2018 IEEE Winter Conference on Applications of Computer Vision

DGSAC: Density Guided SAmpling and Consensus

Lokender Tiwari Saket Anand
IIIT-Delhi
{lokendert, anands}@iiitd.ac.in

Abstract

In this paper, we present an automatic multi-model fitting pipeline that can robustly fit multiple geometric models present in the corrupted and noisy data. Our approach can handle large data corruption and requires no user input, unlike most state-of-the-art approaches. The pipeline can be used as an independent block in many geometric vision applications like 3D reconstruction, motion and planar segmentation. We use residual density as the primary tool to guide hypothesis generation, estimate the fraction of inliers, and perform model selection. We show results for a diverse set of geometric models like planar homographies, fundamental matrices and vanishing points, which often arise in various computer vision applications. Despite being fully automatic, our approach achieves competitive performance compared to state-of-the-art approaches in terms of accuracy and computational time.

1. Introduction

The problem of fitting geometric models such as planar homographies, fundamental matrices or vanishing points is frequently encountered in computer vision. Accurate estimation of these model parameters facilitate a simple and interpretable representation of scene geometry, rigid body motion or camera pose, and can aid scene understanding. However, most features used for estimating these models are obtained from low-level processing, e.g., detection of keypoints, edges or lines. Feature extraction techniques oblivious to the underlying scene geometry, typically produce a significant number of outliers, i.e., detections that do not adhere to any genuine model instances. In many scenarios, e.g., in autonomous driving, it is unreasonable to expect user input about number of model instances or fraction of outliers. It is important to automatically discover multiple models from images or videos but it requires overcoming several challenges including noise scale estimation, outlier rejection and model selection.

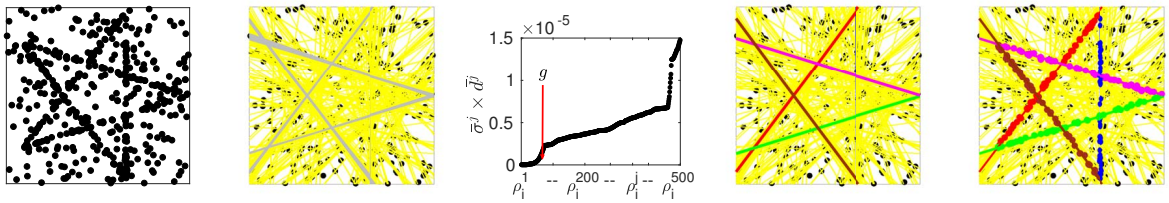
We consider model fitting scenarios where the data is generated from multiple model instances (inlier structures) and is corrupted by noise and outliers. Data points that follow a given model are *inliers* for that model and act as *pseudo-outliers* for all other genuine models, while the ones that do not follow *any* model are the *gross-outliers*. The goal is to identify *all* genuine model instances and their corresponding inliers while rejecting the gross outliers. Once the inliers are identified, standard techniques can be used for parameter estimation of each recovered model.

Single model fitting is often solved by consensus maximization with heuristic methods like Random Sample Consensus (RANSAC) [8] and its variants or by applying globally optimal techniques [2]. However, both approaches critically depend on the correct estimate of the scale of inlier noise (or equivalently, the fraction of inliers), often taken as user input. For single model fitting, this process can be automated by applying scale estimation techniques [26, 25, 15], but in case of multi-model fitting the problem is challenging due to additional unknowns. Several proposed approaches [18, 3, 28, 11, 12, 13, 14, 16] assume the *number of models* or *inlier scale* or *both* are known a priori, which introduces user dependency.

In this paper, we achieve automatic robust multi-model fitting by exploiting the *residual density* [22], which is a key differentiator between inliers and outliers. The inlier residuals are relatively small and form a dense cluster near zero, while the outliers have larger residual values and a much lower density. The Density Guided Sampling and Consensus (DGSAC) is an automatic technique for multi-model fitting and is illustrated using a synthetic example in Fig. 1. Our specific contributions are summarized below:

- *Hypothesis generation*: We present a novel iterative approach that is driven by residual density based point correlations and is shown to generate more relevant model hypotheses from *all* inlier structures.
- *Model Selection*: We propose a novel greedy model selection approach that uses the residual density to uniquely identify a model hypothesis from each inlier structure and rejects redundant hypotheses based on the Spearman-Footrule distance.
- *Inlier Fraction Estimation*: We identify the inlier/outlier boundary by applying a simple yet effec-

978-1-5386-4886-5/18/$31.00 ©2018 IEEE
DOI 10.1109/WACV.2018.00112



(a) Data (b) Hypothesis Generation (c) Fraction Estimation (d) Model Selection (e) Final Result

Figure 1: **DGSAC**: Illustrative example (Multiple Line Fitting). Refer text in, (b) Sec. 3.2, (c) Sec. 3.3, (d) and (e) Sec. 3.4, point membership is color coded, outliers in black.

tive heuristic based on residual dispersion and residual density.

- *Multi-model fitting*: We combine the above three modules to engineer an end-to-end automatic multi-model fitting solution that eliminates the need for user input.

The remainder of the paper is organized as follows. Sec. 2 discusses recent approaches for multi-model fitting and their limitations, followed by details of the proposed method, DGSAC, in Sec. 3. We show extensive experimental evaluation showing the performance of DGSAC over state of the art approaches in Sec. 4 and conclude with a discussion in Sec. 5.

2. Related Work

In this section, we review relevant literature that address different aspects of the robust multi-model fitting problem in computer vision.

Clustering based approaches that operate in a conceptual space [23, 13, 14, 16] have gained popularity over more traditional extensions of RANSAC that work on the *fit-and-remove* principle like sequential-RANSAC and multi-RANSAC [29]. Both class of techniques generate hypotheses by sampling *minimal sample sets* (MSS), but the former class of techniques implicitly assume the availability of several relevant model hypotheses[1] from an inlier structure, which are often achieved through guided sampling approaches. Guided sampling strategies can be broadly classified into the following two categories.

Domain knowledge based guided sampling utilize the score of feature matches across images and/or their spatial local neighborhood to bias the sampling process. Guided-MLESAC [24], PROSAC [4], and EVSAC [9] assume potential inliers would have high feature matching scores, while SCRAMSAC [19] and NAPSAC [17] assume features matches with similar spatial neighborhood in the image space are more likely to be inliers. Such assumptions of higher matching scores or spatial consistency could be violated when repeated patterns are present in the image or with a higher density of feature points, where outliers may also satisfy the spatial consistency check. Moreover, both these assumptions restrict the application to models that directly work with image pairs and are difficult to extend to more general geometric models like vanishing points.

Preference analysis based guided sampling techniques are oblivious to domain information and rely only on ordering of points (or hypotheses) according to the residuals. A point (hypothesis) with a lower residual is said to be preferred by a hypothesis (point) over another with a higher residual value. Both DHF[28] and ITKSF[28] improve upon Multi-GS[3] and use preferences to learn conditional distributions which influence the probability of points sampled for the MSS. While these methods generate a good proportion of relevant model hypotheses, they operate within a time budget framework. This limits their ability to generate hypotheses from *all* inlier structures, particularly when the priors on inlier fractions are skewed. In the presence of smaller inlier structures, i.e., model instances with only a few inliers, the time budget needs to be quite large in order to generate a sufficient number of relevant hypotheses for the subsequent clustering process to successfully identify smaller structures. Besides, it is difficult to set an appropriate time budget without prior knowledge of structures and their corresponding inlier fractions.

Given a set of model hypotheses, one needs to perform model selection, i.e., identifying one hypothesis for each genuine inlier structure. Recently, matrix factorization based approaches like RPA [14] or NMU [21] have been proposed for model selection and have shown to outperform clustering based approaches. RPA assumes the knowledge of the inlier noise scale (albeit for the entire dataset) and the number of structures to be known a priori. It constructs a data point similarity matrix which is decomposed using symmetric NMF [10], and used with a preference matrix for final model selection. NMU outperforms RPA by enforcing an additional constraint of under-approximation, however, it also requires a user specified noise scale estimate.

In this paper, we deviate from the time budget framework and propose a novel guided sampling technique using a residual density based stopping criterion, which makes the guided sampling completely data-driven. Since our proposed method DGSAC, has downstream tasks of inlier fraction estimation and model selection that adhere more to the consensus maximization framework rather than the cluster-

[1] hypothesis fitted to all inliers minimal sample set

ing or matrix factorization based approaches, it suffices for the sampling process to identify only one relevant model hypothesis from each structure.

3. DGSAC

Notation: A matrix and its corresponding i^{th} row and j^{th} column are represented by $\mathbf{R}$, $\mathbf{r}_i$ and $\mathbf{r}^j$ respectively. Element of a matrix is denoted by r_i^j. First k elements of a column vector $\mathbf{r}^j$ is represented by $\mathbf{r}^j_{[1:k]}$. Elements of a column $\mathbf{r}^j$ whose indices are in a set w is represented by $\mathbf{r}^j_{\{w\}}$ and the cardinality of a set a is denoted by $|a|$.

Problem Statement: Let there be a total of κ inlier structures present in a given set of n data points, $\mathbf{X} = \{x_i,\ i = 1, \ldots, n,\ x_i \in \mathbb{R}^d\}$. The fraction of inliers of each structure is denoted by $f_i = \frac{|\mathcal{I}_i|}{n},\ i = 1, ..., \kappa$, where $\mathcal{I}_i$ is an index set of inlier points of i^{th} structure and $f_0 = 1 - \sum_{i=1}^{\kappa} f_i$ denotes the fraction of outliers. Our goal is to output a set of κ tuples $(\mathbf{h}_1, \mathcal{I}_1), (\mathbf{h}_2, \mathcal{I}_2), ..., (\mathbf{h}_\kappa, \mathcal{I}_\kappa)$, where $\mathbf{h}_i$ and $\mathcal{I}_i$ are the estimated model parameters and inlier set of the i^{th} structure respectively.

3.1. Building Blocks

Let $\mathbf{h}_i$ be a hypothesis generated by fitting the model to s_i, a minimal sample subset (MSS) of cardinality η, which is the minimum number of data points required to uniquely fit a model. Further, let $\mathbf{H}$ be the set of m model hypotheses, for which the residual vector $\mathbf{r}_i$ (1) is computed using the residual function defined as $\psi(\mathbf{h}_i, x_j) : \mathbb{R}^d \rightarrow \mathbb{R}_+$.

$$\mathbf{r}_i = [r_i^1 = \psi(\mathbf{h}_i, x_1), ..., r_i^n = \psi(\mathbf{h}_i, x_n)] \qquad (1)$$

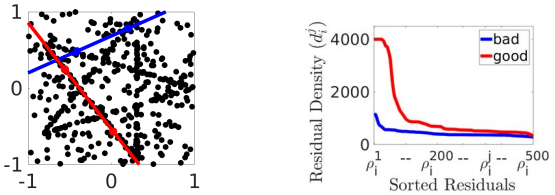
We find a permutation set $\mathbf{q}_i = [q_i^1, q_i^2, ..., q_i^n]$ for the model hypothesis $\mathbf{h}_i$ such that $r_i^{q_i^1} \leq r_i^{q_i^2} \leq, ..., \leq r_i^{q_i^n}$ and thus get a *ordered residual* vector $\boldsymbol{\rho}_i = [r_i^{q_i^1}, r_i^{q_i^2}, ..., r_i^{q_i^n}]$, which is then smoothed using an averaging filter of size $\lceil 0.025 \times n \rceil$. Since the permutation set $\mathbf{q}_i$ ranks the data points and encodes the preferences of the hypothesis $\mathbf{h}_i$ therefore we refer it as *hypothesis preference set*.

3.1.1 Residual Density

We compute the residual density [22] for a given hypothesis $\mathbf{h}_i$ at each data point x_j as shown in (2). The number of data points having residual smaller than ρ_i^j is considered as the mass and accordingly the residual density d_i^j is the mass per unit volume[2].

$$d_i^j = \frac{j}{\rho_i^j + \varepsilon} = \frac{j}{r_i^{q_i^j} + \varepsilon}, \qquad j = 1, ..., n \qquad (2)$$

The residual density plot of a model hypothesis is shown in Fig. 2b. As expected, it can be observed that density is higher near the regression surface (zero residual) and decreases as the residual increases.



(a) Hypotheses $\mathbf{h}_i$ (b) Residual Density (d_i^j)

Figure 2: Residual density plot of a good (red) and a bad (blue) model hypothesis

3.1.2 Point Preferences

Analogous to the hypothesis preference set $\mathbf{q}_i$, we find a permutation $\mathbf{u}^j = [u_1^j, u_2^j, ..., u_m^j]$ for a data point x_j such that $r^j_{u_1^j} \leq r^j_{u_2^j} \leq, ..., \leq r^j_{u_m^j}$. This permutation vector $\mathbf{u}^j$ ranks the hypotheses, it encodes the preferences of a point x_i for model hypotheses. Since the ordering is based on smallest residual first, we refer to $\mathbf{u}^j$ as *residual based point preferences*. Similarly, we find *density based point preferences* for x_j as the permutation $\mathbf{v}^j = [v_1^j, v_2^j, ..., v_m^j]$ such that $d^j_{v_1^j} \geq d^j_{v_2^j} \geq, ..., \geq d^j_{v_m^j}$. Note that hypotheses are ordered in decreasing order of their density values.

Point preferences can be used to compute correlation between point pairs by applying intersection kernels (or a similar measure) over the top-T preferences as in (3) below, which shows the point correlation between x_i and x_j using density based preferences. Similarly, residual based point correlations can be computed using $\mathbf{u}^i_{1:T}$.

$$p_i^j = \frac{\mathbf{v}^i_{[1:T]} \cap \mathbf{v}^j_{[1:T]}}{T} \qquad (3)$$

Ideally, a point correlation measure should be such that an inlier point from one structure is highly correlated with inliers from the same structure, while having zero correlation with points from different structures or outliers. With this observation, using the *breadcubechips* example from the AdelaideRMF dataset [27], we illustrate the advantage of using point correlations based on density as opposed to residual preferences in Fig. 3. The residual and density based pairwise point correlation matrix (PCM) is shown in Fig. 3(a) and Fig. 3(b) respectively, with brighter pixels indicating higher correlation. For ease of illustration, the points are ordered by structure membership, with the first set (top rows) being the gross outliers. The value of T is set after thorough empirical validation independently for both the residual and density cases. Once selected, this value of T is fixed across *all* experiments. Our choice of $T = \lfloor 0.1m \rfloor$ for residual based correlation is consistent with prior work [3, 28] and we set $T = 5$ for the density case.

We estimate the quality of a point correlation measure by computing the percentage of outliers (both gross and

[2] ε is a small constant added to the denominator to restrict the numerical instability of the density.

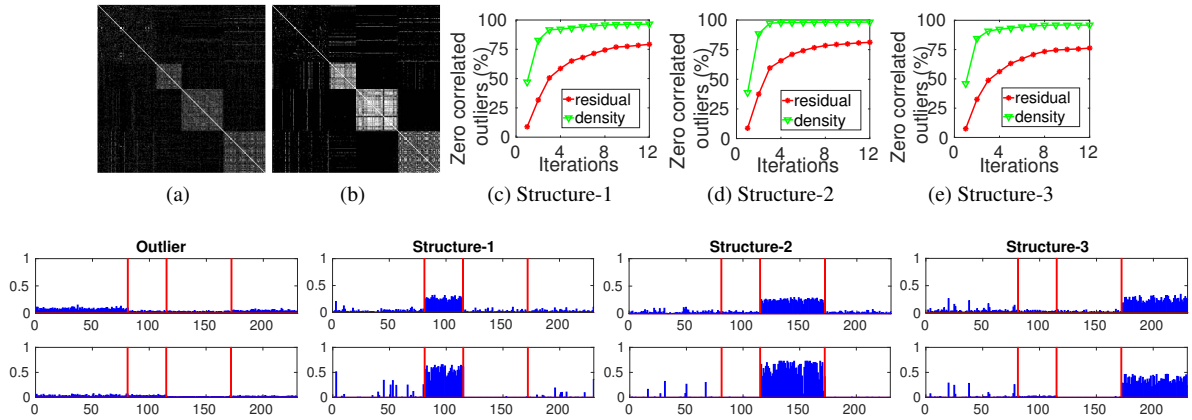


Figure 3: **Residual vs. Density based point correlation**: *breadcubechips* sequence of AdelaideRMF [27] dataset (a) Residual based and (b) Density based point correlation matrix with rows and columns ordered by structure membership. (c), (d) and (e) are percentage of *uncorrelated* outliers varying with iterations of (Algo. 1) for each of the three structures *bread*, *cube* and *chips* respectively. Rows 2 and 3 show structure-wise averaged correlation (see Sec. 3.1.2) for details.

pseudo-outliers) that are *uncorrelated*, i.e., have exactly zero correlation with inliers of a structure. We average the rows of the PCM over all inliers from a single structure and compute the percentage of uncorrelated outliers from this averaged correlation. Note that we use the ground truth for computing the averaged correlation only to compare the density and residual based approaches, not in our algorithm described later. Fig. 3, row-2, shows the plot of this averaged correlation for the residual PCM with each column heading (Outlier, Structure-1, etc.) indicating the rows over which the averaging is performed. Similarly, the plots for density based PCM are shown in row-3. In our guided sampling based hypothesis generation, we update the PCM as new model hypotheses are sampled (see Sec. 3.2 for details). At each iteration of sampling and hypothesis generation, the percentage of *uncorrelated* outliers is computed from the updated PCM. Fig. 3(c)-(e) show how the percentage of outliers evolve over the iterations for density based PCM (green curve) and the residual based PCM (red curve). The density based correlation for outliers quickly converge to zero for most outliers, while the same is not true for residual based correlations. In our experiments we have found this observation to be typical for different types of data, which is not surprising given that density is the key differentiator between inliers and outliers. This sparsity observed in the density based correlation values for outliers is a very useful property and we exploit it in our proposed density based guided sampling, which we discuss next.

3.2. Density Guided Sampling (DGS)

The goal of the guided sampling algorithm is to sample many *pure* MSS (with elements that are inliers to the *same* structure). Most recent guided sampling algorithms take the following approach: sample the first element from a uniform distribution and sample the other elements from a conditional probability mass function (PMF), which is usually derived from the PCM [3, 28].

In our proposed Density Guided Sampling (DGS) algorithm, we follow a similar strategy except that we select the first element deterministically follwed by iterative improvement of the PCM, which subsequently helps generate better model hypotheses. As summarized in Algo. 1, DGS starts with initializing the set $\nu = \{1, \ldots, n\}$ and the PCM $\mathbf{P}$, with a matrix of ones (Algo. 1, line 1), i.e., every point pair is equally correlated. We use the routine `generateHypotheses` (Algo. 2) with inputs $\mathbf{P}, \nu$ and

Algorithm 1: Density Guided Sampling (DGS)

Input : Data$(\mathbf{X}_{d\times n}, \eta)$, Params$(T, \alpha)$
Output: $\mathbf{H}$, $\mathbf{R}$, $\mathbf{D}$, $\mathbf{P}$

1 $\mathbf{P} \leftarrow \mathbb{1}_{n\times n}, \mathbf{H} = \mathbf{R} = \mathbf{D} = \{\emptyset\}$
2 $\nu \leftarrow \{1, \ldots, n\}$, $\boldsymbol{\tau}^{curr} = \boldsymbol{\tau}^{prev} = \mathbf{0}_{n\times 1}$
3 **while** $\nu \neq \{\emptyset\}$ **do**
4 $\quad [\widehat{\mathbf{H}}, \widehat{\mathbf{R}}, \widehat{\mathbf{D}}] \leftarrow$ `generateHypotheses`$(\mathbf{P}, \nu, \eta)$
5 $\quad \mathbf{H} \leftarrow \{\mathbf{H} \cup \widehat{\mathbf{H}}\}$, $\mathbf{R} \leftarrow \{\mathbf{R} \cup \widehat{\mathbf{R}}\}$, $\mathbf{D} \leftarrow \{\mathbf{D} \cup \widehat{\mathbf{D}}\}$
6 $\quad \mathbf{V} \leftarrow$ `getDensityPointPreferences`$(\mathbf{D})$
7 $\quad \boldsymbol{\tau}^{prev} \leftarrow \boldsymbol{\tau}^{curr}$
8 $\quad \tau_j^{curr} = \sum_{t=1}^{T} d^j_{v_t^j}$, $j \in \{1, \ldots, n\}$
9 $\quad \nu \leftarrow \{j \,|\, |\tau_j^{curr} - \tau_j^{prev}| \geq \alpha \tau_j^{prev}, j = 1, \ldots, n\}$
10 $\quad \mathbf{P} \leftarrow$ `updatePointCorrelation`$(\mathbf{P}, \nu)$
11 **end**
12 $\Theta \leftarrow \bigcup_{j=1}^{N} \mathbf{v}^j_{[1:T]}$
13 $\mathbf{H} \leftarrow \mathbf{H}_{\{\Theta\}}$, $\mathbf{R} \leftarrow \mathbf{R}_{\{\Theta\}}$, $\mathbf{D} \leftarrow \mathbf{D}_{\{\Theta\}}$

η to generate a new model hypothesis for each point $j \in \nu$ by sampling η points using the PCM $\mathbf{P}$. The key steps in `generateHypotheses` are lines 5-7 (Algo. 2), where for each $j \in \nu$, we sample $\eta - 1$ points *without replacement* (enforced by Algo. 2, line 6) using the conditional PMF c, which in turn is obtained from $\mathbf{P}_j$. Using the η sample points, we generate a model $\mathbf{h}$ using the function `getModel`, which is known a priori for different model fitting tasks (plane, homography, vanishing point, etc.). We collect the $|\nu|$ hypotheses in $\mathbf{H}$ and compute the corresponding residual matrix $\mathbf{R}$ and residual density matrix $\mathbf{D}$.

Algorithm 2: `generateHypotheses`

Input : $\mathbf{P}, \nu, \eta$
Output: $\mathbf{H}$, $\mathbf{R}$, $\mathbf{D}$

1 **foreach** $j \in \nu$ **do**
2 $s \leftarrow \emptyset$
3 $s \leftarrow \{s \cup x_j\}$
4 **for** $k \leftarrow 2$ *to* η **do**
5 $c \leftarrow \mathbf{P}_j$
6 $c_{\{s\}} \leftarrow 0$
7 $c \leftarrow \frac{c}{\Sigma c}$
8 $s \leftarrow \{s \cup \texttt{getSample}(c)\}$
9 **end**
10 $\mathbf{h} \leftarrow \texttt{getModel}(s)$
11 $\mathbf{H} = \{\mathbf{H} \cup \mathbf{h}\}$
12 **end**
13 $\mathbf{R} \leftarrow \texttt{computeResiduals}(\mathbf{H})$
14 $\mathbf{D} \leftarrow \texttt{computeDensity}(\mathbf{R})$

We combine the hypotheses generated and the corresponding matrices $\mathbf{R}$ and $\mathbf{D}$ in the current iteration with our previous set (Algo. 1, line 5). With a slight abuse of notation we use set union even with the matrices, only to emphasize uniqueness of rows after the update. We then use the updated density matrix to recompute the rows of point preference matrix $\mathbf{V}$ as described in Sec. 3.1.2. In order to identify the points in ν that are well explained by the hypotheses in $\mathbf{H}$, we introduce a measure τ_j, which is the sum of residual densities corresponding to the j^{th} point's top-T preferences (Algo. 1, line 8). If this measure does not change significantly (Algo. 1, line 9), we assume the point is well explained and remove it from the set ν. We update the PCM $\mathbf{P}$ for all points in ν (Algo. 1, line 10) and repeat the process (Algo. 1, lines 4-10) until the set ν is empty. Finally, we retain the unique set of hypotheses which are in the top-T preferences of all points and the corresponding residual and density matrices.

While it is difficult to *guarantee* the termination of Algo. 1 with an upper bound on the number of generated hypotheses, our experiments provide strong empirical evidence that it terminates in a few iterations. In Sec. 4, we report the total run time and show it to be competitive with most recent guided sampling approaches. The main advantage of taking this point-wise deterministic and iterative approach is the reduction in bias towards generating more hypotheses from structures with larger number of inliers. It is straightforward to combine DGS with additional constraints like a time budget or an initial prior on inliers.

3.3. Fraction Estimation

Instead of estimating the scale of inlier noise, we equivalently estimate the fraction of inliers for each of the generated hypotheses in $\mathbf{H}$. Given a model hypothesis, using its preference set $\mathbf{q}$, we reduce the inlier fraction estimation to the problem of finding an index $g \in \{1, \ldots, n\}$ that partitions $\mathbf{q}$ into an inlier subset $[q_i^1, q_i^2, ..., q_i^g]$ and an outlier subset $[q_i^{g+1}, ...q_i^{n-1}, q_i^n]$. The index g then marks the inlier/outlier boundary for the corresponding model hypothesis and the corresponding fraction of inliers is simply $\frac{g}{n}$.

We note that the boundary index will always lie beyond the index of maximum density, i.e., $g \geq k_1$, where $k_1 = \arg\max_j d_i^j$. We ignore extreme outliers by considering boundary candidates that have residuals smaller than a certain (adaptive) threshold. Let k_2 be the index having the largest residual smaller than $\mathrm{b}\rho_i^{2\eta}$, where b is a large constant (in our case, always 50), then we only consider boundary candidate indices in the set $\{k_1, \ldots, k_2\}$.

Since inliers are expected to have a significantly higher residual density than outliers, a reliable property of a candidate boundary point is a large density difference with respect to the maximum density, $\rho_i^{k_1}$. While considering moderately large outliers and inliers, the dispersion of residuals computed over a local neighborhood window w_j is expected to be large at the inlier/outlier boundary. Based on these intuitions, we define a confidence score using two factors: the normalized local dispersion of residual $\bar{\sigma}_i^j$ and the normalized density difference $\bar{d}_i^j$ (Algo. 3, line 4). For the i^{th} hypothesis, the boundary index g_i and the fraction f_i are computed as shown in Algo. 3, lines 5 and 6 respectively.

We emphasize that estimating the fraction is a very difficult problem, if not an ill-posed one. Instead of having a

Algorithm 3: `estimateFraction`

Input : $\mathbf{d}_i, \boldsymbol{\rho}_i, \eta, b$
Output: f_i, g_i

1 $\sigma_i^j \leftarrow \texttt{std}\big(\rho_i^{\{w_j\}}\big), \forall j \in \{1, \ldots, n\}$
2 $k_1 \leftarrow \arg max_j\ d_i^j$
3 $k_2 \leftarrow \arg max_j\ (\rho_i^j \leq \mathrm{b}\rho_i^{2\eta})$
4 $\bar{\sigma}_i^j \leftarrow \frac{\sigma_i^j}{\Sigma_{a=k_1}^{a=k_2}\sigma_i^a},\ \bar{d}_i^j \leftarrow \frac{|d_i^j - d_i^{k_1}|}{\Sigma_{a=k_1}^{a=k_2}|d_i^a - d_i^{k_1}|}\ \forall j \in \{k_1, \ldots, k_2\}$
5 $g_i \leftarrow \arg max_j(\bar{\sigma}_i^j \times \bar{d}_i^j) \quad \forall j \in \{k_1, \ldots, k_2\}$
6 $\mathrm{f}_i \leftarrow \frac{g_i}{n}$

complex fraction estimation technique, we employ a simple and intuitive heuristic which empirically yields reasonable results for good model hypotheses. We expect to make some errors in fraction estimation, however we do not attempt to fix it at this stage and rely on the downstream model selection module to reject such hypotheses with poor fraction estimates.

3.4. Model Selection

We use the fraction estimated by Algo. 3 and identify the inlier set and re-fit all models in $\mathbf{H}$ using residual density based weighted least-squares followed by re-estimating the inlier fraction. For each $\mathbf{h}_i \in \mathbf{H}$, the corresponding inlier set is obtained as $\mathcal{I}_i = \{q_i^1, q_i^2, ..., q_i^{g_i}\}$ from the hypothesis preference set $\mathbf{q}_i$. Due to the nature of the hypothesis generation process, there may be multiple model hypotheses that explain the same inlier structure. The goal of model selection is to retain the most representative model hypothesis and discard the redundant ones. To identify the best model, we need a measure to quantify the goodness of a model. Since the number of structures are not known a priori, we need to measure redundancy between model hypotheses before discarding them. For the latter, we estimate pairwise correlation between hypotheses by computing the Spearman-Footrule (SF) distance [28, 6] between their corresponding inlier only preference lists.

For a hypothesis pair $\mathbf{h}_i$ and $\mathbf{h}_j$, let $\bar{\mathbf{q}}_i = [q_i^1, q_i^2, ..., q_i^{\bar{g}}]$ and $\bar{\mathbf{q}}_j = [q_j^1, q_j^2, ..., q_j^{\bar{g}}]$ denote the top-$\bar{g}$ inlier only preference lists, where $\bar{g}$=min(g_i, g_j). The Spearman-Footrule distance is computed using (4), where $Y(\bar{\mathbf{q}}_i)$ denotes the data points with indexes in $\bar{\mathbf{q}}_i$ and $k^{\bar{\mathbf{q}}_i}$ denotes the position of the data point (k) in the preference list $\bar{\mathbf{q}}_i$. We use $k+1$ for $k^{\bar{\mathbf{q}}_i}$ if $k \notin Y(\bar{\mathbf{q}}_i)$. Variables for $\bar{\mathbf{q}}_j$ are similarly defined.

$$\mathcal{F}(\bar{\mathbf{q}}_i, \bar{\mathbf{q}}_j) = \Sigma_{k \in Y(\bar{\mathbf{q}}_i) \cup Y(\bar{\mathbf{q}}_j)} \; |k^{\bar{\mathbf{q}}_i} - k^{\bar{\mathbf{q}}_j}| \tag{4}$$

$$z_i^j = 1 - \frac{1}{\bar{g} \times (\bar{g}+1)} \mathcal{F}(\bar{\mathbf{q}}_i, \bar{\mathbf{q}}_j) \tag{5}$$

Pairwise hypothesis correlation between $\mathbf{h}_i$ and $\mathbf{h}_j$ is computed using (5) as $z_i^j \in [0, 1]$. A perfect correlation of $z_j^i = 1$ indicates that both $\mathbf{h}_i$ and $\mathbf{h}_j$ have identical inlier only preference lists, while $z_j^i = 0$ indicates completely dissimilar. We say a pair of hypotheses are similar if both have score z_j^i greater than some threshold δ. We construct a binary similarity matrix $\mathbf{C}$ by thresholding $z_j^i \geq \delta$.

For the goodness of the model, we define two scoring functions, both leveraging the residual density. The first score captures how densely are the inliers packed around the structure defined by the model. Since we have to compare this score across multiple hypotheses, simply using d_i^j does not suffice, so we compute the normalized density as $\hat{d}_i^j = \frac{d_i^j}{\mathcal{A}_i}$, where $\mathcal{A}_i$ is the total area under the residual density curve. The final score is computed as $\mathcal{S}_i = \frac{\sum_{k=1}^{k=g_i} \hat{d}_i^j}{\bar{r}_{g_i}}$ and a higher score implies a denser structure.

Algorithm 4: Greedy Model Selection (GMS)

Input : $\ell, \pi, \mathcal{S}, \mathbf{C}$
Output: ϑ
1 **Initialization:** $\vartheta \leftarrow \varnothing$
2 **while** $\ell \neq \varnothing$ **do**
3 $\quad i \leftarrow \text{argmax}_j \mathcal{S}_j, \quad \forall j \in \ell$
4 $\quad k \leftarrow \{j \,|c_i^j = 1\}, \quad \forall j \in \ell$
5 $\quad \varrho \leftarrow \text{argmax}_j \pi_j, \quad \forall j \in k$
6 $\quad \vartheta \leftarrow \{\vartheta \cup \varrho\}$
7 $\quad \ell \leftarrow \{\ell \setminus k\}$
8 **end**

Apart from absolute density, we want the second scoring function π to capture the disparity in the residual density of inliers and outliers. For a model hypothesis $\mathbf{h}_i$, we define $\pi_i = \frac{\text{med}\left(d_i^{[1:g_i]}\right)^2}{\text{med}(d_i^{\gamma})}$, where $\gamma_i = [g_i + 1, ..., \beta]$ is the index set of top-β outliers of $\mathbf{h}_i$. The complete greedy model selection (GMS) algorithm is explained in Algorithm 4. It takes inputs $\ell, \pi, \mathcal{S}$ and $\mathbf{C}$, where ℓ is the index set of all generated hypotheses. The algorithm first picks the densest hypothesis based on the highest score of $\mathcal{S}$ (Algo. 4, line 3) and identifies all similar hypotheses in the set k using the binary similarity matrix $\mathbf{C}$ (Algo. 4, line 4). Of all the similar hypotheses, we select the one that has the highest score π (Algo. 4, line 5). The best hypothesis retained in ϑ and the redundant ones removed from the universal set ℓ. The process is repeated until the set ℓ is empty. The final set of fitted models are in ϑ. At this stage, some of the data points may be members of multiple sets $\mathcal{I}_i$ and $\mathcal{I}_j$ for $i, j \in \vartheta$. This is acceptable for soft partitioning, however, we reassign the points based on the density to achieve hard partitioning of data points.

4. Experimental Results

We compare DGSAC[3] with Tlnk[13], Cov[16], NMU[21], RPA[14], DPA[22], RCM[18] and Jlnk[23] as a complete end-to-end multi-model fitting pipeline. We use publicly available implementation of the competing approaches and tune them for best performance as suggested in the respective publications. All competing approaches are provided the required user inputs (ϵ= inlier threshold,κ = no. of structures) tabulated in the table 1 computed from the ground truth.

Table 1: User input: (✓= Required, ×= Not Required)

	RPA[14]	**Tlnk**[13]	**RCM**[18]	**DPA**[22]	**Cov**[16]	**NMU**[21]	**DGSAC**
ϵ	✓	✓	✓	✓	✓	✓	×
κ	✓	✓	×	×	✓	×	×

[3]Source code available at https://bitbucket.org/lokender/dgsac

Table 2: [**Real Examples- Motion Segmentation**] Quantitative Analysis: Classification Accuracy(CA) in (%), Time in seconds, **out**= Outliers Percentage, κ = #Structures, μ=mean, *med*=median.

	biscuit	biscuitbook	bishopbox	boardgame	book	brdcartoychips	breadcube	brdcubechips	breadtoy	breadtoycar	carchipscube	cube	cubebrdtoychips	cubechips	cubetoy	dinobooks	game	gamebiscuit	toycubecar	CA(%)		Time(s)	
out	57.16	47.51	37.21	42.48	21.48	35.20	32.19	35.22	37.41	34.15	36.59	69.49	28.03	51.62	41.42	44.54	73.48	51.54	36.36	μ	*med*	μ	*med*
κ	1	2	3	3	1	4	2	3	2	3	3	1	4	2	2	3	1	2	3				
Tlnk[13]	83.09	97.77	88.80	83.73	82.57	80.51	85.62	82.00	96.81	84.70	88.00	46.29	80.18	95.14	78.80	78.56	77.60	70.61	70.70	81.65	82.57	12.81	11.69
RCM[18]	95.15	92.52	83.71	78.46	94.01	78.82	87.27	83.17	78.37	83.07	78.85	87.98	81.62	90.32	89.64	72.28	90.77	85.40	83.45	84.99	83.71	04.62	03.83
RPA[14]	98.36	96.42	95.83	87.53	97.54	91.73	95.95	95.57	97.15	92.17	94.30	97.15	93.21	96.48	96.31	84.78	95.97	96.95	91.70	94.47	95.95	39.25	38.75
DPA [22]	82.12	97.24	95.14	83.69	90.16	91.56	94.09	94.61	90.59	88.67	86.30	96.89	87.28	92.92	93.61	84.17	97.47	90.95	85.65	90.64	90.95	50.34	46.78
Cov[16]	98.35	97.58	93.99	77.82	97.16	87.34	95.92	88.59	82.37	89.18	88.72	97.12	90.68	93.59	95.50	68.66	92.39	95.52	82.07	90.14	92.39	54.71	47.29
NMU[21]	97.58	98.83	98.07	82.80	100	94.94	97.11	97.39	97.92	92.17	97.58	98.01	87.16	98.59	97.99	84.44	98.71	92.07	91.50	94.89	97.58	399.3	399.2
DGSAC	98.18	98.58	97.58	82.59	99.03	87.88	97.73	93.04	90.74	89.49	85.52	96.76	88.57	97.38	97.32	83.92	95.02	98.19	90.35	93.05	95.02	24.17	20.21

Table 3: [**Real Examples- Planar Segmentation**] Quantitative Analysis: Notations are same as of table 2.

	barrsmith	bonhall	bonython	elderhalla	elderhallb	hartley	johnsona	johnsonb	ladysymon	library	napiera	napierb	neem	nese	oldclassics	physics	sene	unihouse	unionhouse	CA(%)		Time(s)	
out	68.87	06.17	73.73	60.74	47.84	61.56	20.91	12.01	32.48	55.34	62.91	39.51	36.51	33.51	32.45	45.28	47.20	16.55	76.50	μ	*med*	μ	*med*
κ	2	6	1	2	3	2	4	7	2	2	2	3	3	2	2	1	2	5	1				
Tlnk[13]	57.93	60.41	64.34	69.53	57.80	71.63	57.80	70.72	77.72	82.51	81.26	67.76	53.03	53.70	73.77	68.49	84.32	71.86	77.29	68.99	70.72	492.5	81.33
RCM[18]	84.81	81.67	87.27	75.23	71.45	77.38	83.00	79.41	75.27	77.02	70.66	74.29	71.87	77.56	92.45	54.53	71.68	97.02	90.06	78.55	77.38	5.31	3.39
RPA[14]	62.90	52.88	84.34	99.07	81.96	81.38	91.10	66.84	79.16	63.53	73.25	75.14	78.51	99.21	76.73	100	99.44	88.00	76.14	80.50	79.16	967.2	247.19
DPA[22]	97.68	77.98	96.57	96.17	85.90	96.91	87.08	74.36	90.46	95.21	80.56	83.63	80.21	97.40	96.33	98.40	99.76	93.17	98.34	90.90	95.21	37.68	30.06
Cov[16]	70.74	68.62	99.74	77.92	82.76	91.75	86.12	65.18	93.83	92.92	86.62	74.05	72.61	90.77	79.20	99.51	80.40	91.19	99.45	84.18	82.76	145.9	53.19
NMU[21]	89.63	84.27	98.48	98.13	86.67	98.44	90.62	75.04	96.20	98.14	94.70	78.38	95.85	97.64	98.42	79.25	99.60	94.72	99.10	92.28	95.9	499.6	298.1
DGSAC	69.64	73.01	98.19	96.85	88.31	97.84	94.93	77.51	91.87	94.18	92.87	82.57	90.61	99.15	94.18	99.42	98.58	92.97	97.12	91.04	94.18	114.72	23.24

As is typical in multi-model fitting literature, we use *classification accuracy* (CA) as an evaluation metric i.e., percentage of data points correctly assigned to their corresponding structure or otherwise marked as gross outliers. All results are averaged over 10 runs. While we have also reported the running time, it is not a strictly fair comparison as the programming language varies across the competing approaches. DGSAC is implemented in Matlab therefore further improvement in running time is possible with an optimized implementation.

4.1. Real Examples

We show results on four real examples. 1. Motion segmentation, 2. Homography based planar segmentation, 3. Vanishing Point(VP) estimation and line classification and 4. Plane fitting to 3D point cloud.

Motion Segmentation, Planar Segmentation: We use AdelaideRMF [27] dataset, which comprises 19 image pairs each for homography based planar segmentation and fundamental matrix based motion segmentation. A quantitative analysis of DGSAC with the competing approaches for each of the 19 image pairs of motion segmentation and planar segmentation is reported in the Tables 2 and 3 respectively. NMU achieves the highest accuracy for both motion and planar segmentation experiments but it takes highest running time and requires inlier threshold from the user. DGSAC achieves the next best accuracy, lagging behind by a margin of $< 2\%$, without any user input at all. Moreover, in terms of average time to run, DGSAC is nearly $16\times$ faster than the NMU in motion segmentation, and $5\times$ in planar segmentation, with the median run times being even better. We report some sample qualitative results in Fig. 4 where point membership is color coded.

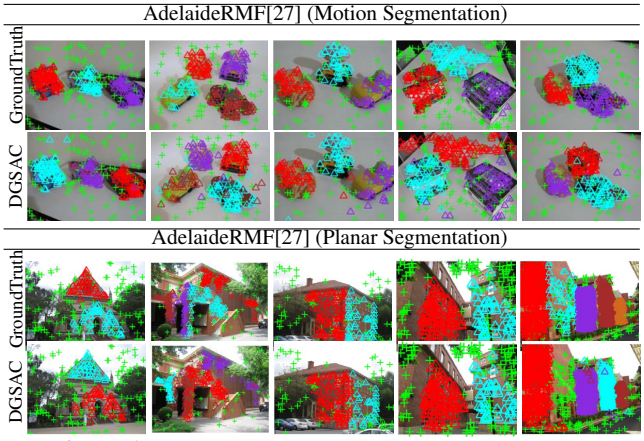


Figure 4: **Real Examples**: Sample Qualitative Results.

VP Estimation and Line Classification: We compare DGSAC with state of the art approaches RPA [14], Cov [16] and L1-NMF [20] for two vanishing point datasets. The York urban line segment [5] and the Toulouse Vanishing Points [1] datasets comprising 102 and 110 images of urban scenes respectively. We report the results in Table 4 and

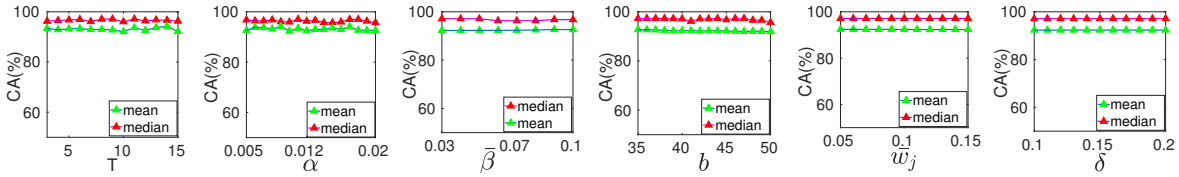


Figure 5: **Parameters Sensitivity Analysis** for AdelaideRMF motion segmentation dataset. Refer section 4.2 for details.

Fig. 6 and observe that DGSAC outperforms the competing approaches in both datasets.

Table 4: [**Real Examples- VP Estimation**] Quantitative Analysis: Notations are same as in table 2.

	York[5] Dataset				Toulouse[1] Dataset			
	CA(%)		**Time(s)**		**CA(%)**		**Time(s)**	
	μ	*med*	μ	*med*	μ	*med*	μ	*med*
RPA[14]	95.39	97.90	04.36	02.36	55.32	54.55	00.76	00.72
Cov[16]	95.57	97.44	01.24	00.26	51.78	50.00	00.04	00.07
L1-NMF[20]	94.12	96.66	00.71	00.31	74.13	75.00	00.07	00.54
DGSAC	95.66	97.96	07.75	01.89	91.58	95.35	00.24	00.23

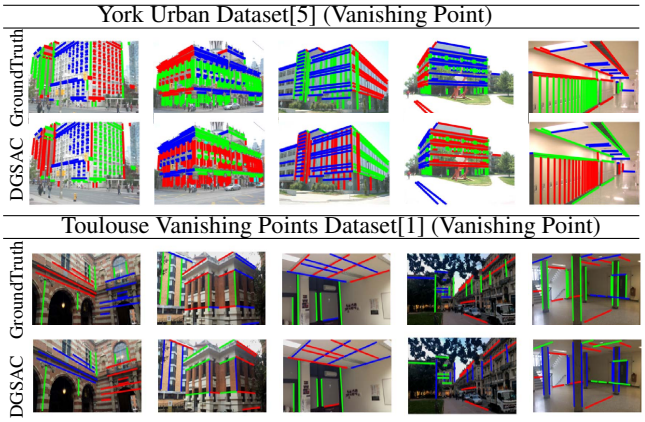


Figure 6: **Real Examples**: Sample Qualitative Results.

Plane Fitting to 3D Point Cloud: We use $CastelVechio$, $PozzoVeggiani$ and $PiazzaDante$ real examples of SAMANTHA [7] data set. Since ground truth labeling is not available for 3D point clouds, therefore, we presented only qualitative results in figure 7. Only JLnk [23] and DGSAC are able to recover planes correctly.

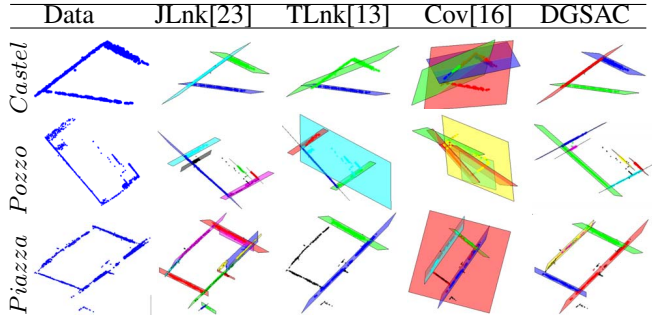


Figure 7: **Real Examples:** (Plane Fitting to 3D Point Cloud): Sample Qualitative Results. Point membership is color coded.

4.2. Analysis of Parameters

In the complete DGSAC pipeline, we have introduced several parameters, which adds to the complexity of the approach. However, our choice of parameters are based on making data-driven decisions and reasonable perturbations do not affect DGSAC's performance. All the parameters, specifically, T and α in Algo. 1, b and w_j in Algo. 3 and δ and β in the Sec. 3.4 are fixed for *all model types and experiments*. We set the values as $T = 5$, $\alpha = 0.01$, $b = 50$, $\delta = 0.15$, $\beta = \lfloor \bar{\beta} \times n \rfloor = \lfloor 0.05 \times n \rfloor$ and $\lfloor w_j \rfloor = \lfloor \bar{w}_j \times n \rfloor = \lfloor 0.1 \times n \rfloor$. We emphasize that DGSAC is not very sensitive to these parameters and plot the mean and median accuracy on motion segmentation AdelaideRMF dataset in the figure5 as empirical evidence.

5. Discussion & Conclusion

Inlier noise scale (inlier fraction) and the number of inlier structures are the two most important parameters in the multi-model fitting process. For methods like TLnk [13] and Cov [16], scale estimate is required for each image pair, which makes the multi model fitting process inefficient and limits it applicability where automatic execution is a requirement. NMU [21] is currently the best performing method for motion and planar segmentation but requires an inlier threshold and is very sensitive to it (as shown in the experiments of the original paper).

We reiterate that if the disparity between inlier and outlier density is absent, it is impossible to distinguish between inliers and outliers. DGSAC utilizes the residual density as a key differentiator between inliers and outliers and presents a solution for all the modules of a multi-model fitting pipeline. Using the residual density based sampling, DGSAC generates more relevant hypotheses and performs model selection by employing residual density based scoring functions. We believe that residual density is a key feature in multi-model fitting tasks and plan to continue our investigations. Specifically, we plan to get a better understanding of termination conditions of the density guided sampling algorithm. We also plan to work on improving the heuristic for inlier fraction estimation by exploring non-parametric hypothesis testing techniques.

6. Acknowledgment

The authors thank the Infosys Center for Artificial Intelligence, IIIT-Delhi for this supporting this project.

References

[1] V. Angladon, S. Gasparini, and V. Charvillat. The toulouse vanishing points dataset. In *Proceedings of the 6th ACM Multimedia Systems Conference (MMSys '15)*, Portland, OR, United States, Mar. 2015.

[2] T.-J. Chin, P. Purkait, A. Eriksson, and D. Suter. Efficient globally optimal consensus maximisation with tree search. In *CVPR*, pages 2413–2421, 2015.

[3] T.-J. Chin, J. Yu, and D. Suter. Accelerated hypothesis generation for multistructure data via preference analysis. *TPAMI*, 34(4):625–638, 2012.

[4] O. Chum and J. Matas. Matching with prosac-progressive sample consensus. In *CVPR*, volume 1, pages 220–226. IEEE, 2005.

[5] P. Denis, J. H. Elder, and F. J. Estrada. Efficient edge-based methods for estimating manhattan frames in urban imagery. In *European conference on computer vision*, pages 197–210. Springer, 2008.

[6] R. Fagin, R. Kumar, and D. Sivakumar. Comparing top k lists. *SIAM Journal on discrete mathematics*, 17(1):134–160, 2003.

[7] M. Farenzena, A. Fusiello, and R. Gherardi. Structure-and-motion pipeline on a hierarchical cluster tree. In *ICCV Workshops*, pages 1489–1496. IEEE, 2009.

[8] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[9] V. Fragoso, P. Sen, S. Rodriguez, and M. Turk. Evsac: Accelerating hypotheses generation by modeling matching scores with extreme value theory. In *ICCV*, December 2013.

[10] D. Kuang, S. Yun, and H. Park. Symnmf: nonnegative low-rank approximation of a similarity matrix for graph clustering. *Journal of Global Optimization*, 62(3):545–574, 2015.

[11] T. Lai, H. Wang, Y. Yan, G. Xiao, and D. Suter. Efficient guided hypothesis generation for multi-structure epipolar geometry estimation. *CVIU*, 2016.

[12] T. Lai, H. Wang, Y. Yan, and L. Zhang. A unified hypothesis generation framework for multi-structure model fitting. *Neurocomputing*, 222:144–154, 2017.

[13] L. Magri and A. Fusiello. T-linkage: A continuous relaxation of j-linkage for multi-model fitting. In *CVPR*, pages 3954–3961, 2014.

[14] L. Magri and A. Fusiello. Robust multiple model fitting with preference analysis and low-rank approximation. In *BMVC*, volume 20, page 12, 2015.

[15] L. Magri and A. Fusiello. Scale estimation in multiple models fitting via consensus clustering. In *International Conference on Computer Analysis of Images and Patterns*, pages 13–25. Springer International Publishing, 2015.

[16] L. Magri and A. Fusiello. Multiple model fitting as a set coverage problem. In *CVPR*, pages 3318–3326, 2016.

[17] D. Nasuto and J. B. R. Craddock. Napsac: High noise, high dimensional robust estimation-its in the bag. 2002.

[18] T. T. Pham, T.-J. Chin, J. Yu, and D. Suter. The random cluster model for robust geometric fitting. *TPAMI*, 36(8):1658–1671, 2014.

[19] T. Sattler, B. Leibe, and L. Kobbelt. Scramsac: Improving ransac's efficiency with a spatial consistency filter. In *ICCV*, pages 2090–2097. IEEE, 2009.

[20] M. Tepper and G. Sapiro. Fast l1-nmf for multiple parametric model estimation. *arXiv preprint arXiv:1610.05712*, 2016.

[21] M. Tepper and G. Sapiro. Nonnegative matrix underapproximation for robust multiple model fitting. *arXiv preprint arXiv:1611.01408*, 2016.

[22] L. Tiwari, S. Anand, and S. Mittal. Robust multi-model fitting using density and preference analysis. In *ACCV*, pages 308–323. Springer, 2016.

[23] R. Toldo and A. Fusiello. Robust multiple structures estimation with j-linkage. *ECCV*, pages 537–547, 2008.

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

[25] H. Wang, J. Cai, and J. Tang. Amsac: An adaptive robust estimator for model fitting. In *2013 IEEE International Conference on Image Processing*, pages 305–309, 2013.

[26] H. Wang, T. J. Chin, and D. Suter. Simultaneously fitting and segmenting multiple-structure data with outliers. *IEEE TPAMI*, 34(6):1177–1192, June 2012.

[27] H. S. Wong, T.-J. Chin, J. Yu, and D. Suter. Dynamic and hierarchical multi-structure geometric model fitting. In *ICCV*, pages 1044–1051. IEEE, 2011.

[28] H. S. Wong, T.-J. Chin, J. Yu, and D. Suter. A simultaneous sample-and-filter strategy for robust multi-structure model fitting. *CVIU*, 117(12):1755–1769, 2013.

[29] M. Zuliani, C. S. Kenney, and B. Manjunath. The multi-ransac algorithm and its application to detect planar homographies. In *ICIP*, volume 3, pages III–153. IEEE, 2005.