of the improved K-means*++ clustering algorithm. RCMSA outperforms the other six competing methods for the image pairs including a large number of model instances (i.e., the Unihouse, Bonhall and Johnsonb image pairs), but it fails in fitting the image pairs with unbalanced data (e.g., the Barrsmith, Elderhallb and Napierb image pairs).

4.6 Two-view based Motion Segmentation

For the task of two-view based motion segmentation, we use the other 19 image pairs of the AdelaideRMF dataset to evaluate the performance of the seven competing fitting methods. We show the average results in Table 3, and we also show some fitting results obtained by CLSA2 in Fig. 9.

From Fig. 9 and Table 3, we can see that RansaCov achieves high average segmentation errors and it fails to fit model instances for most cases. This is because that RansaCov considers all the generated model hypotheses, which include a large number of ineffective hypotheses. Note that the minimal subset for the two-view based motion segmentation task consists of 7 or 8 data points, and it is hard to sample an all-inlier minimal subset. Thus, it requires sampling a large number of minimal subsets to increase the probability of covering all model instances in data, but this will generate more ineffective model hypotheses as well. As the results show, the ineffective model hypotheses have significant influence on the fitting results obtained by RansaCov. Both T-linkage and RPA are able to achieve low segmentation errors for most of the image pairs, but their computational costs are relatively high, due to the use of the complicated clustering procedure. MSHF is able to achieve reasonably low average segmentation errors for 12 out of the 19 image pairs, and CLSA1 achieves reasonably low average segmentation errors for 8 out of the 19 image pairs. RCMSA is also able to achieve low average segmentation errors for most cases. Compared with the other competing methods, CLSA2 obtains the lowest segmentation errors for 16 out of the 19 image pairs and it obtains the lowest mean and median values of overall segmentation errors. In addition, CLSA2 is able to achieve very stable segmentation errors for all the 19 image pairs, when we repeat each experiment 50 times (note that each experiment uses different generated model hypotheses). This excellent performance mainly results from the steps of latent semantic space construction, outlier removal and inlier segmentation in CLSA2. Note that, CLSA1/CLSA2 are able to implement the process of model selection within 2 second for all the image pairs, which is much faster than the other five competing methods.

5 Discussion

5.1 The comparison between CLSA1 and CLSA2

In this paper, we proposed a basic approach with two identified variants. Both variants perform well - usually being the fastest and the most accurate, something uncommon in accuracy/cost tradeoffs. CLSA1 is the variant one would choose in general, if speed is a high priority, as it is usually faster and generally not so inferior in performance (still usually beating the competitors other than our slower variant). CLSA2 is one that would be recommended if speed is less of an issue and higher performance in segmentation has high priority.

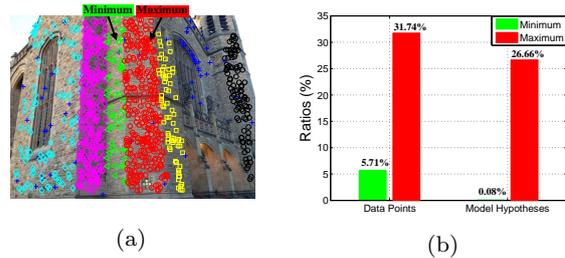


Fig. 10 An example showing a unbalanced data and the corresponding ratios of inliers and model hypotheses. (a) The minimum (in green) and maximum (in red) inlier cases of the “Bonhall” image pair. (b) The minimum and maximum ratios of inliers to all data points and the ratios of the generated model hypotheses corresponding to the two model instances in all generated model hypotheses.

Compared to the other competing methods, the proposed CLSA2 is more generalized, for handling the model fitting problem, and it only requires a few parameters (whose values do not significantly affect the accuracy of CLSA2). CLSA2 is also robust to different guided sampling algorithms and different numbers of the generated model hypotheses.

From the experimental results of line fitting based segmentation, we can see that CLSA2 is able to effectively deal with the data with different numbers of model instances, and a large number of outliers. From the experimental results of homography based segmentation and two-view based motion segmentation, CLSA2 also shows good performance compared to the other competing methods. Note that, CLSA2 shows significant advantages over the other competing methods for two-view based motion segmentation over homography based segmentation. One of the reasons is that, the task for two-view based motion segmentation is more challenging since it requires dealing with more ineffective model hypotheses. Moreover, we cannot ignore the fact that some fitting methods (e.g. MSHF) directly deal with the fitting task in the parameter space, and those methods are able to more effectively handle the data points located at the intersection of two structures. However, the fitting performance of those methods largely depends on the quality of the generated model hypotheses. Thus, the proposed CLSA2 achieves much better performance than those methods for two-view based motion segmentation.

Although the proposed CLSA2 significantly improves the performance of several state-of-the-art fitting methods on dealing with unbalanced data (that is, the numbers of inliers belonging to different model instances are unbalanced), it may not be effective to deal with heavily unbalanced data. We show an example of a unbalanced data in Fig. 10. We can see that, the minimum and maximum ratios of inliers belonging different model in-

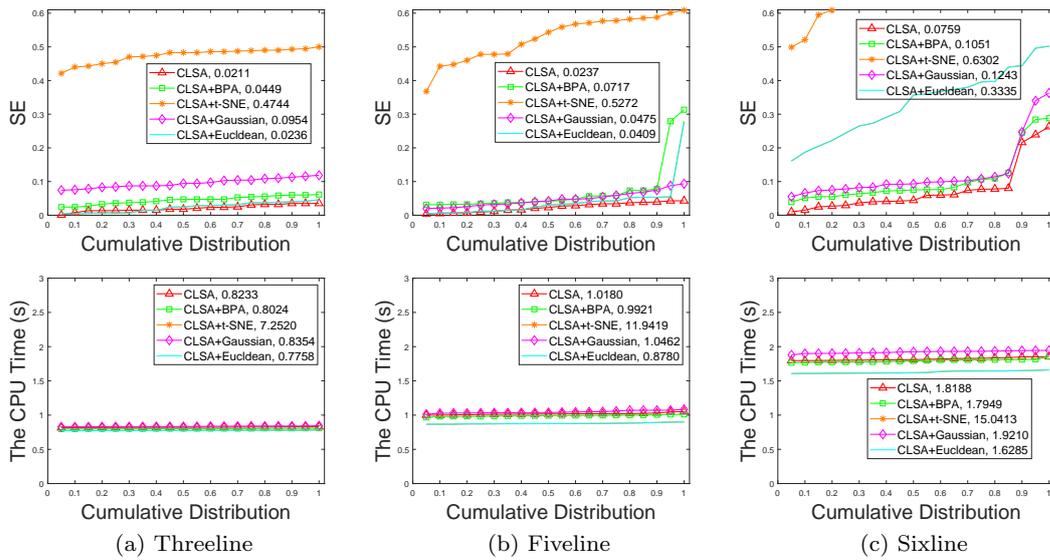


Fig. 11 Quantitative comparisons of different versions of the proposed method for line fitting based segmentation. In each subfigure, we show the segmentation errors obtained by and the CPU time used by the proposed methods with respect to the cumulative distribution. A point on the curve with coordinate (x, y) represents that there are $100 \times x$ percents of image pairs which have values no more than y .

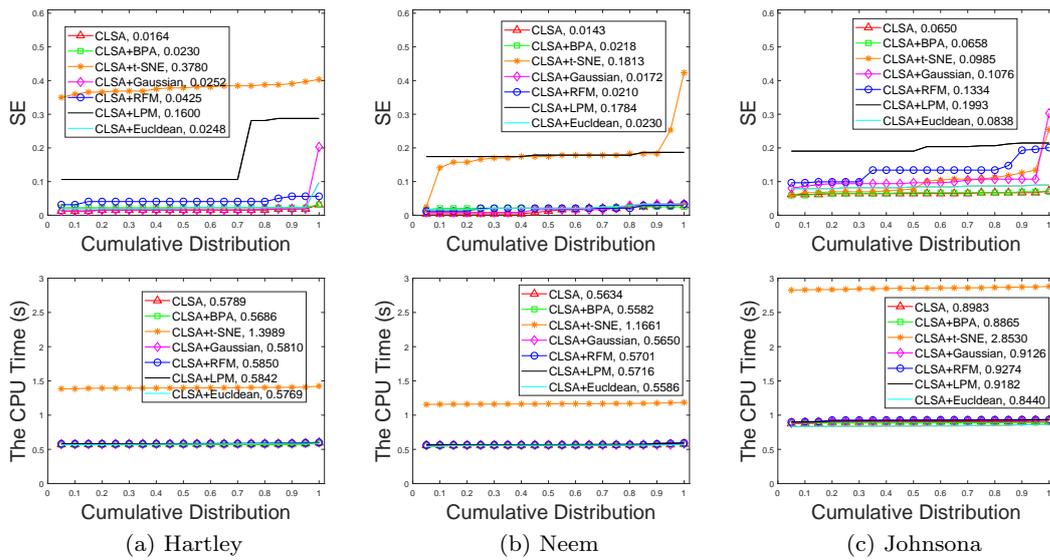


Fig. 12 Quantitative comparisons of different versions of the proposed method for homography based segmentation. In each subfigure, we show the segmentation errors obtained by and the CPU time used by the proposed methods with respect to the cumulative distribution.

stances to all data points for the “Bonhall” image pair are significantly different, and the ratios of the generated model hypotheses corresponding to the two model instances to all model hypotheses are much more different. In this case, the preference function (used in CLSA2) is not effective for the inliers of the model instance with the minimum inlier ratio. As a result, the proposed CLSA2 can not fit this data very well. However, all of the other competing fitting methods also have this limitation. Note that, sampling non-minimal sub-

sets may help alleviate the unbalance of the generated model hypotheses, as discussed in [34]. However, it is still a very challenging problem.

5.2 The Influence of different components of the proposed method

It is worth pointing that, in this paper, we argue that the proposed CLSA fitting method with the combina-

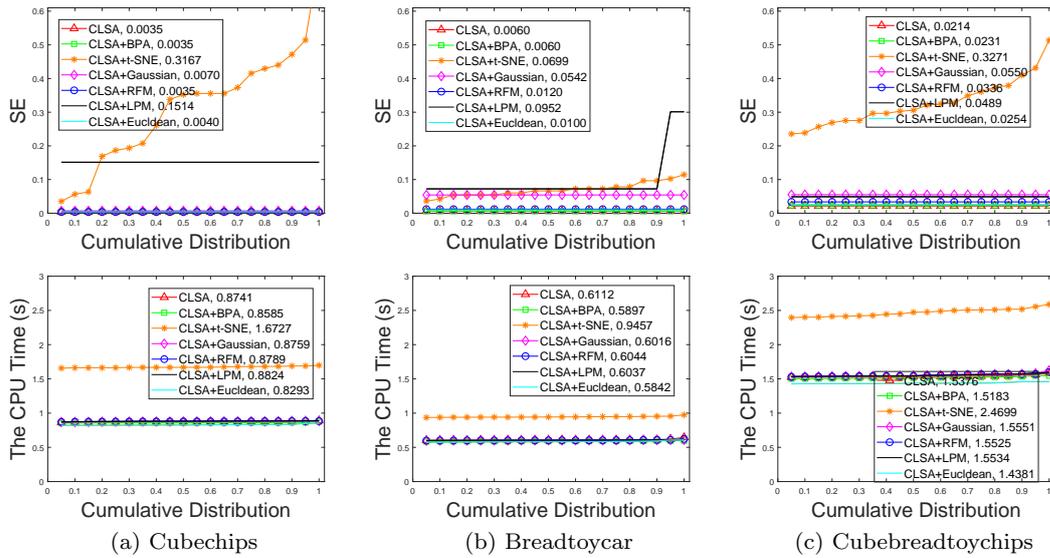


Fig. 13 Quantitative comparisons of different versions of the proposed method for two-view based motion segmentation. In each subfigure, we show the segmentation errors obtained by and the CPU time used by the proposed methods with respect to the cumulative distribution.

tion of latent semantic analysis (LSA) and continuous preference analysis (CPA) is an important improvement over that using either LSA or CPA. To verify this, we evaluate three different versions of CLSA, i.e., CLSA with LSA and CPA, CLSA+BPA with LSA and binary preference analysis (BPA), and CLSA+t-SNE with CPA and t-SNE [19], for line fitting based segmentation, homography based segmentation and two-view based motion segmentation. We also test different outlier removal strategies (i.e., 1D Mixture Gaussian Model [5], RFM [13] and LPM [18]), for the proposed method with CLSA+Gaussian, CLSA+RFM and CLSA+LPM, respectively (here, RFM and LPM only work for the two-view model fitting problem, thus, we test CLSA+RFM and CLSA+LPM only for homography based segmentation and two-view based motion segmentation). In addition, to compare the Tanimoto distance, we test a version (i.e. CLSA+Euclidean) with the Euclidean distance. To test different qualities of preference matrices, we generate 20 groups of model hypotheses for all versions of CLSA. We show the quantitative comparisons of different versions of CLSA in Figs. 11, 12 and 13 for line fitting based segmentation, homography based segmentation and two-view based motion segmentation, respectively.

From Figs. 11, 12 and 13, we can see that, CLSA is able to achieve better performance on the segmentation error than CLSA+BPA for most of the datasets, especially for the task of line fitting based segmentation (see Fig. 11), where the data include a large number of outliers (and they use the similar CPU time for all datasets). This shows the effectiveness of CPA for mod-

el fitting. For CLSA and CLSA+t-SNE, which deal with the preference matrix using LSA and t-SNE, respectively, CLSA is able to achieve lower segmentation errors with much less CPU time than CLSA+t-SNE, for all the three tasks (see Figs. 11, 12 and 13). This shows the effectiveness of LSA for model fitting. For different outlier removal strategies, CLSA achieves lower segmentation errors than CLSA+Gaussian, CLSA+RFM and CLSA+LPM for all the three tasks, while they spend the similar CPU time. This further shows the effectiveness of our proposed outlier removal strategy for model fitting. For different distance measures, CLSA is able to show the superiority over CLSA+Euclidean on the performance of the segmentation error for all the three tasks, while they also spend the similar CPU time. Thus, this shows the effectiveness of the Tanimoto distance for model fitting. For different qualities of preference matrices, CLSA is able to achieve stable fitting results for all the three tasks.

5.3 The estimation of the number of model instances

The estimation of the number of model instances in data is also an important and challenging task for model fitting. We note that if two model hypotheses correspond to the same model instance, they will share most of their inliers. Thus, we provide a promising solution for adaptively estimating the number of model instances in data by analyzing the inlier distribution and the quality of the estimated model instances. Specifically, we set the value of k in Algorithm 2 to be

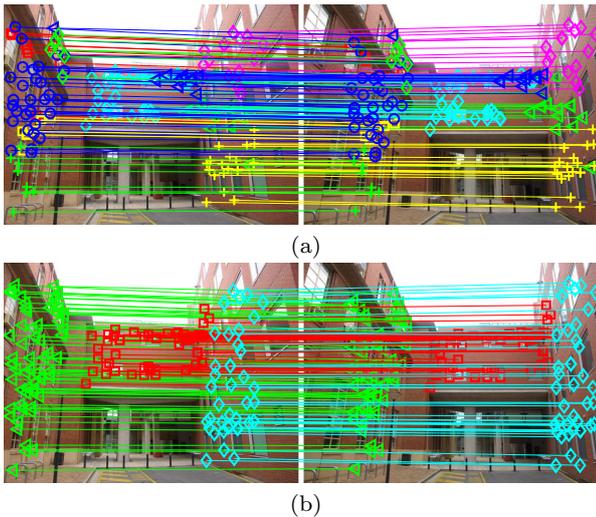


Fig. 14 An example of estimating the number of model instances on an image pair. (a) The over-segmented result when we input $k = 10$. (b) The segmentation result after estimating the number of model instances by using the proposed solution.

a reasonable large number so that the maximum number of model instances in data is less than k . In this way, it is much easier to set the k value than to require an user to input the truth number of model instances in data. After obtaining k model hypothesis candidates by Algorithm 2 (note that these model hypothesis candidates are significant model hypotheses since most of outliers have been removed in Step 4 of Algorithm 2, but the result may be over-segmented as k is usually larger than the true number of model instances in data), we compute the ratio of the common inliers shared by any two model hypothesis candidates, and group the two model hypothesis candidates into one cluster if the ratio of the common inliers shared by the two model hypothesis candidates is high (which means they are more likely to correspond to the same model instance). This process will fuse the model hypothesis candidates corresponding to the same model instance. Then, we estimate the number of model instances by counting the number of the model hypothesis groups. In Fig. 14, we show an example for the process of estimating the number of model instances. We can see that, the result in Fig. 14(a) is over-segmented when we input $k = 10$ (that is, we assume that the maximum number of model instances in the data is less than 10, which can be satisfied in most common cases). In contrast, we obtain the good segmentation result in Fig. 14(b), where we keep the model hypothesis candidate that has the best quality (measured by weighting scores [35]) in each cluster as the parameter estimation of the corresponding model instance, after estimating the number of model instances by the proposed solution.

6 Conclusion

In this paper, we propose a continuous latent semantic analysis based fitting method (called CLSA), which exploits the advantages of both latent semantic analysis (LSA) and continues preference analysis (CPA). LSA is able to significantly increase the efficiency and reduce the computational cost of the proposed CLSA method. CPA can be used by CLSA to effectively represent the relationship between model hypotheses and data points. By exploiting the advantages of both LSA and CPA, we construct a novel latent semantic space (LSS), where we map gross outliers close to the origin of LSS, while mapping inliers of different model instances to several independent directions in LSS. After that, we employ an information theoretic approach to remove the points corresponding to gross outliers in LSS. Then we treat the complex model fitting problem as a simple clustering problem (which is much easier to be dealt with because the data points corresponding to gross outliers are removed). We also propose an effective but simple clustering algorithm for inlier segmentation. Extensive experimental results show that the proposed CLSA method is able to achieve better performance than several state-of-the-art fitting methods on both segmentation accuracy and computational efficiency for different model fitting tasks.

References

1. Amayo, P., Pinis, P., Paz, L.M., Newman, P.: Geometric multi-model fitting with a convex relaxation algorithm. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8138–8146 (2018)
2. Arthur, D., Vassilvitskii, S.: K-means++: The advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms, pp. 1027–1035 (2007)
3. Barath, D., Matas, J., Noskova, J.: Magsac: Marginalizing sample consensus. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 10,197–10,205 (2019)
4. Barath, D., Noskova, J., Ivashechkin, M., Matas, J.: Magsac++, a fast, reliable and accurate robust estimator. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1304–1312 (2020)
5. Chin, T.J., Wang, H., Suter, D.: Robust fitting of multiple structures: The statistical learning approach. In: Proceedings of the IEEE international conference on computer vision, pp. 413–420 (2009)
6. Chin, T.J., Yu, J., Suter, D.: Accelerated hypothesis generation for multistructure data via preference analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(4), 625–638 (2012)
7. Chum, O., Matas, J.: Matching with prosac-progressive sample consensus. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 220–226 (2005)

8. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. *Journal of the American Society for Information Science* **41**(6), 391–407 (1990)
9. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. *IEEE Transactions on Pattern Analysis and Machine Intelligence* pp. 226–231 (1996)
10. Ferraz, L., Felip, R., Martínez, B., Binefa, X.: A density-based data reduction algorithm for robust estimators. In: *Proceedings of Iberian conference on pattern recognition and image analysis*, pp. 355–362 (2007)
11. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* **24**(6), 381–395 (1981)
12. Isack, H., Boykov, Y.: Energy-based geometric multi-model fitting. *International Journal of Computer Vision* **97**(2), 123–147 (2012)
13. Jiang, X., Ma, J., Jiang, J., Guo, X.: Robust feature matching using spatial clustering with heavy outliers. *IEEE Transactions on Image Processing* **29**, 736–746 (2020)
14. Kanazawa, Y., Kawakami, H.: Detection of planar regions with uncalibrated stereo using distributions of feature points. In: *Proceedings of British machine vision conference*, pp. 2701–2710 (2004)
15. Le, H.M., Chin, T.J., Eriksson, A., Do, T.T., Suter, D.: Deterministic approximate methods for maximum consensus robust fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **pp**(1), 1–14 (2019)
16. Lee, J.K., Yoon, K.J.: Joint estimation of camera orientation and vanishing points from an image sequence in a non-manhattan world. *International Journal of Computer Vision* **127**(10), 1426–1442 (2019)
17. Ma, J., Jiang, X., Fan, A., Jiang, J., Yan, J.: Image matching from handcrafted to deep features: A survey. *International Journal of Computer Vision* **129**(1), 23–79 (2021)
18. Ma, J., Zhao, J., Jiang, J., Zhou, H., Guo, X.: Locality preserving matching. *International Journal of Computer Vision* **127**(5), 512–531 (2019)
19. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. *Journal of Machine Learning Research* **9**(11), 2579–2605 (2008)
20. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, pp. 281–296 (1967)
21. Magri, L., Fusiello, A.: T-linkage: A continuous relaxation of j-linkage for multi-model fitting. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3954–3961 (2014)
22. Magri, L., Fusiello, A.: Multiple model fitting as a set coverage problem. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3318–3326 (2016)
23. Magri, L., Fusiello, A.: Multiple structure recovery via robust preference analysis. *Image and Vision Computing* **67**, 1–15 (2017)
24. Magri, L., Fusiello, A.: Fitting multiple heterogeneous models by multi-class cascaded t-linkage. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7460–7468 (2019)
25. Meer, P.: Robust estimation techniques. *Computer Vision: A Reference Guide* pp. 1–8 (2020)
26. Mittal, S., Anand, S., Meer, P.: Generalized projection-based m-estimator. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(12), 2351–2364 (2012)
27. Pham, T.T., Chin, T.J., Yu, J., Suter, D.: The random cluster model for robust geometric fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36**(8), 1658–1671 (2014)
28. Purkait, P., Chin, T.J., Ackermann, H., Suter, D.: Clustering with hypergraphs: The case for large hyperedges. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**(9), 1697–1711 (2017)
29. Rodriguez, A., Laio, A.: Clustering by fast search and find of density peaks. *Science* **344**(6191), 1492–1496 (2014)
30. Tanimoto, T.: An elementary mathematical theory of classification and prediction. Internal technical report, IBM (1957)
31. Tennakoon, R., Bab-Hadiashar, A., Cao, Z., Hoseinezhad, R., Suter, D.: Robust model fitting using higher than minimal subset sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **38**(2), 350–362 (2016)
32. Toldo, R., Fusiello, A.: Robust multiple structures estimation with j-linkage. In: *Proceedings of European conference on computer vision*, pp. 537–547 (2008)
33. Torr, P.H., Murray, D.W.: The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision* **24**(3), 271–300 (1997)
34. Tran, Q.H., Chin, T.J., Chojnacki, W., Suter, D.: Sampling minimal subsets with large spans for robust estimation. *International Journal of Computer Vision* **106**(1), 93–112 (2014)
35. Wang, H., Chin, T.J., Suter, D.: Simultaneously fitting and segmenting multiple-structure data with outliers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(6), 1177–1192 (2012)
36. Wang, H., Xiao, G., Yan, Y., Suter, D.: Searching for representative modes on hypergraphs for robust geometric model fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **41**(3), 697–711 (2019)
37. Wong, H.S., Chin, T.J., Yu, J., Suter, D.: Dynamic and hierarchical multi-structure geometric model fitting. In: *Proceedings of the IEEE international conference on computer vision*, pp. 1044–1051 (2011)
38. Wu, Y., Li, Y., Hu, Z.: Detecting and handling unreliable points for camera parameter estimation. *International Journal of Computer Vision* **79**(2), 209–223 (2008)
39. Xiao, G., Wang, H., Lai, T., Suter, D.: Hypergraph modelling for geometric model fitting. *Pattern Recognition* **60**(1), 748–760 (2016)
40. Xiao, G., Wang, H., Yan, Y., Suter, D.: Superpixel-guided two-view deterministic geometric model fitting. *International Journal of Computer Vision* **127**(4), 323–329 (2019)
41. Xu, L., Oja, E., Kultanen, P.: A new curve detection method: randomized hough transform (rht). *Pattern Recognition Letter* **11**(5), 331–338 (1990)
42. Xu, X., Cheong, L.F., Li, Z.: 3d rigid motion segmentation with mixed and unknown number of models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **pp**(1), 1–14 (2019)
43. Yang, X., Meer, P.M.J.: A new approach to robust estimation of parametric structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **pp**(1), 1–14 (2020)

Segmentation by Continuous Latent Semantic Analysis for Multi-structure Model Fitting

Guobao Xiao · Hanzi Wang · Jiayi Ma · David Suter

Received: date / Accepted: date

Abstract In this paper, we propose a novel continuous latent semantic analysis fitting method, to efficiently and effectively estimate the parameters of model instances in data, based on [latent semantic analysis and continuous preference analysis](#). Specifically, we construct a new latent semantic space (LSS): where inliers of different model instances are mapped into several independent directions, while gross outliers are distributed close to the origin of LSS. After that, we analyze the data distribution to effectively remove gross outliers in LSS, and propose an improved clustering algorithm to segment the remaining data points. On the one hand, the proposed fitting method is able to achieve excellent fitting results; due to the effective continuous preference analysis in LSS. On the other hand, the proposed method can efficiently obtain final fitting results due to the dimensionality reduction in LSS. Experimental results on both synthetic data and real images demonstrate that the proposed method achieves significant superiority over several state-of-the-art model fitting

methods on both fitting accuracy and computational speed.

Keywords Latent semantic analysis · preference analysis · geometric model fitting · multi-structure data.

1 Introduction

Geometric model fitting is one of the most challenging tasks in computer vision, and it has been applied to several types of computer vision tasks, such as, homography/fundamental matrix estimation [40], vanishing point detection [16], 3D reconstruction [43], and motion segmentation [42]. One of the main challenges in model fitting is to robustly estimate the model parameters for data involving a large number of outliers and multiple structures.

1.1 Background

1.1.1 Several Primary Definitions

We firstly [provide](#) some definitions in model fitting to address the model fitting problem.

A *geometric model* is usually specified before model fitting. Several popular geometric models have been used, such as, line, circle, homography/fundamental matrix, etc.. Once a type of geometric model is specified, one can explain data with the model.

A *model instance* (also called a structure) is an instance of the specified type of model [in the data](#). Multi-structural data contain multiple model instances.

A *minimal subset* is the minimum number of data points that are required [generating](#) a model hypothesis,

✉ Hanzi Wang
E-mail: hanzi.wang@xmu.edu.cn

Guobao Xiao
Fujian Key Laboratory of Sensing and Computing for Smart City, School of Informatics, Xiamen University, China; Fujian Provincial Key Laboratory of Information Processing and Intelligent Control, College of Computer and Control Engineering, Minjiang University, China

Hanzi Wang
Fujian Key Laboratory of Sensing and Computing for Smart City, School of Informatics, Xiamen University, China

Jiayi Ma
Electronic Information School, Wuhan University, China

David Suter
School of Science, Edith Cowan University, Australia

e.g., 2 data points for line fitting, 4 data points for homography fitting.

A *model hypothesis* is generated based on each minimal subset. The subset is sampled using different types of sampling strategies. For data with outliers, many model hypotheses are usually required being generated to estimate the model instances in data.

Based on the above definitions, the model fitting problem can be described as: Given a set of input data points and a specified type of model, the goal is to estimate the parameters of model instances that best explain the input data [38, 15, 25, 17].

Generally, there are two steps included in the model fitting procedure: 1) Sample a number of minimal subsets to generate a set of model hypotheses; 2) Estimate the parameters of model instances according to the generated model hypotheses by model selection.

Some methods have been proposed to improve either the quality of sampling (e.g., Proximity [14], PROSAC [7], MultiGS [6] and MAGSAC [3, 4]) or the performance of model selection (e.g., HMSS [31] and RansaCov [22]).

In this paper, we mainly focus on both the effectiveness and computational efficiency of model selection.

1.1.2 Prior Work

Existing fitting methods can be roughly classified into two categories, i.e., the consensus analysis based and the preference analysis based fitting methods. The consensus analysis based fitting methods (e.g., RANSAC [11], RansaCov [22], RHT [41], GPbM [26], AKSWH [35] and MSHF [36]) directly select model instances from a number of generated model hypotheses according to some evaluation criterions (such as, the number of inliers or the weighting scores of model hypotheses), and then segment data points according to the user-specified or estimated inlier noise scale.

Among these methods, RANSAC is one of the most popular fitting methods due to its simplicity and robustness to outliers. The main steps of RANSAC are described as follows:

- Randomly sample a number of minimal subsets of data points and then generate the corresponding model hypotheses.
- Select the model hypothesis that includes the maximum number of inliers as the estimated model instance.

RANSAC is simple yet robust. However, it suffers from several limitations. For example, it requires randomly sampling a large number of minimal subsets for model hypothesis generation to “hit”¹ a model instance in

¹ That is, among the generated model hypotheses, there is at least one model hypothesis (generated by a minimal subset) corresponding to a true model instance in data.

data with high proportions of outliers. However, most of the sampled minimal subsets are not significant (i.e., most of the sampled minimal subsets include outliers). This will cause high computational complexity, especially for high dimensional models. Moreover, RANSAC sequentially selects the best model hypothesis as the estimated model instance for multi-structural data via a “fit-and-remove” framework, which is not very effective and computationally efficient [35].

To improve the fitting performance, RansaCov formulates the model fitting problem as a problem of set coverage and it searches for the best sets that include the maximum number of inliers. RHT [41] assigns each generated model hypothesis a score, and then selects the model hypothesis with the maximum score as the estimated model instance. GPbM [26] formulates the model fitting problem as an optimization problem. AKSWH [35] directly clusters the generated model hypotheses, and then selects the model hypothesis with the maximum weighting score from each cluster as an estimated model instance. MSHF [36] combines hypergraphs and mode-seeking to directly select model hypotheses as the estimated model instances according to their weight scores and a similarity matrix. Although these fitting methods are not very sensitive to data distribution (i.e., their performance is relatively less affected by unbalanced data distribution), their fitting performance largely depends on the quality of the generated model hypotheses.

The preference analysis based fitting methods (e.g., SWS [28], HF [39], RPA [23], J-linkage [32], T-linkage [21] and MCT [24]) estimate model instances by labeling data points based on the generated model hypotheses. Specifically, SWS [28] adopts large hyperedges to represent the relationship between model hypotheses and data points for model fitting. HF [39] introduces a novel hypergraph model to formulate the model fitting problem as a hypergraph partition problem. RPA [23] combines principal component analysis and non negative matrix factorization to deal with multi-structure model fitting. J-linkage [32]/T-linkage [21] directly cluster data points according to their preference sets. MCT [24] extends T-linkage to handle different nested classes of models.

Moreover, there are also some energy-based model fitting methods, e.g., PEaRL [12], RCMSA [27] and CORAL [1]. We can see that, most of the currently existing fitting methods estimate the fitting results from the residual matrix, which is derived from model hypotheses and data points. However, to cover all model instances in data, they often require sampling a large number of model hypotheses. Then, the fitting methods may suffer from high computational complexity when

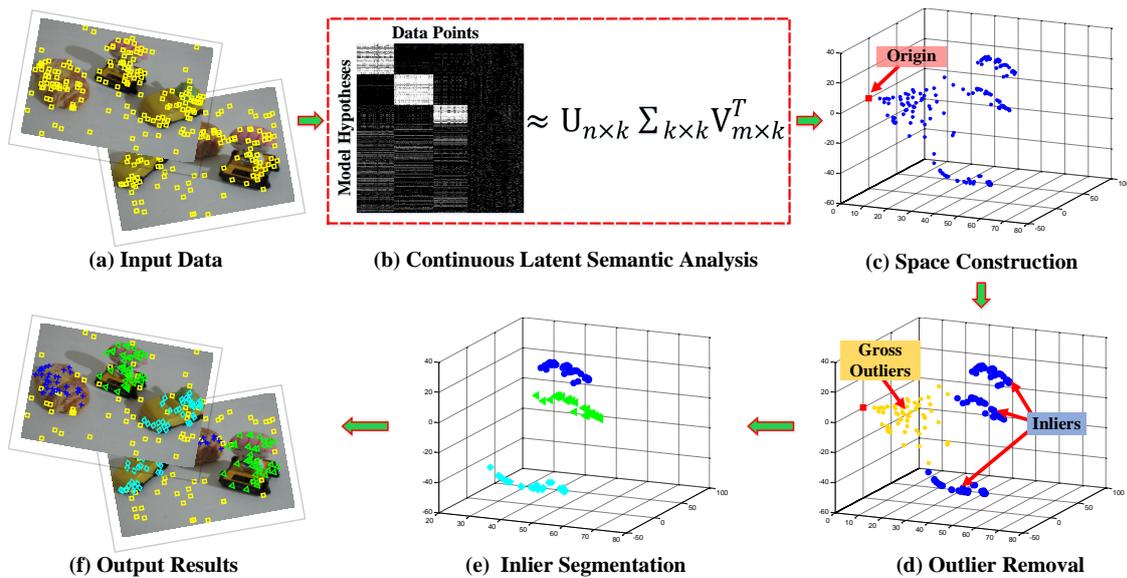


Fig. 1 Overview of the proposed method for two-view based motion segmentation.

they compute the similarities between all data points, especially for large-size data.

1.2 Motivations and Contributions

In this paper, we propose a continuous latent semantic analysis method (called as CLSA) for robust multi-structure fitting. We aim to exploit the advantage of [preference analysis](#) (that is able to achieve good fitting accuracy) and yet reduce the high computational complexity of multi-structural model fitting by applying latent semantic analysis (LSA) [8] on data points ([involving a large number of outliers](#)). Note that LSA is a valuable analysis tool, which [was](#) originally used to construct a low-dimensional subspace by analyzing the relationships between a set of documents and terms for word processing, to effectively reduce the dimensions of word-document count vectors. Inspired by that work, we construct a new low-dimensional latent semantic space (LSS), to analyze the relationships among data points, by both LSA and continuous preference analysis (CPA)² implemented by T-linkage [21], for solving the model fitting problem.

It is worth pointing out that the combination of LSA and CPA is an important improvement over either LSA or CPA. That is, for LSA, CPA can capture more accurate relationships between model hypotheses and data points (note that LSA has no notion of these). For CPA, LSA can be used to reduce the high dimension of a preference matrix generated by CPA while keep-

ing its principal preference information. Thus, both LSA and CPA jointly lead to better fitting performance and higher computational efficiency. To the best of our knowledge, we are the first to exploit the relationships between data points and model hypotheses via both LSA and CPA, for robust multi-structure model fitting.

Moreover, we analyze the constructed LSS to remove gross outliers, and cluster the remaining data points into independent directions in LSS. Note that there usually are two kinds of outliers, i.e., gross outliers and pseudo outliers in data points. Gross outliers are the data points that do not belong to the inliers of any model instance in data; while pseudo outliers are the data points that belong to the inliers of one model instance, but they are the outliers to the other model instances in data. In our work, we aim to effectively remove gross outliers and cluster/segment inliers (belonging to different model instances) in LSS. Firstly, we adaptively set a threshold based on an information theoretic approach [10] to remove gross outliers in LSS. Then, by analyzing the distribution of the remaining data points in LSS, we cluster these points into different directions via an improved clustering algorithm. Note that, in LSS, the high-order model fitting problem (e.g., homography estimation and two-view based motion segmentation) is formulated as a simple clustering problem, which is much easier to deal with.

We show the overview of the proposed CLSA method applied to two-view based motion segmentation in Fig. 1. In Fig. 1(b), a preference matrix, whose entries are the preference values of data points to model hypotheses, and the key step of continuous latent semantic analysis are shown. From Fig. 1(c), we can see that, each

² CPA is a soft generalization of the binary preference analysis.

of the input feature matches is mapped to a point in LSS by the proposed method. In LSS, the points corresponding to the inliers and gross outliers are distributed differently. That is, the points corresponding to the gross outliers are distributed close to the origin of LSS, while the points corresponding to the inliers of different model instances are distributed in their respective directions and they are far from the origin of LSS (see Fig. 1(d)). After removing the gross outliers from the input data, the complex model fitting problem becomes a much easier clustering problem, which can be solved in an effective and efficient manner (see Fig. 1(e)).

In this paper we are primarily concerned with proposing a model fitting method to segment data. In model fitting, one tries to jointly determine both the parameters of the underlying models, and the segmentation induced by the assignment of data to the models they fit. Since our main aim is to segment the data, the accuracy of the estimated model instances is only a means to an end. In contrast, for model extraction/estimation, the accuracy of concern is that of the model parameter estimates, not the induced segmentation per se. Despite the focus of this work on segmentation, the ideas could also be used in areas where the model parameters are more of interest but that is beyond the scope of this paper.

The key contributions of this work are summarized as follows:

- We propose a way to analyze the complex relationships between data points and model hypotheses by exploiting the advantage of both latent semantic analysis and continuous preference analysis, to improve the performance on both segmentation accuracy and computational efficiency for robust model fitting.
- We propose an adaptive gross outlier removal strategy in the new latent semantic space, where inliers and gross outliers are separately distributed. Then we formulate the complex model fitting problem as a simple clustering problem on the remaining data without gross outliers.
- We improve a traditional clustering method [20] by introducing a preference function to effectively seed the initial centers, which achieves more stable clustering results for the task of model fitting.

Our experimental evaluations and comparisons demonstrate the significant superiority of the proposed method over several state-of-the-art geometric model fitting methods for different model fitting tasks.

The rest of the paper is organized as follows: We describe the components of the proposed fitting method

in Sec. 2. We summarize the complete method in Sec. 3. We present the experimental results on both synthetic and real data in Sec. 4. We discuss the proposed method in Sec. 5 and draw conclusions in Sec. 6.

2 The Methodology

In this section, we describe the details of the proposed continuous latent semantic analysis fitting method (called CLSA). We first describe the problem formulation in Sec. 2.1, and then review the background of latent semantic analysis (LSA) [8] (Sec. 2.2). We then construct a novel latent semantic space by LSA and the continuous preference analysis (CPA) in Sec. 2.3. After that, we analyze the data distribution in the constructed latent semantic space, and then propose an adaptive gross outlier removal strategy (Sec. 2.4). Then we propose an improved clustering algorithm to segment the remaining data points (Sec. 2.5).

2.1 Problem Formulation

Given a set of data points $X = \{x_1, \dots, x_n\}$ and a set of model hypotheses $\Theta = \{\theta_1, \dots, \theta_n\}$ generated by sampling, we define the *preference function* [21], which represents the preference of a data point x_i to a model hypothesis θ_j , as follows:

$$f(x_i, \theta_j) = \exp\left\{-\frac{d(x_i, \theta_j)}{\psi}\right\}, \quad (1)$$

where $d(x_i, \theta_j)$ denotes the residual value measured with the Sampson distance [33] from a data point x_i to a model hypothesis θ_j . ψ is a threshold (and the influence of its value on the proposed CLSA will be discussed with more details in Sec. 4.1).

By using Eq. (1), we compute the preference matrix, which is used to estimate the parameters of model instances in the data. Then, we also segment input data points belonging to different model instances into different groups. The cost function is defined as [22,26]:

$$SE = \frac{\# \text{ mislabeled data points}}{\# \text{ data points}}, \quad (2)$$

where $\# \text{ mislabeled data points}$ is the number of mislabeled data points, and $\# \text{ data points}$ is the number of input data points. Note that, if the estimated model instance is close to the true model instance, the value of SE will be small; Otherwise, the value of SE will be large.

2.2 Latent Semantic Analysis (LSA)

Latent semantic analysis (LSA) was proposed to analyze a binary matrix of documents/terms, via singular value decomposition (SVD), for document representation.

Specifically, given documents $Do = \{do_1, \dots, do_n\}$ with terms $Te = \{te_1, \dots, te_m\}$, generate a binary matrix \mathbf{TD} , which denotes the relationship between documents and terms (i.e., $td(i, j) = 1$ if the i -th document belongs to j -th term; Otherwise, $td(i, j) = 0$). The key idea of LSA is to map documents to a latent space, where documents are close if they share meaningful association, based on \mathbf{TD} by SVD.

LSA firstly decomposes \mathbf{TD} by SVD as follows:

$$\mathbf{TD}_{n \times m} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \quad (3)$$

where \mathbf{U} and \mathbf{V} are the orthogonal matrices of left and right singular vectors $\mathbf{U}\mathbf{U}^T = \mathbf{V}\mathbf{V}^T = \mathbf{I}$ (here \mathbf{I} is an identity matrix), respectively. $\mathbf{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_r)$ is a diagonal matrix with singular values on the diagonal and the singular values are listed in non-increasing order, i.e., $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$, $r = \text{rank}(\mathbf{TD})$.

Then \mathbf{TD} is approximately computed by the top k singular values in $\mathbf{\Sigma}$, i.e.,

$$\mathbf{TD} = \tilde{\mathbf{TD}} \approx \mathbf{U}_{n \times r_k} \mathbf{\Sigma}_{r_k \times r_k} \mathbf{V}_{m \times r_k}^T, \quad (4)$$

where r_k is the number of considering $\text{rank}(\mathbf{TD})$. After that, LSA uses the rows of $\mathbf{U}_{n \times r_k} \mathbf{\Sigma}_{r_k \times r_k}$ as the coordinates for documents in the latent space. This is because the document-to-document inner products are given by:

$$\tilde{\mathbf{TD}}\tilde{\mathbf{TD}}^T = \mathbf{U}_{n \times r_k} \mathbf{\Sigma}_{r_k \times r_k} \underbrace{\mathbf{V}_{m \times r_k}^T \mathbf{V}_{m \times r_k}}_{\mathbf{I}} \mathbf{U}_{n \times r_k}^T \mathbf{\Sigma}_{r_k \times r_k}. \quad (5)$$

Based on the latent space, LSA is able to effectively analyze the relationship between documents.

2.3 Latent Semantic Space Construction

Inspired by LSA, we propose to analyze an inlier indexing matrix (generated by considering both data points and model hypotheses) via SVD for the model fitting problem. A binary inlier indexing matrix is used to represent the relationship between data points and model hypotheses. Here, we use a continuous preference matrix [21] (derived from a preference function) instead of a binary one to improve its fitting performance.

For n input data points and m generated model hypotheses, a singular value decomposition of a continuous preference matrix $\mathbf{F}_{n \times m}$ (each entry is derived from

Eq. (1)) has the following form:

$$\mathbf{F}_{n \times m} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \quad (6)$$

i.e.

$$\begin{aligned} & \begin{bmatrix} f(x_1, \theta_1) & \cdots & f(x_1, \theta_m) \\ \vdots & \ddots & \vdots \\ f(x_n, \theta_1) & \cdots & f(x_n, \theta_m) \end{bmatrix} \\ &= \left[\begin{bmatrix} u_1 \\ \vdots \\ u_r \end{bmatrix} \cdots \begin{bmatrix} u_r \\ \vdots \\ u_r \end{bmatrix} \right] \cdot \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_r \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ \vdots \\ v_r \end{bmatrix} \end{aligned} \quad (7)$$

Recall that, a preference matrix denotes the preference between data points and model hypotheses. Specifically, if the value of $f(x_i, \theta_j)$ is large, it means that the data point x_i prefers the model hypothesis θ_j . Thus, after decomposition by SVD, a left singular vector u_i denotes the preference of all data points to the i -th topic model of model hypotheses. A topic model is a estimated model instance for model fitting. Accordingly, a right singular vector v_j denotes the preference of all model hypotheses to the j -th topic model of data points.

Each singular value σ_i denotes the strength of correlation between a data point and a model hypothesis. That is, if the σ_i value is large, then it is a high possibility that the i -th topic model of data points and the i -th topic model of model hypotheses correspond to the same model instance in data; Otherwise, the i -th topic model of data points and the i -th topic model of model hypotheses are usually insignificant and thus they can be ignored. This is because an inlier and the corresponding model instance have much stronger correlation than the other cases. Therefore, we only need to analyze the top r_k data points and model hypotheses instead of analyzing all data points and model hypotheses. Here, r_k is usually set to a small value, i.e., $r_k \ll r$ (we also further discuss its influence in Sec. 4.1). Thus, Eq. (6) can be rewritten as:

$$\mathbf{F}_{n \times m} = \tilde{\mathbf{F}}_{n \times m} \approx \mathbf{U}_{n \times r_k} \mathbf{\Sigma}_{r_k \times r_k} \mathbf{V}_{m \times r_k}^T. \quad (8)$$

Eq. (8) is also consistent with the property of SVD that the top 10% (even 1%) of singular values account for over 99% of all singular values.

Then, to simplify the complex model fitting problems, we construct a latent space where the data points belonging to the same model instance (these data points share similar preference to the same model hypothesis) are close to each other. That is, the value of a data point x_i in the constructed space only relates to the left singular vectors and has nothing with the origin value of x_i . Here we call the constructed space the ‘‘latent semantic space (LSS)’’, because the space involves the latent

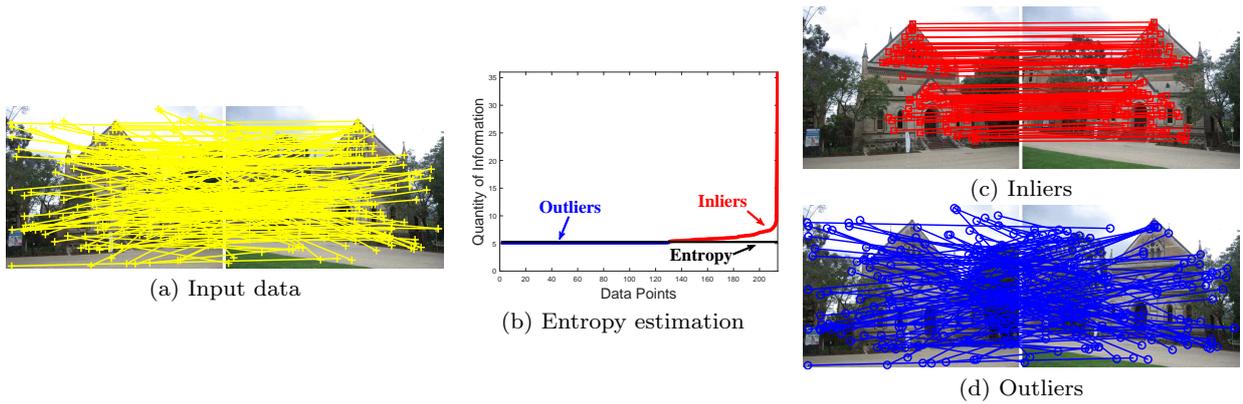


Fig. 2 An example of outlier removal based on the information theoretic approach for homography estimation. (a) The input “Elderhalla” image pair. (b) The proposed outlier removal strategy. (c) The estimated inliers. (d) The removed gross outliers. The gross outliers are marked in blue, and the inliers of different model instances are marked in red.

semantic information (the inliers belonging to the same model instance in the data usually correspond to the mapped points in one direction of LSS).

Note that the points-to-points inner products based on Eq. (8) are given by:

$$\begin{aligned} \tilde{\mathbf{F}}_{n \times n} &= \tilde{\mathbf{F}}_{n \times m} \tilde{\mathbf{F}}_{n \times m}^T \\ &= \mathbf{U}_{n \times r_k} \Sigma_{r_k \times r_k}^2 \mathbf{U}_{n \times r_k}^T. \end{aligned} \quad (9)$$

Now, we have a $n \times n$ matrix $\tilde{\mathbf{F}}$ to determine singular values Σ and left singular vectors \mathbf{U} . Thus, we use the rows of $\mathbf{U}_{n \times r_k} \Sigma_{r_k \times r_k}$ as defining coordinates for data points in LSS.

After that, we no longer analyze the preference matrix but LSS, and the dimension is reduced from a large m value of a preference matrix to a much smaller r_k value of LSS. It is also worth pointing out that, the value of each data point in LSS has its specific physical interpretation. That is, the value of a data point x_i in the j -th dimension of LSS corresponds to the preference of x_i to the j -th topic model of model hypotheses. Thus, the complex model fitting problem is reduced to a much easier LSS analysis problem.

2.4 Adaptive Outlier Removal

Based on the constructed LSS, we propose a novel simple but effective outlier removal strategy in this subsection.

In LSS, a data point x_i corresponding to an inlier is assigned a large value in a direction (which means the data point is distributed far from the origin of LSS), while x_i corresponding to a gross outlier is assigned a small value. The reason behind this is that, the value of x_i in the j -th dimension of LSS corresponds to the preference of x_i to the j -th topic model of model

hypotheses. Obviously, a topic model of model hypotheses prefers x_i if it is an inlier while **not any** topic model of model hypotheses prefers x_i if it is a gross outlier. Thus, in LSS, the points corresponding to inliers of model instances are distributed in several independent directions while the points corresponding to gross outliers are distributed close the origin (\mathbf{O}) of LSS. That is, the points corresponding to inliers are far from the origin \mathbf{O} while the points corresponding to gross outliers are close to \mathbf{O} . Thus, we remove the points corresponding to gross outliers by using a threshold, according to the distance between the mapped points and \mathbf{O} in LSS. However, it is difficult to manually set a proper threshold value for different input data.

In this paper, we propose to use information theory [10] to adaptively set the threshold value for outlier removal. Firstly, given a set of mapped points $\hat{\mathbf{X}} = \{\hat{x}_i\}_{i=1, \dots, n}$ in LSS, we compute the Euclidean distance $\hat{d}_{\hat{x}_i}^o$ between each mapped point \hat{x}_i and \mathbf{O} in LSS. Then, we compute the gap g_i between $\hat{d}_{\hat{x}_i}^o$ and the maximum distance of the mapped points to \mathbf{O} as:

$$g_i = \max\{\mathbf{D}^o\} - \hat{d}_{\hat{x}_i}^o, \quad (10)$$

where $\mathbf{D}^o = \{\hat{d}_{\hat{x}_i}^o\}_{i=1, \dots, n}$. After that, we measure the quantity of information provided by each mapped point \hat{x}_i :

$$Q(\hat{x}_i) = -\log p(\hat{x}_i), \quad (11)$$

where $p(\hat{x}_i)$ is the prior probability of each mapped point \hat{x}_i to correspond to an outlier, and it is computed by normalizing the gap:

$$p(\hat{x}_i) = g_i / \sum_{i=1}^n g_i. \quad (12)$$

The corresponding entropy is computed as:

$$L = - \sum_{i=1}^n p(\hat{x}_i) \log p(\hat{x}_i). \quad (13)$$

We remove the mapped points with quantity of information lower than the estimated entropy value L and retain the points with quantity of information higher than the entropy value L :

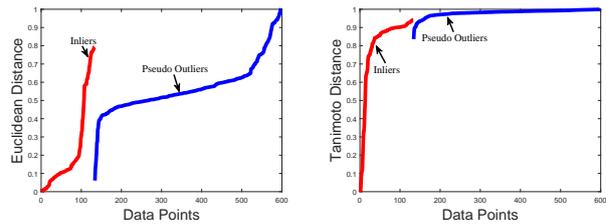
$$\hat{\mathbf{X}}_I = \{x_i | Q(\hat{x}_i) > L\}. \quad (14)$$

To illustrate the outlier removal strategy, in Fig. 2, we show an example of outlier removal (based on our information theoretic approach) for homography estimation on the ‘‘Elderhalla’’ image pair. We can see that, inliers are assigned a large quantity of information while gross outliers are assigned a smaller quantity of information (see Fig. 2(b)). The proposed outlier removal strategy successfully distinguishes the mapped points in LSS (see Fig. 2(c) and (d)).

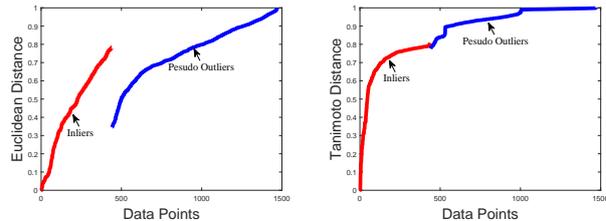
It is worth pointing out that, although [35, 36] and the proposed outlier removal strategy in this paper, use information theory similar to that in [10] to improve the fitting performance, they are significantly different in that: 1) They deal with different objects, i.e., the proposed strategy in this paper uses information theory to remove gross outliers from all data points while [35, 36] use it to remove insignificant model hypotheses from all generated model hypotheses; 2) They work in different spaces, i.e., the proposed strategy works in LSS while [35, 36] work in the parameter space.

2.5 Inlier Segmentation

The constructed latent semantic space is well approximated by multiple low-dimensional spaces. Thus, after removing the points corresponding to gross outliers (those with high probabilities in LSS), we propose to cluster the remaining mapped points, where each cluster corresponds to a model instance and the corresponding data points are the inliers of the model instance. From the distribution of the mapped points in LSS (see Fig. 1(d)), we can see that the points belonging to different clusters have obvious discrimination. The reason behind this is that, the inliers belonging to the same model instance, i.e., the same topic model of model hypotheses, will prefer the same directions in LSS. Thus, we can use a clustering method (e.g., K-means [20], DBSCAN [9] and Clusterdp [29]) to cluster the remaining points in LSS. However, these clustering methods have some limitations, e.g., K-means is sensitive to the selection of the initial centers; for DBSCAN and Clusterdp,



(a) Line fitting based segmentation



(b) Homography based segmentation

Fig. 3 Two examples of the distance distribution between an inlier and the other data points (i.e. the other inliers and the pseudo outliers) based on the Euclidean distance (Left) and the Tanimoto distance (Right) for two model fitting tasks.

it is difficult to set their parameters for different input data.

In this paper, we follow the idea of K-means++ [2], which improves K-means by employing a novel technique for seeding the initial centers. Specifically, we propose a new way of seeding the initial centers for K-means by introducing the preference function (see Eq. (1)). We measure the similarity between two points (\hat{x}_i, \hat{x}_j) in LSS based on their Tanimoto distance [30]:

$$s(\hat{x}_i, \hat{x}_j) = 1 - \frac{\langle \mathbf{PS}_i, \mathbf{PS}_j \rangle}{\|\mathbf{PS}_i\|^2 + \|\mathbf{PS}_j\|^2 - \langle \mathbf{PS}_i, \mathbf{PS}_j \rangle}, \quad (15)$$

where \mathbf{PS}_i and \mathbf{PS}_j are the preference sets of the data points \hat{x}_i and \hat{x}_j for all model hypotheses³. $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ denote the standard inner product and the corresponding induced norm, respectively.

Compared with the Euclidean distance, the Tanimoto distance is much more robust to the distribution of data points, for our case. Note that, if two data points share similar preference to model hypotheses, then their Tanimoto distance will be a small value; Otherwise, their Tanimoto distance will be a large value. In Fig. 3, we show two examples of the distance distribution between an inlier and the other data points (including the other inliers and the pseudo outliers) based on the Euclidean distance and the Tanimoto distance on the ‘‘6 lines’’ data and the ‘‘Unihouse’’ data, respectively.

³ The elements of the preference set \mathbf{PS}_i are computed according to the preference function $f(\cdot)$ in Eq. (1), based on the corresponding data point x_i and all the generated model hypotheses. Similarly, the elements of \mathbf{PS}_j can be computed.

Algorithm 1 The K-means*++ Clustering Algorithm

Input: Data points \hat{X} , the number of model instances k

- 1: Select one center c_1 randomly from the input data points in LSS and add it to the center pool.
- 2: Compute the Tanimoto distance (referred to as T-distance) between all data points and the centers in the center pool by Eq. (15).
- 3: Select the data point with the maximum T-distance as a new center and add it to the center pool.
- 4: Repeat Steps 2 to 3 until k centers are selected.
- 5: Proceed with the standard K-means algorithm [20] to obtain the labels of data points.

Output: The labels of data points \hat{X} .

We can see that, many pseudo outliers have the same distance as inliers, when the Euclidean distance is used; while only a few pseudo outliers share the same distance with inliers, when the Tanimoto distance is used. Thus, the Tanimoto distance is able to more effectively capture the distance between inliers from the same model instance than the Euclidean distance, and we employ the Tanimoto distance, rather than the Euclidean distance, in the proposed fitting method.

Then, based on the similarity, we seed the centers from data points, according to their Tanimoto distance to the closest center that is chosen. We summarize the proposed clustering algorithm (called K-means*++) in Algorithm 1. The proposed K-means*++ is an improvement over K-means++ in terms of initial center selection. Compared with the standard K-means [20], both K-means++ and K-means*++ are able to achieve more stable clustering results, due to their effectiveness for seeding the initial centers. However, it is worth pointing out that K-means*++ is able to achieve better clustering results than K-means++ for model fitting. This is because that K-means*++ uses the preference information to measure the similarity between data points, which is more effective than the Euclidean distance used by K-means++, for robust model fitting.

3 The Complete Method and Complexity Analysis

In this section, based on the components described in the previous sections, we summarize the complete Continuous Latent Semantic Analysis (CLSA) fitting method in Algorithm 2.

The proposed CLSA mainly includes three parts: the latent semantic space (LSS) construction, outlier removal, and inlier segmentation. CLSA firstly constructs LSS by both latent semantic analysis and continuous preference analysis, to map inliers of different model instances to several independent directions while gross outliers are mapped to the region close to \mathbf{O} in LSS.

Algorithm 2 The CLSA fitting method

Input: Data points X , the threshold value of ψ and the dimension r_k .

- 1: Sample a number of minimal subsets to generate model hypotheses.
- 2: Compute the preference matrix by Eq. (1).
- 3: Construct the latent semantic space (LSS) and map all data points to LSS according to Eq. (9).
- 4: Remove the points corresponding to gross outliers in LSS by Eq. (14).
- 5: Label the remaining points in LSS via the proposed K-means*++ (see Algorithm 1).
- 6: Estimate the parameters of model instances in data according to the labels of the points in LSS.

Output: The parameters of model instances and the labels of data points.

Then, CLSA analyzes the distribution of the mapped points in LSS and adaptively removes gross outliers by using information theory, which makes the model fitting problem easier to deal with. After that, CLSA labels the remaining points via an improved K-means*++ clustering algorithm, by which the parameters and inliers of model instances are estimated and segmented, respectively. As the experimental results show in Sec. 4, CLSA is able to efficiently and effectively deal with the multi-structure fitting problem.

For the computational complexity of the proposed CLSA, we mainly focus on improving both the effectiveness and computational efficiency of model selection for model fitting in this paper, and thus we do not consider the time for model hypothesis generation (i.e. Step 1). The preference matrix can be directly derived from the residual values between data points and model hypotheses (i.e. Step 2), thus its computational complexity is $O(n)$, where n is the number of data points. For constructing the latent semantic space (i.e. Step 3), the computational complexity is approximately $O(nr_k)$, where r_k is the dimension of LSS, respectively. For outlier removal (i.e. Step 4), the computational complexity is $O(n)$. For Algorithm 1, the computational complexity is $O(\hat{n}m)$, where \hat{n} is the number of estimated inliers. Thus the total complexity of CLSA approximately amounts to $O(nr_k + \hat{n}m)$.

4 Experiments

In this section, we investigate the performance of the proposed CLSA fitting method, and compare it with several state-of-the-art model fitting methods⁴, including the consensus analysis based fitting methods (i.e.,

⁴ For MSHF, we use the code provided by the authors. For RansaCov, T-linkage and RPA, we download the codes from the following website: <http://www.diegm.uniud.it/fusiello/demo>.

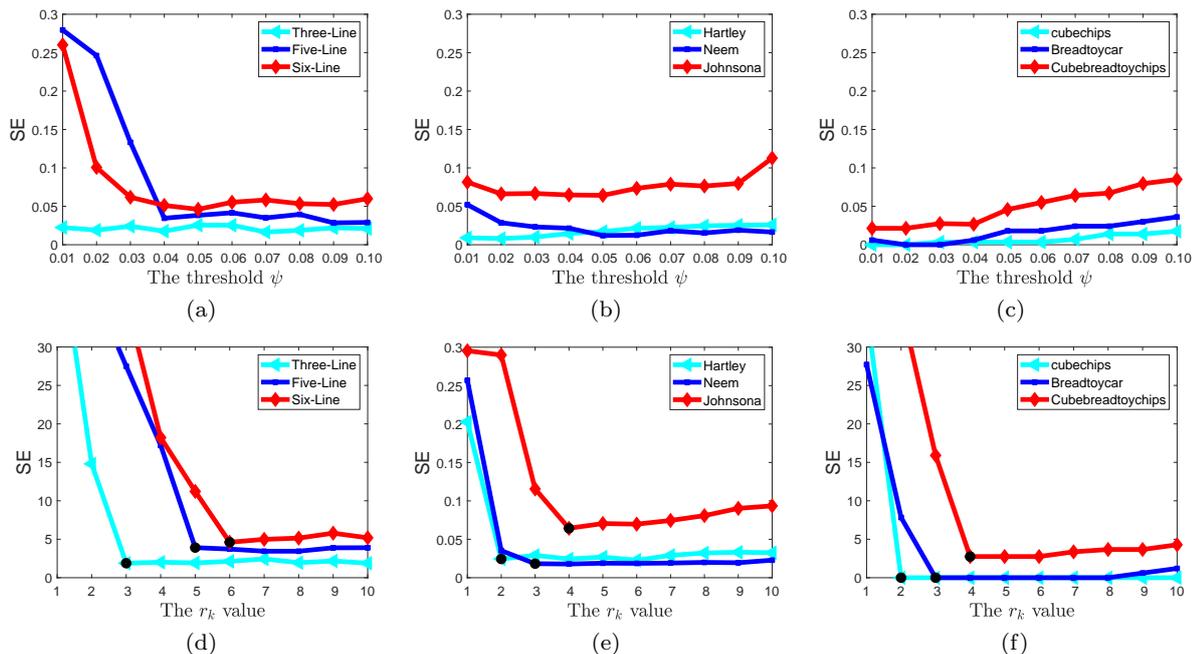


Fig. 4 The segmentation errors obtained by the proposed CLSA2 fitting method with different parameter values for line fitting based segmentation ((a) and (d)), homography based segmentation ((b) and (e)), and two-view based motion segmentation ((c) and (f)). The horizontal coordinate values of the black points in (d), (e) and (f) denote the groundtruth number of model instances in data.

MSHF [36] and RansaCov [22]) and the preference analysis based fitting methods (i.e., T-linkage [21] and R-PA [23]), on both synthetic data and real images. Moreover, to show the effectiveness of the proposed clustering algorithm, we test two versions of CLSA: CLSA1, which uses the traditional K-means algorithm [20] to cluster points in LSS, and CLSA2, which uses the proposed K-means*++ algorithm to cluster points in LSS. We also run RCMSA [27] as a baseline for high-order fitting tasks (i.e., homography based segmentation and two-view based motion segmentation) due to its effectiveness.

All experiments are run on MS Windows 10 with Intel Core i7-3770 CPU 3.4GHz and 16GB RAM. The segmentation error (SE) is measured by the cost function in Eq. (2).

4.1 Parameter Analysis and Settings

In this subsection, we analyze the influence of the parameter (i.e., the threshold ψ in Eq. (1)) of the proposed CLSA2 fitting method on its performance. We test different values of the parameters for line fitting based segmentation, homography based segmentation and two-view based motion segmentation (for each task in Sec. 4.4, Sec. 4.5 and Sec. 4.6, we test **three datasets**, respectively), and show the segmentation errors ob-

tained by CLSA2 with different parameter values in Fig. 4 (a)-(c).

We can see that, CLSA2 is able to achieve low segmentation errors in the three fitting tasks for most cases. Specifically, when the value of the threshold ψ (Fig. 4(a)~(c)) is respectively set to 0.04 ~ 0.10 (for line fitting based segmentation), 0.02 ~ 0.09 (for homography based segmentation) and 0.01 ~ 0.04 (for two-view based motion segmentation), CLSA2 is able to achieve relatively stable results. Thus, we set the value of ψ to 0.04 for all the following experiments.

We have also tested the proposed CLSA2 fitting method with different r_k values in Eq (9) for line fitting based segmentation, homography based segmentation and two-view based motion segmentation. We show the segmentation errors obtained by CLSA2 with different r_k values in Fig. 4 (d)-(f).

We can see that, when the r_k value is less than the ground truth number of model instances in data, CLSA2 fails to fit model instances for all three fitting tasks. In contrast, when the r_k value is equal to or larger than the ground truth number of model instances, CLSA2 is able to achieve low segmentation errors. This is because some inliers of different model instances are wrongly mapped into one direction, if the r_k value is less than the ground truth number of model instances, which causes K-means*++ (used by CLSA2) to fail to correctly label these points. Thus, we set the r_k value to

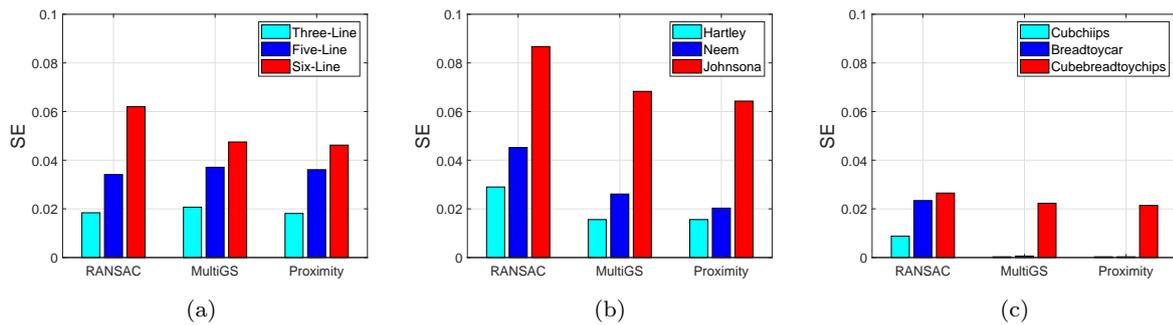


Fig. 5 The segmentation errors obtained by the proposed CLSA2 fitting method with different sampling algorithms (i.e., RANSAC, MultiGS and Proximity) for the tasks of (a) line fitting based segmentation, (b) homography based segmentation and (c) two-view based motion segmentation.

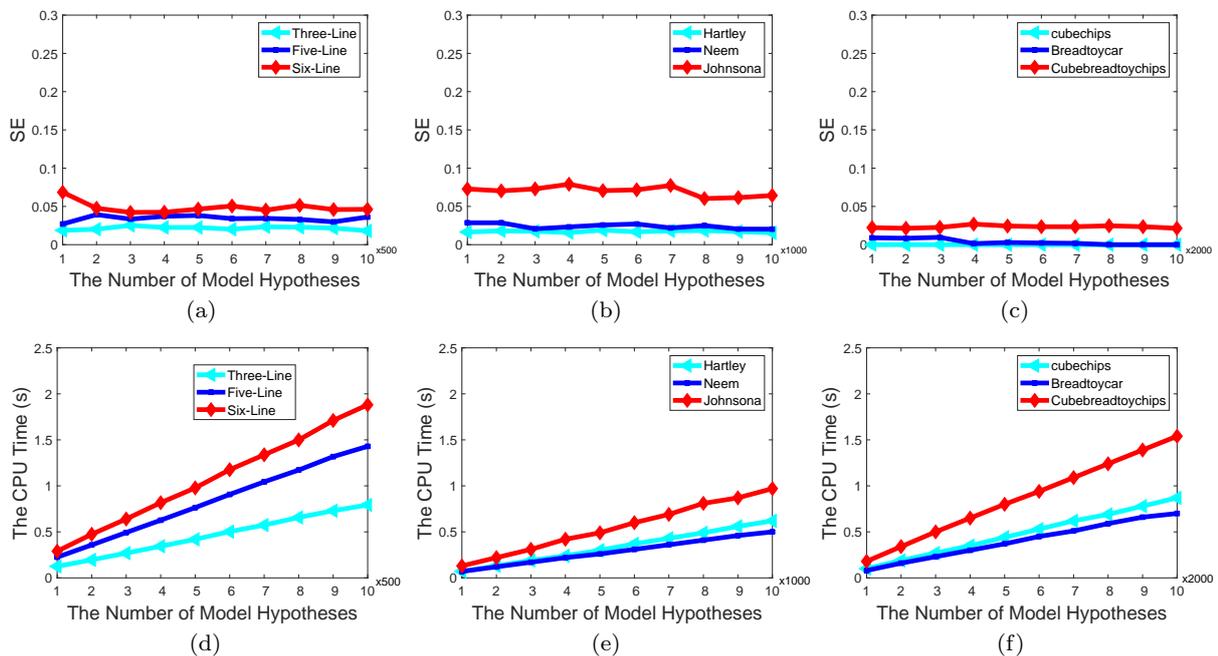


Fig. 6 The segmentation errors obtained by and the CPU time used by the proposed CLSA2 fitting method with different numbers of model hypotheses for the tasks of line fitting based segmentation ((a) and (d)), homography based segmentation ((b) and (e)) and two-view based motion segmentation ((c) and (f)).

a large value, which is obviously larger than the ground truth number of model instances for Eq. (9).

In this paper, we mainly focus on model selection. Thus, we provide the number of model instances to the proposed method as the competing methods (i.e., RansCov, RPA and T-linkage) do. That is, r_k is equal to k in Algorithm 1.

4.2 The Influence of Sampling Algorithms

We also analyze the performance of CLSA2 with three popular sampling algorithms (i.e., RANSAC [11], MultiGS [6] and Proximity [14]) for line fitting based segmentation, homography based segmentation and two-view

based motion segmentation, respectively. For each task in Sec. 4.4, Sec. 4.5 and Sec. 4.6, we test three datasets, respectively, and we also show the segmentation errors obtained by CLSA2 in Fig. 5.

We can see that, for line fitting based segmentation, three versions of CLSA2 using all three sampling algorithms are able to achieve low segmentation errors (see Fig. 5 (a)) compared with the other competing fitting methods. However, when using the random sampling algorithm (which is used in RANSAC), CLSA2 generates a high proportion of bad model hypotheses for high-order fitting tasks. For this case, CLSA2 achieves relatively higher segmentation errors than those achieved by CLSA2 using the guided sampling algorithms (see Fig. 5 (b) and (c)). Among the three sampling algo-

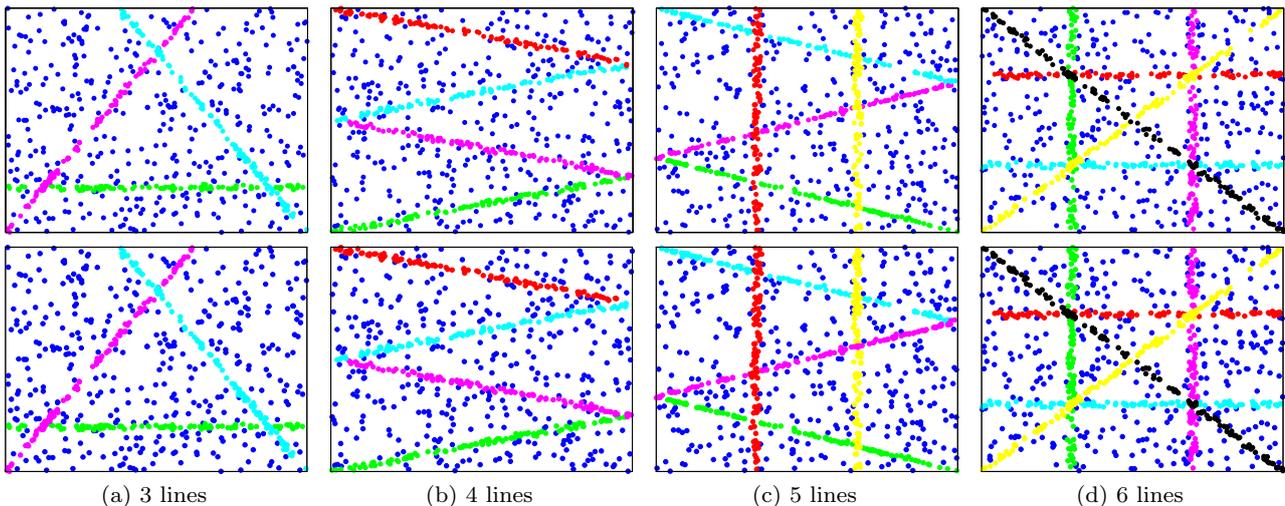


Fig. 7 The results obtained by the proposed methods (CLSA2) for line fitting based segmentation. The first row of each sub-figure shows the ground truth segmentation results and the second row shows the fitting and segmentation results obtained by CLSA2. The inlier noise scale is 1.5 and each line has 100 inliers. The outliers are labeled in blue, and the inliers of each estimated model instance are labeled in different colors (the following figures have the same settings).

gorithms, Proximity generates a similar quality of model hypotheses as MultiGS does, but it is much faster than MultiGS. Therefore, we select Proximity as the sampling algorithm for the proposed method and all the other competing methods in the following experiments.

4.3 The Influence of the Number of Model Hypotheses

In this subsection, we analyze the performance of CLSA2 with different numbers of model hypotheses for line fitting based segmentation, homography based segmentation and two-view based motion segmentation. For each task, we also test the three datasets, and we show the segmentation errors obtained by CLSA2 and the CPU time used by CLSA2 in Fig. 6.

We can see that, CLSA2 is able to achieve low segmentation errors even when we generate a small number of model hypotheses for all three fitting tasks (see Fig. 6 (a)-(c)), and it also achieves stable fitting results (across different numbers of model hypotheses), which further shows the stable properties of CLSA2. For the CPU time, CLSA2 spends more time with increasing number of model hypotheses for the three fitting tasks (see Fig. 6 (d)-(f)). This is because that the preference analysis becomes more complex when more model hypotheses are generated. Therefore, for CLSA2, we can generate a small number of model hypotheses for the three fitting tasks. However, to fairly compare with the other competing methods, we use the same setting of the number of model hypotheses, as that used in [35] and [36]. That is, we generate 5,000, 10,000 and

Table 1 Quantitative comparison results obtained by the six competing methods for line fitting based segmentation on four synthetic data. The best results of SE and CPU time are respectively labeled in magenta and cyan.

Data		MSHF	RansaCov	T-linkage	RPA	CLSA1	CLSA2
3 lines	SE	0.05±0.05	0.03±0.00	0.03±0.01	0.05±0.01	0.03±0.04	0.02±0.01
	Time	1.96	2.88	88.94	175.31	0.65	0.80
4 lines	SE	0.06±0.05	0.03±0.01	0.10±0.08	0.03±0.01	0.08±0.14	0.02±0.00
	Time	2.12	3.46	119.12	222.11	0.83	1.12
5 lines	SE	0.03±0.01	0.03±0.00	0.11±0.11	0.03±0.02	0.08±0.11	0.02±0.01
	Time	2.11	3.51	154.61	360.55	0.99	1.47
6 lines	SE	0.11±0.04	0.08±0.02	0.35±0.10	0.41±0.09	0.10±0.11	0.05±0.02
	Time	2.16	4.37	193.74	430.03	1.24	1.87

20,000 model hypotheses for line fitting based segmentation, homography based segmentation and two-view based motion segmentation, respectively. Moreover, all the competing methods select model instances based on the same model hypotheses.

4.4 Line Fitting based Segmentation

We evaluate the performance of the six fitting methods for line fitting based segmentation on four challenging synthetic data. We repeat each experiment 50 times. We report the average segmentation errors (in the format of *average errors*±*standard deviation*) obtained by the six fitting methods, and the average CPU time⁵ (in seconds) used by the competing methods in Table 1. We also show the fitting results obtained by the proposed method in Fig. 7.

⁵ For the CPU time, we exclude the time used for sampling minimal subsets and generating model hypotheses for all the competing fitting methods in all the following experiments.

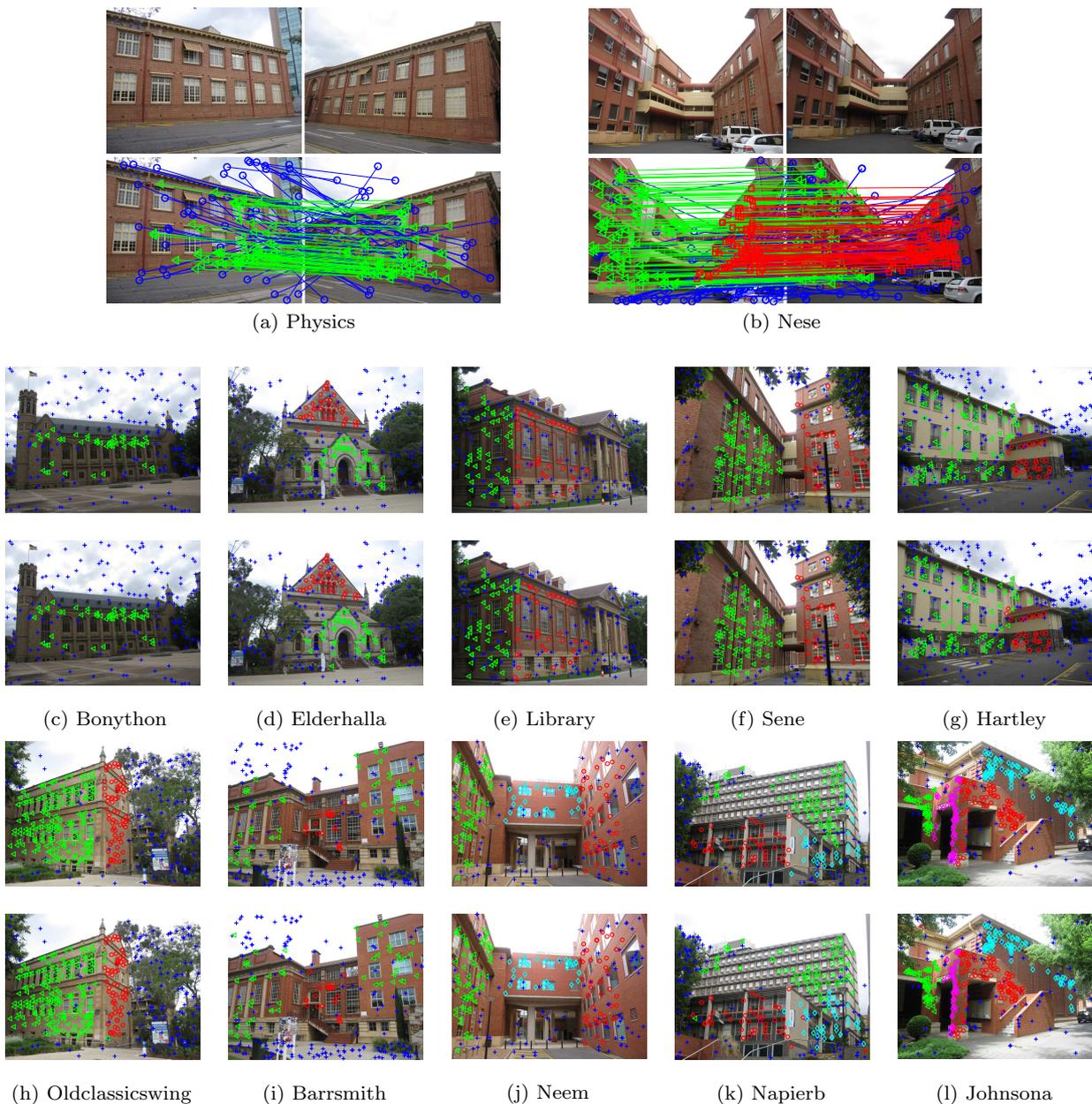


Fig. 8 The results obtained by CLSA2 for homography based segmentation on some of the AdelaideRMF dataset (two views are shown for each case in (a)-(b), and only one of the two views is shown for each case in (c)-(l)). The first row in (a)-(b) shows the original images. The first row in (c)-(l) shows the ground truth segmentation results, and the second row in (a)-(l) shows the fitting results obtained by CLSA2.

From Fig. 7 and Table 1, we can see that the proposed CLSA2 fitting method achieves significantly better results than the other five competing fitting methods, with regard to the average segmentation errors. For the CPU time, CLSA2 achieves similar performance to CLSA1 but it is much faster than the other four fitting methods: CLSA2 is about 1.10 to 229.96 times faster than the other four fitting methods. CLSA1/CLSA2 are faster than the other four competing methods since they effectively reduce the high dimension of the prefer-

ence matrix, by which they are able to greatly improve the computational efficiency. Moreover, they formulate the line fitting problem as a simpler clustering problem, which further improves the computational speed.

For the segmentation accuracy, MSHF, RansaCov and CLSA1 achieve similar low segmentation errors for all the four data. T-linkage and RPA achieve low segmentation errors for the “3 lines”, “4 lines” and “5 lines” data, but both may fail to fit the “6 lines” data among the 50 repetitions because they deal with the

Table 2 Quantitative comparison results obtained by the seven competing methods for homography based segmentation on all 19 image pairs from the AdelaideRMF dataset. “ k ” is the actual number of model instances in each data and “%Outlier” is the percentage of gross outliers. The best results of SE and CPU time are respectively labeled in magenta and cyan.

Data	k	%Outlier		MSHF	RansaCov	T-linkage	RPA	RCMSA	CLSA1	CLSA2
Unionhouse	1	18.78	<i>SE</i>	0.00±0.00	0.00±0.00	0.32±0.25	0.02±0.00	0.04±0.00	0.00±0.00	0.00±0.00
			<i>Time</i>	1.09	7.49	49.54	60.57	20.18	0.21	0.21
Bonython	1	75.13	<i>SE</i>	0.00±0.00	0.00±0.00	0.06±0.09	0.03±0.01	0.05±0.08	0.00±0.00	0.00±0.00
			<i>Time</i>	1.66	12.00	16.57	25.01	1.15	0.28	0.28
Physics	1	46.60	<i>SE</i>	0.00±0.00	0.03±0.00	0.32±0.12	0.02±0.01	0.28±0.08	0.00±0.00	0.00±0.00
			<i>Time</i>	1.65	6.85	3.36	11.19	0.89	0.16	0.16
Elderhalla	2	60.75	<i>SE</i>	0.01±0.00	0.01±0.00	0.02±0.01	0.02±0.01	0.10±0.11	0.01±0.00	0.01±0.00
			<i>Time</i>	1.27	6.97	18.06	44.73	1.15	0.41	0.46
Ladysymon	2	33.48	<i>SE</i>	0.03±0.01	0.04±0.01	0.07±0.02	0.23±0.01	0.06±0.02	0.09±0.03	0.05±0.04
			<i>Time</i>	1.93	11.41	24.69	55.04	1.29	0.14	0.17
Library	2	56.13	<i>SE</i>	0.02±0.01	0.06±0.01	0.06±0.01	0.06±0.03	0.03±0.00	0.06±0.01	0.05±0.02
			<i>Time</i>	1.25	5.47	18.37	52.65	1.13	0.41	0.48
Nese	2	30.29	<i>SE</i>	0.00±0.00	0.04±0.00	0.02±0.00	0.04±0.00	0.01±0.00	0.00±0.00	0.00±0.00
			<i>Time</i>	1.93	11.19	28.71	51.59	1.12	0.35	0.35
Sene	2	44.49	<i>SE</i>	0.01±0.00	0.02±0.00	0.00±0.00	0.00±0.01	0.01±0.00	0.00±0.00	0.00±0.00
			<i>Time</i>	1.56	9.19	27.67	55.59	1.14	0.47	0.54
Napiera	2	64.73	<i>SE</i>	0.28±0.03	0.15±0.01	0.07±0.01	0.33±0.04	0.22±0.06	0.25±0.04	0.14±0.03
			<i>Time</i>	1.19	13.98	42.32	91.66	1.73	0.53	0.62
Hartley	2	62.22	<i>SE</i>	0.02±0.00	0.03±0.00	0.03±0.01	0.12±0.00	0.06±0.11	0.02±0.00	0.02±0.00
			<i>Time</i>	1.22	7.97	46.33	94.73	1.60	0.59	0.62
Oldclassicswing	2	32.23	<i>SE</i>	0.01±0.00	0.05±0.05	0.02±0.01	0.20±0.00	0.01±0.00	0.02±0.01	0.01±0.01
			<i>Time</i>	2.30	7.78	67.18	98.73	1.55	0.67	0.67
Barrsmith	2	69.79	<i>SE</i>	0.24±0.10	0.10±0.02	0.22±0.04	0.12±0.00	0.16±0.13	0.02±0.00	0.01±0.00
			<i>Time</i>	1.44	14.43	29.16	66.02	1.72	0.56	0.65
Neem	3	37.83	<i>SE</i>	0.02±0.00	0.11±0.02	0.15±0.09	0.22±0.12	0.06±0.02	0.03±0.04	0.02±0.01
			<i>Time</i>	1.74	9.29	24.98	82.02	1.13	0.45	0.56
Elderhallb	3	49.80	<i>SE</i>	0.03±0.01	0.12±0.02	0.13±0.02	0.18±0.02	0.18±0.11	0.09±0.07	0.08±0.06
			<i>Time</i>	1.51	13.28	29.23	97.94	1.24	0.71	0.86
Napierb	3	37.13	<i>SE</i>	0.13±0.04	0.16±0.03	0.18±0.01	0.19±0.01	0.19±0.04	0.17±0.02	0.11±0.01
			<i>Time</i>	1.65	16.24	26.58	77.72	1.08	0.47	0.57
Johnsona	4	21.25	<i>SE</i>	0.03±0.01	0.06±0.01	0.06±0.02	0.16±0.00	0.04±0.01	0.15±0.09	0.06±0.00
			<i>Time</i>	2.22	16.48	65.09	152.66	1.68	0.64	0.91
Unihouse	5	18.78	<i>SE</i>	0.09±0.00	0.07±0.03	0.14±0.00	0.05±0.02	0.04±0.00	0.19±0.06	0.18±0.01
			<i>Time</i>	5.70	62.14	1,833.87	3,133.69	20.24	3.47	4.29
Bonhall	6	6.43	<i>SE</i>	0.32±0.09	0.21±0.02	0.33±0.07	0.39±0.08	0.10±0.04	0.37±0.06	0.32±0.03
			<i>Time</i>	3.55	45.96	650.09	1,111.15	6.72	1.52	1.91
Johnsonb	7	12.02	<i>SE</i>	0.17±0.05	0.13±0.03	0.19±0.06	0.25±0.10	0.10±0.01	0.40±0.15	0.26±0.06
			<i>Time</i>	2.73	26.38	206.05	539.91	3.37	1.71	1.76
<i>Mean</i>				0.074	0.080	0.131	0.139	0.091	0.099	0.070
<i>Median</i>				0.024	0.060	0.074	0.124	0.058	0.032	0.020

crossing model instances ineffectively (note that the “6 lines” data includes many crossing model instances). In contrast to CLSA2, CLSA1 is not very stable. This is because that K-means used by CLSA1 is sensitive to the initialization. Thus, CLSA1 is not able to achieve low standard deviations of error. In contrast, CLSA2 succeeds in fitting all the four data, and achieves the lowest average segmentation errors among the six competing fitting methods, for all the four data. Compared with CLSA1, CLSA2 is also very stable and achieves relatively lower standard deviations due to the effectiveness of the proposed K-means*++ (used by CLSA2).

4.5 Homography based Segmentation

We evaluate the competing methods for homography based segmentation on the 19 real image pairs from the AdelaideRMF dataset [37]⁶ (the dataset contains 19 image pairs for homography based segmentation, and 19 image pairs for two-view based motion segmentation, which will be evaluated in Sec. 4.6). We repeat each experiment 50 times, and report the average results obtained by the seven competing methods in Table 2. We also show some fitting results obtained by CLSA2 for homography based segmentation in Fig. 8.

⁶ <http://cs.adelaide.edu.au/~hwong/doku.php?id=data>

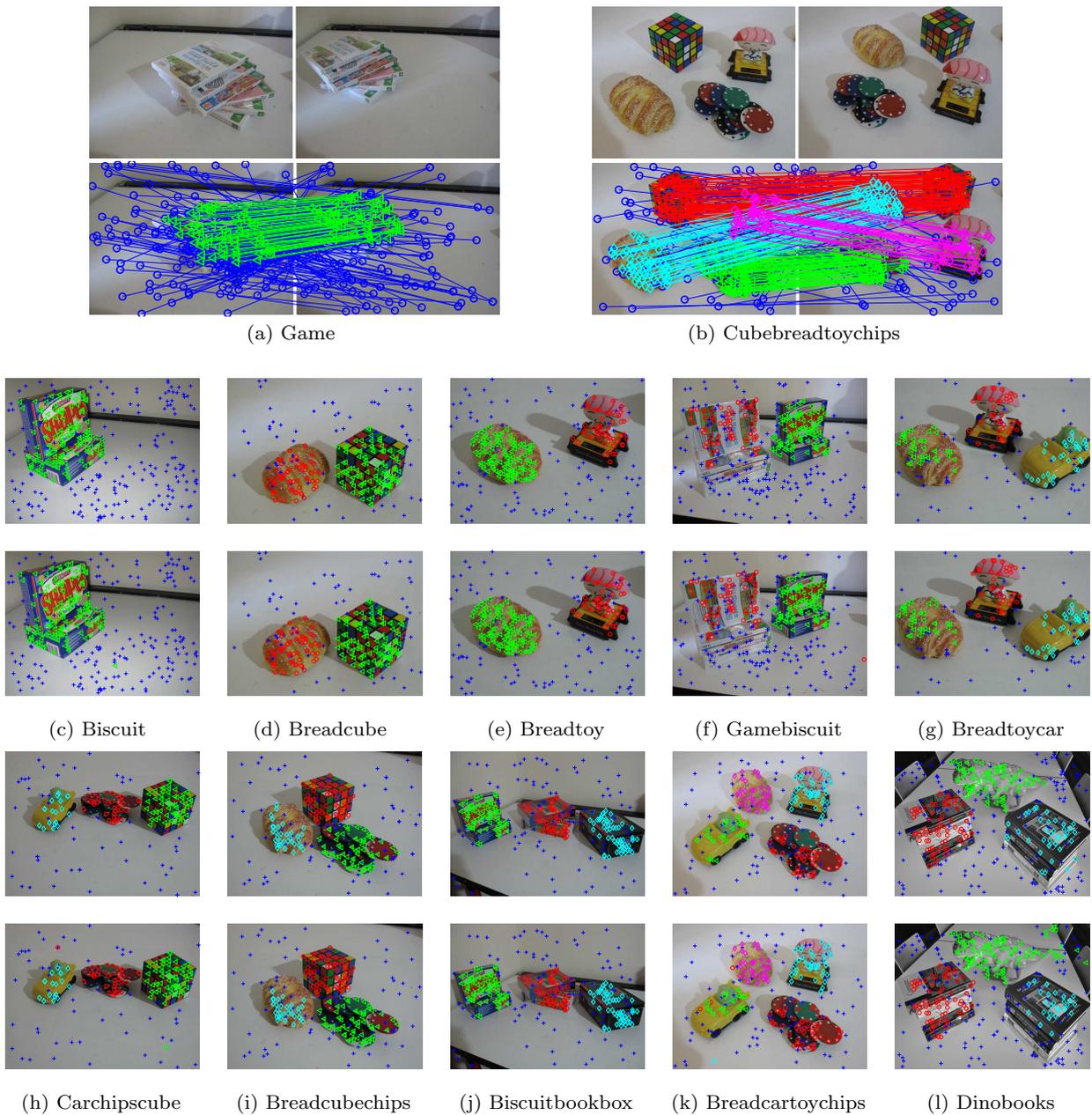


Fig. 9 Some results obtained by CLSA2 for two-view based motion segmentation on the AdelaideRMF dataset (two views are shown for each case in (a)-(b), and only one of the two views is shown for each case in (c)-(l)). The first row in (a)-(b) shows the original images. The first row in (c)-(l) shows the ground truth segmentation results, and the second row in (a)-(l) shows the fitting results obtained by CLSA2.

From Fig. 8 and Table 2, we can see that CLSA2 is able to achieve good results, obtaining the lowest average segmentation errors for 11 out of the 19 image pairs and obtaining the top three lowest segmentation errors for 7 out of the other 8 image pairs among all the seven competing fitting methods. Considering all the 19 image pairs, CLSA2 also achieves the lowest mean and median values of overall segmentation errors. It is worth pointing out that CLSA1/CLSA2 im-

plement the process of model selection within one second for 16 out of the 19 image pairs, which is much less than the time used by the other five competing methods. In contrast, MSHF only succeeds in fitting 13 out of the 19 image pairs. RansaCov also achieves low segmentation errors for 11 out of the 19 image pairs. However, RansaCov is slower than MSHF for all image pairs and it produces a higher median value of overall segmentation errors than MSHF. This is because

Table 3 Quantitative comparison results obtained by the seven competing methods for two-view based motion segmentation on 19 image pairs from the AdelaideRMF dataset.

Data	k	%Outlier		MSHF	RansaCov	T-linkage	RPA	RCMSA	CLSA1	CLSA2	
Cube	1	69.49	<i>SE</i>	0.02±0.01	0.04±0.01	0.06±0.03	0.05±0.02	0.06±0.04	0.03±0.00	0.03±0.00	
			<i>Time</i>	5.00	9.14	75.73	42.61	3.48	0.81	0.81	
Book	1	44.32	<i>SE</i>	0.01±0.00	0.06±0.01	0.13±0.08	0.05±0.02	0.02±0.02	0.00±0.00	0.00±0.00	
			<i>Time</i>	4.34	10.14	24.59	16.78	2.56	0.53	0.53	
Biscuit	1	57.68	<i>SE</i>	0.01±0.01	0.06±0.01	0.05±0.04	0.09±0.03	0.00±0.00	0.00±0.00	0.00±0.00	
			<i>Time</i>	5.08	12.72	95.21	51.14	3.93	0.88	0.88	
Game	1	73.48	<i>SE</i>	0.02±0.01	0.07±0.02	0.05±0.03	0.09±0.02	0.04±0.02	0.00±0.00	0.00±0.00	
			<i>Time</i>	4.74	4.07	40.62	24.98	3.10	0.41	0.41	
Biscuitbook	2	47.51	<i>SE</i>	0.03±0.02	0.05±0.01	0.02±0.01	0.06±0.02	0.03±0.01	0.00±0.00	0.00±0.00	
			<i>Time</i>	4.99	13.60	100.96	64.28	2.96	0.91	1.12	
Breadcube	2	32.19	<i>SE</i>	0.02±0.01	0.05±0.01	0.03±0.01	0.06±0.02	0.15±0.10	0.00±0.00	0.00±0.00	
			<i>Time</i>	4.53	10.45	49.61	33.99	2.42	0.68	0.82	
Breadtoy	2	37.41	<i>SE</i>	0.19±0.08	0.33±0.02	0.04±0.01	0.12±0.06	0.05±0.06	0.00±0.00	0.00±0.00	
			<i>Time</i>	5.01	14.86	74.39	48.91	2.60	0.76	0.86	
Cubechips	2	51.62	<i>SE</i>	0.04±0.01	0.07±0.01	0.01±0.01	0.08±0.02	0.04±0.01	0.00±0.00	0.00±0.00	
			<i>Time</i>	4.66	10.63	67.19	45.72	2.64	0.76	0.87	
Cubetoy	2	41.42	<i>SE</i>	0.02±0.01	0.08±0.01	0.03±0.02	0.04±0.03	0.06±0.04	0.00±0.00	0.00±0.00	
			<i>Time</i>	4.38	9.14	52.17	36.01	2.36	0.70	0.83	
Gamebiscuit	2	51.54	<i>SE</i>	0.02±0.01	0.08±0.04	0.03±0.01	0.08±0.01	0.05±0.03	0.01±0.00	0.01±0.00	
			<i>Time</i>	4.97	11.20	91.71	59.04	2.91	0.88	1.05	
Breadtoyocar	3	34.15	<i>SE</i>	0.08±0.02	0.23±0.01	0.08±0.01	0.14±0.02	0.14±0.08	0.14±0.15	0.00±0.00	
			<i>Time</i>	4.12	3.41	19.03	25.65	1.91	0.48	0.59	
Carchipscube	3	36.59	<i>SE</i>	0.26±0.08	0.24±0.10	0.16±0.04	0.32±0.07	0.07±0.06	0.10±0.11	0.03±0.00	
			<i>Time</i>	3.95	6.14	18.61	23.78	1.81	0.60	0.77	
Toycubecar	3	36.36	<i>SE</i>	0.14±0.07	0.35±0.09	0.10±0.02	0.14±0.03	0.13±0.02	0.11±0.02	0.08±0.00	
			<i>Time</i>	4.18	8.20	29.75	30.60	2.03	0.85	0.93	
Breadcubechips	3	35.22	<i>SE</i>	0.04±0.02	0.13±0.08	0.04±0.01	0.10±0.01	0.09±0.07	0.21±0.10	0.01±0.00	
			<i>Time</i>	4.55	6.81	42.99	38.97	2.44	0.83	1.03	
Biscuitbookbox	3	37.21	<i>SE</i>	0.03±0.01	0.23±0.02	0.03±0.02	0.04±0.02	0.03±0.01	0.11±0.13	0.01±0.00	
			<i>Time</i>	4.76	16.20	57.73	47.22	2.47	0.89	1.11	
Cubebreadtoychips	4	28.03	<i>SE</i>	0.13±0.07	0.24±0.05	0.04±0.12	0.07±0.02	0.11±0.05	0.11±0.08	0.02±0.00	
			<i>Time</i>	4.83	9.57	93.86	83.49	3.12	1.09	1.53	
Breadcartoychips	4	35.20	<i>SE</i>	0.25±0.06	0.16±0.07	0.06±0.09	0.10±0.02	0.09±0.03	0.11±0.06	0.05±0.00	
			<i>Time</i>	4.36	5.70	43.37	47.24	2.60	0.84	1.16	
Dinobooks	3	44.54	<i>SE</i>	0.18±0.02	0.31±0.02	0.23±0.01	0.16±0.02	0.20±0.06	0.16±0.06	0.10±0.00	
			<i>Time</i>	4.95	12.46	114.17	91.99	3.43	1.43	1.87	
Boardgame	3	42.48	<i>SE</i>	0.22±0.08	0.22±0.03	0.24±0.06	0.14±0.03	0.16±0.04	0.21±0.02	0.17±0.00	
			<i>Time</i>	4.56	9.45	65.25	52.16	2.56	0.87	1.48	
				Mean	0.102	0.158	0.076	0.100	0.079	0.069	0.027
				Median	0.050	0.126	0.048	0.085	0.063	0.030	0.006

MSHF removes most of the ineffective model hypotheses before performing the process of model selection, while RansaCov considers all generated model hypotheses during the model selection procedure. Although T-linkage and RPA are preference analysis based fitting methods (like CLSA2), both are not able to achieve low segmentation errors (compared to CLSA2) for most of the image pairs. This is because that CLSA2 only keeps the principal preference information while removing the redundant information by latent semantic analysis. Moreover, CLSA2 uses an effective clustering algorithm (i.e., K-means*++) to label data points. Note that, CLSA1 achieves the same average segmentation errors as CLSA2 for 7 out of the 19 image pairs, but it produces higher average segmentation errors than CLSA2 for the other 12 image pairs, which further shows the ef-

fectiveness of the improved K-means*++ clustering algorithm. RCMSA outperforms the other six competing methods for the image pairs including a large number of model instances (i.e., the Unihouse, Bonhall and Johnsonb image pairs), but it fails in fitting the image pairs with unbalanced data (e.g., the Barrsmith, Elderhallb and Napierb image pairs).

4.6 Two-view based Motion Segmentation

For the task of two-view based motion segmentation, we use the other 19 image pairs of the AdelaideRMF dataset to evaluate the performance of the seven competing fitting methods. We show the average results in Table 3, and we also show some fitting results obtained by CLSA2 in Fig. 9.

From Fig. 9 and Table 3, we can see that RansaCov achieves high average segmentation errors and it fails to fit model instances for most cases. This is because that RansaCov considers all the generated model hypotheses, which include a large number of ineffective hypotheses. Note that the minimal subset for the two-view based motion segmentation task consists of 7 or 8 data points, and it is hard to sample an all-inlier minimal subset. Thus, it requires [sampling](#) a large number of minimal subsets to increase the probability of covering all model instances in data, but this will generate more ineffective model hypotheses as well. As the results show, the ineffective model hypotheses have significant influence on the fitting results obtained by RansaCov. Both T-linkage and RPA are able to achieve low segmentation errors for most of the image pairs, but their computational costs are relatively high, due to the use of the complicated clustering procedure. MSHF is able to achieve reasonably low average segmentation errors for 12 out of the 19 image pairs, and CLSA1 achieves reasonably low average segmentation errors for 8 out of the 19 image pairs. RCMSA is also able to achieve low average segmentation errors for most cases. Compared with the other competing methods, CLSA2 obtains the lowest segmentation errors for 16 out of the 19 image pairs and it obtains the lowest mean and median values of overall segmentation errors. In addition, CLSA2 is able to achieve very stable segmentation errors for all the 19 image pairs, when we repeat each experiment 50 times (note that each experiment uses different generated model hypotheses). This excellent performance mainly [results](#) from the steps of latent semantic space construction, outlier removal and inlier segmentation in CLSA2. Note that, CLSA1/CLSA2 are able to implement the process of model selection within 2 second for all the image pairs, which is much faster than the other five competing methods.

5 Discussion

5.1 The comparison between CLSA1 and CLSA2

In this paper, we proposed a basic approach with two identified variants. Both variants perform well - usually being the fastest and the most accurate, something uncommon in accuracy/cost tradeoffs. CLSA1 is the variant one would choose in general, if speed is a high priority, as it is usually faster and generally not so inferior in performance (still usually beating the competitors other than our slower variant). CLSA2 is one that would be recommended if speed is less of an issue and higher performance in segmentation has high priority.

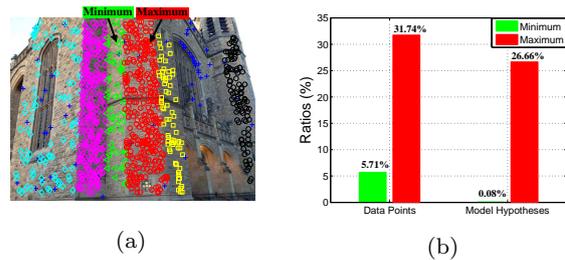


Fig. 10 An example showing a unbalanced data and the corresponding ratios of inliers and model hypotheses. (a) The minimum (in green) and maximum (in red) inlier cases of the “Bonhall” image pair. (b) The minimum and maximum ratios of inliers to all data points and the ratios of the generated model hypotheses corresponding to the two model instances in all generated model hypotheses.

Compared to the other competing methods, the proposed CLSA2 is more generalized, for handling the model fitting problem, and it only requires a few parameters (whose values do not significantly affect the accuracy of CLSA2). CLSA2 is also robust to different guided sampling algorithms and different numbers of the generated model hypotheses.

From the experimental results of line fitting based segmentation, we can see that CLSA2 is able to effectively deal with the data with different numbers of model instances, and a large number of outliers. From the experimental results of homography based segmentation and two-view based motion segmentation, CLSA2 also shows good performance compared to the other competing methods. Note that, CLSA2 shows significant advantages over the other competing methods for two-view based motion segmentation [over](#) homography based segmentation. One of the reasons is that, the task for two-view based motion segmentation is more challenging since it requires [dealing](#) with more ineffective model hypotheses. Moreover, we cannot ignore the fact that some fitting methods (e.g. MSHF) directly deal with the fitting task in the parameter space, and those methods are able to more effectively handle the data points located at the intersection of two structures. However, the fitting performance of those methods largely depends on the quality of the generated model hypotheses. Thus, the proposed CLSA2 achieves much better performance than those methods for two-view based motion segmentation.

Although the proposed CLSA2 significantly improves the performance of several state-of-the-art fitting methods on dealing with unbalanced data (that is, the numbers of inliers belonging to different model instances are unbalanced), it may not be effective to deal with heavily unbalanced data. We show an example of a unbalanced data in Fig. 10. We can see that, the minimum and maximum ratios of inliers belonging different model in-

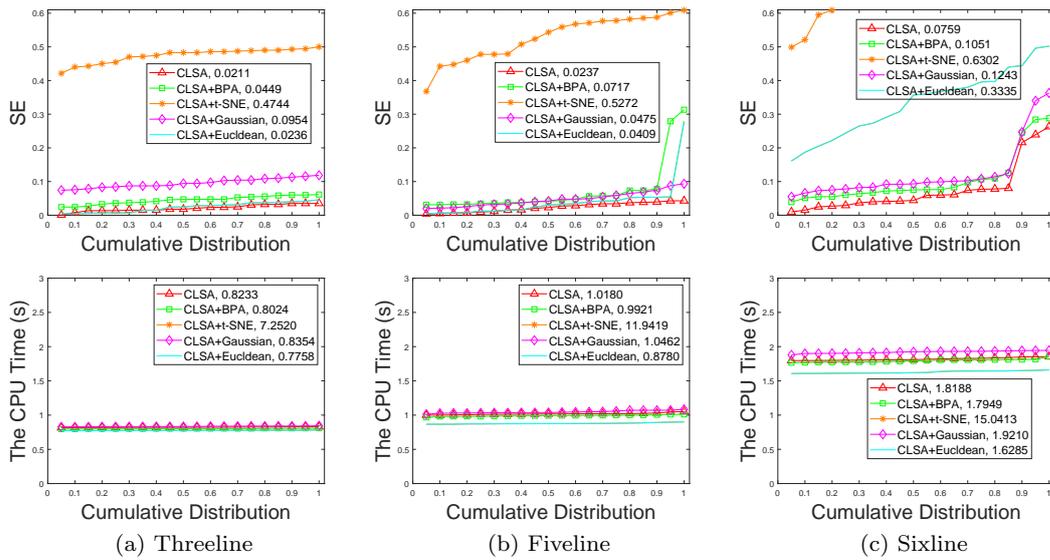


Fig. 11 Quantitative comparisons of different versions of the proposed method for line fitting based segmentation. In each subfigure, we show the segmentation errors obtained by and the CPU time used by the proposed methods with respect to the cumulative distribution. A point on the curve with coordinate (x, y) represents that there are $100 \times x$ percents of image pairs which have values no more than y .

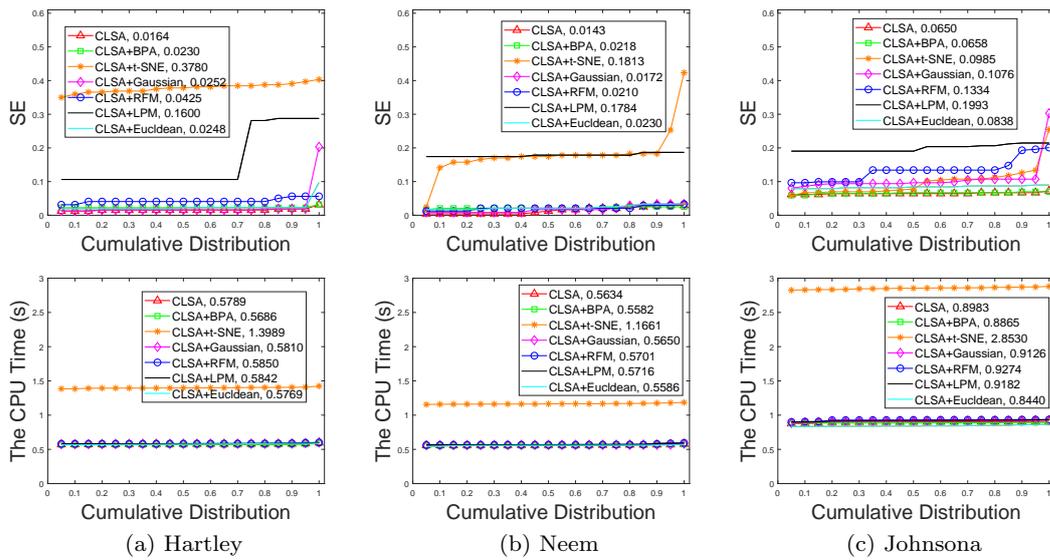


Fig. 12 Quantitative comparisons of different versions of the proposed method for homography based segmentation. In each subfigure, we show the segmentation errors obtained by and the CPU time used by the proposed methods with respect to the cumulative distribution.

stances to all data points for the “Bonhall” image pair are significantly different, and the ratios of the generated model hypotheses corresponding to the two model instances to all model hypotheses are much more different. In this case, the preference function (used in CLSA2) is not effective for the inliers of the model instance with the minimum inlier ratio. As a result, the proposed CLSA2 can not fit this data very well. However, all of the other competing fitting methods also have this limitation. Note that, sampling non-minimal sub-

sets may help alleviate the unbalance of the generated model hypotheses, as discussed in [34]. However, it is still a very challenging problem.

5.2 The Influence of different components of the proposed method

It is worth pointing that, in this paper, we argue that the proposed CLSA fitting method with the combina-

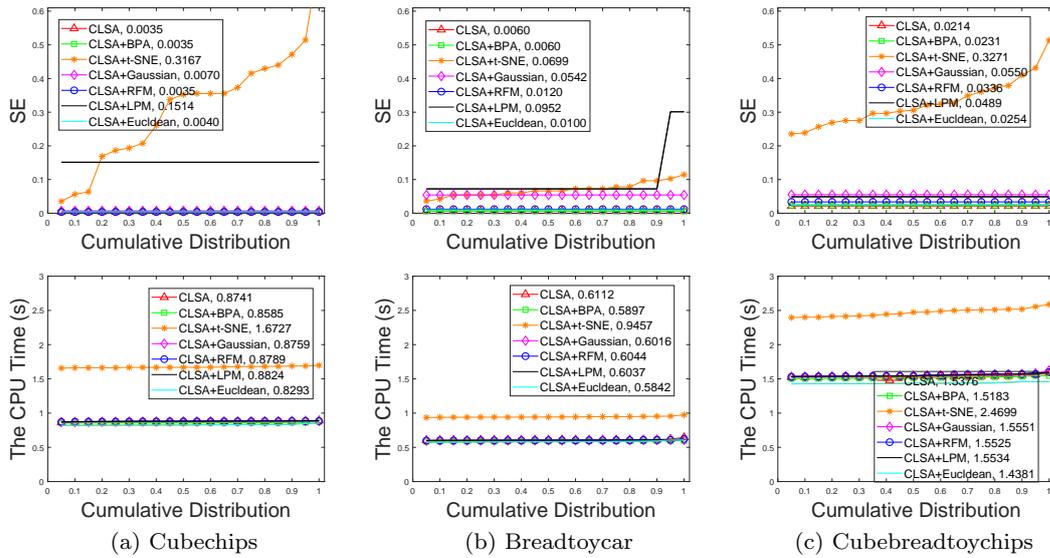


Fig. 13 Quantitative comparisons of different versions of the proposed method for two-view based motion segmentation. In each subfigure, we show the segmentation errors obtained by the proposed methods and the CPU time used by the proposed methods with respect to the cumulative distribution.

tion of latent semantic analysis (LSA) and continuous preference analysis (CPA) is an important improvement over that using either LSA or CPA. To verify this, we evaluate three different versions of CLSA, i.e., CLSA with LSA and CPA, CLSA+BNA with LSA and binary preference analysis (BNA), and CLSA+t-SNE with CPA and t-SNE [19], for line fitting based segmentation, homography based segmentation and two-view based motion segmentation. We also test different outlier removal strategies (i.e., 1D Mixture Gaussian Model [5], RFM [13] and LPM [18]), for the proposed method with CLSA+Gaussian, CLSA+RFM and CLSA+LPM, respectively (here, RFM and LPM only work for the two-view model fitting problem, thus, we test CLSA+RFM and CLSA+LPM only for homography based segmentation and two-view based motion segmentation). In addition, to compare the Tanimoto distance, we test a version (i.e. CLSA+Euclidean) with the Euclidean distance. To test different qualities of preference matrices, we generate 20 groups of model hypotheses for all versions of CLSA. We show the quantitative comparisons of different versions of CLSA in Figs. 11, 12 and 13 for line fitting based segmentation, homography based segmentation and two-view based motion segmentation, respectively.

From Figs. 11, 12 and 13, we can see that, CLSA is able to achieve better performance on the segmentation error than CLSA+BPA for most of the datasets, especially for the task of line fitting based segmentation (see Fig. 11), where the data include a large number of outliers (and they use the similar CPU time for all datasets). This shows the effectiveness of CPA for mod-

el fitting. For CLSA and CLSA+t-SNE, which deal with the preference matrix using LSA and t-SNE, respectively, CLSA is able to achieve lower segmentation errors with much less CPU time than CLSA+t-SNE, for all the three tasks (see Figs. 11, 12 and 13). This shows the effectiveness of LSA for model fitting. For different outlier removal strategies, CLSA achieves lower segmentation errors than CLSA+Gaussian, CLSA+RFM and CLSA+LPM for all the three tasks, while they spend the similar CPU time. This further shows the effectiveness of our proposed outlier removal strategy for model fitting. For different distance measures, CLSA is able to show the superiority over CLSA+Euclidean on the performance of the segmentation error for all the three tasks, while they also spend the similar CPU time. Thus, this shows the effectiveness of the Tanimoto distance for model fitting. For different qualities of preference matrices, CLSA is able to achieve stable fitting results for all the three tasks.

5.3 The estimation of the number of model instances

The estimation of the number of model instances in data is also an important and challenging task for model fitting. We note that if two model hypotheses correspond to the same model instance, they will share most of their inliers. Thus, we provide a promising solution for adaptively estimating the number of model instances in data by analyzing the inlier distribution and the quality of the estimated model instances. Specifically, we set the value of k in Algorithm 2 to be

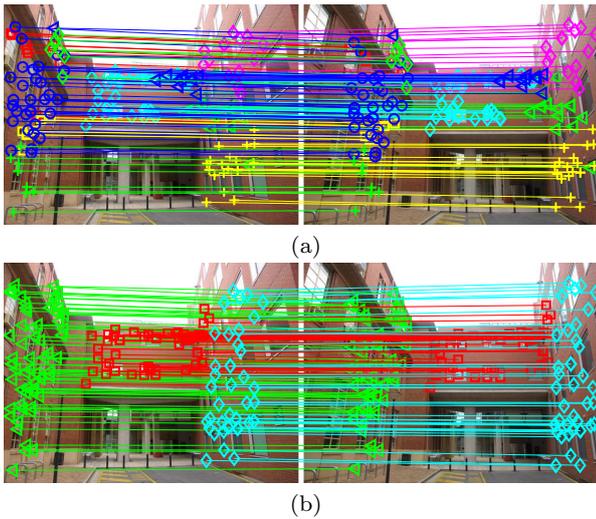


Fig. 14 An example of estimating the number of model instances on an image pair. (a) The over-segmented result when we input $k = 10$. (b) The segmentation result after estimating the number of model instances by using the proposed solution.

a reasonable large number so that the maximum number of model instances in data is less than k . In this way, it is much easier to set the k value than to require an user to input the truth number of model instances in data. After obtaining k model hypothesis candidates by Algorithm 2 (note that these model hypothesis candidates are significant model hypotheses since most of outliers have been removed in Step 4 of Algorithm 2, but the result may be over-segmented as k is usually larger than the true number of model instances in data), we compute the ratio of the common inliers shared by any two model hypothesis candidates, and group the two model hypothesis candidates into one cluster if the ratio of the common inliers shared by the two model hypothesis candidates is high (which means they are more likely to correspond to the same model instance). This process will fuse the model hypothesis candidates corresponding to the same model instance. Then, we estimate the number of model instances by counting the number of the model hypothesis groups. In Fig. 14, we show an example for the process of estimating the number of model instances. We can see that, the result in Fig. 14(a) is over-segmented when we input $k = 10$ (that is, we assume that the maximum number of model instances in the data is less than 10, which can be satisfied in most common cases). In contrast, we obtain the good segmentation result in Fig. 14(b), where we keep the model hypothesis candidate that has the best quality (measured by weighting scores [35]) in each cluster as the parameter estimation of the corresponding model instance, after estimating the number of model instances by the proposed solution.

6 Conclusion

In this paper, we propose a continuous latent semantic analysis based fitting method (called CLSA), which exploits the advantages of both latent semantic analysis (LSA) and continues preference analysis (CPA). LSA is able to significantly increase the efficiency and reduce the computational cost of the proposed CLSA method. CPA can be used by CLSA to effectively represent the relationship between model hypotheses and data points. By exploiting the advantages of both LSA and CPA, we construct a novel latent semantic space (LSS), where we map gross outliers close to the origin of LSS, while mapping inliers of different model instances to several independent directions in LSS. After that, we employ an information theoretic approach to remove the points corresponding to gross outliers in LSS. Then we treat the complex model fitting problem as a simple clustering problem (which is much easier to be dealt with because the data points corresponding to gross outliers are removed). We also propose an effective but simple clustering algorithm for inlier segmentation. Extensive experimental results show that the proposed CLSA method is able to achieve better performance than several state-of-the-art fitting methods on both segmentation accuracy and computational efficiency for different model fitting tasks.

References

1. Amayo, P., Pinis, P., Paz, L.M., Newman, P.: Geometric multi-model fitting with a convex relaxation algorithm. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8138–8146 (2018)
2. Arthur, D., Vassilvitskii, S.: K-means++: The advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms, pp. 1027–1035 (2007)
3. Barath, D., Matas, J., Noskova, J.: Magsac: Marginalizing sample consensus. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 10,197–10,205 (2019)
4. Barath, D., Noskova, J., Ivashechkin, M., Matas, J.: Magsac++, a fast, reliable and accurate robust estimator. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1304–1312 (2020)
5. Chin, T.J., Wang, H., Suter, D.: Robust fitting of multiple structures: The statistical learning approach. In: Proceedings of the IEEE international conference on computer vision, pp. 413–420 (2009)
6. Chin, T.J., Yu, J., Suter, D.: Accelerated hypothesis generation for multistructure data via preference analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(4), 625–638 (2012)
7. Chum, O., Matas, J.: Matching with prosac-progressive sample consensus. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 220–226 (2005)

8. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. *Journal of the American Society for Information Science* **41**(6), 391–407 (1990)
9. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. *IEEE Transactions on Pattern Analysis and Machine Intelligence* pp. 226–231 (1996)
10. Ferraz, L., Felip, R., Martínez, B., Binefa, X.: A density-based data reduction algorithm for robust estimators. In: *Proceedings of Iberian conference on pattern recognition and image analysis*, pp. 355–362 (2007)
11. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* **24**(6), 381–395 (1981)
12. Isack, H., Boykov, Y.: Energy-based geometric multi-model fitting. *International Journal of Computer Vision* **97**(2), 123–147 (2012)
13. Jiang, X., Ma, J., Jiang, J., Guo, X.: Robust feature matching using spatial clustering with heavy outliers. *IEEE Transactions on Image Processing* **29**, 736–746 (2020)
14. Kanazawa, Y., Kawakami, H.: Detection of planar regions with uncalibrated stereo using distributions of feature points. In: *Proceedings of British machine vision conference*, pp. 2701–2710 (2004)
15. Le, H.M., Chin, T.J., Eriksson, A., Do, T.T., Suter, D.: Deterministic approximate methods for maximum consensus robust fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **pp**(1), 1–14 (2019)
16. Lee, J.K., Yoon, K.J.: Joint estimation of camera orientation and vanishing points from an image sequence in a non-manhattan world. *International Journal of Computer Vision* **127**(10), 1426–1442 (2019)
17. Ma, J., Jiang, X., Fan, A., Jiang, J., Yan, J.: Image matching from handcrafted to deep features: A survey. *International Journal of Computer Vision* **129**(1), 23–79 (2021)
18. Ma, J., Zhao, J., Jiang, J., Zhou, H., Guo, X.: Locality preserving matching. *International Journal of Computer Vision* **127**(5), 512–531 (2019)
19. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. *Journal of Machine Learning Research* **9**(11), 2579–2605 (2008)
20. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, pp. 281–296 (1967)
21. Magri, L., Fusiello, A.: T-linkage: A continuous relaxation of j-linkage for multi-model fitting. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3954–3961 (2014)
22. Magri, L., Fusiello, A.: Multiple model fitting as a set coverage problem. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3318–3326 (2016)
23. Magri, L., Fusiello, A.: Multiple structure recovery via robust preference analysis. *Image and Vision Computing* **67**, 1–15 (2017)
24. Magri, L., Fusiello, A.: Fitting multiple heterogeneous models by multi-class cascaded t-linkage. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7460–7468 (2019)
25. Meer, P.: Robust estimation techniques. *Computer Vision: A Reference Guide* pp. 1–8 (2020)
26. Mittal, S., Anand, S., Meer, P.: Generalized projection-based m-estimator. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(12), 2351–2364 (2012)
27. Pham, T.T., Chin, T.J., Yu, J., Suter, D.: The random cluster model for robust geometric fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36**(8), 1658–1671 (2014)
28. Purkait, P., Chin, T.J., Ackermann, H., Suter, D.: Clustering with hypergraphs: The case for large hyperedges. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**(9), 1697–1711 (2017)
29. Rodriguez, A., Laio, A.: Clustering by fast search and find of density peaks. *Science* **344**(6191), 1492–1496 (2014)
30. Tanimoto, T.: An elementary mathematical theory of classification and prediction. Internal technical report, IBM (1957)
31. Tennakoon, R., Bab-Hadiashar, A., Cao, Z., Hoseinezhad, R., Suter, D.: Robust model fitting using higher than minimal subset sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **38**(2), 350–362 (2016)
32. Toldo, R., Fusiello, A.: Robust multiple structures estimation with j-linkage. In: *Proceedings of European conference on computer vision*, pp. 537–547 (2008)
33. Torr, P.H., Murray, D.W.: The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision* **24**(3), 271–300 (1997)
34. Tran, Q.H., Chin, T.J., Chojnacki, W., Suter, D.: Sampling minimal subsets with large spans for robust estimation. *International Journal of Computer Vision* **106**(1), 93–112 (2014)
35. Wang, H., Chin, T.J., Suter, D.: Simultaneously fitting and segmenting multiple-structure data with outliers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(6), 1177–1192 (2012)
36. Wang, H., Xiao, G., Yan, Y., Suter, D.: Searching for representative modes on hypergraphs for robust geometric model fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **41**(3), 697–711 (2019)
37. Wong, H.S., Chin, T.J., Yu, J., Suter, D.: Dynamic and hierarchical multi-structure geometric model fitting. In: *Proceedings of the IEEE international conference on computer vision*, pp. 1044–1051 (2011)
38. Wu, Y., Li, Y., Hu, Z.: Detecting and handling unreliable points for camera parameter estimation. *International Journal of Computer Vision* **79**(2), 209–223 (2008)
39. Xiao, G., Wang, H., Lai, T., Suter, D.: Hypergraph modelling for geometric model fitting. *Pattern Recognition* **60**(1), 748–760 (2016)
40. Xiao, G., Wang, H., Yan, Y., Suter, D.: Superpixel-guided two-view deterministic geometric model fitting. *International Journal of Computer Vision* **127**(4), 323–329 (2019)
41. Xu, L., Oja, E., Kultanen, P.: A new curve detection method: randomized hough transform (rht). *Pattern Recognition Letter* **11**(5), 331–338 (1990)
42. Xu, X., Cheong, L.F., Li, Z.: 3d rigid motion segmentation with mixed and unknown number of models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **pp**(1), 1–14 (2019)
43. Yang, X., Meer, P.M.J.: A new approach to robust estimation of parametric structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **pp**(1), 1–14 (2020)