Incremental Top-k List Comparison Approach to Robust Multi-Structure Model Fitting

Hoi Sim Wong, Tat-Jun Chin, Jin Yu and David Suter

School of Computer Science, The University of Adelaide, South Australia, 5005

Abstract

Random hypothesis sampling lies at the core of many popular robust fitting techniques such as RANSAC. In this paper, we propose a novel hypothesis sampling scheme based on incremental computation of distances between partial rankings (top-k lists) derived from residual sorting information. Our method simultaneously (1) guides the sampling such that hypotheses corresponding to all true structures can be quickly retrieved and (2) filters the hypotheses such that only a small but very promising subset remain. This permits the usage of simple agglomerative clustering on the surviving hypotheses for accurate model selection. The outcome is a highly efficient multi-structure robust estimation technique. Experiments on synthetic and real data show the superior performance of our approach over previous methods.

Keywords: Model fitting, guided sampling, multi-structure data, top-k list

1. Introduction

Robust model fitting techniques play an integral role in computer vision since the observations or measurements are frequently contaminated with outliers. Major applications include the estimation of various projective entities from multi-view data [1] which often contain false correspondences. At the core of many robust techniques is random hypothesis generation, i.e., iteratively generate many hypotheses of the geometric model from randomly sampled minimal subsets of the data. The hypotheses are then scored according to a robust criterion (e.g., RANSAC [2]) or clustered (e.g., Mean Shift [3]) to find the most promising model(s). Success rests upon retrieving
Figure 1: (a) Input data with 5 structures (lines) with 100 points per structure and 250 gross outliers. The inlier scale is 0.01. (b) 500 hypotheses are generated with the proposed multi-structure guided sampling scheme and simultaneous hypothesis filtering, producing 180 good hypotheses as shown in the figure. (c) Simple agglomerative clustering of the remaining 180 hypotheses gives the final fitting result.

This paper addresses two major issues affecting the current paradigm of robust estimation. The first is that hypothesis generation tends to be time consuming for heavily contaminated data. Previous methods attempted to improve sampling efficiency by guiding the sampling such that the probability of selecting all-inlier minimal subsets is increased. These methods often depend on assumptions or domain knowledge of the data, e.g., inliers have higher keypoint matching scores [4, 5] or are correspondences that respect local geometry patterns [6]. Most methods, however, are not optimized for data with multiple instances (or structures [7]) of the geometric model. This is because they sample based on estimated inlier probabilities alone while ignoring the fact that only inliers from the same structure should be included in the same minimal subset. Such methods may inefficiently generate a large number of samples before obtaining an all-inlier minimal subset for each genuine structure in the data.

The second crucial issue is the lack of a principled approach to fit the multiple structures in the data. Many previous works [8, 9] simply apply RANSAC sequentially, i.e., fit one structure, remove corresponding inliers, then repeat. This is risky because inaccuracies in the initial fits will be amplified in the subsequent fits [10]. Moreover, finding a stopping criterion for sequential fitting that accurately reflects the true number of structures
is non-trivial. Methods based on clustering [11] or mode detection [8] [12] given the generated hypotheses are not affected by the dangers of sequential fitting. However, if there are insufficient hypotheses corresponding to the true structures, the genuine clusters will easily be overwhelmed by the irrelevant hypotheses. Consequently, these methods often miss the true structures or find spurious structures.

The inability to retrieve “good” hypotheses at sufficiently large quantities represents the fundamental obstacle to the satisfactory performance of previous methods. To address this limitation, we propose a novel hypothesis sampling scheme based on incremental computation of distances between partial rankings or top-k lists [13] derived from residual sorting information. Our approach enhances hypothesis generation in two ways: (1) The computed distances guide the sampling such that inliers from a single coherent structure are more likely to be simultaneously selected. This dramatically improves the chances of hitting all-inlier minimal subsets for each structure in the data. (2) The qualities of the generated hypotheses are evaluated based on the computed distances. This permits an on-the-fly filtering scheme to reject “bad” hypotheses. The outcome is a set of only the most promising hypotheses which facilitate a simple agglomerative clustering step to fit all the genuine structures in the data. Fig. 1 summarizes the proposed approach.

The rest of the paper is organized as follows: Sec. 2 describes how to derive data similarities from residual sorting information by comparing top-k lists. Sec. 3 describes our guided sampling scheme with simultaneous hypothesis filtering and incremental computations of distances between top-k lists. Sec. 4 describes how multi-structure fitting can be done by a simple agglomerative clustering on the promising hypotheses returned by our method. Sec. 5 presents results on synthetic and real data which validate our approach. Finally, we draw conclusions in Sec. 6.

2. Data Similarity by Comparing Top-k Lists

A key ingredient of our guided sampling scheme is a data similarity measure. This section describes how to derive such a measure from residual sorting information.

2.1. Top-k Lists from Residual Sorting Information

We measure the similarity between two input data based on the idea that if they are inliers from the same structure, then their preferences to the
hypotheses as measured by residuals will be similar. Such preferences can be effectively captured by lists of ranked residuals.

Let $X = \{x_i\}_{i=1}^N$ be a set of $N$ input data and $\Theta = \{\theta_j\}_{j=1}^M$ a set of $M$ hypotheses, where each hypothesis $\theta_j$ is fitted from a minimal subset of $p$ data (e.g., $p=2$ for line fitting). For each datum $x_i$, we compute its absolute residual as measured to $M$ hypotheses to form a residual vector

$$r_i = [r_1^{(i)}, r_2^{(i)}, \ldots, r_M^{(i)}].$$  \hspace{1cm} (1)

We sort the elements in $r_i$ to obtain a sorted residual vector

$$\tilde{r}_i = [r_{\lambda_1}^{(i)}, \ldots, r_{\lambda_M}^{(i)}]$$  \hspace{1cm} (2)

such that $r_{\lambda_1}^{(i)} \leq \cdots \leq r_{\lambda_M}^{(i)}$. The permutation $[\lambda_1^{(i)}, \ldots, \lambda_M^{(i)}]$ encapsulate the data preference of $x_i$ to the hypotheses, i.e., $x_i$ is more likely to be an inlier to the hypotheses which have higher rank.

The top-$k$ list of data $x_i$ is defined as the first $k$ elements in the permutation $[\lambda_1^{(i)}, \ldots, \lambda_M^{(i)}]$, i.e.,

$$\tau_i = [\lambda_1^{(i)}, \ldots, \lambda_k^{(i)}].$$  \hspace{1cm} (3)

The top-$k$ list $\tau_i$ essentially gives the top-$k$ hypotheses preferred by $x_i$.

Fig 2 illustrate our idea to measure the data similarity using data preferences to the hypotheses. For data $x_1$ and $x_2$, they are inliers from the same structure and their corresponding top-10 lists $\tau_1$ and $\tau_2$ are similar, e.g., hypotheses 6,7 and 8 are highly ranked. For two inliers $x_1$ and $x_3$ from different structure, their corresponding top-10 lists $\tau_1$ and $\tau_3$ are inconsistent, e.g., hypotheses 2 and 4 are highly ranked by $x_3$ but not by $x_1$.

2.2. The Spearman Footrule Distance

Given the top-$k$ lists, we measure their similarity using the Spearman Footrule (SF) distance [13]. Let $\tau$ be a top-$k$ list and $D_\tau$ be a set of elements contained in $\tau$. Denote the position of the element $m \in D_\tau$ in $\tau$ by $\tau(m)$. The SF distance between two top-$k$ lists $\tau_i$ and $\tau_{i'}$ is defined as

$$F^{(\ell)}(\tau_i, \tau_{i'}) = \sum_{m \in D_{\tau_i} \cup D_{\tau_{i'}}} |\tau_i'(m) - \tau_{i'}'(m)|,$$  \hspace{1cm} (4)

where $\ell > 0$ is the so-called location parameter (often set to $k + 1$), $\tau_i'(m) = \tau_i(m)$ if $m \in D_{\tau_i}$; otherwise $\tau_i'(m) = \ell$, and $\tau_{i'}'$ is similarly obtained from $\tau_{i'}$.  

4
2.3. Measuring Similarity between Data

To measure the similarity between two data, we use the SF distance (Eq. 4) between their corresponding top-k lists. The similarity value between two data \( x_i \) and \( x_{i'} \) is defined as

\[
d(\tau_i, \tau_{i'}) = 1 - \frac{1}{k} \times \ell F^{(\ell)}(\tau_i, \tau_{i'}). \tag{5}
\]

Note that we normalize \( F^{(\ell)}(\tau_i, \tau_{i'}) \) such that \( d(\tau_i, \tau_{i'}) \) is between 0 (dissimilar) and 1 (identical). By comparing the top-k lists between all data, we obtain a \( N \times N \) similarity matrix \( K \) with

\[
K(i, i') = d(\tau_i, \tau_{i'}), \tag{6}
\]

where \( K(i, i') \) denotes the element at its \( i \)-th row and \( i' \)-th column.

Fig. 3(a) shows an example of \( K \), which is generated from the input data shown in Fig. 1(a). The evident block structures in \( K \) correspond to the 5 lines in Fig. 1(a). As shown in Fig. 3(b) across various \( k \), the average of similarity values between two inliers from the same structure (SS) is higher than that from different structure (DS). It supports the observation...
Figure 3: (a) Similarity matrix $K$ for data shown in Fig. 1(a) (data is arranged according to structure membership for representation only) (b) Under various $k$, the average of similarity values between two inliers from the same structure (SS), two inliers from different structures (DS), an inlier and a gross outlier (IO), two gross outliers (OO).

from our given example (Fig. 2) that inliers from the same structure gives consistent preferences to the hypotheses. From Fig. 3(b), it also shows that the inconsistent preferences to hypotheses occur between an inlier/a gross outlier(IO) and two gross outliers (OO), hence, their corresponding averages of similarity values are smaller.

3. Guided Sampling with Hypothesis Filtering

This section describes our guided sampling scheme which involves a simultaneous hypothesis filtering scheme. We also provide an efficient incremental update for computing the sampling weights. Furthermore, we provide an improvement on our guided sampling scheme based on the result from our hypothesis filtering scheme.

3.1. Guided Sampling

We use the similarity matrix $K$ (Eq. 6) to sample data in a guided fashion. Let $Q = \{s_u\}_{u=1}^p$ be the indices of data in a minimal subset of size $p$, where $s_u$ are indexed by the order in which they are sampled. The first element $s_1$ in $Q$ is randomly selected from $X$. To sample the next element $s_2$, we use $K(s_1,:)$ as the weight to guide the sampling, i.e., the similarity values of all input data with respect to $s_1$. We set $K(s_1,s_1)$ to 0 to avoid sampling the same data again. Fig. 4 shows the example of such sampling weights.
Suppose data \( s_1, \ldots, s_u \) have been selected, then the next datum \( s_{u+1} \) is chosen conditionally on the selected data. Its sampling weight is defined as

\[
K'(s_1,:) \cdot K'(s_2,:) \cdot \ldots \cdot K'(s_u,:),
\]

where \( \cdot \) is the element-wise multiplication and \( K'(s_u,:) \) is just \( K(s_u,:) \) with \( K(s_u, s_u) = 0 \). Eq. 7 means that in order to have higher probabilities of being sampled, a datum need to be similar (measured by Eq. 5) to all the data that have been selected into the minimal subset.

### 3.2. Incremental Top-k Lists Comparison

Our sampling method computes an update to the similarity matrix \( K \) (Eq. 6) once a block (of size \( b \)) of new hypotheses are generated. This involves comparing top-\( k \) lists of ranked residuals. The computation of top-\( k \) lists can be done efficiently via merge sort. However, comparing top-\( k \) lists between all data, i.e., constructing \( K \), can be computationally expensive. Here we provide efficient incremental updates for \( K \) that can substantially accelerate the computation.

As proved in [13], the SF distance (Eq. 4) can be equivalently computed as

\[
F^{(\ell)}(\tau_i, \tau_j) = 2(k - |Z|)\ell + \sum_{m \in Z} |\tau_i(m) - \tau_j(m)| - \sum_{m \in S} \tau_i(m) - \sum_{m \in T} \tau_j(m),
\]

(8)
where $Z = D_{\tau_i} \cap D_{\tau_j}$, $S = D_{\tau_i} \backslash D_{\tau_j}$, and $T = D_{\tau_j} \backslash D_{\tau_i}$. In fact, $S$ is simply the elements in $D_{\tau_i}$ but not in $Z$, i.e., $S = D_{\tau_i} \backslash Z$, similarly for $T$. Hence, we have

$$\sum_{m \in S} \tau_i(m) = \sum_{m=1}^{k} m - \sum_{m \in Z} \tau_i(m) = \frac{1}{2} k(k + 1) - \sum_{m \in Z} \tau_i(m), \quad (9)$$

similarly for $\sum_{m \in T} \tau_j(m)$. By setting $\ell = k + 1$ and using Eq. 9 we can rewrite Eq. 8 to be in terms of $Z$ only,

$$F^{(k+1)}(\tau_i, \tau_j) = (k + 1)(k - 2|Z|) + \sum_{m \in Z} (|\tau_i(m) - \tau_j(m)| + \tau_i(m) + \tau_j(m)). \quad (10)$$

Let $A$ and $B$ be two $N \times N$ symmetric matrices and set the elements at the $i$-th row and the $j$-th column of $A$ and $B$ to

$$A(i, j) = |Z| \quad \text{and} \quad B(i, j) = \sum_{m \in Z} (|\tau_i(m) - \tau_j(m)| + \tau_i(m) + \tau_j(m)). \quad (11)$$

From Equations 5, 6, and 10 the similarity matrix $K$ can be constructed by

$$K = 1 - \frac{1}{k} (k - 2A) - \frac{1}{k(k + 1)} B, \quad (12)$$

where $I_N$ is an $N \times N$ identity matrix. Observe from Eq. 11 that the matrices $A$ and $B$ can be efficiently updated by keeping track of the elements that move into or out of $Z$. This information is readily available from the merge sort. Once $A$ and $B$ are updated, $K$ can be updated via Eq. 12.

### 3.3. Simultaneous Hypothesis Filtering

During sampling, we want to simultaneously filter hypotheses such that only a small but very promising subset remains. We make use the data preferences and hypothesis preferences to identify the good hypotheses.

Similar to the definition of top-$k$ hypotheses in Eq. 3 we define the top-$h$ data of a hypotheses by sorting the residual information. For each hypothesis $\theta_j$, we sort its absolute residual $r_j = [r_{1}^{(j)}, r_{2}^{(j)}, \ldots, r_{N}^{(j)}]$ as measured to $N$ data and its sorted residual vector $\tilde{r}_j = [r_{\pi_{1}^{(j)}}^{(j)}, \ldots, r_{\pi_{N}^{(j)}}^{(j)}]$ such that $r_{\pi_{1}^{(j)}}^{(j)} \leq \cdots \leq$
The top-\(h\) data of hypothesis \(\theta_j\) is defined as the first \(h\) elements in the permutation \([\pi_1^{(j)}, \ldots, \pi_N^{(j)}]\), i.e.,

\[
\sigma_j = [\pi_1^{(j)}, \ldots, \pi_h^{(j)}].
\] (13)

The top-\(h\) data \(\sigma_j\) gives the \(h\) data preferred by the hypothesis \(\sigma_j\) to be its inliers, i.e., the higher \(x_i\) is ranked, the more likely \(x_i\) is the inlier to it. The value of \(h\) is conservatively set to 0.1 in all experiments. The assumption is that at least 10\% of data are inliers.

For each hypothesis \(\theta_j\), we construct the feature vector using the data and hypothesis preferences.

\[
f_j = \left[ f_j^{(1)}, f_j^{(2)} \right] = \left[ \frac{\sum_{(i,i') \in E} K(i,i')}{|E|}, \frac{\sum_{(i,i') \in \Sigma} K(i,i')}{|\Sigma|} \right],
\] (14)

where \(E = \{(i,i')| i \neq i' \text{ and } x_i, x_{i'} \in \Omega_j\}\) with \(\Omega_j = \{x_i \in X | m \in \tau_i\}\), \(\Sigma = \{(i,i')| i \neq i' \text{ and } i,i' \in \sigma_j\}\) and \(K\) is the similarity matrix computed by Eq. 12. The set \(\Omega_j\) contains all data that include the hypothesis \(j\) in their top-\(k\) lists. If the hypothesis \(j\) is “good”, then \(\Omega_j\) should contain many inliers from a structure. Hence, \(f_j^{(1)}\), the average of similarity values between all data in \(\Omega_m\) should be high. Moreover, the top-\(h\) data of hypotheses should contain data which are similar to each other (from the same structure), the average of similarity values between its top-\(h\) data should be high, i.e., high \(f_j^{(2)}\). Therefore, we want to find a set of hypotheses which have high value in both \(f_m^{(1)}\) and \(f_j^{(2)}\). To this end, we apply k-means on the feature vectors (Eq. 14) to separate “good” and “bad” hypotheses. As illustrated in Fig. 5(a), the cluster whose center has larger norm (circles) contains good hypotheses. We incrementally maintain a set of “good” hypotheses as the guided sampling proceeds.

3.4. Selection of the First Datum in Minimal Subset

The guided sampling scheme in Sec. 3.1 can be further improved by the result from the hypothesis filtering scheme.

In Sec. 3.1, the first element in the minimal subset is selected randomly. The subsequent sampling the data of the minimal subset becomes unprofitable if the first element is a gross outlier. Here, we can make use the result from hypothesis filtering to increase the probability of obtaining an
Figure 5: Hypothesis filtering: (Left) Feature space where x axis is $f_m^{(1)}$ and y axis is $f_m^{(2)}$ (Best view in color). (b) An example of “good” (denoted “g”) and “bad” (denoted “b”) hypotheses.

inlier. Note that if a hypothesis is “good”, its minimal subset should mainly contain inliers. Hence, we simply select the first element from the minimal subsets of the “good” hypotheses.

4. Multi-Structure Fitting

By leveraging the simultaneous hypothesis filtering, a set of “good” hypotheses is immediately available once the sampling is done. This set of “good” hypotheses allows us to easily cluster the hypotheses using a simple agglomerative clustering method.

We use the agglomerative clustering (See [14] for a detailed description) to cluster the hypotheses. For each hypothesis $\theta_j$, we aggregate the similarity of its top-$h$ data as measured to $N$ data and represent each hypothesis $\theta_j$ by a $1 \times N$ feature vector

$$\alpha_j = \sum_{i \in \sigma_j} K(i,:). \quad (15)$$

The distance between two hypotheses $\theta_j$ and $\theta_{j'}$ is given by

$$d(\theta_j, \theta_{j'}) = \| \alpha_j - \alpha_{j'} \|_2, \quad (16)$$
where $\|\cdot\|_2$ denotes the $L_2$ norm. The $d(\theta_j, \theta_j')$ is smaller if two hypotheses $\theta_j$ and $\theta_j'$ is similar, i.e., explaining the same structure. Using this distance measure, the clustering is then performed through the standard agglomerative clustering mechanism.

Each cluster of hypotheses contains the hypotheses overlapping on the same structure. For each cluster of hypotheses, we can simply select the hypothesis with the minimum of sum of squared residuals over its top-$h$ data.

5. Experiments

We test the proposed method on homography and fundamental matrix estimation using real data. To evaluate the efficiency of the proposed guided sampling scheme, we compare our method against 6 sampling techniques: Uniform random sampling in RANSAC (Random) [2], proximity sampling (Proximity) [9, 11], LO-RANSAC(LRANSAC) [15], Guided-MLESAC (GMLESAC) [4] and PROSAC [5]. Our proposed method is denoted by ITKSF (Sec. 3.1) and its extension is denoted by ITKSF-S (Sec. 3.4).

In all experiments, we fix $b = 100$ and $k = \lceil 0.1 \times t \rceil$ throughout, $b$ being the block size (cf Sec. 3.2) and $t$ being the number of hypotheses generated so far. All experiments are run on a machine with 2.53GHz Intel Core 2 Duo processor and 4GB RAM.

5.1. Data Set: AdelaideRMF

We created a data set for robust model fitting, called AdelaideRMF. It contains a collection of image pairs for homography and fundamental matrix estimation on single and multi-structure data. For each image pair, we use SIFT [16] to obtain the keypoint correspondences and manually labelled each keypoint correspondence.

5.2. Homography Estimation

This experiment involves estimating multiple planar homographies. The data used for this experiment is shown in Fig. 6. We use 4 correspondences to estimate a homography using Direct Linear Transformation [1]. Each method is given 50 random runs, each for 5 CPU seconds.

---

1 AdelaideRMF is publicly available from [http://cs.adelaide.edu.au/~hwong/doku.php?id=data](http://cs.adelaide.edu.au/~hwong/doku.php?id=data)
Table 1 shows the performance of guided sampling methods and also the hypothesis filtering result from our proposed method.

Figure 6: Data for homography estimation. Red crosses indicate the gross outliers and other colored markers indicate the structure membership. (Best viewed in color)
Table 1: Performance of guided sampling methods in homography estimation. We record the number of hypotheses generated (M), the number of all-inlier minimal subset found on each structure (Structures) and the percentage of all-inlier minimal subsets (IS) found within the time budget. We also record the CPU time to hit at least one all-inlier minimal subset on each structure (HIT) which is penalized by the time budget if a method fails. For ITKSF and ITKSF-S, the hypothesis filtering result is shown in bracket. The results represent the average over 50 runs with the best result boldfaced.

| Data          | Method | M     | HIT(s) | Structures       | IS(%)     |
|---------------|--------|-------|--------|------------------|-----------|
| **Union**     |        |       |        |                  |           |
|               | Random | 2465  | 0.74   | [7]              | 0.29      |
|               | Proximity | 2296  | 0.51   | [10]             | 0.45      |
|               | LRANSAC | 2426  | 0.83   | [27]             | 1.10      |
|               | GMLESAC | 2409  | 0.17   | [29]             | 1.20      |
|               | PROSAC  | 2400  | **0.00** | [80]         | 3.34      |
|               | Multi-GS | 835   | 0.27   | [51]             | 6.05      |
|               | ITKSF   | 1546(231) | 0.37 | [206(170)]     | 13.42(74.94) |
|               | ITKSF-S | 1539(258) | 0.32 | [386(174)]      | **24.87(67.75)** |
| **Symon**     |        |       |        |                  |           |
|               | Random | 2532  | 0.87   | [108,5]         | 4.47      |
|               | Proximity | 2363  | **0.34** | [162,12]    | 7.34      |
|               | LRANSAC | 2490  | 1.09   | [141,5]         | 5.88      |
|               | GMLESAC | 2473  | 0.77   | [264,5]         | 10.88     |
|               | PROSAC  | 2478  | 3.27   | **722.3**       | 29.26     |
|               | Multi-GS | 997   | **0.34** | [253,48]     | 30.22     |
|               | ITKSF   | 1738(148) | 0.37 | [513(30),125(42)] | 36.71(52.86) |
|               | ITKSF-S | 1737(131) | 0.35 | [594(15),156(64)] | **43.18(52.86)** |
| **Hartley**   |        |       |        |                  |           |
|               | Random | 2458  | 4.72   | [15,0]          | 0.63      |
|               | Proximity | 2375  | 3.67   | [23,1]          | 1.06      |
|               | LRANSAC | 2411  | 4.76   | [34,0]          | 1.42      |
|               | GMLESAC | 2395  | 3.12   | [45,1]          | 1.93      |
|               | PROSAC  | 2399  | **0.29** | [122,3]      | 5.23      |
|               | Multi-GS | 846   | 1.08   | [101,5]         | 12.52     |
|               | ITKSF   | 1592(286) | 0.69 | [272(175),31(14)] | 19.19(66.42) |
|               | ITKSF-S | 1583(346) | 0.57 | [380(181),52(29)] | **27.12(60.95)** |
| **NEEM**      |        |       |        |                  |           |
|               | Random | 2512  | 2.70   | [11,2,3]        | 0.64      |
|               | Proximity | 2339  | 1.40   | [37,4,9]        | 2.13      |
|               | LRANSAC | 2470  | 2.94   | [20,3,4]        | 1.10      |
|               | GMLESAC | 2454  | 1.21   | [22,6,5]        | 1.36      |
|               | PROSAC  | 2469  | 1.03   | [57,23,13]      | 3.76      |
|               | Multi-GS | 1011  | 0.56   | [110,26,56]     | 18.91     |
|               | ITKSF   | 1717(334) | 0.61 | [251(128),79(16),153(115)] | 28.14(77.71) |
|               | ITKSF-S | 1704(334) | **0.54** | [300(116),100(21),187(131)] | **34.42(80.00)** |
| **Johnson**   |        |       |        |                  |           |
|               | Random | 2436  | 3.55   | [4,9,2,1]       | 0.68      |
|               | Proximity | 2261  | 1.55   | [11,15,5,5]     | 1.61      |
|               | LRANSAC | 2386  | 3.54   | [4,18,3,2]      | 1.13      |
|               | GMLESAC | 2373  | 3.18   | [6,12,2,2]      | 0.91      |
|               | PROSAC  | 2382  | 3.97   | [7,29,1,4]      | 1.73      |
|               | Multi-GS | 805   | **0.50** | [53,71,16,31] | 21.15     |
|               | ITKSF   | 1522(434) | 0.57 | [162(84),211(129),57(7),100(83)] | 34.83(68.71) |
|               | ITKSF-S | 1507(385) | **0.50** | [207(76),271(113),63(7),128(96)] | **44.37(74.47)** |
5.3. Fundamental Matrix Estimation

We also evaluate the proposed method on fundamental matrix estimation. The data used for this experiment is shown in Fig. 7. We use the 8-point algorithm \[^2\] to estimate a fundamental matrix. Each method is given 50 random runs and each for 10 CPU seconds.

Table 2 shows the performance of guided sampling methods and also the hypothesis filtering result from our proposed method.

\[^2\]http://www.robots.ox.ac.uk/~vgg/hzbook/code/
Table 2: Performance of guided sampling methods in fundamental matrix estimation. The notation is same as Table 1.

| Data   | Method     | M  | HIT(s) | Structures | IS(%) |
|--------|------------|----|--------|------------|-------|
|        |            |    |        |            |       |
|        | Random     | 5083 | 8.19  | [1]        | 0.01  |
|        | Proximity  | 4741 | 3.29  | [3]        | 0.06  |
|        | LRANSAC    | 4950 | 8.28  | [1]        | 0.01  |
|        | GMLESAC    | 4960 | 5.08  | [2]        | 0.03  |
|        | PROSAC     | 4904 | 1.73  | [3]        | 0.05  |
|        | Multi-GS   | 981  | 0.30  | [134]      | 13.68 |
|        | ITKSF      | 2903(654) | 0.34 | [483(338)] | 16.63(51.98) |
|        | ITKSF-S    | 2896(636) | 0.37 | [532(334)] | 18.36(52.74) |
| Cube   |            |    |        |            |       |
| (Fig. 7(a)) | Random     | 6130 | 9.41  | [0,6]      | 0.09  |
|        | Proximity  | 5640 | 3.06  | [4,52]     | 1.00  |
|        | LRANSAC    | 5926 | 9.74  | [0,8]      | 0.13  |
|        | GMLESAC    | 5973 | 8.66  | [0,32]     | 0.54  |
|        | PROSAC     | 5855 | 8.46  | [0,28]     | 0.48  |
|        | Multi-GS   | 1108 | 0.24  | [129,24]   | 33.69 |
|        | ITKSF      | 3402(545) | 0.32 | [506(220),867(144)] | 40.36(69.48) |
|        | ITKSF-S    | 3393(497) | 0.31 | [310,534](188),933(153) | 43.34(66.88) |
| Bread  |            |    |        |            |       |
| (Fig. 7(b)) | Random     | 8277 | 10.00 | [0,0,1]    | 0.01  |
|        | Proximity  | 7449 | 10.00 | [0,0,11]   | 0.16  |
|        | LRANSAC    | 7903 | 10.00 | [0,0,1]    | 0.01  |
|        | GMLESAC    | 8026 | 10.00 | [0,0,11]   | 0.14  |
|        | PROSAC     | 7782 | 10.00 | [0,0,6]    | 0.08  |
|        | Multi-GS   | 1382 | 0.92  | [18,81,28] | 27.56 |
|        | ITKSF      | 4212(520) | 0.77 | [157(47),319(12),996(361)] | 34.95(81.08) |
|        | ITKSF-S    | 4198(531) | 0.91 | [96(30),374(25),1270(369)] | 41.45(80.16) |
| Car    |            |    |        |            |       |
| (Fig. 7(c)) | Random     | 6361 | 10.00 | [0,0,0]    | 0.00  |
|        | Proximity  | 5836 | 10.00 | [0,1,0]    | 0.02  |
|        | LRANSAC    | 6141 | 10.00 | [0,0,0]    | 0.01  |
|        | GMLESAC    | 6180 | 10.00 | [0,1,0]    | 0.02  |
|        | PROSAC     | 6065 | 10.00 | [0,1,0]    | 0.02  |
|        | Multi-GS   | 1142 | 0.64  | [37,13,127] | 26.17 |
|        | ITKSF      | 3405(812) | 0.49 | [184(63),521(263),504(187)] | 35.53(65.55) |
|        | ITKSF-S    | 3395(736) | 0.49 | [190(61),534(238),546(157)] | 37.42(64.82) |
| Chips  |            |    |        |            |       |
| (Fig. 7(d)) | Random     | 6361 | 10.00 | [0,0,0]    | 0.00  |
|        | Proximity  | 5836 | 10.00 | [0,1,0]    | 0.02  |
|        | LRANSAC    | 6141 | 10.00 | [0,0,0]    | 0.01  |
|        | GMLESAC    | 6180 | 10.00 | [0,1,0]    | 0.02  |
|        | PROSAC     | 6065 | 10.00 | [0,1,0]    | 0.02  |
|        | Multi-GS   | 1142 | 0.64  | [37,13,127] | 26.17 |
|        | ITKSF      | 3405(812) | 0.49 | [184(63),521(263),504(187)] | 35.53(65.55) |
|        | ITKSF-S    | 3395(736) | 0.49 | [190(61),534(238),546(157)] | 37.42(64.82) |
| ToysA  |            |    |        |            |       |
| (Fig. 7(e)) | Random     | 6202 | 10.00 | [0,0,0]    | 0.00  |
|        | Proximity  | 5706 | 10.00 | [0,0,0]    | 0.02  |
|        | LRANSAC    | 5990 | 10.00 | [0,0,0]    | 0.00  |
|        | GMLESAC    | 6040 | 10.00 | [0,0,0]    | 0.00  |
|        | PROSAC     | 5926 | 10.00 | [0,0,0]    | 0.00  |
|        | Multi-GS   | 1104 | 2.16  | [39,8,21,104] | 21.03 |
|        | ITKSF      | 3406(637) | 1.13 | [214(109),95(40),361(271),448(34)] | 32.84(71.78) |
|        | ITKSF-S    | 3399(664) | 1.56 | [288(132),72(32),403(273),485(29)] | 35.25(79.06) |
| ToysB  |            |    |        |            |       |
| (Fig. 7(f)) | Random     | 6202 | 10.00 | [0,0,0]    | 0.00  |
|        | Proximity  | 5706 | 10.00 | [0,0,0]    | 0.02  |
|        | LRANSAC    | 5990 | 10.00 | [0,0,0]    | 0.00  |
|        | GMLESAC    | 6040 | 10.00 | [0,0,0]    | 0.00  |
|        | PROSAC     | 5926 | 10.00 | [0,0,0]    | 0.00  |
|        | Multi-GS   | 1104 | 2.16  | [39,8,21,104] | 21.03 |
|        | ITKSF      | 3406(637) | 1.13 | [214(109),95(40),361(271),448(34)] | 32.84(71.78) |
|        | ITKSF-S    | 3399(664) | 1.56 | [288(132),72(32),403(273),485(29)] | 35.25(79.06) |
6. Conclusions

We propose a novel guided sampling scheme based on the distances between top-\(k\) lists that are derived from residual sorting information. In contrast to many existing sampling enhancement techniques, our method does not rely on any domain-specific knowledge, and is capable of handling multiple structures. Moreover, while performing sampling, our method simultaneously filters the hypotheses such that only a small but very promising subset remains. This permits the use of simple agglomerative clustering on the surviving hypotheses for accurate model selection. Experiments on synthetic and real data show the superior performance of our approach over previous methods.

References

[1] R. I. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, 2nd Edition, Cambridge University Press, 2004.

[2] M. A. Fischler, R. C. Bolles, Ransac: A paradigm for model fitting with applications to image analysis and automated cartography, Comm. of the ACM 24 (1981) 381–395.

[3] D. Comaniciu, P. Meer, Mean shift: A robust approach toward feature space analysis, IEEE Transactions on pattern analysis and machine intelligence 24 (5) (2002) 603–619.

[4] B. J. Tordoff, D. W. Murray, Guided-mlesac: Faster image transform estimation by using matching priors, TPAMI 27 (10) (2005) 1523–1535.

[5] O. Chum, J. Matas, Matching with PROSAC- progressive sample consensus, in: CVPR, 2005.

[6] T. Sattler, B. Leibe, L. Kobbelt, Scramsac: Improving ransac’s efficiency with a spatial consistency filter, in: Computer Vision, 2009 IEEE 12th International Conference on, IEEE, 2010, pp. 2090–2097.

[7] C. V. Stewart, Robust parameter estimation in Computer Vision, SIAM Review 41 (3) (1999) 513–537.
[8] E. Vincent, R. Laganiere, Detecting planar homographies in an image pair, in: Image and Signal Processing and Analysis, 2001. ISPA 2001. Proceedings of the 2nd International Symposium on, 2001, pp. 182–187.

[9] Y. Kanazawa, H. Kawakami, Detection of planar regions with uncalibrated stereo using distributions of feature points, in: BMVC, 2004.

[10] M. Zuliani, C. Kenney, B. Manjunath, The multiransac algorithm and its application to detect planar homographies, in: Image Processing, 2005. ICIP 2005. IEEE International Conference on, Vol. 3, IEEE, 2006.

[11] R. Toldo, A. Fusiello, Robust multiple structures estimation with j-linkage, in: European Conference on Computer Vision, Springer, 2008, pp. 537–547.

[12] L. Xu, E. Oja, P. Kultanen, A new curve detection method: randomized hough transform (rht), Pattern Recognition Letters 11 (5) (1990) 331–338.

[13] R. Fagin, R. Kumar, D. Sivakumar, Comparing top k lists, in: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics, 2003, p. 36.

[14] T. Hastie, R. Tibshirani, J. Friedman, J. Franklin, The elements of statistical learning: data mining, inference and prediction, The Mathematical Intelligencer 27 (2) (2005) 83–85.

[15] O. Chum, J. Matas, J. Kittler, Locally optimized RANSAC, in: DAGM, 2003.

[16] D. Lowe, Distinctive image features from scale-invariant keypoints, IJCV 60 (2) (2004) 91–110.