Accelerated Hypothesis Generation for Multi-structure Robust Fitting

Tat-Jun Chin, Jin Yu, and David Suter

School of Computer Science, The University of Adelaide, South Australia
{tjchin,jin.yu,dsuter}@cs.adelaide.edu.au

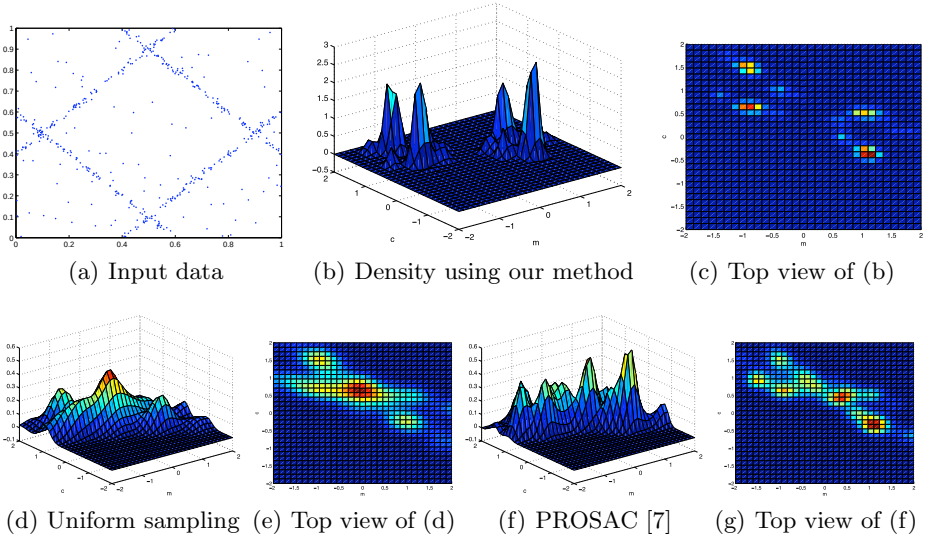
Abstract. Random hypothesis generation underpins many geometric model fitting techniques. Unfortunately it is also computationally expensive. We propose a fundamentally new approach to accelerate hypothesis sampling by guiding it with information derived from residual sorting. We show that residual sorting innately encodes the probability of two points to have arisen from the same model and is obtained without recourse to domain knowledge (*e.g.* keypoint matching scores) typically used in previous sampling enhancement methods. More crucially our approach is naturally capable of handling data with multiple model instances and excels in applications (*e.g.* multi-homography fitting) which easily frustrate other techniques. Experiments show that our method provides superior efficiency on various geometric model estimation tasks. Implementation of our algorithm is available on the authors' homepage.

1 Introduction

Random hypothesis sampling is central to many state-of-the-art robust estimation techniques. The procedure is often embedded in the "hypothesise-and-verify" framework commonly found in methods such as Random Sample Consensus (RANSAC) [1] and Least Median Squares (LMedS) [2]. The goal of sampling is to generate many putative hypotheses of a given geometric model (*e.g.* fundamental matrix, homography) from randomly chosen minimal subsets of the input data. The hypotheses are then scored in the verification stage according to a robust criterion (*e.g.* number of inliers, median of squared residuals).

The underlying principle of random hypothesis sampling is to "hit" at least one all-inlier subset corresponding to a particular genuine instance of the geometric model. Unfortunately the total number of hypotheses required such that this happens with significant chance scales with the fraction of outlier contamination. For heavily contaminated data hypothesis generation easily becomes the computational bottleneck. Moreover in data with *multiple* instances of the geometric model (also called "structures" [3]) the inliers of one structure behave as *pseudo*-outliers to the other structures, thus further compounding the problem.

Due to the widespread usage of robust estimators in Computer Vision there have been many innovations [4–9] to speed-up random hypothesis generation. These methods aim to guide the sampling process such that the probability of hitting an all-inlier subset is improved. The trick is to endow each input datum



(a) Input data (b) Density using our method (c) Top view of (b)

(d) Uniform sampling (e) Top view of (d) (f) PROSAC [7] (g) Top view of (f)

Fig. 1. Given the input data in (a) where there are 4 lines with 100 points per line and 100 gross outliers, we sample 100 line hypotheses using the proposed method, uniform sampling (à la the original RANSAC [1]) and PROSAC [7], yielding the parameter space density plotted respectively in (b), (d) and (f). Notice that the hypotheses of our method are concentrated on the correct models yielding 4 distinct peaks. Results from uniform sampling and PROSAC however contain many false peaks and irrelevant hypotheses. As Sec. 4 shows, in multi-structure data our method can successfully "hit" all true models using considerably less time than previous techniques.

a prior probability of being an inlier and to sample such that data that have high probabilities are more likely to be *simultaneously* selected. Such prior probabilities are often derived from domain-specific knowledge. For example Guided-MLESAC [6] and PROSAC [7] concentrate the sampling effort on correspondences with higher keypoint matching scores, the rationale being that inlier correspondences originate from confident keypoint matches (recall that in geometry estimation one correspondence consists of two matching points in different views). SCRAMSAC [9] further imposes a spatial consistency filter so that only correspondences which respect local geometry get sampled. Other works assume that inliers form dense clusters [5] or lie in meaningful image segments [8].

A crucial deficiency of previous methods lies in regarding the inlier probability of a datum to be *independent* of the other data. This is untrue when there are multiple structures. Given that an inlier of one structure is chosen, the probability that a second datum is an inlier (and thus should be chosen as well) depends on whether the second datum arose from the *same* structure. In other words, it is very possible that two correspondences with high keypoint matching scores are inliers from *different* valid structures. Methods that ignore this point are bound to wastefully generate many invalid *cross-structure* hypotheses. Our results on real and synthetic data (see Fig. 1) prove that this is indeed the case.

We also argue that the domain knowledge used in previous guided sampling techniques do not translate into convincing prior inlier probabilities. For example, inliers of a valid homography relation do not necessarily cluster together, while false or irrelevant correspondences can have high matching scores especially on scenes with repetitive textures. In the general case it is often questionable whether some usable and *reliable* domain knowledge is always available.

In this paper we propose a fundamentally novel technique to accelerate random hypothesis generation for robust model fitting. Our guided sampling scheme is driven only by residual sorting information and does not require domain- or application-specific knowledge. The scheme is encoded in a series of inlier probabilities which are updated on-the-fly. Most importantly our inlier probabilities are conditional on the selected data and thus encourages only inliers from the *same* structure to be simultaneously chosen for estimation. As our results demonstrate (see Sec. 4), our technique provides superior sampling efficiency especially on multi-structure data where other methods simply breakdown.

The rest of the paper is organised as follows: Sec. 1.1 surveys related work to put this paper in the right context. Sec. 2 describes the basic principles leading to our novel hypothesis generation scheme in Sec. 3. Sec. 4 outlines our experimental results and Sec. 5 draws conclusions.

1.1 Related Work

Many previous enhancements on the hypothesise-and-verify framework occur in the context of the RANSAC method. A recent survey [10] categorises them roughly into three groups. The first group of methods [4–9] aim to improve the random hypothesis sampling routine such that the chances of hitting an all-inlier sample is increased. In LO-RANSAC [4] an inner RANSAC loop is introduced into the main RANSAC body such that hypotheses may be generated from the set of inliers found so far, thus improving the consensus score more rapidly. Guided-MLESAC [6] and PROSAC [7] focus the sampling on more promising data based on keypoint matching scores, and this is extended to include spatial verification in SCRAMSAC [9]. In [5] sampling is concentrated on neighbouring correspondences, and in a similar spirit GroupSAC [8] focusses sampling on groups of data obtained using image segmentation. We emphasise that our work belongs to this category with the novelty of being domain-independent and optimised for accelerated hypothesis sampling in multi-structure data.

The second group of innovations [11–14] speed-up the hypothesis verification stage by minimising the time expended for evaluating unpromising hypotheses. The $T_{d,d}$ test [11] evaluates a hypothesis on a small random subset of the input data. This may mistakenly reject good hypotheses thus a much larger number of samples are required. However the overall time can potentially be reduced since the verification now consumes less time. Bail-Out test [12] and WaldSAC [13, 14] respectively apply catch-and-release statistics and Wald's theory of sequential decision making to allow early termination of the verification of a bad hypothesis.

The third category [10, 15] considers RANSAC in a real-time setting. The goal is to find the best model from a fixed number of hypotheses afforded by

the allotted time interval. Given a set of hypotheses, Preemptive RANSAC [15] scores them in a breadth-first manner such that unpromising hypotheses can be quickly filtered out from the subsequent passes. ARRSAC [10] performs a partially breadth-first verification such that the number of hypotheses may be modified according to the inlier ratio estimate while still bounding the runtime.

We are also aware of recent work [16, 17] that side-steps the hypothesise-and-verify framework and solves robust estimation directly as a global optimisation problem. While providing globally optimal solutions, these methods require considerably more time than RANSAC, especially for higher order geometric models. Our concern in this paper is to efficiently fit a geometric model onto noisy data with minimal loss to accuracy, and therefore our aims are different to [16, 17]. We also note that these methods [16, 17] currently cannot handle multi-structure data which make up a significant proportion of practical problems.

2 Inlier Probabilities from Sorting Information

We first describe how inlier probabilities can be derived from residual sorting information. Let $\mathcal{X} := \{\boldsymbol{x}_i\}_{i=1}^{N}$ be a set of N input data. Under the hypothesise-and-verify framework, a series of tentative models (or hypotheses) $\{\theta_1, \ldots, \theta_M\}$ are generated from minimal subsets of the input data where M is the number of hypotheses generated. For each datum $\boldsymbol{x}_i$ we compute its absolute residuals as measured to the M hypotheses to form the residual vector

$$\boldsymbol{r}^{(i)} := [\ r_1^{(i)}\ \ r_2^{(i)}\ \ \cdots\ \ r_M^{(i)}\ ]. \tag{1}$$

Note that the hypotheses do not lie in any particular order except the order in which they are generated. We then find the permutation

$$\boldsymbol{a}^{(i)} := [\ a_1^{(i)}\ \ a_2^{(i)}\ \ \cdots\ \ a_M^{(i)}\ ] \tag{2}$$

such that the elements in $\boldsymbol{r}^{(i)}$ are sorted in non-descending order, *i.e.*,

$$p < q \Longrightarrow r^{(i)}_{a_p^{(i)}} <= r^{(i)}_{a_q^{(i)}}. \tag{3}$$

The sorting $\boldsymbol{a}^{(i)}$ essentially ranks the M hypotheses according to the *preference* of $\boldsymbol{x}_i$; the higher a hypothesis is ranked the more likely $\boldsymbol{x}_i$ is an inlier to it.

Intuitively, two data $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ will share many common hypotheses at the top of their preference list $\boldsymbol{a}^{(i)}$ and $\boldsymbol{a}^{(j)}$ *if* they are inliers from the *same* structure. This is independent of whether $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ coexist in the same neighbourhood or whether they are correspondences with high keypoint matching scores.

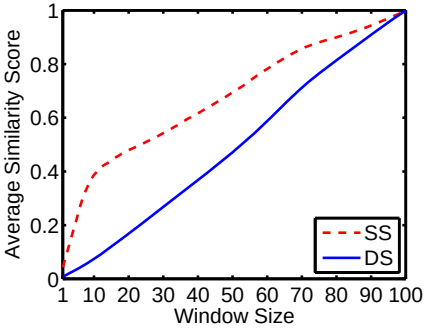
To illustrate this, first let $\boldsymbol{a}^{(i)}_{1:h}$ be the vector with the first-h elements of $\boldsymbol{a}^{(i)}$. We define the following function as the "intersection" between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$:

$$f(\boldsymbol{x}_i, \boldsymbol{x}_j) := \frac{1}{h}\left|\boldsymbol{a}^{(i)}_{1:h} \cap \boldsymbol{a}^{(j)}_{1:h}\right|, \tag{4}$$

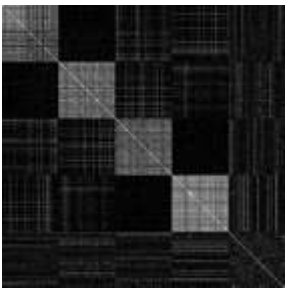
where $|\boldsymbol{a}_{1:h}^{(i)} \cap \boldsymbol{a}_{1:h}^{(j)}|$ finds the number of identical elements shared by $\boldsymbol{a}_{1:h}^{(i)}$ and $\boldsymbol{a}_{1:h}^{(j)}$. Window size h with $1 \leq h \leq M$ specifies the number of leading hypotheses to take into account. Note that $f(\boldsymbol{x}_i, \boldsymbol{x}_j)$ ranges between 0 and 1 and is symmetric with respect to its inputs. Also $f(\boldsymbol{x}_i, \boldsymbol{x}_i) = 1$ for all i.

The window size h controls the discriminability of the intersection score given by (4). It is found empirically that across a wide range of h values this score is discriminative. For the data in Fig. 1(a) where $M = 100$ we obtain the responses of $f(\boldsymbol{x}_i, \boldsymbol{x}_j)$ while h is varied from $1, \ldots, M$. Fig. 2(a) plots the mean of the responses which are separated according to whether the two input data are inliers from the same structure (denoted "SS") or otherwise (denoted "DS"). The result clearly shows that inliers from the same structure have higher intersection values relative to other possible pairs of inputs. Based on the result in Fig. 2(a), we set $h = \lceil 0.1 \times M \rceil$ by default for the intersection function unless mentioned otherwise.

We then obtain the $N \times N$ matrix K where the element at the i-th row and j-th column is simply $f(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Fig. 2(b) displays the matrix K by rearranging the points according to their structure membership, i.e., $\boldsymbol{x}_1$ to $\boldsymbol{x}_{100}$ are inliers from structure 1, $\boldsymbol{x}_{101}$ to $\boldsymbol{x}_{200}$ are inliers from structure 2 and so on. The gross outliers are $\boldsymbol{x}_{401}$ to $\boldsymbol{x}_{500}$. This makes visible a block diagonal pattern which confirms that strong mutual support occur among inliers of the same structure. We emphasise that such an arrangement is purely to aid in presentation and is unnecessary for $f(\boldsymbol{x}_i, \boldsymbol{x}_j)$ or our subsequent steps to work.



(a)



(b)

Fig. 2. (a) Average intersection values for the data in Fig. 1(a) while h is varied. (b) Matrix K of size 100×100 corresponding to $h = 10$.

We further analyse the results by plotting in Fig. 3 the values of selected rows of K. Unsurprisingly at a row corresponding to an inlier the significant values concentrate mostly on other inliers from the same structure, while for a gross outlier the values are generally low and appear to be randomly distributed. Therefore, given that a datum is selected, our idea is to use the intersection values of the datum as weights to sample a second datum. This yields inlier probabilities that encourages sampling within coherent structures. We emphasise that this phenomenon or idea is independent of the *type* of the geometric model.

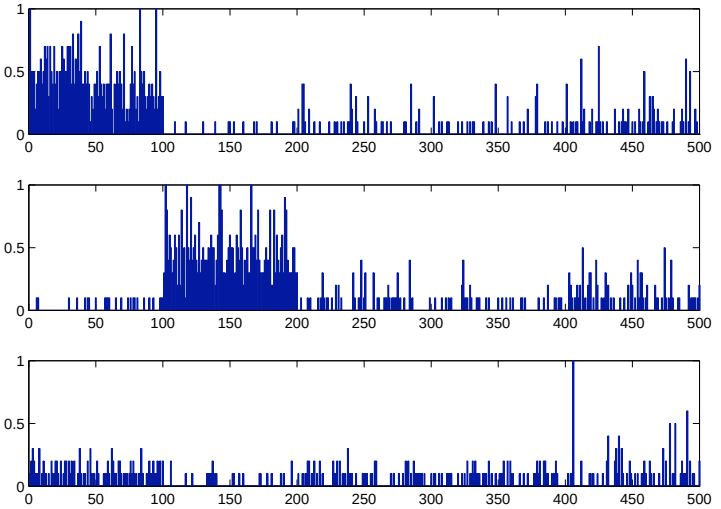


Fig. 3. Values at rows of K corresponding to an inlier from structure 1 (Top), an inlier from structure 2 (Center) and a gross outlier (Bottom)

More generally the observations in Figs. 2 and 3 indicate considerable potential in residual sorting information. This concept has been pursued in constructing statistical learning-based model fitting [18] and robust clustering methods [19]. In the next section we illustrate how residual sorting can be exploited to drive a very efficient hypothesis generation scheme.

3 Guided Sampling for Multi-structure Data

We use the similarity function (4) to design a guided sampling scheme (Multi-GS, Algorithm 1.) that is optimised for multi-structure data.

The Weighting Function Assume M model hypotheses have been generated so far and we wish to sample the next hypotheses (in a guided fashion). Let the model to be fitted be determined by a minimal subset $\mathcal{S} := \{\boldsymbol{s}_k\}_{k=1}^{p} \subseteq \mathcal{X}$ of p data, where $\boldsymbol{s}_k$ are indexed by the order in which they are sampled. The first datum $\boldsymbol{s}_1$ is selected randomly. We then define a *basis weighting function*

$$w(\boldsymbol{x}_i, \boldsymbol{x}_j) := \begin{cases} f(\boldsymbol{x}_i, \boldsymbol{x}_j) & \text{if } \boldsymbol{x}_i \neq \boldsymbol{x}_j, \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

where f is the intersection function (4). Given the first selection $\boldsymbol{s}_1$, the conditional probability $P(\boldsymbol{x}_j|\boldsymbol{s}_1)$ of selecting $\boldsymbol{x}_j$ as the second datum in the current minimal subset is then determined by the following monotonic relation:

$$w_1(\boldsymbol{x}_m) \geq w_1(\boldsymbol{x}_n) \Longrightarrow P(\boldsymbol{x}_m|\boldsymbol{s}_1) \geq P(\boldsymbol{x}_n|\boldsymbol{s}_1), \tag{6}$$

where $w_1(\cdot)$ is a *weighting function* conditioned on $\boldsymbol{s}_1$

$$w_1(\cdot) := w(\cdot, \boldsymbol{s}_1). \tag{7}$$

The monotonic relation (6) says that data that are consistent with $\boldsymbol{s}_1$ (according to the intersection score (4)) are more likely to be selected. Effectively $w_1(\boldsymbol{x}_j)$ for $j = 1, \ldots, N$ is a set of *sampling weights* to choose the second datum.

The remaining members (i.e., $\boldsymbol{s}_3, \boldsymbol{s}_4, \ldots, \boldsymbol{s}_p$) are also chosen *conditionally* on the data that are already drawn into the current minimal subset. Specifically the sampling weights for the $(k+1)$-th member of the minimal subset is

$$w_k(\boldsymbol{x}_j) := \prod_{i=1}^{k} w(\boldsymbol{x}_j, \boldsymbol{s}_i). \tag{8}$$

This is the element-wise multiplication of the rows of matrix K (see Fig. 2(b)) corresponding to data that have already been selected $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_k$. The conditional probability $P(\boldsymbol{x}_j|\boldsymbol{s}_1, \cdots, \boldsymbol{s}_k)$ of selecting $\boldsymbol{x}_j$ then follows from the rule

$$w_k(\boldsymbol{x}_m) \geq w_k(\boldsymbol{x}_n) \Longrightarrow P(\boldsymbol{x}_m|\boldsymbol{s}_1, \cdots, \boldsymbol{s}_k) \geq P(\boldsymbol{x}_n|\boldsymbol{s}_1, \cdots, \boldsymbol{s}_k). \tag{9}$$

This continues until p data have been selected. The $(M+1)$-th hypothesis is then estimated from the new minimal subset. Note that since (5) imposes $w_k(\boldsymbol{s}_k) = 0$ a datum cannot be chosen more than once into the *same* minimal subset.

Updating of Sampling Weights. Theoretically the sampling weights (8) are updated as soon as a new hypothesis is produced since (5) uses all available

Algorithm 1. Guided-Sampling for Multi-structure Robust Fitting (Multi-GS)

input input data $\mathcal{X}$, total number of hypotheses T, size of a minimal subset $p > 0$ and block size $b > 0$.
output a set Θ of T model hypotheses.
for $t := 1, 2, \cdots, T$ **do**
  if $t <= b$ **then**
    randomly sample p data and store as $\mathcal{S}$
  else
    select a datum $\boldsymbol{s}_1$ from $\mathcal{X}$ and initialise $\mathcal{S} := \{\boldsymbol{s}_1\}$
    for $k := 1, 2, \cdots, (p-1)$ **do**
      sample $\boldsymbol{s}_{k+1}$ from $\mathcal{X}$ by following the rule (9)
      $\mathcal{S} := \mathcal{S} \cup \{\boldsymbol{s}_{k+1}\}$
    end for
  end if
  $\Theta := \Theta \cup \{\text{Hypothesis instantiated from } \mathcal{S}\}$
  if $t >= b$ and $\texttt{mod}(t, b) == 0$ **then**
    update the permutation $\boldsymbol{a}^{(i)}$ (2) for all data
  end if
end for
return Θ

hypotheses. From a computational standpoint this is inefficient because a single new hypothesis does not add much information about inlier probabilities. Our proposed algorithm thus updates the weighting function only after a block (of size b) of new hypotheses are generated; see Step 14–16 in Algorithm 1..

The weighting function is practically updated by modifying the permutations $\boldsymbol{a}^{(i)}$ to account for the new hypotheses. We propose an efficient strategy to perform this step. Firstly, assume that we have the absolute residuals $\{\boldsymbol{r}^{(i)}\}_{i=1}^{N}$ for the M hypotheses sampled so far. Each of these is sorted increasingly to obtain the permutation vectors $\{\boldsymbol{a}^{(i)}\}_{i=1}^{N}$ of which only the top-h elements $\{\boldsymbol{a}_{1:h}^{(i)}\}_{i=1}^{N}$ partake in the computation of the sampling weights. The key to efficient updating is to fully retain $\{\boldsymbol{r}^{(i)}\}_{i=1}^{N}$ and $\{\boldsymbol{a}^{(i)}\}_{i=1}^{N}$. After b new hypotheses become available their absolute residuals to the dataset are computed and inserted using *binary search* into the sorted $\{\boldsymbol{r}^{(i)}\}_{i=1}^{N}$. The new leading hypotheses $\{\boldsymbol{a}_{1:h}^{(i)}\}_{i=1}^{N}$ are then extracted from the updated sorting. A binary search insertion into a vector of length M scales as $O(\log M)$ and we have b of these per datum. Therefore the total cost of updating $\boldsymbol{a}^{(i)}$ for N data is $O(Nb \log M)$.

On the surface it seems that the somewhat higher computational cost constitutes a weakness. However our algorithm conducts a more informed sampling given a unit of time in comparison to other techniques. The result is that we require less total CPU time to hit at least one all-inlier subset of *all* valid structures in the data; this is validated by our experiments in Sec. 4. In contrast the other methods are much slower because they unproductively generate many invalid cross-structure hypotheses. In single structure data the proposed algorithm performs comparably to other guided sampling techniques.

4 Experiments

We evaluated the performance of the proposed method (Multi-GS, Algorithm 1.) on both synthetic and real image datasets. We compared against other state-of-the-art sampling enhancement schemes: LO-RANSAC [4], proximity sampling [5, 20] (denoted "Exp"), Guided-MLESAC [6], and PROSAC [7]. Uniform random sampling as in the original RANSAC [1] (denoted "Random") is used as the baseline. We implemented all algorithms in MATLAB. All experiments were run on a Linux machine with 2.67GHz Intel quad core processors and 4 GB of RAM.

In all experiments the inlier threshold required by LO-RANSAC was set to the average residual of inliers as measured to their corresponding structures; T_N in PROSAC was set to 5×10^4. The scale parameter of Exp (σ^2 as in Equation 1 in [20]) was set to twice the squared average nearest neighbour distance. We implemented Guided-MLESAC such that data points with higher quality scores (*e.g.* keypoint matching scores) are given higher probabilities to be drawn. For our method we consistently fixed the block size $b = 10$ and the window size $h = \lceil 0.1 \times t \rceil$, t being the number hypotheses generated so far.

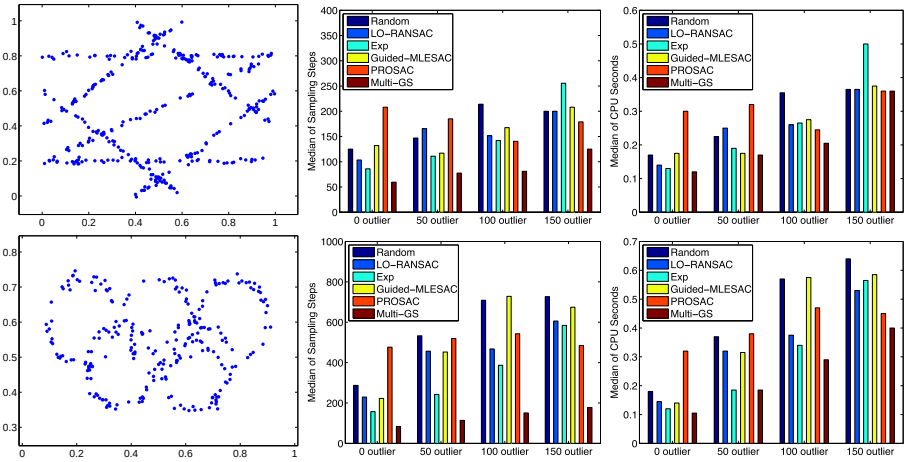


Fig. 4. The performance of various sampling methods on 2D geometric data. The left column shows the generated inliers. The centre and right columns respectively show the median number of sampling steps and total CPU seconds needed to hit at least one all-inlier subset in each structure as the number of gross outliers is varied.

4.1 Multiple Line and Circle Fitting

We first compare our algorithm against previous methods on multiple line and circle fitting in 2D. Fig. 4 (left column) depicts the synthetically generated inliers (respectively 7 lines and 5 circles). We also add random gross outliers to increase the difficulty of the problem. The inlier noise scale and the number of inliers per structure were fixed to 10^{-2} and 50, respectively. PROSAC and Guided-MLESAC require each datum to be associated with a quality score; we simulate this by probabilistically assigning inliers with higher scores than gross outliers.

Each method is given 20 random runs. The centre and right columns in Fig. 4 respectively show the median of sampling steps (i.e., number of hypotheses) and total CPU seconds required to recover *at least* one all-inlier minimal subset for each structure *vs* the number of gross outliers in the data. It can be seen that our method is the most efficient in terms of the required sampling steps for both lines and circles. For instance, in the case of circle fitting, Multi-GS typically takes no more than half of the sampling steps needed by the other methods. In terms of total time expended Multi-GS still require less CPU seconds than the others, especially so for circle fitting. This suggests that the performance gap between our method and previous approaches would widen for higher order geometric models. Indeed, we demonstrate this in the next two experiments.

4.2 Homography Estimation

Our second set of experiments involve estimating planar homographies on real image data.[1] Putative keypoint correspondences and their corresponding scores

[1] `http://www.robots.ox.ac.uk/~vgg/data`

(a) Left: College-1. Right: College-2.

(b) Left: College-3. Right: College-4.

Fig. 5. Images pairs used in the experiments of Sec. 4.2 with marked *inlier* keypoints

were obtained by SIFT matching [21].[2] Fig. 5 shows the image pairs used in our experiments with marked keypoints. Note that for clarity we show only the *true inliers* in Fig. 5; there actually exist a large number of false correspondences (ranging between 20 to 100 depending on the images) which represent gross outliers due to incorrect SIFT matches. We use 4 correspondences to estimate a homography via Direct Linear Transformation (DLT, [22]). For each method, 50 random runs were performed, each for 60 CPU seconds.

Our experiments start with a relatively easy task which involves estimating a single homography for the planar structure marked in the College-1 image pair (Fig. 5, top left). The data contain 70 inlier correspondences with an approximately 41% inlier ratio. As can be seen in Table 1, by leveraging the SIFT matching scores, PROSAC hits an all-inlier sample at the very first iteration, costing nearly zero CPU time, whereas Exp and Guided-MLESAC perform better than the others in terms of the total number of all-inlier samples found within the given time budget. Overall, the performance of all methods are comparable on this simple *single structure* recovery task.

We now move to the setting of multi-structure fitting. The performance of various sampling methods was evaluated on the image pairs that contain 2–4 planar structures (Fig. 5, top right and bottom rows). Table 1 shows that the proposed Multi-GS is superior in terms of CPU seconds required to hit at least one all-inlier subset from all structures. For instance, on College-3 and College-4, Multi-GS requires around 80% *less* time than the best performing competing method. Moreover, within the given CPU time limit the total number of all-inlier subsets found by our method is typically much more than that of other methods.

[2] We used the code given on `http://www.vlfeat.org/~vedaldi/code/sift.html`

Table 1. Performance of various sampling methods over 50 random runs, each for 60 CPU seconds. We report the median of CPU time (CPU) (*resp.* sampling steps (Iter)) that is required to find at least one all-inlier minimal subset for each structure present in the data in Fig. 5. The average number of all-inlier samples found within 60 CPU seconds is listed separately for each structure I-i, i $= 1, 2, \cdots$. The number of inliers and the inlier ratio for each I-i is given in the parenthesis. The top result with respect to each performance measure are boldfaced.

Data		Random	LO-RANSAC	Exp	Guided-MLESAC	PROSAC	Multi-GS
College-1	CPU	0.06	0.02	0.02	0.01	$\mathbf{< 10^{-3}}$	0.02
	Iter	33	13	7	7	**1**	13
	I-1 (70, 41%)	978	1012	2782	**3591**	1380	1119
College-2	CPU	1.39	0.47	0.65	0.39	0.14	**0.11**
	Iter	836	261	354	229	77	**47**
	I-1 (70, 34%)	450	438	**1183**	1164	576	867
	I-2 (36, 17%)	30	29	63	96	43	**350**
College-3	CPU	1.03	0.75	0.62	0.67	0.69	**0.14**
	Iter	592	418	336	374	374	**54**
	I-1 (71, 22%)	72	71	140	134	96	**292**
	I-2 (80, 24%)	116	115	266	401	166	**494**
	I-3 (78, 24%)	101	100	199	82	88	**286**
College-4	CPU	5.23	5.34	2.42	3.49	1.62	**0.31**
	Iter	3060	3105	1309	2029	897	**113**
	I-1 (42, 15%)	18	17	47	29	24	**160**
	I-2 (42, 15%)	17	17	37	49	25	**171**
	I-3 (47, 17%)	28	27	57	60	39	**237**
	I-4 (42, 15%)	15	17	42	26	19	**91**

4.3 Fundamental Matrix Estimation

We also applied our sampling method to accelerate the estimation of fundamental matrices. Images of multiple moving objects were obtained from the web.[3] The keypoint correspondences and matching scores were obtained by SIFT matching. Hypotheses were generated from 7 keypoint correspondences via the standard 7-point estimation method [23].[4] For each method, 50 random runs were performed, each for 60 CPU seconds. Table 2 summarizes the performance of all methods on the three image pairs in Fig. 6. Again note that for clarity we only show the true inliers in Fig. 6. There exist from incorrect SIFT matchings many false correspondences which constitute gross outliers in the data.

Similar to the previous set of experiments, existing sampling methods are effective in sampling from single-structure data (*cf.* results on the Book data in Table 2), while they fail disastrously when more than one structure is present. Along with their inability to distinguish keypoint correspondences from different

[3] `http://www.iu.tu-darmstadt.de/datasets`
[4] `http://www.robots.ox.ac.uk/~vgg/hzbook/code/`

(a) Book (b) Desk

(c) Office

Fig. 6. Image pairs used in the experiments of Sec. 4.3 with marked *inlier* keypoints

Table 2. Performance of various sampling methods on image pairs in Fig. 6 (50 random runs with 60 CPU seconds per run). The same notations as used in Table 1 are used. In addition, we record the number of times a method fails to find at least one all-inlier sample for each structure within the given 60 CPU seconds (Fail). The reported median of CPU time (*resp.* sampling steps) is taken over successful runs only.

Data		Random	LO-RANSAC	Exp	Guided-MLESAC	PROSAC	Multi-GS
Book	CPU	0.03	0.01	0.01	$\mathbf{< 10^{-3}}$	$\mathbf{< 10^{-3}}$	0.02
	Iter	11	4	4	3	**1**	8
	I-1 (28, 58%)	1426	1429	5879	**9152**	1778	184
	Fail	0	0	0	0	0	0
Desk	CPU	19.13	18.44	24.61	7.45	17.41	**0.18**
	Iter	5716	5572	7604	2983	5458	**41**
	I-1 (48, 27%)	7	7	79	40	8	**355**
	I-2 (28, 16%)	1	1	2	5	1	**175**
	Fail	47	47	15	2	44	**0**
Office	CPU	40.48	44.56	9.9	16.23	46.83	**0.17**
	Iter	13883	15442	3456	5719	15861	**38**
	I-1 (81, 24%)	2	2	10	6	2	**167**
	I-2 (78, 23%)	2	2	9	3	2	**234**
	I-3 (84, 24%)	2	3	10	6	3	**193**
	Fail	35	22	0	7	23	**0**

structures, the increase in the size of the minimal subset (from 4 in homography estimation to the current 7) makes the sampling from multi-structure data an extremely challenging task for previous methods. For instance, random sampling, LO-RANSAC, and PROSAC fail to find an all-inlier subset for each structure

in 44%-94% of the 50 random runs. As can be seen in Table 2, Multi-GS dramatically outperforms other methods in terms of all performance measures on the two multi-structure data: Desk and Office. It hits at least an all-inlier subset in each structure over an order of magnitude faster in terms of both CPU time and sampling steps. Moreover, within the given time limit, the overall all-inlier subsets found by our methods are up to over two orders of magnitude more than that obtained by other methods.

5 Conclusions

We propose a fundamentally new approach to accelerate hypothesis generation by guiding information derived from residual sorting. In contrast to existing sampling techniques, our approach is naturally capable of handling data with multiple structures. We also do not require potentially confusing domain knowledge needed by other techniques. We demonstrated, and compared our method on various multi-structure geometric modelling tasks. Our results show that the proposed method significantly outperforms previous techniques in terms of the total CPU time required to recover all valid structures in multi-structure data.

Acknowledgements

This work is supported by the Australian Research Council grant DP0878801.

References

1. Fischler, M.A., Bolles, R.C.: RANSAC: A paradigm for model fitting with applications to image analysis and automated cartography. Comm. of the ACM 24, 381–395 (1981)
2. Rousseeuw, P.J., Leroy, A.M.: Robust regression and outlier detection. Wiley, Chichester (1987)
3. Stewart, C.V.: Robust parameter estimation in Computer Vision. SIAM Review 41, 513–537 (1999)
4. Chum, O., Matas, J., Kittler, J.: Locally optimized RANSAC. In: Michaelis, B., Krell, G. (eds.) DAGM 2003. LNCS, vol. 2781, pp. 236–243. Springer, Heidelberg (2003)
5. Kanazawa, Y., Kawakami, H.: Detection of planar regions with uncalibrated stereo using distributions of feature points. In: BMVC (2004)
6. Tordoff, B.J., Murray, D.W.: Guided-MLESAC: Faster image transform estimation by using matching priors. TPAMI 27, 1523–1535 (2005)
7. Chum, O., Matas, J.: Matching with PROSAC- progressive sample consensus. In: CVPR (2005)
8. Ni, K., Jin, H., Dellaert, F.: GroupSAC: Efficient consensus in the presence of groupings. In: ICCV (2009)
9. Sattler, T., Leibe, B., Kobbelt, L.: SCRAMSAC: Improving RANSAC's efficiency with a spatial consistency filter. In: ICCV (2009)

10. Raguram, R., Frahm, J.M., Pollefeys, M.: A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 500–513. Springer, Heidelberg (2008)
11. Matas, J., Chum, O.: Randomized RANSAC with $t_{d,d}$ test. In: Image and Vision Computing (2004)
12. Capel, D.: An effective bail-out test for RANSAC consensus scoring. In: BMVC (2005)
13. Matas, J., Chum, O.: Randomized RANSAC with sequential probability ratio test. In: ICCV (2005)
14. Chum, O., Matas, J.: Optimal randomized RANSAC. TPAMI 30, 1472–1482 (2008)
15. Nister, D.: Preemptive RANSAC for live structure and motion estimation. In: ICCV (2003)
16. Enqvist, O., Kahl, F.: Two view geometry estimation with outliers. In: BMVC (2009)
17. Li, H.: Consensus set maximization with guaranteed global optimality for robust geometry estimation. In: ICCV (2009)
18. Chin, T.J., Wang, H., Suter, D.: Robust fitting of multiple structures: The statistical learning approach. In: ICCV (2009)
19. Chin, T.J., Wang, H., Suter, D.: The ordered residual kernel for robust motion subspace clustering. In: NIPS (2009)
20. Toldo, R., Fusiello, A.: Robust multiple structures estimation with j-linkage. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 537–547. Springer, Heidelberg (2008)
21. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV 60, 91–110 (2004)
22. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2003)
23. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004) ISBN: 0521540518