A New Approach to Robust Estimation of Parametric Structures

Xiang Yang Peter Meer *Fellow, IEEE* and Jonathan Meer

Abstract—Most robust estimators require tuning the parameters of the algorithm for the particular application, a bottleneck for practical applications. The paper presents the Multiple Input Structures with Robust Estimator (MISRE), where each structure, inlier or outlier, is processed independently. The same two constants are used to find the scale estimates over expansions for each structure. The inlier/outlier classification is straightforward since the data is processed and ordered with the relevant inlier structures listed first. If the inlier noises are similar, MISRE's performance is equivalent to RANSAC-type algorithms. MISRE still returns the correct inlier estimates when inlier noises are very different, while RANSAC-type algorithms do not perform as well. MISRE's failures are gradual when too many outliers are present, beginning with the least significant inlier structure. Examples from 2D images and 3D point clouds illustrate the estimation.

Index Terms—scale estimation, density based classification, structures segmentation

1 Introduction

This paper describes the Multiple Input Structures with Robust Estimator (MISRE). MISRE has three advantages relative to other robust estimators. First, each structure, inlier or outlier, is processed independently. Second, the same two constants are used in every estimation, rather than being specified by the user. Finally, its failures are gradual when too many outliers are present in the data, with the stronger structure(s) still recovered. MISRE's performance is equivalent to RANSAC-type algorithms when inlier scales are similar, but MISRE is superior when inlier scales are very different.

A *structure* is defined as the estimated points in an iteration. Inlier structures have an *objective function* which can be linear, e.g., a 3D plane, or nonlinear, e.g., a homography between two 2D images. The outliers do not have a defined configuration. Robust estimators are NP-hard [6] and have to be approximated in an algorithm. The building block in robust regressions is the *elemental subset*. An elemental subset is a randomly chosen minimum number of input points required to estimate the objective function. The returned parameters are correct only for inlier structures.

RANdom SAmple Consensus (RANSAC) [15] was the first algorithm for robust estimation in computer vision. Before estimation, the user must specify a scale for the inliers. RANSAC can fail if there are multiple inlier structures, if an image is resized, if in a sequence of images the scale changes greatly, or if the relevance of a hypothesis is not explicitly considered [19]. Modern cameras and sensors generally keep the inlier scales small and therefore predictable; the scale is generally not explicitly given. However, the scale threshold is always present in the code, even if the user is not aware of it.

The literature on variants of RANSAC is enormous. Reviews published in [8], [36] on PROSAC [9], MLESAC [45], Lo-RANSAC [10] describe how these algorithms use different ways to generate random sampling and/or probabilistic relationships. More recently, binomial constraints [4], maximum consensus [25], [41], graph-cut RANSAC [1], latent-RANSAC [23], convolutional neural networks for robust estimation [5], [31], have been tried.

The scale threshold for inliers can be based on Gaussian distributions in an universal framework for RANSAC (USAC) [35]. Using a statistical distribution for the inliers is not valid at all times, and USAC was outperformed using probabilistic reasoning [26].

Two algorithms, J-linkage [44] and T-linkage [28], use RANSAC to obtain the inliers by clustering through an iterative process. Sec.6.1 shows that these two algorithms cannot handle the estimation of 2D lines if the inlier scales are different.

A soft-thresholding RANSAC [30] obtains the correct result with 90% outliers. In a more sophisticated example, [46] use RANSAC combined with a structure from motion algorithm and extended Kalman filter to tolerate 60% outliers. However, the validity of these algorithms depends on the constants chosen, necessitating input from the user. As another approach, the k-th ordered absolute residual can yield useful results [47], but only if the number of inlier structures is specified before estimation [13], a major limitation.

Propose Expand and Re-estimate Labels (PEaRL) applies an energy-minimization-based procedure to computer vision [21]. Beginning with RANSAC, alternative steps of expansion for inlier classification and re-estimation of the errors are done in sequence. The Random Cluster Model SAmpler (RCMSA) [34] is similar, but uses simulated annealing for the energy minimization.

In some cases, constants are found only in the code and

• *X. Yang, Dept. of Mechanical and Aerospace Engineering.*
Present address: *ACME, Muyun Industrial Zone, Changsha City, Hunan Province, 410118, China. E-mail: xiang.yang@yahoo.com.*
P. Meer, Dept. of Electrical and Computer Engineering, Rutgers University, NJ 08854, USA. E-mail: meer@soe.rutgers.edu
J. Meer, Dept. of Economics, Texas A&M, TX 77843, USA. E-mail: jmeer@tamu.edu

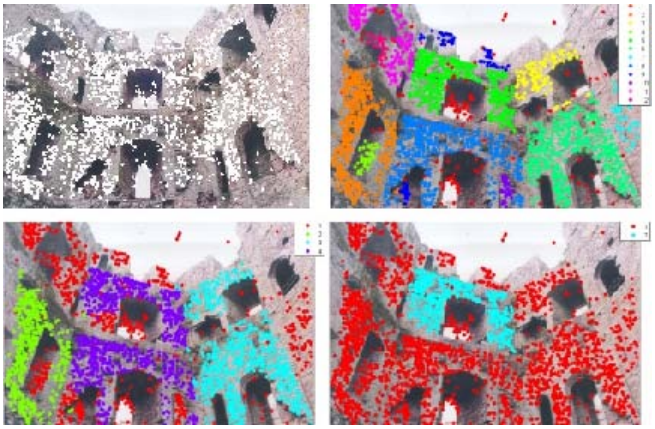


Fig. 1. Homography estimation in RCMSA [34]. Top left: first image with input points. Top right: second image classified with model complexity value 10. Bottom left: model complexity 50. Bottom right: model complexity 100.

must be adjusted to tailor an algorithm for a particular task. For example, model complexity in RCMSA influences the estimation, shown for *Raglan castle* (Fig.1). The required model complexity for homography is 10, but for fundamental matrix, it is 100. The estimator does not work if the model complexity is not correctly chosen.

Building a robust estimator with each inlier structure estimated independently fell short in [29]. The generalized projection-based M-estimator (gpbM) localized the inlier scales in dense regions using a cumulative distribution type function computed with *all points* still active. But the estimator uses only a small, given percentage of the points [29, Fig.3], making it not completely independent between the structures. The algorithm stops once the processing fell below an other given scalar constant.

While the goal of robust estimators is the same, to find inlier structures with maximum point supports while thresholding out outliers, the aforementioned RANSAC-type estimators rely on application-dependent thresholds. The Multiple Input Structures with Robust Estimator (MISRE) takes a different approach, using the same two constants to identify structures and succeeds in estimating the scale of each structure, inlier or outlier, independently. The algorithm first rewrites the nonlinear objective function of the input into a linear function. The set-up of the estimation for the linear expression is presented in Section 2.

The detailed algorithm is described in Section 3. In *all* the 2D or 3D experiments, MISRE uses the same two constants to estimate the scales. The number of elemental subsets is given by the user, but MISRE is not sensitive to this value above a certain level, as shown in Section 3.5.

In Section 4, several applications are presented, both for 2D images and 3D point clouds. The use of MISRE in the structure from motion algorithm (SfM) is also described.

Like all robust estimators, MISRE fails if the number of outliers increases above a limit defined by the input data. Section 5 shows that failures start with the least significant inlier structure, since the structures are estimated independently.

Section 6.1 compares the performance of several other robust estimators to MISRE. Extensions of MISRE are sketched in Section 6.2.

2 From Input to Parameter Estimation

When a nonlinear objective function $f(\mathbf{y})$ is transformed into a linear function $\mathbf{x}^T\theta - \alpha$, the products of the elements of an input measurement also become separate variables. The linear function's coefficients are called *carriers*

$$f(\mathbf{y}) \quad \longrightarrow \quad \mathbf{x}^\top \boldsymbol{\theta} - \alpha = \sum_{i=1}^{m} x_i \theta_i - \alpha \tag{1}$$

and the parameters are transformed into the vector $\boldsymbol{\theta}$ and scalar α. The number of unknowns in $\boldsymbol{\theta}$ are equal to the number of unknowns derived from $f(\mathbf{y})$.

For example, $f(\mathbf{y})$ is a nonlinear objective function for an ellipse

$$f(\mathbf{y}) = (\mathbf{y} - \mathbf{y}_c)^\top \mathbf{Q} (\mathbf{y} - \mathbf{y}_c) - 1 \tag{2}$$

if $\mathbf{Q}$ is a symmetric 2×2 positive definite matrix and $\mathbf{y}_c$ is the position of the ellipse center. From the input variable $\mathbf{y} = [x \;\; y]^\top \in \mathbb{R}^2$, dimension $l = 2$, the carrier vector $\mathbf{x} = [x \;\; y \;\; x^2 \;\; xy \;\; y^2]^\top \in \mathbb{R}^5$, dimension $m = 5$, is obtained. The condition $4\theta_3\theta_5 - \theta_4^2 > 0$ has to be satisfied for (2) to represent an ellipse.

A single input $\mathbf{y}$ can result in multiple carrier vectors

$$\mathbf{x}^{[c]\top} \boldsymbol{\theta} - \alpha \qquad c = 1, \ldots \zeta \tag{3}$$

corresponding to ζ different $\mathbf{x}^{[c]}$. For example, the objective function of 2D homography

$$\mathbf{f}(\mathbf{y}) = [x' \;\; y' \;\; 1]^\top - \mathbf{H} \, [x \;\; y \;\; 1]^\top \tag{4}$$

connects the projective coordinates of two planes in two 2D images and the 3×3 matrix $\mathbf{H}$ has to be found. The input variable $\mathbf{y} = [x \;\; y \;\; x' \;\; y']^\top$ has $\zeta = 2$ carrier vectors because there are x and y correspondences.

The set of equations in (3) are set equal to zero for an elemental subset. $m_e = \lceil \frac{m}{\zeta} \rceil$ are needed input points to define $\boldsymbol{\theta}$ and α. The intercept α is the average projection of the m carrier vectors $m^{-1} \sum_{c=1}^{\zeta} \sum_{i=1}^{m_e} \mathbf{x}_i^{[c]\top} \boldsymbol{\theta}$. An ellipse $(l = 2, m = 5, \zeta = 1)$ needs five points. A homography $(l = 4, m = 8, \zeta = 2)$ needs four point pairs to give eight correspondences. The constraint $\boldsymbol{\theta}^T \boldsymbol{\theta} = 1$ reduces the ambiguity of $\boldsymbol{\theta}$ to orthonormal matrices. The input is normalized and each obtained structure is mapped back to the original space [18, Sec.4.4.4].

An inlier structure has an $l \times l$ covariance matrix $\sigma^2 \mathbf{C}_\mathbf{y}$, where σ is unknown. The σ can change with each structure. The matrix $\mathbf{C}_\mathbf{y}$ has to be provided before estimation, which is possible only if there is additional information about the inliers. Otherwise, the inliers are set as independent and identically distributed with $\mathbf{C}_\mathbf{y}$ equal the identity matrix $\mathbf{I}_{l \times l}$. Section 6.2 sketches a solution for finding $[\sigma_1^2 \ldots \sigma_l^2] \mathbf{I}_{l \times l}$.

The $\mathbf{y}$ and $\mathbf{x}^{[c]}$ define the $m \times l$ Jacobian matrix for a carrier vector. Each column of the Jacobian matrix contains the derivatives of the m carriers with respect to one of the l input measurements. For nonlinear objective functions, the Jacobian

depends on the input points. For example, the transpose of the 5×2 Jacobian matrix for the ellipse is

$$\mathbf{J}^{\top}_{\mathbf{x}_i|\mathbf{y}_i} = \begin{bmatrix} 1 & 0 & 2x_i & y_i & 0 \\ 0 & 1 & 0 & x_i & 2y_i \end{bmatrix}. \quad (5)$$

The $m \times m$ covariance of a carrier vector $\sigma^2\mathbf{C}_i^{[c]}$, with $\mathbf{C}_{\mathbf{y}} = \mathbf{I}_{l\times l}$, is

$$\sigma^2\mathbf{C}_i^{[c]} = \sigma^2\mathbf{J}_{\mathbf{x}_i^{[c]}|\mathbf{y}_i}\,\mathbf{J}^{\top}_{\mathbf{x}_i^{[c]}|\mathbf{y}_i} \quad (6)$$

with the scale σ of the structure unknown.

For a $\boldsymbol{\theta}$, the c-s carrier vector is projected to the scalar $z_i^{[c]} = \mathbf{x}_i^{[c]\top}\boldsymbol{\theta}$. The variance of $z_i^{[c]}$ is $\sigma^2 H_i^{[c]} = \sigma^2\boldsymbol{\theta}^{\top}\mathbf{C}_i^{[c]}\boldsymbol{\theta}$. The Mahalanobis distance, without the scale σ, specifies how far the projection $z_i^{[c]}$ is from α

$$\begin{aligned} d_i^{[c]} &= \sqrt{\left(\mathbf{x}_i^{[c]\top}\boldsymbol{\theta} - \alpha\right)^{\top}\left(H_i^{[c]}\right)^{-1}\left(\mathbf{x}_i^{[c]\top}\boldsymbol{\theta} - \alpha\right)} \quad (7)\\ &= \frac{|\mathbf{x}_i^{[c]\top}\boldsymbol{\theta} - \alpha|}{\sqrt{\boldsymbol{\theta}^{\top}\mathbf{C}_i^{[c]}\boldsymbol{\theta}}} \geq 0 \qquad c = 1,\dots\zeta \quad i = 1,\dots n \end{aligned}$$

being zero for the elemental subset.

Each input point $\mathbf{y}_i$ gives a ζ-dimensional vector

$$\mathbf{d}_i = \left[\, d_i^{[1]} \;\; \dots \;\; d_i^{[\zeta]} \,\right]^{\top}. \quad (8)$$

To be conservative, we retain the *largest Mahalanobis distance* $d_i^{[\tilde{c}_i]} = \tilde{d}_i$ among the ζ values

$$\tilde{c}_i = \underset{c=1,\dots,\zeta}{\arg\max}\; d_i^{[c_i]}. \quad (9)$$

The carrier vector $\mathbf{x}_i^{[\tilde{c}_i]} = \tilde{\mathbf{x}}_i$ yields the covariance matrix $\widetilde{\mathbf{C}}_i$, the scalar projection $\tilde{z}_i$ with variance $\widetilde{H}_i$, and the largest Mahalanobis distance $\tilde{d}_i$ (without σ). Since the variance is the same for each component of $\mathbf{y}_i$, it does not matter which d_i is chosen for $\tilde{d}_i$. For the same $\boldsymbol{\theta}$, but a different i, the $\tilde{d}_i$ can be different and the Mahalanobis distances are therefore no longer rotational invariant. For each structure, inlier or outlier, the scale σ has to be estimated.

3 Multiple Structures Recovery

Each structure corresponds to an iteration with $n \leq n_T$ points, where n_T is the total number of data points.

The largest Mahalanobis distances $\tilde{d}_i$, $i = 1,\dots,n$, are used in each of the M elemental subset trials. The user specifies M, as discussed in Section 3.5. MISRE uses the minimum sum of ascendingly ordered Mahalanobis distances, corresponding to a *fixed percent* of the entire data, as the starting point for the expansion for a structure. The current expansion terminates when the average number of points in the processed segments is at least twice the number of points in the next segment. When the next expansion no longer can begin, the current structure's expansions end. The largest Mahalanobis distance in the region of interest is the scale estimate. The iterations continue until insufficient data remains and then all the structures are sorted. This strategy separates the scale estimates of the inliers from those of the outliers if there are not too many outliers present.

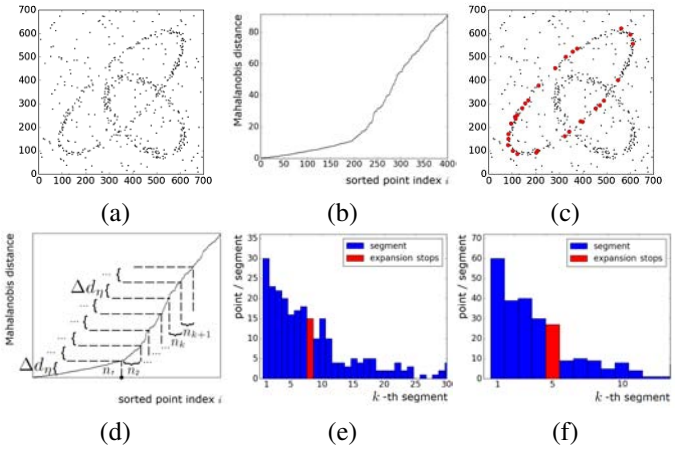


Fig. 2. Two synthetic ellipses and outliers. (a) Input data. (b) Mahalanobis distances of the first 400 points of the first $\tilde{d}_{[i](w)}$. (c) The n_ϵ points in red in the structure. (d) Stylized example of a $\tilde{d}_{[i](w)}$ sequence. (e) Expansion with Δd_5. (f) Expansion with Δd_{10}.

An iteration consists of three steps: scale estimation (Section 3.1), refinement with mean shift (Section 3.2) and finding the density of the structure (Section 3.3). The same two constants are used for every scale estimate. When too few data points remain for a new scale estimation, the structures are classified by their densities (Section 3.4). Multiple Input Structures with Robust Estimator (MISRE) does *not* distinguish between inlier and outlier structures; designating structures as inliers or outliers is done by the user.

3.1 Scale Estimation

The *first* MISRE constant is the number n_ϵ, satisfying two conditions. First, it is five percent of the total data, $0.05n_T$; second, it must be five times larger than the number of points in an elemental subset [18, p.182]. The second condition applies when the elemental subset in large, e.g., 2D homography, and the number of total samples is small. The value of n_ϵ is the same for all the iterations. The five percent condition is conservative, since estimated inlier structures normally make up more than five percent of the data. Section 3.5 shows that this level is an adequate starting point.

For an elemental subset, the largest Mahalanobis distances $\tilde{d}_i$, $i = 1,\dots,n$, are sorted in ascending order $\tilde{d}_{[i]}$. The M different elemental subsets give $\tilde{d}_{[i](j)}$, $j = 1,\dots,M$ sequences. The sequence with the *minimum sum* of Mahalanobis distances for n_ϵ points

$$\min_{j\in M}\sum_{i=1}^{n_\epsilon}\tilde{d}_{[i](j)} \quad (10)$$

defines the *working* sequence $\tilde{d}_{[i](w)}$, $i = 1,\dots,n$, with the parameters $\hat{\boldsymbol{\theta}}_w$ and $\hat{\alpha}_w$. For a sufficiently large M, the sorted points at the beginning of the working sequence come from the same structure, if there still is a significant inlier structure. Subsequent structures have different $\tilde{d}_{[i](w)}$, until the number of points becomes less than n_ϵ.

Two synthetic ellipses, each with $n_{in} = 200$ inlier points, along with and $n_{out} = 200$ outliers, are used to illustrate the MISRE algorithm throughout this section (Fig.2a). The inlier

points are corrupted by Gaussian noise with $\sigma_g = 5, 10$, but the specific distribution is not relevant. The number of elemental subsets is $M = 2000$ per iteration.

From the total of 600 input points, the first 400 sorted Mahalanobis distances for the first $\tilde{d}_{[i](w)}$ are shown in Fig.2b. The $n_\epsilon = 0.05 \times 600 = 30$ points are drawn in red for the structure with $\sigma_g = 5$ (Fig.2c).

Divide the sequence $\tilde{d}_{[i](w)}$, $i = 1, \ldots, n$, into equal Mahalanobis distances of Δd_η, where Δd_η corresponds to the first η percentage of the points (Fig.2d). The number of points in the first segment is $n_1 = \eta\% \, n_T$. For an η, the expansion increases each time by one Δd_η. The k-thΔd_η segment has n_k points.

The *second* MISRE constant is the *ratio_to_stop* (*r_to_s*), equal to two: if an expansion's average number of points in the of already-processed segments is larger than *twice* the number of points in the next segment, the expansion finishes

$$\frac{1}{k}\sum_{i=1}^{k} n_i > 2\, n_{k+1} \qquad k = 1, 2, 3 \ldots \tag{11}$$

When (11) is satisfied, the boundary between the structure and the rest of the points has been found, $k = k_{t_\eta}$. Section 3.5 shows that MISRE's performance is not meaningfully affected by different values of *r_to_s*.

Due to the randomness of the data, the scale estimate cannot be obtained only from one expansion. In Fig.2e, the expansion with Δd_5 stops at $k_{t_5} = 8$ (red bar) giving $\hat{\sigma} = 8.06$. In Fig.2f, the expansion with Δd_{10} stops at $k_{t_{10}} = 5$, giving $\hat{\sigma} = 11.10$.

The *region of interest* is defined from $\Delta d_\eta = \Delta d_5$, the lowest limit in the scale estimation, to the first η where Δd_η no longer can expand. If the first expansion begins at $\eta > 5\%$, the region of interest begins there as well.

Each iteration starts from the same number of points corresponding to n_ϵ eliminating the potential bias between a structure detected first and a structure detected later. The independent expansions increase by 1% each time to $\eta = 6\%, 7\% \ldots$ where the percentages correspond to the total number of data points.

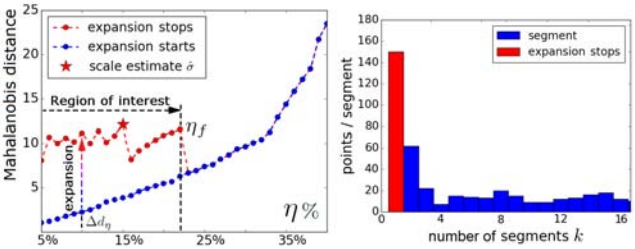
In Fig.3a, the blue points indicate the Mahalanobis distances corresponding to the length of Δd_η. An expansion process ends at the red point where condition (11) is met. The region of interest in the example is from Δd_5 to Δd_{η_f}, which is 22% in this example. Condition (11) holds for $\eta = 23\%$ and Δd_{23} can no longer expand, as seen in Fig.3b.

The estimated scale is the largest Mahalanobis distance in the region of interest

$$\hat{\sigma} = \max_{\eta=5\%,\ldots,\eta_f} k_{t_\eta} \Delta d_\eta. \tag{12}$$

In Fig.3a, the scale estimate is $\hat{\sigma} = 12.54$ with $n_{\hat{\sigma}}$ points between $\hat{\alpha}_w \pm \hat{\sigma}$. The scale estimate returned in an iteration is the maximum among many scales in the region of interest. This is similar to a nonparametric bootstrap-type estimator because only the σ-s in the region of interest participate [12].

Fig.2b is a typical example for an inlier structure. Once η is large enough that the second segment of Δd_η takes points from the steeper slope given by outliers (Fig.3a), the region of interest will end shortly (Fig.3b).



(a) (b)

Fig. 3. Finding the scale estimate. (a) Independent expansions in the region of interest. (b) For $\eta\% = 23\%$, Δd_η no longer can expand.

3.2 Refinement with Mean Shift

From the $n_{\hat{\sigma}}$ data points falling *inside* the scale estimate (12) another $N = M/10$ elemental subsets are generated. Since $\hat{\sigma}$ is known, this number of trials is sufficient.

The complete variance of $\tilde{z}_i$ is

$$\widetilde{B}_i = \hat{\sigma}^2 \widetilde{H}_i = \hat{\sigma}^2 \boldsymbol{\theta}^\top \tilde{\mathbf{C}}_i \boldsymbol{\theta} = \hat{\sigma}^2 \boldsymbol{\theta}^\top \mathbf{J}_{\tilde{\mathbf{x}}_i|\mathbf{y}_i} \mathbf{J}_{\tilde{\mathbf{x}}_i|\mathbf{y}_i}^\top \boldsymbol{\theta}. \tag{13}$$

All the $n > n_{\hat{\sigma}}$ data points participate in the mean shift [11]. With $\tilde{z}_i = \tilde{\mathbf{x}}_i^\top \boldsymbol{\theta}$, $i = 1, \ldots, n$, the Epanechnikov kernel has the profile for nonnegative squared Mahalanobis distances

$$\kappa(u) = \begin{cases} 1 - u & (z - \tilde{z}_i)^\top \widetilde{B}_i^{-1} (z - \tilde{z}_i) \le 1 \\ 0 & (z - \tilde{z}_i)^\top \widetilde{B}_i^{-1} (z - \tilde{z}_i) > 1. \end{cases} \tag{14}$$

Mean shift returns the modes of the function

$$\arg\max_{\hat{\alpha}} \sum_{i=1}^{n} \kappa\left((z - \tilde{z}_i)^\top \widetilde{B}_i^{-1} (z - \tilde{z}_i) \right) \tag{15}$$

and we look for the *closest mode* from α, the scalar estimate.

The derivative $g(u) = -\kappa'(u)$

$$g(u) = 1 \quad 0 \le u \le 1 \qquad g(u) = 0 \quad u > 1 \tag{16}$$

and all the points inside a window contribute equally. Let the current value be $z = z_{old}$. The next value z_{new} is computed by taking the gradient of (15) equal to zero

$$z_{new} = \left[\sum_{i=1}^{n} g\left(u_i\right) \right]^{-1} \left[\sum_{i=1}^{n} g\left(u_i\right) \tilde{z}_i \right] \tag{17}$$

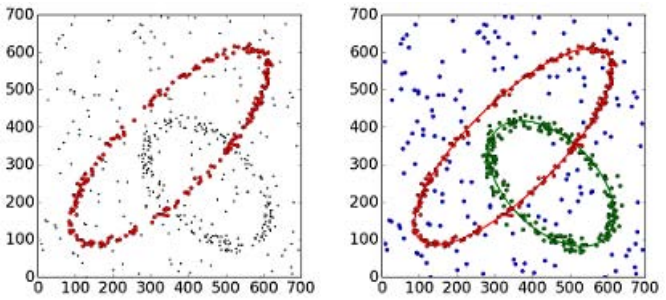
with $\tilde{z}_i$-s more distant from z_{old} than $\pm\sqrt{\widetilde{B}_i}$ having weights $g(u_i)$ equal to zero. Many of the n points do not converge. The mode estimate comes from the elemental subset whose window at convergence has the most points $\tilde{z}_i$, giving $\hat{\alpha}$.

From the points converging to $\hat{\alpha}$, the nonrobust total least squares (TLS) estimate finds $\hat{\boldsymbol{\theta}}^{tls}$, $\hat{\alpha}^{tls}$ and $\hat{\sigma}^{tls}$. There are n_{st} points in the region $\hat{\alpha}^{tls} \pm \hat{\sigma}^{tls}$. In Fig.4a the estimated structure has $n_{st} = 219$ points in red and $\hat{\sigma}^{tls} = 12.37$.

3.3 Density of the Structure

The density for a structure is the ratio between the number of points in the structure and the TLS scale of the structure

$$\rho = \frac{n_{st}}{\hat{\sigma}^{tls}}. \tag{18}$$



(a) (b)

Fig. 4. Recovered structures after mean shift. (a) First structure. (b) The three recovered structures.

For Fig.4a the density is $\rho = 17.7$. The n_{st} points are removed from the input and the processing of a next structure begins.

3.4 Sorting Based on Density

The processing continues until the remaining input data becomes smaller than n_ϵ. The detected structures are sorted in *descending* order based on the densities. Until this point, we did not distinguish between inlier and outlier structures.

In Fig.4b three structures were estimated

	red	*green*	*blue*
nr. points :	219	210	163
TLS *scale* :	12.37	28.4	708.7
density :	17.7	7.4	0.23

The significant inlier structures are first because the inlier scale estimates are much smaller than the outlier scale estimates. If there are too many outliers present in the data, there can be interaction between the inliers and outliers. However, this occurs in a different way than in RANSAC, as discussed in Section 5.

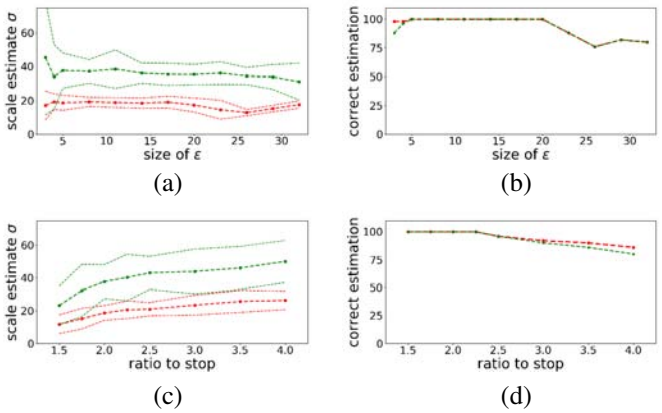
The user must specify the cutoff between inlier structures and outliers after estimation by noting where the increase in $\hat{\sigma}^{tls}$ is substantial and marked from one structure to the next. This is equivalent with the number of returned inlier structures. If the number of outliers is not too great, the increase in $\hat{\sigma}^{tls}$ is *always* much larger when moving from the weakest inlier structure to the outliers.

In the example above, there is a large jump in $\hat{\sigma}^{tls}$ from the second (green) to the third (blue) structure, indicating that the third structure is outliers.

3.5 Sensitivity to Parameter Values

The constants used in MISRE, ϵ and *ratio_to_stop* (*r_to_s*), provide a wider range of robustness than other algorithms. This is illustrated in Fig.5, which shows how estimation varies with values of ϵ and *r_to_s*. There are two inlier structures, the ellipses in Fig.2a: $n_{in} = 200$ with $\sigma_g = 5$, drawn in red, and $\sigma_g = 10$, drawn in green. There are $n_{out} = 200$. Each inlier structure is 33% of the data. Five times each ellipse's elemental subset, $5 \times 5 = 25$ points, are only equal to 4.17% of the total data. Therefore $n_\epsilon = 0.05 \times 600 = 30$. Each measurement is based on 100 trials.

In Fig.5a ϵ varies from 3% to 32%, while *r_to_s* is set to 2 and $M = 2000$. A structure containing fewer than 50% inlier points is considered "missed." Below 5%, the second condition



(a) (b) (c) (d)

Fig. 5. The ϵ and *r_to_s* with $\pm\sigma$, using Fig.2a as the example. The red and green graphs correspond to ellipses with different levels of noise, described in Section 3.5. (a) *r_to_s* = 2, $M = 2000$ and ϵ from 3% to 32%. (b) The output function of ϵ. (c) $\epsilon = 5\%$, $M = 2000$ and *r_to_s* from 1.5 to 4. (d) The output function of *r_to_s*.

for n_ϵ is not satisfied. For $\epsilon = 3\%$ the standard deviation is large and MISRE can return a few structures which are correct but consist of fewer points. The $\sigma_g = 10$ structure is missed several times (Fig.5b). For $\epsilon = 4\%$ and higher, the standard deviation is stable and the output is correct. Once the estimator begins with expansions larger than 20%, however, the two inlier structures are not always returned correctly (Fig.5b). If the starting size ϵ is too small, it may not capture an inlier structure. If ϵ is large enough that it is close to the number of points of the inlier structure, it may begin to capture outliers as well. As such, we choose 5% as the minimum to start the expansions. In practice, though, an inlier structure does contain such a small percentage of the data.

In Fig.5c, the *r_to_s* varies from 1.5 to 4, while $\epsilon = 5\%$ and $M = 2000$. When an inlier structure is analyzed, *r_to_s* establishes the border between the inliers and the outliers. As *r_to_s* increases beyond 2.25, not every trial ends correctly (Fig.5d), though performance does not degrade quickly nor catastrophically, even as *r_to_s* doubles. The structure with $\sigma_g = 10$ has somewhat more errors. Performance diminishes as it becomes more difficult to distinguish the difference between inliers and outliers with larger *r_to_s*. We choose the conservative *r_to_s* = 2, which errs on the side of detecting inliers.

The number of elemental subsets M depends on the size of the input data, the complexity of the objective function, the noise levels of the inlier structures, the interaction among inlier structures, the amount of outliers, etc. If there is no additional information about the inlier structures, no theoretical M can be set.

The M varies here from 50 to 5000, while $\epsilon = 5\%$ and *r_to_s* = 2.

M	50	100	200	500	1000	2000	5000
correct	56	78	84	94	100	100	100

The user should set an initial value for M and experiment with larger values. Once the results of an estimation do not differ for increasing values of M, further increases do not

refine the solution. We choose $M = 2000$.

When the estimation gives a stable result, the estimate can only be improved through more elaborate pre-processing. If the number of outliers is reduced without significantly reducing the number of inliers, the estimation begins from a "better" input. For example, [40] shows that when the number of consistent matches increases before estimation, the estimator was better than PROSAC [9].

In real scenes, the inlier points are very rarely sufficiently separated from the outliers. An estimator that can take a wide range of inputs cannot achieve bechmark performances in all situations, even if the thresholds are adjusted. MISRE estimates the correct inlier structures in a large variety of conditions. However, the output may not return 100% of the structure; post-processing with specific thresholds is needed to achieve a particular output.

3.6 Pseudocode of the MISRE Algorithm

Input: $\mathbf{y}_i$, $i = 1, \dots, n_T$. Covariance $\mathbf{C}_\mathbf{y} = \mathbf{I}_\mathbf{y}$.
M elemental subsets for each structure.
Output: Sorted structures.

- For $\mathbf{y}_i$, $i = 1, \dots, n_T$ compute the carriers $\mathbf{x}_i^{[c]}$ $c = 1, \dots, \zeta$ and the Jacobians $\mathbf{J}_{\mathbf{x}_i^{[c]}|\mathbf{y}_i}$.

$\odot$ n input points. M elemental subsets.

  - For each $\boldsymbol{\theta}$ and α find:
    - The largest Mahalanobis distance $\tilde{d}_i$.
    - Sort $\tilde{d}_i$ in ascending order, $\tilde{d}_{[i]}$.
  - Minimum sum of Mahalanobis distances for n_ϵ points, $\tilde{d}_{[i](w)}$ the working sequence for $i = 1, \dots, n$.

- Region of expansions are from n_ϵ points until the first η with no expansion. The largest expansion is $\hat{\sigma}$.
- Between $\hat{\alpha}_w \pm \hat{\sigma}$ there are $n_{\hat{\sigma}}$ points.
- $n_{\hat{\sigma}}$ points define another $N = 0.1M$ elemental subsets.
  - Mean shift trials use $n > n_{\hat{\sigma}}$ points:
    - Find the closest mode to α for each trial.
  - The strongest mode is $\hat{\alpha}$.
  - TLS estimate given by the converged points.
  - Between $\hat{\alpha}^{tls} \pm \hat{\sigma}^{tls}$ there are n_{st} points.
  - Compute the density of the structure $\rho = n_{st}/\hat{\sigma}^{tls}$.
  - Remove n_{st} points from the input.
- Back to $\odot$ until fewer than n_ϵ points remain.
- Sort by decreasing densities. Return the results.

The program is available at

`https://github.com/MISRE`

written in Python/C++ with a few 2D and 3D examples. For robust processing in 3D, access to a complete structure from motion (SfM) algorithm or the Autodesk commercial software program is also needed.

4 Experiments

In RANSAC-type estimators, users must specify parameters before estimation, which can differ depending on the task. MISRE always uses the same constants, making the algorithm more straightforward. When using real images, the scales of the inlier structures are often similar and small. Both RANSAC-type estimators and MISRE will correctly estimate the number of inlier structures in these cases, as will be seen in Section 6.1.

In other data, like the synthetic experiments shown below, the scales can be very different for each inlier structure. Section 6.1 will show how RANSAC-type estimators can fail in such cases. Further, while some RANSAC-type algorithms assume Gaussian noise, e.g. [35], MISRE does not take the type of noise distribution into account.

Examples from 2D images and 3D point clouds are estimated here with MISRE. Lines, ellipses, fundamental matrices, and homographies are the 2D examples. For the 3D examples, planes, spheres, and cylinders are estimated. A 3D point cloud is obtained from a sequence of 2D images, either by structure from motion (SfM) algorithm or by the software program *ReMake* from Autodesk [37]. The interaction between SfM and MISRE for estimation of the 3D point cloud is described in Section 4.5.

Pre-processing and post-processing are *not part* of MISRE. Pre-processing increases the number of inliers relative to the number of outliers. Post-processing takes the given output and attempts either to fuse several similar inlier structures which are judged by the user to be a single structure, or to recover additional inlier structures. These processes are context-specific with thresholds and therefore are not part of the generality of MISRE. The fusion problem is discussed at the end of Section 4.4.

The processing times are measured with an i7-2617M 1.5GHz processor. The estimated structures are colored in this order: red, green, blue, cyan, yellow, purple, sorted by descending densities.

4.1 Lines Estimated in 2D

A 2D line has a linear objection function

$$f(\mathbf{y}) = \theta_1 x + \theta_2 y - \alpha \tag{19}$$

with the input variable $\mathbf{y} = [x \; y]^\top$ identical to $\mathbf{x}$, the carrier vector. $M = 1000$ for all 2D line experiments.

Five synthetic lines with $n_{in} = 300, 250, 200, 150, 100$ inlier points are corrupted with 2D Gaussian noise, $\sigma_g = 3, 6, 9, 12, 15$ and these are $n_{out} = 350$ outliers (Fig.6a). The "weakest" line has $n_{in} = 100$ points and is corrupted with $\sigma_g = 15$. The estimation stops when the number of points is less than $n_\epsilon = 68$.

In Fig.6a the lines intersect and the expansions can pick-up alternatively one of the lines. The scale estimate being the maximum in the region of interest this gives the correct value for one of the lines.

In Fig.6b, six estimated structures are identified

	red	*green*	*blue*	*cyan*	*yellow*	*purple*
nr. points :	321	282	240	161	106	240
TLS scale :	9.6	18.7	28.1	37.1	44.2	370.8
density :	33.4	15.1	8.5	4.3	2.4	0.6

where the first five are inlier structures, followed by outliers with $\hat{\sigma}^{tls}$ being much larger than the others.

Running 100 tests, the first four lines are correctly segmented every time and the "weakest" line having the smallest

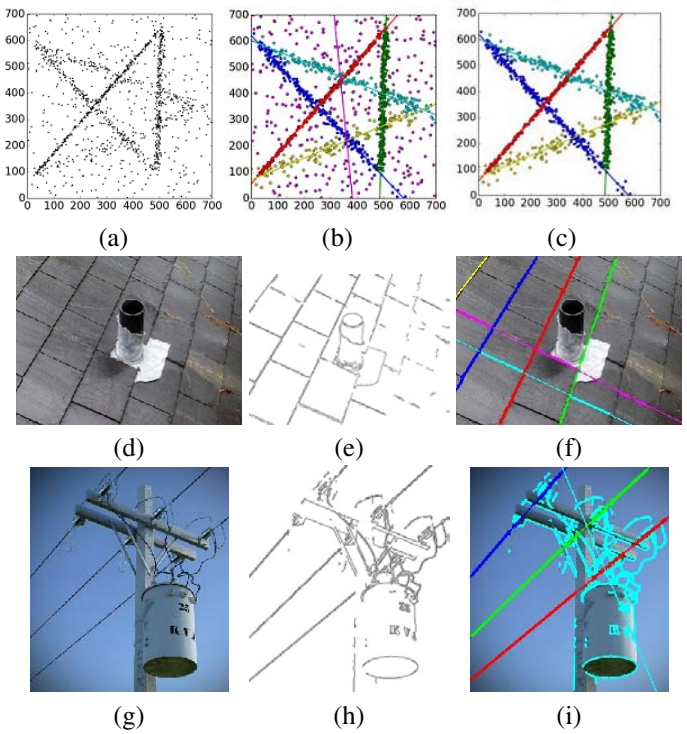


Fig. 6. Estimation of lines in 2D. (a) $n_{out} = 350$. Five synthetic inlier lines. (b) First five structures are inliers followed by outliers. (c) The inlier structures. (d) *Roof*. (e) Canny edges. (f) First six structures are inliers. (g) *Pole*. (h) Canny edges. (i) First three structures are inliers followed by outliers.

density becomes outliers in six out of 100 estimations. The average processing time is 0.58 seconds.

The *roof* image (Fig.6d) and the *pole* image (Fig.6g) extract similar-sized input data with Canny edge detection, with 8310 in Fig.6e and 8072 points in Fig.6h. The estimations stop when the number of points are less than $n_\epsilon = 416$ or 404, respectively.

For the *roof*, the first six structures, shown in Fig.6f, are inliers. The purple and cyan lines are not continuous in the roof image itself. The processing time is 7.44 seconds. The outliers have several short lines and an ellipse. The short lines might be recovered by post-processing. In Section 6.2, we will sketch how more than one type of inlier structures can be recovered through multiple estimation.

For the *pole*, the first three inlier structures are followed by outliers in Fig.6i. The processing time is 4.35 seconds. The outliers are more diverse. A few shorter lines around the two wooden crossbars might be estimated by post-processing. The two ellipses might be recovered by multiple estimation.

4.2 Ellipses Estimated in 2D

Ellipse estimation was introduced in Section 2. The input variable $\mathbf{y}$ has a nonlinear objective function

$$f(\mathbf{y}) = (\mathbf{y} - \mathbf{y}_c)^\top \mathbf{Q} (\mathbf{y} - \mathbf{y}_c) - 1 \tag{20}$$

satisfying several conditions. $\mathbf{y} = [x \;\; y]^\top$ gives the carrier vector $\mathbf{x} = [x \;\; y \;\; x^2 \;\; xy \;\; y^2]^\top$. Ellipse estimation is biased, especially if only the large curvature part of an ellipse is given.

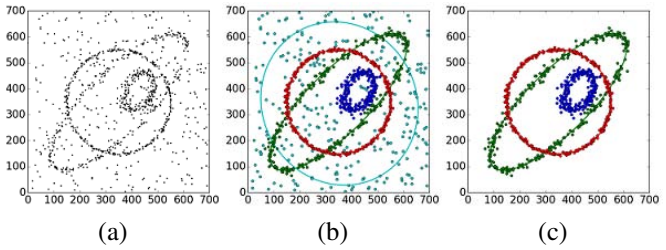


Fig. 7. Estimation in 2D of three synthetic ellipses. (a) $n_{out} = 350$. (b) First three structures are inliers followed by outliers. (c) The inlier structures.

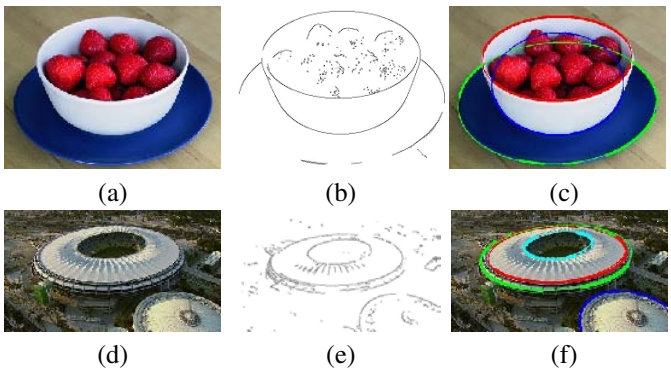


Fig. 8. Ellipses in 2D real images. (a) *Strawberries*. (b) Canny edges. (c) First three are inlier structures. (d) *Stadium*. (e) Canny edges. (f) First four structures. See the text also.

Taking the covariance matrix of the Gaussian inlier noise for each point into account does not eliminate the bias, e.g., [22], [42].

To avoid classifying line segments as very flat ellipses, we assume the major axis cannot be more than 10 times longer than the minor axis. The Jacobian matrix was given in (5). For all ellipse experiments, $M = 5000$.

Three synthetic ellipses with $n_{in} = 300, 250, 200$ inlier points are corrupted with Gaussian noise, $\sigma_g = 3, 6, 9$ and $n_{out} = 350$ outliers (Fig.7a). The smallest ellipse has $n_{in} = 200$ and is corrupted with the largest noise $\sigma_g = 9$. The estimation stops when the number of points is less than $n_\epsilon = 55$.

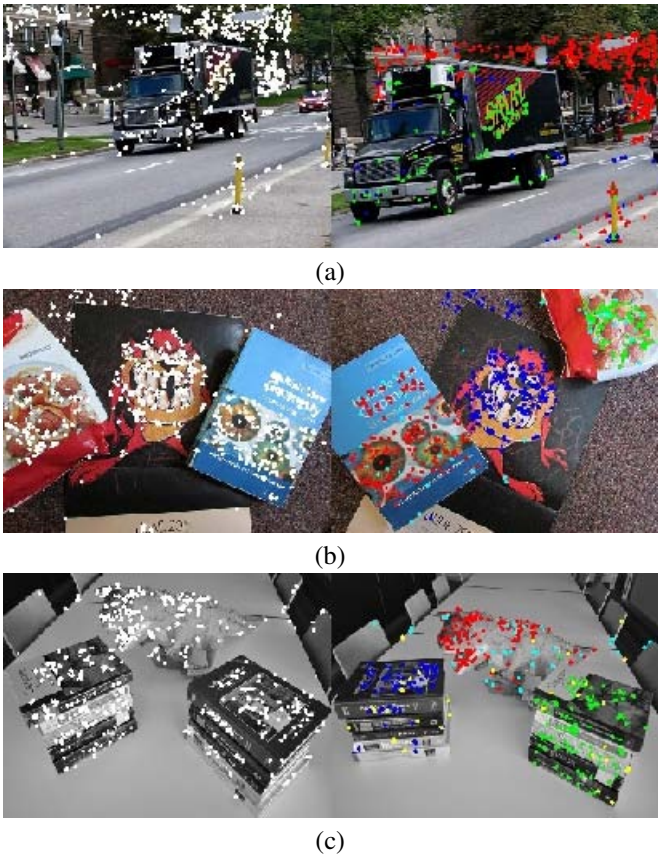
The four structures in Fig.7b have three inlier structures followed by outliers (Fig.7c). The nonlinear transformation of the input influences the estimated inlier scales.

	red	*green*	*blue*	*cyan*
nr. points :	337	292	222	248
TLS scale :	12.1	28.9	48.0	1321.2
density :	28.0	10.1	4.6	0.2

Repeating the test 100 times, the smallest ellipse (blue) becomes outliers ten times, the middle ellipse (green) becomes outliers six times, while the strongest ellipse (red) is always correct. The average processing time is 3.28 seconds.

The *strawberries* image (Fig.8a) and the *stadium* images (Fig.8d) extract similar-sized input data with Canny edge detection, with 4343 in Fig.8b and 4579 input points in Fig.8e. The estimations stop when the number of points are less than $n_\epsilon = 218$ or 229, respectively.

The first three inlier structures for the *strawberries* are



(a) (b) (c)

Fig. 9. Fundamental matrices estimation. (a) *Truck* on a street. (b) *Books* on a table. (c) *Dinabooks* from [34]

drawn in Fig.8c. The processing time is 18.90 seconds. The blue ellipse is estimated based only on its support.

The first four structures for *stadium* are shown in Fig.8f. The processing time is 23.14 seconds. By running 100 test, with only with the elemental subsets changing, the first two ellipses (red and green) are estimated reliably 98 times. The other two ellipses (blue and cyan) are less stable and only pre-processing can help to obtain better results.

4.3 Estimation of Fundamental Matrices

The nonlinear objective function for the fundamental matrix

$$f(\mathbf{y}) = [x' \quad y' \quad 1] \ \mathbf{F} \ [x \quad y \quad 1]^\top \tag{21}$$

connects projective point correspondences between two 2D images and the estimation returns a 3×3 matrix $\mathbf{F}$ of rank-2.

The two 2D images are projections from 3D scene. Objects staying together in 3D have one structure in 2D. Objects moving separately in 3D need separate fundamental matrices in 2D. In [34, Sec. 6.2.2], fundamental matrices are estimated under the title "Two-view motion segmentation".

If only quasi-translational motions are present in 3D, only homography can be used instead of fundamental matrix estimation [2]. See also [49, Sec. 4.2.1]. Some algorithms consider this problem but do not explicitly address it [34].

The input variable $\mathbf{y} = [x \ y \ x' \ y']^\top$ has an eight-dimensional carrier vector $\mathbf{x} = [x \ y \ x' \ y' \ xx' \ xy' \ x'y \ yy']^\top$. The transpose of the 8×4 Jacobian matrix is

$$\mathbf{J}^\top_{\mathbf{x}_i|\mathbf{y}_i} = \begin{bmatrix} 1 & 0 & 0 & 0 & x'_i & y'_i & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & x'_i & y'_i \\ 0 & 0 & 1 & 0 & x_i & 0 & y_i & 0 \\ 0 & 0 & 0 & 1 & 0 & x_i & 0 & y_i \end{bmatrix}. \tag{22}$$

Eight point pairs are required for the 8-point algorithm to define $\boldsymbol{\theta}$ and α. The processing is in the projective framework and additional information is needed to recover the Euclidean framework.

All fundamental matrix experiments have $M = 5000$. The scale-invariant feature transform (SIFT) [27] is used for the point correspondences. The distance ratio is 0.8 and some correspondences are outliers. For repetitive features use [38].

The three example, *truck* in Fig.9a, *books* in Fig.9b, and *dinobooks* in Fig.9c, have 608, 614, and 457 point pairs, respectively. The estimations stop when the number of points are less than $n_\epsilon = 40$, five times the required points for a fundamental matrix. For each pair, the image on the left shows all the processed points, while the image on the right shows the overlayed structures. The tables below report the results in the same order as the three examples.

	red	*green*	*blue*		*red*	*green*	*blue*	*cyan*
nr. points :	407	101	51	||	192	96	221	47
TLS scale :	0.56	0.73	11.78	||	0.46	0.40	1.12	10.42
density :	727.3	139.3	4.3	||	413.4	242.8	196.8	4.5

	red	*green*	*blue*	*cyan*	*yellow*
nr. points :	135	117	84	48	43
TLS scale :	0.22	0.72	0.65	0.70	23.9
density :	623.0	161.5	129.0	68.2	1.8

The processing times are between 1.75 and 2.3 seconds. Since all inlier scale estimates are small, a scale value of, for example, two pixels for RANSAC returns the correct output in all three cases. MISRE's advantage is in estimating each structure separately with the same two constants.

In Fig.9a, the first two structures are inliers followed by outliers. The red structure is only the camera's action on the static background. The green structure is the truck.

In Fig.9b, the first three structures are inliers followed by outliers. The red and green structures are moved separately. The blue structure also contain points from the static background of the table, which is to be expected since it did not shift much.

In Fig.9c, the first four structures are inliers followed by outliers. The blue and green structures are moved separately. The first (red) and the fourth (cyan) inlier structures might be fused together in post-processing. With more elaborate preprocessing, the points which are now outliers (yellow) around the blue structure, might be detected as belonging to the blue structure.

4.4 Homographies Estimated in 2D

The nonlinear objective function for 2D homography

$$\mathbf{f}(\mathbf{y}) = [x' \quad y' \quad 1]^\top - \mathbf{H} \ [x \quad y \quad 1]^\top \tag{23}$$

connects the projective coordinates between two 2D image planes and the 3×3 matrix $\mathbf{H}$ has to be found.

The homography is based on projections from 3D scenes. Well-separated planes in 3D may not be well-separated in 2D and the homography will return just a single plane correspondence; see below for an example.

The input variable $\mathbf{y} = [x \quad y \quad x' \quad y']^\top$ give two carrier vectors $\mathbf{x}^{[1]}$, $\mathbf{x}^{[2]}$ for x and y correspondences, where $(\boldsymbol{\theta}, \alpha)$ is $\text{vec}\,\mathbf{H}^\top = \mathbf{h}$

$$\begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & x'x & x'y & x' \\ 0 & 0 & 0 & -x & -y & -1 & y'x & y'y & y' \end{bmatrix} \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{bmatrix} \quad (24)$$

The homography has $\alpha = 0$ and the transposes of 9×4 Jacobian matrices are

$$\mathbf{J}^\top_{\mathbf{x}_i^{[1]}|\mathbf{y}_i} = \begin{bmatrix} -\mathbf{I}_{2\times 2} & & x_i'\mathbf{I}_{2\times 2} & \mathbf{0}_2 \\ \mathbf{0}_2^\top & \mathbf{0}_{4\times 4} & x_i \quad y_i & 1 \\ \mathbf{0}_2^\top & & \mathbf{0}_2^\top & 0 \end{bmatrix}$$

$$\mathbf{J}^\top_{\mathbf{x}_i^{[2]}|\mathbf{y}_i} = \begin{bmatrix} & -\mathbf{I}_{2\times 2} & & y_i'\mathbf{I}_{2\times 2} & \mathbf{0}_2 \\ \mathbf{0}_{4\times 3} & \mathbf{0}_2^\top & \mathbf{0}_4 & \mathbf{0}_2^\top & 0 \\ & \mathbf{0}_2^\top & & x_i \quad y_i & 1 \end{bmatrix}. \quad (25)$$

For a $\boldsymbol{\theta}$ the larger Mahalanobis distance, $\tilde{d}_i$, is used for each $\mathbf{y}_i$, $i = 1, \ldots, n$. For all 2D homography experiments, $M = 2000$.

Two examples are from the *Hopkins 155* dataset, the *street* in Fig.10a and the *table* in Fig.10b. These 2D image pairs have clear homographies because they have relatively small translations in 3D.

The SIFT correspondences return 990 and 482 point pairs, respectively. The estimations stop when the number of points are less than $n_\epsilon = 50$ or 40, where the latter number is bound by the minimum number required for estimation. For each pair, the image on the left shows all the processed points, and the image on the right shows the overlayed structures.

	red	*green*	*blue*	*cyan*	|| *red*	*green*	*blue*	*cyan*
nr. points :	713	101	67	105	|| 107	88	165	89
TLS scale :	1.62	1.12	4.49	203.11	|| 0.20	0.17	0.56	5.16
density :	440.2	89.5	14.9	0.5	|| 529.6	517.7	293.4	17.3

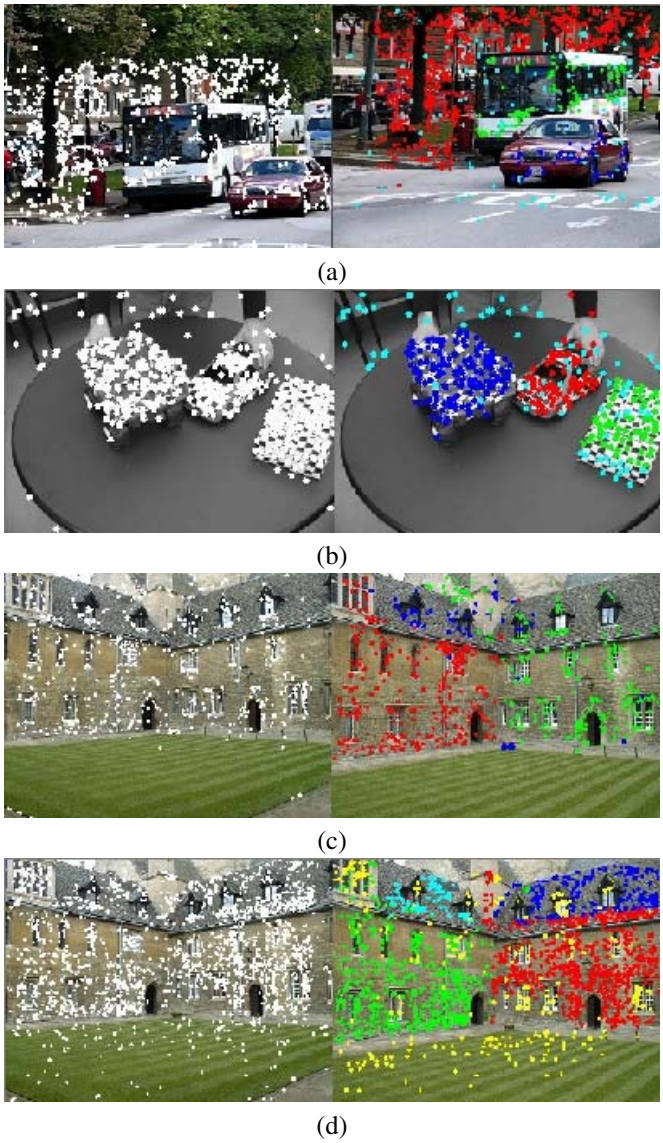
The processing times are 1.12 and 1.09 seconds.

In Fig.10a the first three structures are inliers followed by outliers. The two orthogonal 3D planes on the bus are estimated as one plane in 2D. A user-given RANSAC scale of two pixels ($\sigma = 2$) will not capture all of the inlier points on the car (the blue structure), which has $\hat{\sigma}^{tls} = 4.49$.

In Fig.10b, the first three structures are inliers followed by outliers. The quasi-affine viewpoint results in the objects being described with only one plane each. Several outliers (cyan) are around the green structure. Better pre-processing might recover these outliers as part of the green structure.

The importance of having more inliers is illustrated in Fig.10c and Fig.10d with the *Merton College 2* image pair from the *Oxford Visual Geometry Group* archives. The SIFT gives 713 and 1940 point correspondences, respectively, with the second correspondence equal to the value given in [34]. MISRE continues until the number of points are less than $n_\epsilon = 40$ or 97.

Both MISRE and RCMSA [34] return the same number



(a) (b) (c) (d)

Fig. 10. Homography estimation in 2D. (a) *Street*. (b) *Table*. (c) *Merton College 2*. 713 point pairs. (d) 1940 point pairs.

of estimated inlier structures. In Fig.10c, MISRE estimates the first two structures as inliers followed by outliers (blue). In Fig.10d, the first four structures are inliers followed by outliers (yellow). If there are more inlier points in the input, more inlier structures can be detected.

A single inlier structure, as judged by the user, can appear as several similar inlier structures in the figure. Post-processing with user-specified thresholds is required. The carriers do not explicitly represent the nonlinearities and the post-processing should be executed in the input space. The examples in this paper did not require fusion.

For two lines or two planes, the orientations of two structures and the distance between them is sufficient to determine the thresholds for fusion. For two 2D ellipses, determining the overlap area is enough [20]. The fundamental matrices and 2D homographies are executed in a projective framework from scenes projected from 3D. Without additional information about the 3D relation of the two 2D images, recovery of an euclidean framework is not possible [18, Chap.19].

4.5 Structure from Motion

A structure from motion algorithm (SfM) starts with a 2D sequence of images and ends with a 3D point cloud [14], [16]. See also [49, Chap.5]. The programs are taken from [48]. In this subsection we just want to show how MISRE interacts with SfM when the 3D point cloud is built.

The procedure is represented in Fig.11a from [39]. The construction of SfM is incremental, starting with small parts of the 3D scene defining 3D tracks, then fusing the tracks into a 3D euclidean point cloud.

Seventy 2D images, called also as 2D frames, are selected automatically from a video taken around a *lamp post*. A few images are shown in Fig.11b.

The point correspondences are found by SIFT followed by a hierarchical k-means tree algorithm [32]. All the 2D frames have the same inlier vs. outlier threshold for the correspondence between a pair of 2D images. Distant pair of 2D images are above the threshold and are discarded. Fundamental matrix estimation with MISRE (Section 4.3) is used to eliminate most of the outliers. For example, an image pair with 379 correspondences after the matching are reduced to 314 correspondences after MISRE.

Choose the lowest average error between two 2D frames with a rotation of at least 5°. In our case, the two frames are close in the sequence. Compute the 3D projective coordinates for all matches in the chosen 2D image pair. Starting with this initial 2D pair, do 2D-3D bundle adjustment taking frames on both sides [18, Appen.6]. Once the average reprojection error for a frame is larger than one pixel, the track expansion terminates. For us, a 3D track has maximum 10 to 15 consecutive frames in 2D. In total, 10 overlapping tracks in 3D are obtained from the 70 frames in 2D (Fig.11c).

Each track has a different 3D coordinate system. To merge them, each track has to be multiplied with a 3D homography. A MISRE approach similar to Section 4.4 is used in 3D [49, p.72]. The cheirality of the tracks has to be also checked [18, Chap.21]. For example, the overlap between tracks 4 and 5 (Fig.11d) is reduced from 1247 to 1096 point pairs after the fusion (Fig.11e).

Projective distortions can remain after hierarchical merging around both ends of two fused tracks. The final 2D-3D bundle adjustment is executed with all the 2D frames and camera definitions participating, obtaining the 3D point cloud (Fig.12d).

A stereo algorithm [17] [51] can significantly increase the 3D point cloud after the SfM is finished. Currently, MISRE cannot process such large clouds [49, p.75].

Obtaining the 3D point cloud from the sequence of 2D images can be very long; therefore, the processing time begins when the estimation from the 3D point cloud begins.

4.6 Planes Estimated in 3D

The 3D plane has linear objective function

$$f(\mathbf{y}) = \theta_1 X + \theta_2 Y + \theta_3 Z - \alpha \tag{26}$$

with the input data $\mathbf{y} = [X\ Y\ Z]^\top$ identical with $\mathbf{x}$, the carrier vector. $M = 1000$ for all 3D plane estimation.

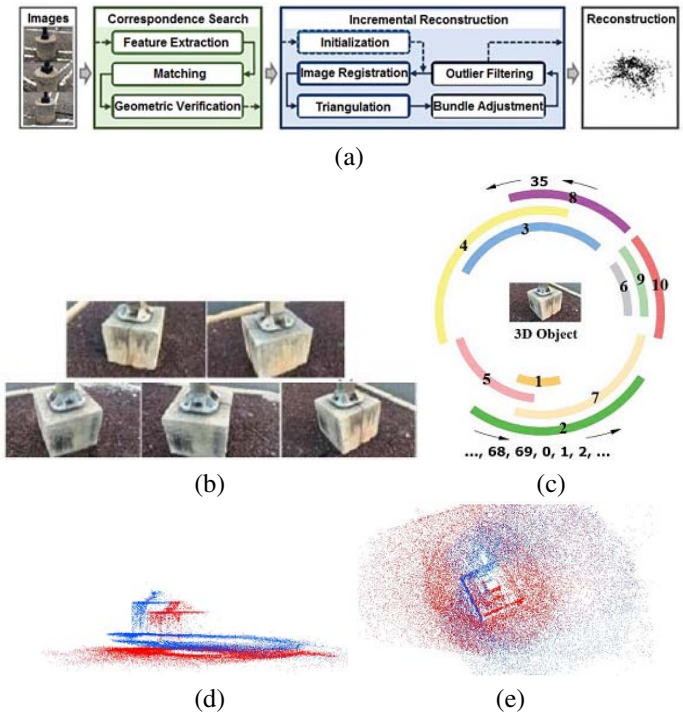


Fig. 11. Structure from motion. (a) The procedure. (b) Few 2D images of the *lamp post*. (c) Different tracks. (d) Track 5 (blue) plotted in the coordinated system of track 4 (red). (e) The merged two tracks seen from the top.

The synthetic pyramid shown in Fig.12a has 5000 points in the 3D point cloud, distributed around five planes (Fig.12b). The base has the most points. The pyramid has length one in all three dimensions. The points are corrupted with 3D Gaussian noise $\sigma_g = 0.01$ and there are no outliers. The type of noise distribution is not taken into account. The estimation stops when the number of points is less than $n_\epsilon = 250$.

In Fig.12c the densities are very large.

	red	$green$	$blue$	$cyan$	$yellow$
$nr.\ points:$	2205	827	817	581	570
$TLS\ scale:$	0.038	0.032	0.037	0.033	0.034
$density/10^3:$	58.0	25.8	22.0	17.6	16.6

The processing time is 1.70 seconds. The base is almost not visible in Fig.12c. In 100 tests the estimation segments the five inlier structures every time.

When the Gaussian noise increases to $\sigma_g = 0.03$, the estimator returns six inlier structures, with one plane appearing as two. Increasing the noise to $\sigma_g = 0.05$, the estimator fails, with two incorrectly-placed planes [49, Sec. 7.2.2]. Increasing M beyond 1000 does not improve the output.

In Fig.12d the 3D point cloud of 23077 points from the SfM output of Fig.11 is shown. The estimation stops when the number of points is less than $n_\epsilon = 1154$. The first six structures, including the ground, are inliers with 21757 points and there are 1320 outliers (Fig.12e). Fig.12f shows a side view with only five planes visible. The processing time is 7.04 seconds.

The commercial software *ReMake* was used for the *cube* sequence. The 2D sequence has only eight images, with only the faces with numbers 1, 2 and 7 in the input. An image

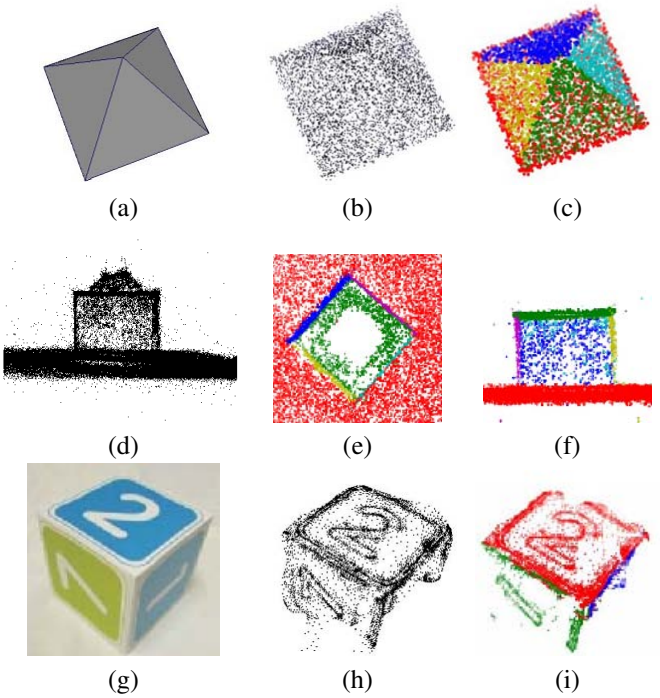


Fig. 12. Plane estimation in 3D. (a) Five synthetic planes. (b) 3D point cloud. (c) Five structures are inliers. See the text also. (d) 3D point cloud from Fig.11. (e) First six structures are inliers. Top view. (f) First five structures are inliers. Side view. (g) *Cube* sequence. A 2D image. (h) 3D point cloud. (i) First three structures are inliers.

is shown in Fig.12g. The 3D point cloud has 5463 points (Fig.12h). The estimation stops only when the number of points is less than $n_\epsilon = 274$. The first three inlier structures with a total of 4718 points are shown in Fig.12i. The 745 outlier points are not shown. The processing time is 2.48 seconds.

4.7 Spheres Estimated in 3D

The nonlinear objective function for the 3D sphere

$$f(\mathbf{y}) = (X-a)^2 + (Y-b)^2 + (Z-c)^2 - r^2 \qquad (27)$$

has the input variable $\mathbf{y} = [X \;\; Y \;\; Z]^\top$ with the carrier vector $\mathbf{x} = \left[X \;\; Y \;\; Z \;\; X^2+Y^2+Z^2\right]^\top$. The center of the sphere is $[a \;\; b \;\; c]^\top$ with the radius r. The transpose of the 4×3 Jacobian matrix is

$$\mathbf{J}^\top_{\mathbf{x}_i|\mathbf{y}_i} = \begin{bmatrix} 1 & 0 & 0 & 2X_i \\ 0 & 1 & 0 & 2Y_i \\ 0 & 0 & 1 & 2Z_i \end{bmatrix}. \qquad (28)$$

$M = 1000$ for all sphere estimation.

In a 12^3 block, two synthetic spheres $n_{in} = 200$ with radii $r = 2, 3$, are corrupted by 3D Gaussian noise $\sigma_g = 0.05, 0.1$ and there are $n_{out} = 200$ outliers (Fig.13a). The estimation stops when the number of points is less than $n_\epsilon = 30$.

Fig.13b shows the $\sigma_g = 0.05$ sphere in red with the scale estimate $\hat{\sigma}^{tls} = 0.079$. The first two structures are inliers, followed by outliers in Fig.13c. The processing time is 5.35 seconds. In 100 tests, the two inlier structures are always estimated.

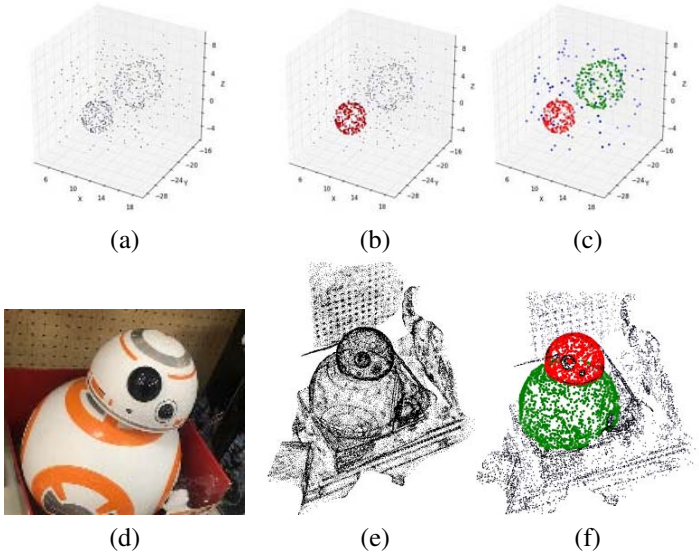


Fig. 13. Sphere estimation in 3D. (a) Two synthetic spheres. (b) The structure estimated with $r = 2$. (c) First two structures are inliers followed by outliers. (d) *Toy* sequence. A 2D image. (e) 3D point cloud. (f) First two structures are inliers.

ReMake processes 36 images of the 2D *toy* sequence, with one shown in Fig.13d, and returns 10854 points in the 3D point cloud (Fig.13e). The estimation stops when the number of points is less than $n_\epsilon = 543$. The processing time is 7.24 seconds. The first two structures are inliers with a total of 3504 points. The number of outliers is 7350, twice as many as the inliers, mostly around the planes in the 3D scene (Fig.13f).

The two small sphere-type objects in Fig.13e are not detected as inlier structures because they are not large enough. In Section 5 we will discuss this type of limitation. Estimation of two different type of inliers, planes and spheres in this case, is discussed in Section 6.2.

4.8 Circular Cylinders Estimated in 3D

Several solutions for the circular cylinder estimation are described in [3] with elemental subsets between five to nine points. We choose the most general, nine-point solution which can estimate any quadric from the 4×4 symmetric matrix $\mathbf{P}$. Applying constrains to this quadric, any particular estimation can be accomplished [18, Sec.3.2.4].

Start with a cylinder aligned with the Z-axis

$$(X-a)^2 + (Y-b)^2 - r^2 \qquad (29)$$

where $[a \;\; b]^\top$ is the center in the XY-plane and r is the radius. In 3D, this is equivalent with function $[\mathbf{y} \;\; 1]\, \mathbf{P}'\, [\mathbf{y} \;\; 1]^\top$, where $\mathbf{P}'$ is a 4×4 symmetric matrix

$$\mathbf{P}' = \begin{bmatrix} \mathbf{D}' & \mathbf{d}' \\ \mathbf{d}'^T & a^2+b^2-r^2 \end{bmatrix} \quad \mathbf{D}' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{d}' = \begin{bmatrix} -a \\ -b \\ 0 \end{bmatrix} \qquad (30)$$

A rigid 3D transformation

$$\mathbf{M} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \qquad \mathbf{P} = \mathbf{M}^{-T}\mathbf{P}'\mathbf{M}^{-1} = \begin{bmatrix} \mathbf{D} & \mathbf{d} \\ \mathbf{d}^T & d \end{bmatrix} \qquad (31)$$

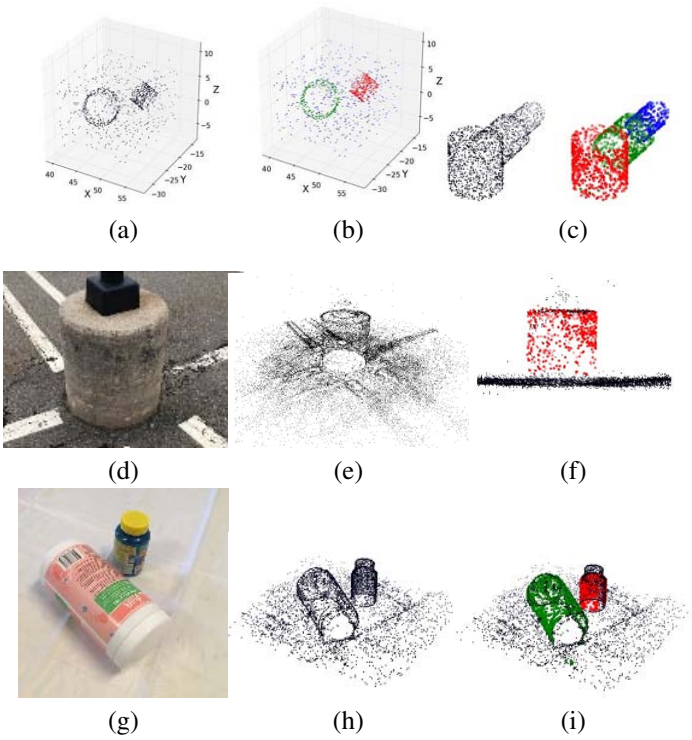


Fig. 14. Cylinder estimation in 3D. (a) Two synthetic cylinders. (b) Estimated three stuctures. (c) Three synthetic cylinders: input and estimation. (d) *Circular pole* sequence. A 2D image. (e) 3D point cloud. (f) First estimated structure is inliers. (g) *Medicine* sequence. A 2D image. (h) 3D point cloud. (i) First two estimated structures are inliers.

obtains the desired cylinder $f(\mathbf{y}) = [\mathbf{y} \;\; 1]\ \mathbf{P}\ [\mathbf{y} \;\; 1]^\top$ with a 4×4 symmetric matrix $\mathbf{P}$ having nine parameters.

The input variable $\mathbf{y} = [X \; Y \; Z]^\top$ gives the carrier vector $\mathbf{x} = [X \;\; Y \;\; Z \;\; X^2 \;\; XY \;\; XZ \;\; Y^2 \;\; YZ \;\; Z^2]^\top$. The transpose of the 9×3 Jacobian matrix is

$$\mathbf{J}^\top_{\mathbf{x}_i|\mathbf{y}_i} = \begin{bmatrix} 1 & 0 & 0 & 2X_i & Y_i & Z_i & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & X_i & 0 & 2Y_i & Z_i & 0 \\ 0 & 0 & 1 & 0 & 0 & X_i & 0 & Y_i & 2Z_i \end{bmatrix}. \quad (32)$$

A circular cylinder has five degrees of freedom, four for the axis and one for a radius. From (30) and (31), two of the three singular values of matrix $\mathbf{D}$ are identical and the third one is zero. The vector $\mathbf{d}$ is an eigenvector of $\mathbf{D}$. These constraints have to be verified for each elemental subset. $M = 2000$ for all cylinder estimation except Fig.14a, which needs $M = 5000$ because this example has a much greater corruption of the inliers.

In a 16^3 block, two synthetic cylinders, $n_{in} = 400, 300$ with radii $r = 2, 3$, are corrupted by 3D Gaussian noise $\sigma = 0.06, 0.1$ and there are $n_{out} = 500$ outliers (Fig.14a). The rotation axis is randomly generated. The estimation stops when the number of points is less than $n_\epsilon = 60$.

The estimated structures in Fig.14b are

	red	*green*	*blue*
nr. points :	413	337	449
TLS scale :	0.28	0.48	5.56
density :	1475	702	80.6

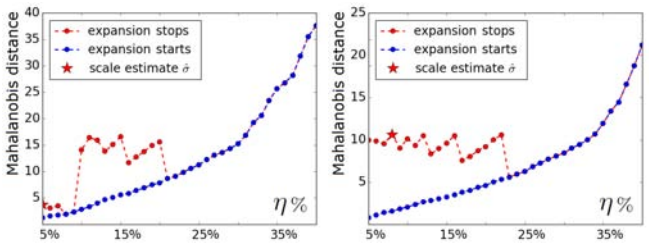


Fig. 15. Ratio of inliers/outliers limits recovery. Based on Fig.2a. (a) Unstable estimate, $n_{in} = 200, n_{out} = 400$. (b) Stable estimate, $n_{in} = 400, n_{out} = 400$.

with the first two structures as inliers, followed by outliers. The processing time is 25.02 seconds. The height of the cylinders can be recovered from the inlier points. In 100 tests, the weaker cylinder becomes outliers six times; in four of those tests, the stronger cylinder does so as well.

Having no outliers in the input does not guarantee that there are no outliers in the output. A synthetic 3D point cloud of 2000 points ($n_\epsilon = 100$) has three cylinders with radii $r = 1, 2, 3$, corrupted with 3D Gaussian noise $\sigma = 0.01$ with no outliers (Fig.14c). The first three structures are inliers with total of 1556 points; the estimated scales are between 0.06 and 0.08. However, the other 444 points are outliers having a larger scale estimate $\hat{\sigma}^{tls} = 0.27$. The processing time is 15.83 seconds.

A 2D image from 54 images in the *circular pole* sequence is shown in Fig.14d. The SfM algorithm returns 7241 points in the 3D point cloud (Fig.14e). The estimation stops when the number of points is less than $n_\epsilon = 362$. The first structure is inliers, containing 568 points, and 6673 points are outliers, mostly around the ground plane (Fig.14f). The processing time is 18.55 seconds. There are many more outliers than inliers.

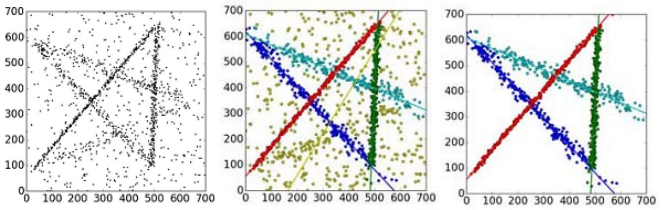
The 22 images in the *medicine* sequence, with one shown in Fig.14g, are processed with *ReMake*. The 3D point cloud has 6500 points (Fig.14h). The estimation stops when the number of points is less than $n_\epsilon = 325$. The first two structures are inliers with a total of 2262 points, followed by 4238 outliers (Fig.14i). Processing time is 12.56 seconds. The yellow cap in Fig.14g is too small to be detected as a separate inlier structure. Pre-processing might help to increase the number of points belonging to the cap.

5 Limitations of MISRE

Every robust estimator fails when the amount of outliers increases beyond a certain limit, with that limit depending on the method. Most robust estimators are considered to have failed completely once they do not return a desired inlier structure, e.g., [7], [19], [34].

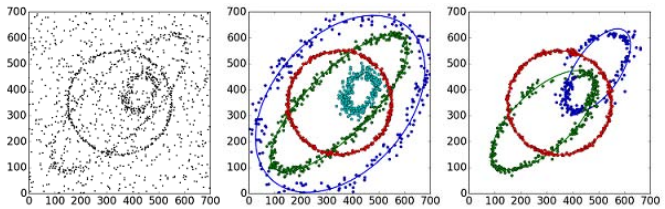
In MISRE the structures are estimated independently. At first, only the inlier structure with the lowest density becomes outliers as the data become more degraded. The stronger-density inlier structures are still estimated correctly.

The synthetic examples used in this section are illustrated in Fig.2a; Fig.6a and Fig.16a; Fig.7a and Fig.17a were the advantage of MISRE can be observed directly.



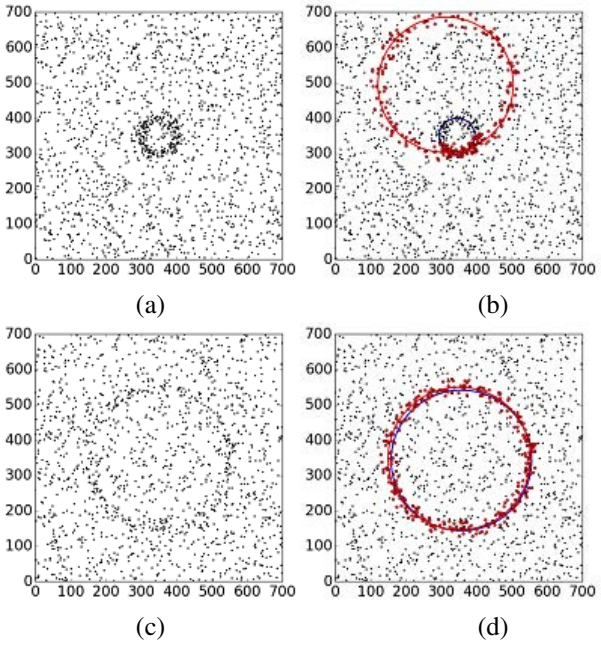
(a) (b) (c)

Fig. 16. Estimation of lines in 2D. (a) $n_{out} = 500$. (b) First four structures are inliers followed by outliers. (c) The inlier structures.



(a) (b) (c)

Fig. 17. Estimation in 2D of three synthetic ellipses. (a) $n_{out} = 800$. (b) First outliers (blue) comes before the weakest ellipse (cyan). (c) Interaction between the two weaker ellipses.



(a) (b) (c) (d)

Fig. 18. The scale estimate is not always sufficient. (a) Circle radius 50. (b) The result after mean shift is incorrect. (c) Circle radius 200. (d) The result is correct.

By increasing the number of outliers to $n_{out} = 400$ in Fig.2a, the scale estimate $\hat{\sigma}$ becomes unstable (Fig.15a). If the number of inliers is also increased to $n_{in} = 400$, the scale estimate becomes stable again (Fig.15b), and remains stable for 100 tests. Having a similar inlier/outlier ratio yields similar performance even with more data.

When five synthetic lines were processed with $n_{out} = 350$ (Fig.6a), the "weakest" inlier structure with the fewest points and largest inlier noise becomes outliers six times out in 100 tests (Fig.6c). When the number of outliers increase to $n_{out} = 500$ in Fig.16a, the "weakest" inlier structure becomes outliers in 34 of 100 tests (Fig.16b), the next structure becomes outliers in only two of those tests. The three strongest-density inlier structures are estimated correctly in all 100 tests. The inlier structures estimated in a specific test are shown in Fig.16c.

Three synthetic ellipses estimated with $n_{out} = 350$ (Fig.7a), have the smallest ellipse become outliers in ten out of 100 tests, while the middle ellipse becomes outliers in six tests. When the number of outliers increase to $n_{out} = 800$ (Fig.17a), the largest ellipse (red) is still correct in 100 repetitions. The smallest ellipse becomes outliers 53 times; in three of those cases, the ellipse is still returned, but not ranked as the third estimate. For example, in Fig.17b, the ellipse has a density of 3.9 (cyan), listed after the first outlier "structure" (blue) with density 4.8. The middle ellipse becomes outliers in 19 cases, including one in which it is recovered but in the incorrect order. In some cases, the two weaker inlier structures can interact when the mean shifts converge to the incorrect modes (Fig.17c). More elaborate pre-processing is needed for more stable results.

The size of an inlier structure is also important in how many outliers can be removed. The two circles with $n_{in} = 200$ inliers but different radii, 50 in Fig.18a and 200 in Fig.18c, are shown with $n_{out} = 1500$ outliers. The circles are corrupted by Gaussian noise, $\sigma_g = 10$. The correct scale estimates $\hat{\sigma}_{50} = 23.65$ and $\hat{\sigma}_{200} = 23.58$ are found in Fig.18b and Fig.18d, shown with blue points. The next step, the mean shift, can give different results based on the radii.

Inside $50 \pm \hat{\sigma}_{50}$ in Fig.18b, there are 241 blue points: 196 true inlier points and 45 outliers. But the highest mode, drawn in red, returns 261 points: 84 true inlier points and 177 outliers. The increase in the number of outliers leads the dense but small nonlinear input to converge to the incorrect mode.

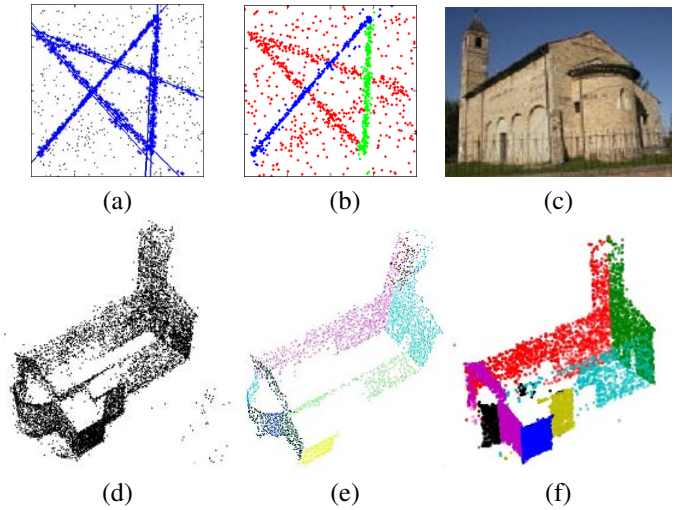
In the case of $r = 200$, the mean shift classifies 346 points as inliers: 190 true inlier points and 156 outliers, drawn with red in Fig.18d. The estimate is stable in 100 tests. This circle has a larger radius but is less visible in Fig.18c.

6 Discussion

This paper introduces MISRE, an algorithm which estimates each structure independently. A predefined threshold between inlier structures and outliers is no longer necessary. The robust estimators discussed in Section 1 can return the same number of inlier structures as MISRE as long as the inlier noises are similar and the estimations are set up correctly. However, MISRE has three significant advantages. First, each structure is estimated independently. Second, MISRE uses the same two constants all the time to estimate the scales. Finally, when MISRE fails, it does so in a predictable way, with the weakest inlier structure becoming outliers first.

6.1 Comparison with Other Robust Estimators

J-linkage and T-linkage algorithms taken from the web and implemented with multi-label optimization and MATLAB wrapper are applied to the data in Fig.6a. All the thresholds are the default settings in the code. In Fig.19a, J-linkage [44]



(a) (b) (c) (d) (e) (f)

Fig. 19. Comparison with J-Linkage and T-Linkage. Fig.6a is the input for (a) J-Linkage. (b) T-Linkage. (c) *Church* sequence. A 2D image. (d) 3D point cloud. (e) J-Linkage. (f) MISRE.

(a) (b) (c)

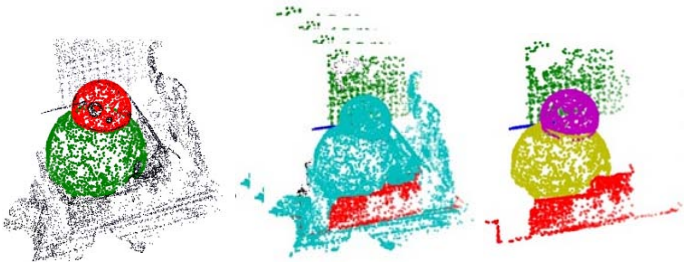
Fig. 20. Three 2D homography estimators. The right image shown only. (a) GpbM. (b) RCMSA. (c) MISRE.

fails completely, because the lines have different scales and J-linkage is ill-suited to such applications. Similarly, T-linkage [28] returns only two inlier structures, the green and blue points. The strongest structure, the red points, is an outlier in Fig19b. The two detected inlier structures have the highest number of data points and the smallest scales in Section 4.1; T-linkage is therefore correct only for the two smallest scales.

A 2D image from the 48 images in the *church* sequence is in Fig.19c. The SfM algorithm returns 11094 points in the 3D point cloud (Fig 19d). The estimated scales are similar and J-linkage also works. The processing time is 330 seconds because of the large 3D input (Fig.19e). The processing time for MISRE is 10.2 seconds (Fig.19f).

The performance in 2D homography is compared for three estimators: gpbM [29], RCMSA [34] and MISRE. The *union house* pair from the *Oxford Visual Geometry Group* has 2084 point pairs: 1739 inliers and 345 outliers. The results are presented with the structures superimposed over the right image of each pair for each estimator (Fig.20).

The gpbM estimates only four inlier structures (Fig.20a). Both RCMSA (Fig.20b) and MISRE (Fig.20c) estimate five inlier structures. In all three cases the inlier structures are followed by outliers. The gpbM is implemented with C++ and MATLAB and computes the estimates in an iterative way. The processing time is 495 seconds. The processing time for RCMSA, running the implementation from the web, is 25.40 seconds. The processing time for MISRE is 3.78 seconds.



(a) (b) (c)

Fig. 21. Post-processing of several inlier structure types. Example: Fig.13e. (a) Two spheres. (b) Three planes. (c) Post-processing together.

An indirect comparison between MISRE and four other robust estimators can be inferred from [34, Table 2], which compares those estimators to RCMSA. The four estimators are: Propose Expand and Re-estimate Labels (PEARL) [21]; Facility Location via meSSage passing (FLoSS) [24]; Quadratic prOgramming to maximize Mutual preFerence (QOMF) [50]; and optimization with Adaptive Reversible Jump Markov Chain (ARJMC) Monte Carlo [33]. These four methods were optimized separately to achieve the best performances. The *median* time of processing is compared for nine pairs of images, each with fifty repetitions. The medians were computed based on the lowest segmentation errors and RCMSA had the fastest processing time. MISRE returns the same number of inlier structures as RCMSA, but without needing to specify parameters.

Statistical measurement of an algorithmic process should not use the median of the estimates. Since up to half of the results may be extreme, this approach can mask substantial variance. The mean of a process gives a more accurate portrayal of the performance. For example, in [43, Table II] the performance metrics for the *checkerboard* sequence from the *Hopkins 155* dataset frequently have significant differences between the mean and median.

In conclusion, MISRE performs as well as other robust algorithms when inlier noises are similar. For varying inlier scales, MISRE still returns the correct estimates, while the other estimators can fail. As discussed in Section 3.5, MISRE's goal is not to maximize the performance of a robust estimator, but rather to be more versatile, usable, and less reliant on user input.

6.2 Future Research

Here, we briefly discuss three other estimation problems whose implementation is left for further experiments: finding multiple types of structures in a scene; independent, identically distributed inputs with different σ-s for each dimension; and objects defined by more than one mathematical relation.

First, in many instances, more than one type of inlier structure has to be estimated; see the examples in Fig.6h, Fig.13e or Fig.14h. Different types of inlier structures often have different scales, too.

Take, as an example, the 3D point cloud of Fig.13e. The two spheres were estimated in Fig.13f and are reproduced in Fig.21a. We can also estimate planes, as in Section 4.6, and

three planes are recovered in Fig.21b. Note the very small plane in the background of the image.

Do post-processing in the input space. The two inlier structure estimations give a total of five inlier structures. Each 3D inlier point is assigned to the *closest* structure, plane or sphere. The resulting five inlier structures are shown in Fig.21c.

Pre-processing can further improve the results for the *toy* sequence (Fig.13d). It can increase the number of points in the background planes and/or recover more points for the two small sphere-type objects which appear as outliers in Fig.21a.

Second, the input $\mathbf{y}$ may have independent, identically distributed inliers having very different variances $[\sigma_1^2 \dots \sigma_l^2]\mathbf{I}_{l\times l}$. For example, the noise along the Z-axis changes with the depth in structure from motion. The σ_j-s cannot be separated in the covariances of the carriers (6).

If $\hat{\sigma}^2\mathbf{I}_{l\times l}$ gives reasonable estimates, run MISRE several times and retain the same inlier structure. The $\hat{\sigma}^{tls}$ is between the smallest $\hat{\sigma}^{tls}_{smt}$ and the largest $\hat{\sigma}^{tls}_{lrt}$ value. The $\hat{\sigma}^{tls}_{lrt}$ has n_{st} points around the estimate $\hat{\boldsymbol{\theta}}_{lrt}$ and $\hat{\alpha}_{lrt}$. Go back to the original space with these values. For each of the l-dimensions, do a separate mean shift with window of size $\hat{\sigma}^{tls}_{smt}$. Similar to Section 3.2, $\hat{\sigma}_j,\ j = 1, \dots l$ are obtained from the points converging to the mode closest to the projection of the estimates. Compute an l-dimensional nonlinear TLS in the input space. Further research is needed to establish the reliability of this procedure.

Third, there may be two different intersecting types of inlier structures, even taking $\mathbf{y}$ with $\sigma^2\mathbf{I}_{l\times l}$. For example, two surfaces in 3D, a cylinder and a plane, intersect in a 2D ellipse in 3D. In the linear space they are

$$\mathbf{y} \Longleftrightarrow \mathbf{x}^{(1)}, \mathbf{x}^{(2)} \qquad \mathbf{x}^{(1)\top}\boldsymbol{\theta}_1 - \alpha_1 \qquad \mathbf{x}^{(2)\top}\boldsymbol{\theta}_2 - \alpha_2.$$

Solve the two relations separately, obtaining $n_{st1}, \hat{\sigma}^{tls}_1$ and $n_{st2}, \hat{\sigma}^{tls}_2$ as the two inlier structures. Retain only those points which are inside *both* structures, n_{st}. Do nonlinear TLS in the input space with the n_{st} points. The validity of this approach should be verified through experiments.

For geometrically-defined objects, the Multiple Input Structures with Robust Estimator (MISRE) estimates each structure independently. The advantages of MISRE become more apparent when the scale estimates for the inlier structures are vastly different. As such, this estimator can be of value in a number of other fields, like mechanical measurement or statistical analysis of economic data.

Acknowledgments We thank the three reviewers for many valuable comments.

REFERENCES

[1] D. Barath and J. Matas, "Graph-Cut RANSAC," in *CVPR'18*, 2018, pp. 6733 – 6741.

[2] S. Basah, A. Bab-Hadiashar, and R. Hoseinnezhad, "Conditions for motion-background segmentation using fundamental matrix," *IET Comput. Vis.*, vol. 3, pp. 189–200, 2009.

[3] C. Beder and W. Förstner, "Direct solutions for computing cylinders from minimal sets of 3D points," in *ECCV'10*, volume 3952, Springer, 2010, pp. 135–146.

[4] J. W. Bian, W.-Y. Lin, Y. Matsushita, S. Yeung, T. D. Nguyen, and M.-M. Cheng, "GMS: Grid-based motion statistics for fast, ultra-robust feature correspondence," in *CVPR'17*, 2017, pp. 4181 – 4190.

[5] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, "DSAC - Differentiable RANSAC for camera localization," in *CVPR'17*, 2017, pp. 6684 – 6692.

[6] T.-J. Chin, Z. Cai, and F. Neumann, "Robust fitting in computer vision: Easy or hard?," in *ECCV'18*, 2018.

[7] J. Choi and G. Medioni, "StaRSaC: Stable random sample consensus for parameter estimation," in *CVPR'09*, 2009, pp. 675 – 682.

[8] S. Choi, T. Kim, and W. Yu, "Performance evaluation of RANSAC family," in *BMVC'09*, 2009, pp. 1 – 12.

[9] O. Chum and J. Matas, "Matching with PROSAC - Progressive sample consensus," in *CVPR'05*, volume I, 2005, pp. 220–226.

[10] O. Chum, J. Matas, and J. Kittler, "Locally optimized RANSAC," in *25th DAGM Symposium*, 2003, pp. 236–243.

[11] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 24, pp. 603–619, 2002.

[12] B. Efron and R. Tibshirani, *An Introduction to the Bootstrap*. Chapman & Hall, 1993.

[13] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 35, pp. 2765–2781, 2013.

[14] M. Farenzena, A. Fusiello, and R. Gherardi, "Structure-and-motion pipeline on a hierarchical cluster tree," in *ICCV Workshops*, 2009, pp. 1489–1496.

[15] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Comm. Assoc. Comp. Mach*, vol. 24, pp. 381–395, 1981.

[16] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multi-view stereopsis," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 32, pp. 1362–1376, 2010.

[17] R. I. Hartley, "Theory and practice of projective rectification," *International J. Computer Vision*, vol. 35, pp. 115–127, 1999.

[18] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[19] T. Hassner, L. Assif, and L. Wolf, "When standard RANSAC is not enough: Cross-media visual matching with hypothesis relevancy," *Machine Vision and Applications*, vol. 25, pp. 971–983, 2014.

[20] G. B. Hughes and M. Chraibi, "Calculating ellipse overlap areas," *Comput. Visual Sci.*, vol. 15, pp. 291–301, 2012.

[21] H. Isack and Y. Boykov, "Energy-based geometric multi-model fitting," *International J. of Computer Vision*, vol. 97, pp. 123–147, 2012.

[22] K. Kanatani, "Ellipse fitting with hyperaccuracy," *IEICE Trans. Inf. & Syst.*, vol. E89-D, pp. 2653–2660, 2006.

[23] S. Korman and R. Litman, "Latent RANSAC," in *CVPR'18*, 2018, pp. 6693 – 6702.

[24] N. Lazic, I. Givoni, B. Frey, and P. Aarabi, "FLoSS: Facility location for subspace segmentation," in *ICCV'09*, 2009, pp. 825–832.

[25] H. Le, T.-J. Chin, and D. Suter, "An exact penalty method for locally convergent maximum consensus," in *CVPR'17*, 2017, pp. 1888 – 1896.

[26] R. Litman, S. Korman, A. Bronstein, and S. Avidan, "Inverting RANSAC: Global model detection via inlier rate estimation," in *CVPR'15*, 2015, pp. 5243–5251.

[27] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International J. of Computer Vision*, vol. 60, pp. 91–110, 2004.

[28] L. Magri and A. Fusiello, "T-linkage: A continuous relaxation of J-linkage for multi-model fitting," in *CVPR'14*, 2014, pp.

3954–3961.
[29] S. Mittal, S. Anand, and P. Meer, "Generalized projection-based M-estimator," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 34, pp. 2351–2364, 2012.
[30] L. Moisan, P. Moulon, and P. Monasse, "Automatic homographic registration of a pair of images, with a contrario elimination of outliers," *Image Proc. Online*, vol. 2, pp. 56–73, 2012.
[31] D. Morley and H. Foroosh, "Improving RANSAC-based segmentation through CNN encapsulation," in *CVPR'17*, 2017, pp. 6338 – 6347.
[32] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *VISAPP'09*, 2009, pp. 331–340.
[33] T. T. Pham, T.-J. Chin, J. Yu, and D. Suter, "Simultaneous sampling and multi-structure fitting with adaptive reversible jump MCMC," in *NIPS'11*, 2011, pp. 540–548.
[34] T. T. Pham, T.-J. Chin, J. Yu, and D. Suter, "The random cluster model for robust geometric fitting," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 36, pp. 1658–1671, 2014.
[35] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J. Frahm, "USAC: A universal framework for random sample consensus," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 35, pp. 2022–2038, 2013.
[36] R. Raguram, J.-M. Frahm, and M. Pollefeys, "A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus," in *ECCV'08*, volume 5303, Springer, 2008, pp. 500–513.
[37] ReMake, "Autodesk." https://www.autodesk.com/products/remake/overview, 2015.
[38] F. Schaffalitzky and A. Zisserman, "Geometric grouping of repeated elements within images," in *Shape, Contour and Grouping in Computer Vision*, Springer, 1999, pp. 165–181.
[39] J. L. Schönberger and J. M. Frahm, "Structure-from-motion revisited," in *CVPR'16*, 2016, pp. 4104–4113.
[40] E. Serradell, M. Özuysal, V. Lepetit, P. Fua, and F. Moreno-Noguer, "Combining geometric and appearance priors for robust homography estimation," in *ECCV'10*, volume 6313, Springer, 2010, pp. 58–72.
[41] P. Speciale, D. P. Paudel, M. R. Oswald, T. Kroeger, L. V. Gool, and M. Pollefeys, "Consensus maximization with linear matrix inequality constraints," in *CVPR'17*, 2017, pp. 4941 – 4949.
[42] Z. L. Szpak, W. Chojnacki, and A. van den Hengel, "Guaranteed ellipse fitting with a confidence region and an uncertainty measure for centre, axes, and orientation," *J. Math. Imaging Vision*, vol. 52, pp. 173–199, 2015.
[43] R. B. Tennakoon, A. Sadri, R. Hoseinnezhad, and A. Bab-Hadiashar, "Effective sampling: Fast segmentation using robust geometric model fitting," *IEEE Trans. Image Processing*, vol. 27, pp. 4182–4194, 2018.
[44] R. Toldo and A. Fusiello, "Robust multiple structures estimation with J-linkage," in *ECCV'08*, 2008, pp. 537–547.
[45] P. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, pp. 138–156, 2000.
[46] A. Vedaldi, H. Jin, P. Favaro, and S. Soatto, "KALMANSAC: Robust filtering by consensus," in *ICCV'05*, volume 1, 2005, pp. 633–640.
[47] H. Wang, T.-J. Chin, and D. Suter, "Simultaneously fitting and segmenting multiple-structure data with outliers," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 34, pp. 1177–1192, 2012.
[48] C. Wu, "Towards linear-time incremental structure from motion," in *Inter. Conf. of 3D Vision*, 2013, pp. 127–134.
[49] X. Yang, "Robust method in photogrammetric reconstruction of geometric primitives in solid modeling." https://sites.rutgers.edu/peter-meer/wp-content/uploads/sites/69/2019/01/xiangyang_theses.pdf, October 2017. PhD dissertation, Rutgers University.
[50] J. Yu, T.-J. Chin, and D. Suter, "A global optimization approach to robust multi-model fitting," in *CVPR'11*, 2011, pp. 2014–2048.
[51] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 22, pp. 1330–1334, 2000.

Xiang Yang received his BE degree in mechanical engineering and automation from Beihang University, Beijing, China, in 2009, his MS in mechanical engineering from University of Bridgeport, Connecticut, in 2012, and his PhD in 2017 from Rutgers University, New Jersey. His research interests include computer aided design, 3D reconstruction, and statistical pattern recognition.

Peter Meer received his Dipl. Engn. degree from the Bucharest Polytechnic Institute, Romania, in 1971, and his D.Sc. degree from the Technion, Israel Institute of Technology, in 1986, both in electrical engineering. From 1971 to 1979, he worked for the Computer Research Institute in Cluj, Romania, and between 1986 and 1990 with the Center for Automation Research at the University of Maryland, College Park. In 1991, he joined the Department of Electrical and Computer Engineering at Rutgers University, New Jersey and now is a Distinguished Professor. He received the 2010 Longuet-Higgins prize for fundamental contributions in computer vision with coauthors Dorin Comaniciu and Visvanathan Ramesh. He is an IEEE Fellow.

Jonathan Meer received his A.B. from Princeton University in 2002 and his Ph.D. from Stanford University in 2009. He is the Private Enterprise Research Center Professor of Economics at Texas A&M University and a Research Associate of the National Bureau of Economic Research. His research interests include the economics of philanthropy and altruism, the economics of education, and low-skill labor markets, and he teaches an online course on the principles of microeconomics that reaches 3000 Texas A&M students every year.