Multi-class Model Fitting by Energy Minimization and Mode-Seeking

Daniel Barath[1,2(✉)] and Jiri Matas[1]

[1] Centre for Machine Perception, Department of Cybernetics, Czech Technical University, Prague, Czech Republic
[2] Machine Perception Research Laboratory, MTA SZTAKI, Budapest, Hungary
barath.daniel@sztaki.mta.hu

Abstract. We propose a general formulation, called Multi-X, for multi-class multi-instance model fitting – the problem of interpreting the input data as a mixture of noisy observations originating from multiple instances of multiple classes. We extend the commonly used α-expansion-based technique with a new move in the label space. The move replaces a set of labels with the corresponding density mode in the model parameter domain, thus achieving fast and robust optimization. Key optimization parameters like the bandwidth of the mode seeking are set automatically within the algorithm. Considering that a group of outliers may form spatially coherent structures in the data, we propose a cross-validation-based technique removing statistically insignificant instances. Multi-X outperforms significantly the state-of-the-art on publicly available datasets for diverse problems: multiple plane and rigid motion detection; motion segmentation; simultaneous plane and cylinder fitting; circle and line fitting.

Keywords: Multi-model fitting · Clustering · Energy minimization

1 Introduction

In multi-class fitting, the input data is interpreted as a mixture of noisy observations originating from multiple instances of multiple model classes, e.g. k lines and l circles in 2D edge maps, k planes and l cylinders in 3D data, multiple homographies or fundamental matrices from correspondences from a non-rigid scene (see Fig. 1). Robustness is achieved by considering assignment to an outlier class.

Multi-model fitting has been studied since the early sixties, the Hough-transform [1,2] being the first popular method for extracting multiple instances of a single class [3–6]. A widely used approach for finding a single instance is RANSAC [7] which alternates two steps: the generation of instance hypotheses and their validation. However, extending RANSAC to the multi-instance case has had limited success. Sequential RANSAC detects instance one after another in a greedy manner, removing their inliers [8,9]. In this approach, data points

V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11220, pp. 229–245, 2018.
https://doi.org/10.1007/978-3-030-01270-0_14

Fig. 1. Multi-class multi-instance fitting examples. Results on simultaneous plane and cylinder (1st), line and circle fitting (2nd), motion (3rd) and plane segmentation (4th).

are assigned to the first instance, typically the one with the largest support for which they cannot be deemed outliers, rather than to the best instance. Multi-RANSAC [10] forms compound hypothesis about n instances. Besides requiring the number n of the instances to be known a priori, the approach increases the size of the minimum sample and thus the number of hypotheses that have to be validated.

Most recent approaches [11–15] focus on the single class case: finding multiple instances of the same model class. A popular group of methods [11,16–19] adopts a two step process: initialization by RANSAC-like instance generation followed by a point-to-instance assignment optimization by *energy minimization* using graph labeling techniques [20]. Another group of methods uses *preference analysis*, introduced by RHA [21], which is based on the distribution of residuals of individual data points with respect to the instances [12,13,15].

The *multiple instance multiple class case* considers fitting of instances that are not necessarily of the same class. This generalization has received much less attention than the single-class case. To our knowledge, the last significant contribution is that of Stricker and Leonardis [22] who search for multiple parametric models simultaneously by minimizing description length using Tabu search. Preference-based methods [12–14] are not directly applicable to the problem since after calculating the preference vectors (or sets), using class-specific distances (or preferences) is not addressed, the type of the distance is thus not maintained. Consequently, instances with "fuzzy" classes, e.g. half line half circle, may emerge.

The proposed Multi-X method finds multiple instances of multiple model classes drawing on progress in energy minimization extended with a new move in the label space: replacement of a set of labels with the corresponding density mode in the model parameter domain. Mode seeking significantly reduces the label space, thus speeding up the energy minimization, and it overcomes the problem of multiple instances with similar parameters, a weakness of state-of-the-art single-class approaches. The assignment of data to instances of different model classes is handled by the introduction of class-specific distance functions. Multi-X can also be seen as an extension or generalization of the Hough transform: (i) it finds modes of the parameter space density without creating an accumulator and locating local maxima there, which is prohibitive in high dimensional spaces, (ii) it handles multiple classes – running Hough transform for each model type in parallel or sequentially cannot easily handle competition

for data points, and (iii) the ability to model spatial coherence of inliers and to consider higher-order geometric priors is added.

Most recent papers [12,14,23] report results tuned for each test case separately. The results are impressive, but input-specific tuning, i.e. semi-automatic operation with multiple passes, severely restricts possible applications. We propose an *adaptive parameter setting* strategy within the algorithm, allowing the user to run Multi-X as a black box on a range of problems with no need to set any parameters. Considering that outliers may form structures in the input, as a post-processing step, a cross-validation-based technique removes insignificant instances.

The contributions of the paper are: (i) A general formulation is proposed for multi-class multi-instance model fitting which, to the best of our knowledge, has not been investigated before. (ii) The commonly used energy minimizing technique, introduced by PEARL [11], is extended with a new move in the label space: replacing a set of labels with the corresponding density mode in the model parameter domain. Benefiting from this move, the minimization is speeded up, terminates with lower energy and the estimated model parameters are more accurate. (iii) The proposed pipeline combines state-of-the-art techniques, such as energy-minimization, median-based mode-seeking, cross-validation, to achieve results superior to the recent multi-model fitting algorithms both in terms of accuracy and processing time. Proposing automatic setting for the key optimization parameters, the method is applicable to various real world problems.

2 Multi-class Formulation

Before presenting the general definition, let us consider a few examples of multi-instance fitting: to find a pair of *line instances* $h_1, h_2 \in \mathcal{H}_l$ interpreting a set of 2D points $\mathcal{P} \subseteq \mathbb{R}^2$. Line class $\mathcal{H}_l$ is the space of lines $\mathcal{H}_l = \{(\theta_l, \phi_l, \tau_l), \theta_l = [\alpha \quad c]^{\mathrm{T}}\}$ equipped with a distance function $\phi_l(\theta_l, p) = |\cos(\alpha)x + \sin(\alpha)y + c|$ ($p = [x \quad y]^{\mathrm{T}} \in \mathcal{P}$) and a function $\tau_l(p_1, ..., p_{m_l}) = \theta_l$ for estimating θ_l from $m_l \in \mathbb{N}$ data points. Another simple example is the fitting n *circle instances* $h_1, h_2, \cdots, h_n \in \mathcal{H}_c$ to the same data. The circle class $\mathcal{H}_c = \{(\theta_c, \phi_c, \tau_c), \theta_c = [c_x \quad c_y \quad r]^{\mathrm{T}}\}$ is the space of circles, $\phi_c(\theta_c, p) = |r - \sqrt{(c_x - x)^2 + (c_y - y)^2}|$ is a distance function and $\tau_c(p_1, ..., p_{m_c}) = \theta_c$ is an estimator. *Multi-line fitting* is the problem of finding multiple line instances $\{h_1, h_2, ...\} \subseteq \mathcal{H}_l$, while the *multi-class* case is extracting a subset $\mathcal{H} \subseteq \mathcal{H}_\forall$, where $\mathcal{H}_\forall = \mathcal{H}_l \cup \mathcal{H}_c \cup \mathcal{H}_. \cup \cdots$. The set $\mathcal{H}_\forall$ is the space of all classes, e.g. line and circle. The formulation includes the outlier class $\mathcal{H}_o = \{(\theta_o, \phi_o, \tau_o), \theta_o = \emptyset\}$ where each instance has constant but possibly different distance to all points $\phi_o(\theta_o, p) = k$, $k \in \mathbb{R}^+$ and $\tau_o(p_1, ..., p_{m_o}) = \emptyset$. Note that considering multiple outlier classes allows interpretation of outliers originating from different sources.

Definition 1 (Multi-class Model). *The multi-class model is a space* $\mathcal{H}_\forall = \bigcup \mathcal{H}_i$, *where* $\mathcal{H}_i = \{(\theta_i, \phi_i, \tau_i) \mid d_i \in \mathbb{N}, \theta_i \in \mathbb{R}^{d_i}, \phi_i \in \mathcal{P} \times \mathbb{R}^{d_i} \rightarrow \mathbb{R}, \tau_i : \mathcal{P}^* \rightarrow \mathbb{R}^{d_i}\}$ *is a single class,* $\mathcal{P}$ *is the set of data points,* d_i *is the dimension of parameter vector* θ_i, ϕ_i *is the distance function and* τ_i *is the estimator of the* i*th class.*

The *objective of multi-instance multi-class model fitting* is to determine a set of instances $\mathcal{H} \subseteq \mathcal{H}_\forall$ and labeling $L \in \mathcal{P} \rightarrow \mathcal{H}$ assigning each point $p \in \mathcal{P}$ to an instance $h \in \mathcal{H}$ minimizing energy E. We adopt energy

$$E(L) = E_d(L) + w_g E_g(L) + w_c E_c(L) \tag{1}$$

to measure the quality of the fitting, where w_g and w_c are weights balancing the different terms described bellow, and E_d, E_c and E_g are the data, complexity terms, and the one considering geometric priors, e.g. spatial coherence or perpendicularity, respectively.

Data term. $E_d : (\mathcal{P} \rightarrow \mathcal{H}) \rightarrow \mathbb{R}$ is defined in most energy minimization approaches as

$$E_d(L) = \sum_{p \in \mathcal{P}} \phi_{L(p)}(\theta_{L(p)}, p), \tag{2}$$

penalizing inaccuracies induced by the point-to-instance assignment, where $\phi_{L(p)}$ is the distance function of $h_{L(p)}$.

Geometric prior term. E_g considers spatial coherence of the data points, adopted from [11], and possibly higher order geometric terms [17], e.g. perpendicularity of instances. The term favoring spatial coherence, i.e. close points more likely belong to the same instance, is defined as

$$E_g(L) : (\mathcal{P} \rightarrow \mathcal{H}) \rightarrow \mathbb{R} = \sum_{(p,q) \in N} w_{pq} [\![L(p) \neq L(q)]\!], \tag{3}$$

where N are the edges of a predefined neighborhood-graph, the Iverson bracket $[\![.]\!]$ equals to one if the condition inside holds and zero otherwise, and w_{pq} is a pairwise weighting term. In this paper, w_{pq} equals to one. For problems, where it is required to consider higher-order geometric terms, e.g. to find three perpendicular planes, E_g can be replaced with the energy term proposed in [17].

A regularization of the number of instances is proposed by Delong et al. [24] as a label count penalty $E_c(L) : (\mathcal{P} \rightarrow \mathcal{H}) \rightarrow \mathbb{R} = |L(\mathcal{P})|$, where $L(\mathcal{P})$ is the set of distinct labels of labeling function L. To handle multi-class models which might have different costs on the basis of the model class, we thus propose the following definition:

Definition 2 (Weighted Multi-class Model). *The weighted multi-class model is a space* $\widehat{\mathcal{H}}_\forall = \bigcup \widehat{\mathcal{H}}_i$, *where* $\widehat{\mathcal{H}}_i = \{(\theta_i, \phi_i, \tau_i, \psi_i) \mid d_i \in \mathbb{N}, \theta_i \in \mathbb{R}^{d_i}, \phi_i \in \mathcal{P} \times \mathbb{R}^{d_i} \rightarrow \mathbb{R}, \tau_i : \mathcal{P}^* \rightarrow \mathbb{R}^{d_i}, \psi_i \in \mathbb{R}\}$ *is a weighted class,* $\mathcal{P}$ *is the set of data points,* d_i *is the dimension of parameter vector* θ_i, ϕ_i *is the distance function,* τ_i *is the estimator, and* ψ_i *is the weight of the* i*th class.*

The term controlling the number of instances is

$$\widehat{E}_c(L) = \sum_{l \in L(\mathcal{P})} \psi_l, \tag{4}$$

instead of E_c, where ψ_l is the weight of the weighted multi-class model referred by label l. Eqs. 2, 3, 4 lead to **overall energy** $\widehat{E}(L) = E_d(L) + w_g E_g(L) + w_c \widehat{E}_c(L)$.

3 Replacing Label Sets

For the optimization of the previously described energy, we build on and extend the PEARL algorithm [11]. PEARL generates a set of initial instances applying a RANSAC-like randomized sampling technique, then alternates two steps until convergence:
(1) Application of α-expansion [25] to obtain labeling L minimizing overall energy $\widehat{E}$ w.r.t. the current instance set.
(2) Re-estimation of the parameter vector θ of each model instance in $\mathcal{H}$ w.r.t. L.
In the PEARL formulation, the only way to remove a label, i.e. to discard an instance, is to assign it to no data points. Experiments show that (i) this removal process is often unable to delete instances having similar parameters, (ii) and makes the estimation sensitive to the choice of label cost w_c. We thus propose a new move in the label space: replacing a set of labels with the density mode in the model parameter domain.

Multi-model fitting techniques based on energy-minimization usually generate a high number of instances $\mathcal{H} \subseteq \mathcal{H}_\forall$ randomly as a first step [11,17] ($|\mathcal{H}| \gg |\mathcal{H}_{\text{real}}|$, where $\mathcal{H}_{\text{real}}$ is the ground truth instance set). Therefore, the presence of many similar instances is typical. We assume, and experimentally validate, that many points supporting the sought instances in $\mathcal{H}_{\text{real}}$ are often assigned in the initialization to a number of instances in $\mathcal{H}$ with similar parameters. The cluster around the ground truth instances in the model parameter domain can be replaced with the modes of the density (see Fig. 2).

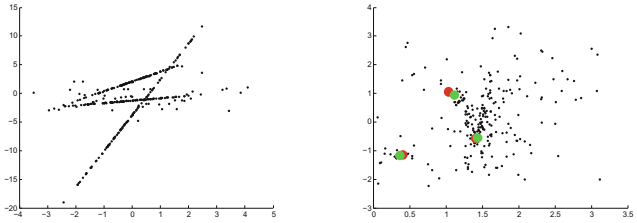


Fig. 2. (**Left**) Three lines each generating 100 points with zero-mean Gaussian noise added, plus 50 outliers. (**Right**) 1000 lines generated from random point pairs, the ground truth instance parameters (red dots) and the modes (green) provided by Mean-Shift shown in the model parameter domain: α angle – vertical, offset – horizontal axis. (Color figure online)

Given a mode-seeking function $\Theta : \mathcal{H}^*_\forall \rightarrow \mathcal{H}^*_\forall$, e.g. Mean-Shift [26], which obtains the density modes of input instance set $\mathcal{H}_i$ in the ith iteration. The proposed move is as

$$\mathcal{H}_{i+1} := \begin{cases} \Theta(\mathcal{H}_i) & \text{if } E(L_{\Theta(\mathcal{H}_i)}) \leq E(L_i), \\ \mathcal{H}_i & \text{otherwise,} \end{cases} \tag{5}$$

where L_i is the labeling in the ith iteration and $L_{\Theta(\mathcal{H}_i)}$ is the optimal labeling which minimizes the energy w.r.t. to instance set $\Theta(\mathcal{H}_i)$. It can be easily seen, that Eq. 5 does not break the convergence since it replaces the instances, i.e. the labels, *if and only if* the energy does not increase. Note that clusters with cardinality one – modes supported by a single instance – can be considered as outliers and removed. This step reduces the label space and speeds up the process.

4 Multi-X

The proposed approach, called Multi-X, combining PEARL, multi-class models and the proposed label replacement move, is summarized in Algorithm 1. Next, each step is described.

Algorithm 1 Multi-X

Input: P – data points
Output: H^* – model instances, L^* – labeling

H_0 := InstanceGeneration(P); $i := 1$;
repeat
  H_i := ModeSeeking(H_{i-1}); ▷ by Median-Shift
  L_i := Labeling(H_i, P); ▷ by α-expansion
  H_i := ModelFitting(H_i, L_i, P); ▷ by Weiszfeld
  $i := i + 1$;
until !Convergence(H_i, L_i)
$H^* := H_{i-1}$, $L^* := L_{i-1}$;
H^*, L^* := ModelValidation(H^*, L^*) ▷ Algorithm 2

1. Instance generation step generates a set of initial instances before the alternating optimization is applied. Reflecting the assumption that the data points are spatially coherent, we use the guided sampling of NAPSAC [27]. This approach first selects a random point, then the remaining ones are chosen from the neighborhood of the selected point. The same neighborhood is used as for the spatial coherence term in the α-expansion. Note that this step can easily be replaced by e.g. PROSAC [28] for problems where the spatial coherence does not hold or favors degenerate estimates, e.g. in fundamental matrix estimation.
2. Mode-Seeking is applied in the model parameter domain. Suppose that a set of instances $\mathcal{H}$ is given. Since the number of instances in the solution – the modes in the parameter domain – is unknown, a suitable choice for mode-seeking is the Mean-Shift algorithm [26] or one of its variants. In preliminary experiments, the most robust choice was the Median-Shift [29] using Weiszfeld- [30] or Tukey-medians [31]. There was no significant difference, but Tukey-median was slightly faster to compute. In contrast to Mean-Shift, it does not generate new elements in the vector space since it always return an element of the input set. With

the Tukey-medians as modes, it is more robust than Mean-Shift [29]. However, we replaced Locality Sensitive Hashing [32] with Fast Approximated Nearest Neighbors [33] to achieve higher speed.

Reflecting the fact that a general *instance-to-instance* distance is needed, we represent instances by point sets, e.g. a line by two points and a homography by four correspondences, and define the *instance-to-instance* distance as the Hausdorff distance [34] of the point sets. Even though it yields slightly more parameters than the minimal representation, thus making Median-Shift a bit slower, it is always available as it is used to define spatial neighborhood of points. Another motivation for representing by points is the fact that having a non-homogeneous representation, e.g. a line described by angle and offset, leads to anisotropic distance functions along the axes, thus complicating the distance calculation in the mode-seeking.

There are many point sets defining an instance and a canonical point set representation is needed. For lines, the nearest point to the origin is used and a point on the line at a fixed distance from it. For a homography $\mathbf{H}$, the four points are $\mathbf{H}[0,0,1]^{\mathrm{T}}$, $\mathbf{H}[1,0,1]^{\mathrm{T}}$, $\mathbf{H}[0,1,1]^{\mathrm{T}}$, and $\mathbf{H}[1,1,1]^{\mathrm{T}}$. The matching step is excluded from the Hausdorff distance, thus speeding up the distance calculation significantly.[1]

The application of Median-Shift Θ_{med} never increases the number of instances $|\mathcal{H}_i|$: $|\Theta_{\mathrm{med}}(\mathcal{H}_i)| \leq |\mathcal{H}_i|$. The equality is achieved *if and only if* the distance between every instance pair is greater than the bandwidth. Note that for each distinct model class, Median-Shift has to be applied separately. According to our experience, applying this label replacement move in the first iteration does not make the estimation less accurate but speeds it up significantly even if the energy slightly increases.

3. Labeling assigns points to model instances obtained in the previous step. A suitable choice for such task is α-expansion [25], since it handles an arbitrary number of labels. Given $\mathcal{H}_i$ and an initial labeling L_{i-1} in the ith iteration, labeling L_i is estimated using α-expansion minimizing energy $\widehat{E}$. Note that L_0 is determined by α-expansion in the first step. The number of the model instances $|\mathcal{H}_i|$ is fixed during this step and the energy must decreases: $\widehat{E}(L_i, \mathcal{H}_i) \leq \widehat{E}(L_{i-1}, \mathcal{H}_i)$. To reduce the sensitivity on the outlier threshold (as it was shown for the single-instance case in [35]), the distance function of each class is included into a Gaussian-kernel.

4. Model Fitting re-estimates the instance parameters w.r.t. the assigned points. The obtained instance set $\mathcal{H}_i$ is re-fitted using the labeling provided by α-expansion. The number of the model instances $|\mathcal{H}_i|$ is constant. L_2 fitting is an appropriate choice, since combined with the labeling step, it can be considered as truncated L_2 norm.

The overall energy $\widehat{E}$ can only decrease or stay constant during this step since it consists of three terms: (1) E_d – the sum of the assignment costs minimized,

[1] Details on the choice of model representation are submitted in the supplementary material.

(2) E_g – a function of the labeling L_i, fixed in this step and (3) $\widehat{E}_c$ – which depends on $|H_i|$ so $\widehat{E}_c$ remains the same. Thus

$$\widehat{E}(L_i, \mathcal{H}_{i+1}) \leq \widehat{E}(L_i, \mathcal{H}_i). \tag{6}$$

5. Model Validation considers that a group of outliers may form spatially coherent structures in the data. We propose a post-processing step to remove statistically insignificant models using cross-validation. The algorithm, summarized in Algorithm 2, selects a minimal subset t times from the inlier points I. In each iteration, an instance is estimated from the selected points and its distance to each point is computed. The original instance is considered stable if the mean of the distances is lower than threshold γ. Note that γ is the outlier threshold used in the previous sections. **Automatic parameter setting** is crucial for Multi-X to be applicable to various real world tasks without requiring the user to set most of the parameters manually. To avoid manual bandwidth selection for **mode-seeking**, we adopted the automatic procedure proposed in [36] which sets the bandwidth ϵ_i of the ith instance to the distance of the instance and its kth neighbor. Thus each instance has its own bandwidth set automatically on the basis of the input.

Algorithm 2 Model Validation.

Input: I – inlier points, t – trial number,
γ – outlier threshold ▷ default $t = 100$
Output: $R \in \{\text{true}, \text{false}\}$ – response

$\widehat{D} := 0$
for $i := 1$ to t **do**
  MSS := SelectMinimalSubset(I)
  H := ModelEstimation(MSS)
  $\widehat{D} := \widehat{D}$+ MeanDistanceFromPoints(H, I) /t
$R := \widehat{D} < \gamma$

Label cost w_c is set automatically using the approach proposed in [17] as follows: $w_c = m \log(|\mathcal{P}|)/h_{\max}$, where m is the size of the minimal sample to estimate the current model, $|\mathcal{P}|$ is the point number and $h_{\max}$ is the maximum expected number of instances in the data. Note that this cost is not required to be high since mode-seeking successfully suppresses instances having similar parameters. The objective of introducing a label cost is to remove model instances with weak supports. In practice, this means that the choice of $h_{\max}$ is not restrictive.

Experiments show that the choice of the **number of initial instances** does not affect the outcome of Multi-X significantly. In our experiments, the number of instances generated was twice the number of the input points.

Spatial coherence weight w_g value 0.3 performed well in the experiments. The common problem-specific outlier thresholds which led to the most accurate results was: homographies (2.4 pixels), fundamental matrices (2.0 pixels), lines and circles (2.0 pixels), rigid motions (2.5), planes and cylinders (10 cm).

5 Experimental Results

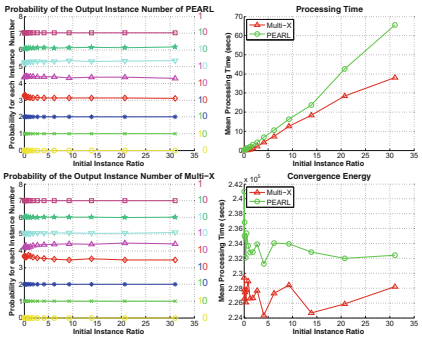
First we compare Multi-X with PEARL [11] combined with the label cost of [24]. Then the performance of Multi-X applied to the following Computer Vision problems is reported: 2D geometric primitive fitting, 3D plane and cylinder fitting to LIDAR point clouds, multiple homography fitting, two-view and video motion segmentation.

Comparison of PEARL and Multi-X. In a test designed to show the effect of the proposed label move, model validation was not applied and both methods used the same algorithmic components described in the previous section. A synthetic environment consisting of three 2D lines, each sampled at 100 random locations, was created. Then 200 outliers, i.e. random points, were generated. Finally, zero-mean Gaussian noise was added to the point coordinates with 3.0 pixel standard deviation.

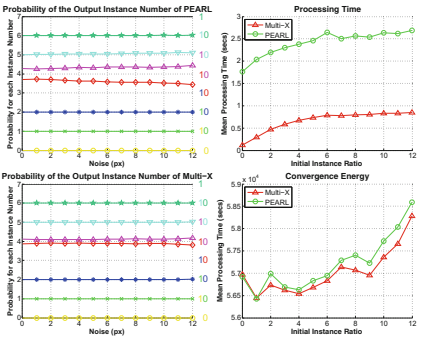
The left column of Fig. 3a shows the *probability of returning an instance number* for Multi-X (top-left) and PEARL (bottom-left) as the function of the initial instance number (horizontal axis; ratio w.r.t. to the input point number; calculated from 1000 runs on each). The numbers next to the vertical axis are the numbers of returned instances. The curve on their right shows the probability ($\in [0, 1]$) of returning them. For example, the red curve of PEARL (top-left) on the right of number 3 is close to the 0.1 probability, while for Multi-X (bottom-left), it is approximately 0.6. Therefore, Multi-X returns the desired number of instances (remember that the ground truth number is 3) in ≈60% of the cases if as many instances are given as points. PEARL achieved ≈10%. The processing times (top-right), and convergence energies (bottom-right) are also reported. The standard deviation of the zero-mean Gaussian-noise added to the point coordinates is 20 pixels. Reflecting the fact that the noise σ is usually not known in real applications, we set the outlier threshold to 6.0 pixels. The maximum model number of the label cost was set to the ground truth value, $h_{\max} = 3$, to demonstrate that suppressing instances only with label cost penalties is not sufficient even with the proper parameters. It can be seen that Multi-X more likely returns the ground truth number of models, both its processing time and convergence energy are superior to that of PEARL.

For Fig. 3b, the number of the generated instances was set to twice the point number and the threshold to 3 pixels. Each reported property is plotted as the function of the noise σ added to the point coordinates. The same trend can be seen as in Fig. 3a: Multi-X is less sensitive to the noise than PEARL. It more often returns the desired number of instances, its processing time and convergence energy are lower.

Synthetic multi-class fitting. In this paragraph, Multi-X is compared with state-of-the-art multi-model fitting techniques on synthetically generated scenes (see Fig. 4) consisting of 2D geometric entities, i.e. lines, parabolas and circles. Each entity was sampled at 100 points and the outlier ratio was 0.33 in all scenes, i.e. 50 outliers were generated for every 100 inliers. For plots (a-c), the task was to find the generated parabolas, lines and circles. For (d), three types of circles were generated: $r_1 = 200$, $r_2 = 100$ and $r_3 = 50$. Different radii were considered



(a) *Increasing instance number.* Zero-mean Gaussian noise with $\sigma = 20$ pixels added to the point coordinates. (**Left**) the probability of returning 0, ..., 7 instances (vertical axis) for PEARL (top) and Multi-X (bottom) plotted as the function of the ratio of the initial instance number and the point number (horizonal axis). (**Right**): the processing time in seconds and convergence energy.



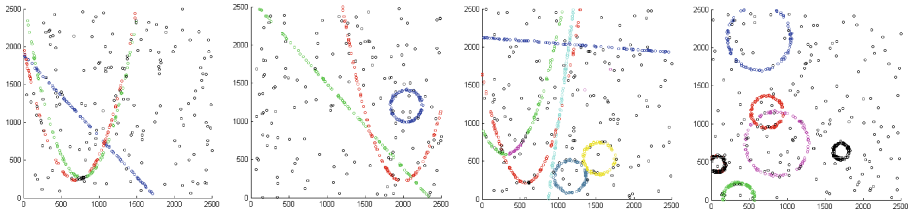
(b) *Increasing noise.* The number of initial instances generated is twice the point number. (**Left**): the probability of returning instance numbers 0, ..., 7 (vertical axis) for PEARL (top) and Multi-X (bottom) plotted as the function of the noise σ (horizonal axis). (**Right**): the processing time in seconds and convergence energy.

Fig. 3. Comparison of PEARL and Multi-X. Three random lines sampled at 100 locations, plus 200 outliers. Parameters of both methods are: $h_{\max} = 3$, and the outlier threshold is (a) 6 and (b) 3 pixels.

as different class. The objective of (d) was slightly different than that of (a-c): to find circles with $r = 200$ and $r = 100$ (± 5 pixels), without applying a post-processing step to remove circles with different radii. Therefore, other circles were considered degenerate, and thus dropped, in the initial instance generation step of all compared methods. Those methods are PEARL [11,16], T-Linkage [12][2] and RPA [13][3] since they can be considered as the state-of-the-art and their implementations are available. PEARL and Multi-X used a fixed setting. Since neither RPA nor T-Linkage are applicable to the multi-class problem, we applied each of them sequentially in all possible ways (e.g. lines first, then circles and parabolas) and selected the best solution. In contrast to PEARL and Multi-X, we tuned the thresholds of RPA and T-Linkage for each problem separately to achieve the best results.

The number of points (Point #), the initial instance number (Inst. #) and fitting results on the problems of Fig. 4, i.e. misclassification error (ME), number false positive (FP) and false negative (FN) instances, are reported in Table 1. The initial instance numbers were calculated by the well-known formula, proposed for RANSAC, from the ground truth inlier ratios requiring 99% confidence. It can be seen that even though the per-problem tuning of RPA and T-Linkage,

[2] http://www.diegm.uniud.it/fusiello/demo/jlk/.
[3] http://www.diegm.uniud.it/fusiello/demo/rpa/.



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

Fig. 4. Estimating 2D geometric classes: lines, parabolas, and circles with radii in given range (a-c) and with fixed radii of 100 and 200 (d); in (d) the small circles are thus structured outliers. Data: 100 points, plus 50 outliers per instance. The Multi-X assignment to instances is color-coded. Multi-X produces zero false negatives (FN) and a single false positive (FP), in (c) (purple points). See Table 1 for results – competing methods have higher FP and FN rates.

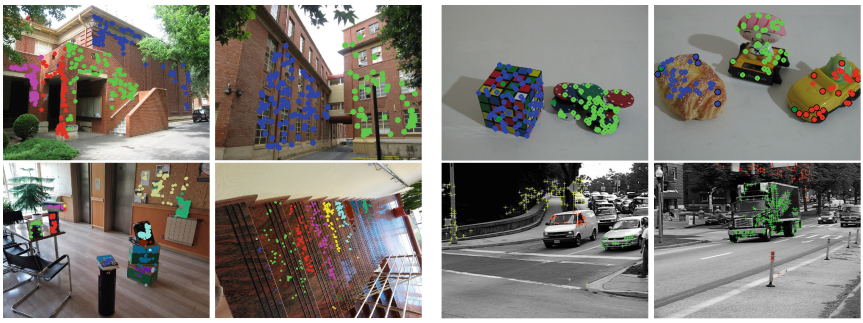
Table 1. Estimating 2D geometric classes: lines, parabolas, and circles. Misclassification errors (ME, in %), number of false positive (FP), false negative (FN) instances, and the processing time (T, in seconds) on the scenes of Fig. 4. Point (#) and initial instance (Inst. #) numbers are shown in the 2nd and 3rd columns. *PEARL, Multi-X used fixed, and T-Linkage, RPA per-problem tuned parameters.*

Figure 4			PEARL [11]				T-Linkage [12]				RPA [13]				**Multi-X**			
	Point #	Inst. #	ME	FP	FN	T	ME	FP	FN	T	ME	FP	FN	T	ME	FP	FN	T
(a)	450	926	10.6	**0**	**0**	88.2	9.9	**0**	**0**	19.5	23.6	**0**	1	61.1	**9.8**	**0**	**0**	**4.1**
(b)	450	926	**2.3**	**0**	**0**	283.5	6.4	**0**	**0**	24.5	4.2	**0**	**0**	221.7	**2.3**	**0**	**0**	**4.6**
(c)	750	8 275	16.4	2	**0**	1186.5	33.2	2	3	172.4	27.2	**1**	2	460.8	**9.7**	**1**	**0**	**7.1**
(d)	750	7 792	28.4	4	**0**	5.3	16.1	1	1	45.3	27.3	**0**	2	100.4	**8.7**	**0**	**0**	**2.4**

both PEARL and Multi-X outperformed them for this multi-class problem. Also, Multi-X results are superior to that of PEARL with significant improvement in processing time.

Multiple homography fitting is evaluated on the AdelaideRMF homography dataset [37] used in most recent publications (see Fig.5a for examples). AdelaideRMF consists of 19 image pairs of different resolutions with ground truth point correspondences assigned to planes (homographies). To generate initial model instances the technique proposed by Barath et al. [19] is applied: a single homography is estimated for each correspondence using the point locations together with the related local affine transformations. Table 2 reports the results of PEARL [25], FLOSS [38], T-Linkage [12], ARJMC [39], RCMSA [18], J-Linkage [15], and Multi-X. To allow comparison with the state-of-the-art, all methods, including Multi-X, are tuned separately for each test and only the same 6 image pairs are used as in [12].

Results using a fixed parameter setting are reported in Table 3 (results, except that of Multi-X, copied from [13]). Multi-X achieves the lowest errors. Compared to results in Table 2 for parameters hand-tuned for each problem, the errors are



(a) AdelaideRMF (1-2) and Multi-H (3-4) examples. Colors indicate the planes Multi-X assigned points to.

(b) AdelaideRMF (1-2) and Hopkins (3-4) examples. Color indicates the motion Multi-X assigned a point to.

Fig. 5. Two-view geometry fitting. First images of the pairs.

significantly higher, but automatic parameter setting is the only possibility in many applications. Moreover, per-image-tuning leads to overfitting.

Table 2. Misclassification error (%) for the two-view plane segmentation on AdelaideRMF test pairs: (1) `johnsonna`, (2) `johnsonnb`, (3) `ladysymon`, (4) `neem`, (5) `oldclassicswing`, (6) `sene`.

	Plane #	PEARL [11]	FLOSS [38]	T-Lnkg [12]	ARJMC [39]	RCMSA [18]	J-Lnkg [15]	**Multi-X**
(1)	4	4.02	4.16	4.02	6.48	5.90	5.07	**3.75**
(2)	6	18.18	18.18	18.17	21.49	17.95	18.33	**4.46**
(3)	2	5.49	5.91	5.06	5.91	7.17	9.25	**0.00**
(4)	3	5.39	5.39	3.73	8.81	5.81	3.73	**0.00**
(5)	2	1.58	1.85	0.26	1.85	2.11	0.27	**0.00**
(6)	2	0.80	0.80	0.40	0.80	0.80	0.84	**0.00**
Avg.		5.91	6.05	5.30	7.56	6.62	6.25	**1.37**
Med.		4.71	4.78	3.87	6.20	5.86	4.40	**0.00**

Two-view motion segmentation is evaluated on the AdelaideRMF motion dataset consisting of 21 image pairs of different sizes and the ground truth – correspondences assigned to their motion clusters.

Figure 5b presents example image pairs from the AdelaideRMF motion datasets partitioned by Multi-X. Different motion clusters are denoted by color. Table 4 shows comparison with state-of-the-art methods when all methods are tuned separately for each test case. Results are the average and minimum misclassification errors (in percentage) of ten runs. All results except that of Multi-X are copied from [23]. For Table 5, all methods use fixed parameters. For both test types, Multi-X achieved higher accuracy than the other methods.

Simultaneous plane and cylinder fitting is evaluated on LIDAR point cloud data (see Fig. 6). The annotated database consists of traffic signs, balusters and

Table 3. Misclassification errors (%, average and median) for two-view plane segmentation on all the 19 pairs from AdelaideRMF test pairs using fixed parameters.

	T-Lnkg [12]	RCMSA [18]	RPA [13]	Multi-H [19]	**Multi-X**
Avg.	44.68	23.17	15.71	14.35	**9.72**
Med.	44.49	24.53	15.89	9.56	**2.49**

Table 4. Misclassification errors (%) for two-view motion segmentation on the AdelaideRMF dataset. All the methods were tuned separately for each video by the authors. Tested image pairs: (1) `cubechips`, (2) `cubetoy`, (3) `breadcube`, (4) `gamebiscuit`, (5) `breadtoycar`, (6) `biscuitbookbox`, (7) `breadcubechips`, (8) `cubebreadtoychips`.

	KF [40]		RCG [41]		T-Lnkg [12]		AKSWH [42]		MSH [23]		**Multi-X**	
	Avg.	Min.	Avg.	Min.	Avg.	Min.	Avg.	Min.	Avg.	Min.	Avg.	Min.
(1)	8.42	4.23	13.43	9.52	5.63	2.46	4.72	2.11	3.80	2.11	**3.45**	**1.41**
(2)	12.53	2.81	13.35	10.92	5.62	4.82	7.23	4.02	3.21	1.61	**2.27**	**0.40**
(3)	14.83	4.13	12.60	8.07	4.96	1.32	5.45	1.42	2.69	0.83	**1.45**	**0.41**
(4)	13.78	5.10	9.94	3.96	7.32	3.54	7.01	5.18	3.72	1.22	**0.61**	**0.30**
(5)	16.87	14.55	26.51	19.54	**4.42**	4.00	9.04	8.43	6.63	4.55	5.24	**1.80**
(6)	16.06	14.29	16.87	14.36	1.93	1.16	8.54	4.99	1.54	1.16	**0.62**	**0.00**
(7)	33.43	21.30	26.39	20.43	**1.06**	0.86	7.39	3.41	1.74	0.43	5.32	**0.00**
(8)	31.07	22.94	37.95	20.80	3.11	3.00	14.95	13.15	4.28	3.57	**2.63**	**1.52**

the neighboring point clouds truncated by a 3-meter-radius cylinder parallel to the vertical axis. Points were manually assigned to signs (planes) and balusters (cylinders). Multi-X is compared with the same methods as in the line and circle fitting section. PEARL and Multi-X fit cylinders and planes simultaneously while T-Linkage and RPA sequentially. Table 6 reports that Multi-X is the most accurate in all test cases except one.

Table 5. Misclassification errors (%, average and median) for two-view motion segmentation on all the 21 pairs from the AdelaideRMF dataset using fixed parameters.

	RPA [13]	RCMSA [18]	T-Lnkg [12]	AKSWH [42]	**Multi-X**
Avg.	5.62	9.71	43.83	12.59	**2.97**
Med.	4.58	8.48	39.42	11.57	**0.00**

Video motion segmentation is evaluated on 51 videos of the Hopkins dataset [43]. Motion segmentation in video sequences is the retrieval of sets of points undergoing rigid motions contained in a dynamic scene captured by a moving camera. It can be considered as a subspace segmentation under the

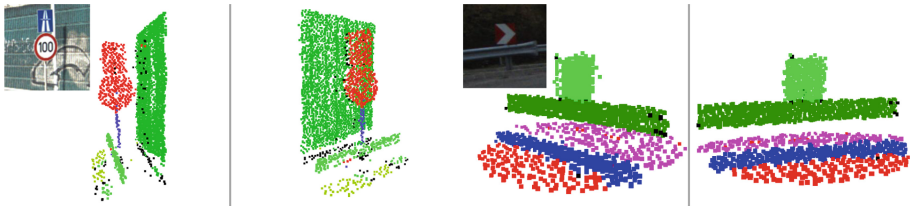


Fig. 6. Results of simultaneous plane and cylinder fitting to LIDAR point cloud in two scenes. Segmented scenes visualized from different viewpoints. There is only one cylinder on the two scenes: the pole of the traffic sign on the top. Color indicates the instance Multi-X assigned a point to.

Table 6. Misclassification error (%) of simultaneous plane and cylinder fitting to LIDAR data. See Fig. 6 for examples.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)
PEARL [11]	10.63	10.88	37.34	38.13	17.20	**17.35**	6.12
T-Lnkg [12]	57.46	41.79	52.97	38.91	51.83	61.77	12.49
RPA [13]	46.83	53.39	61.64	41.41	53.34	51.21	80.45
Multi-X	**8.89**	**4.72**	**2.84**	**19.38**	**16.83**	21.72	**5.72**

assumption of affine cameras. For affine cameras, all feature trajectories associated with a single moving object lie in a 4D linear subspace in $\mathbb{R}^{2F}$, where F is the number of frames [43].

Table 7. Misclassification errors (%, average and median) for multi-motion detection on 51 videos of Hopkins dataset: (1) `Traffic2` – 2 motions, 31 videos, (2) `Traffic3` – 3 motions, 7 videos, (3) `Others2` – 2 motions, 11 videos, (4) `Others3` – 3 motions, 2 videos, (5) `All` – 51 videos.

	SSC [44]		T-Lnkg [12]		RPA [13]		Grdy-RC [14]		ILP-RC [14]		J-Lnkg [15]		**Multi-X**	
	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.	Avg.	Med.
(1)	0.06	**0.00**	1.31	**0.00i**	7.48	**0.00**	7.48	**0.00**	0.54	**0.00**	1.75	**0.00**	**0.09**	**0.00**
(2)	0.76	**0.00**	0.48	0.19	28.65	**0.00**	28.65	1.53	0.35	0.19	1.58	0.34	**0.32**	**0.00**
(3)	3.95	**0.00**	6.47	2.38	8.75	2.44	8.75	0.20	2.40	1.30	5.32	1.30	**1.06**	**0.00**
(4)	2.13	2.13	5.32	5.32	14.89	9.11	14.89	14.89	2.13	2.13	6.91	6.91	**1.06**	**0.16**
(5)	1.08	**0.00**	2.47	**0.00**	10.91	**0.00**	10.91	**0.00**	0.98	**0.00**	2.70	**0.00**	**0.16**	**0.00**

Table 7 shows that Multi-X outperforms the state-of-the-art: SSC [44], T-Linkage [12], RPA [13], Grdy-RansaCov [14], ILP-RansaCov [14], and J-Linkage [15]. Results, except for Multi-X, are copied from [14]. Figure 5b shows two frames of the tested videos.

6 Conclusion

A novel multi-class multi-instance model fitting method has been proposed. It extends an energy minimization approach with a new move in the label space: replacing a set of labels corresponding to model instances by the mode of the density in the model parameter domain. Most of its key parameters are set adaptively making it applicable as a black box on a range of problems. Multi-X outperforms the state-of-the-art in multiple homography, rigid motion, simultaneous plane and cylinder fitting; motion segmentation; and 2D edge interpretation (circle and line fitting). Multi-X runs in time approximately linear in the number of data points, it is an order of magnitude faster than available implementations of commonly used methods.[4]

Limitations. The proposed formulation assumes "non-overlapping" instances, i.e. no shared support, a point can be assigned to a single instance only. Thus, for example, the problem of simultaneously finding a fundamental matrix $\mathbf{F}$ and homographies consistent with it is not covered by the formulation. The problem of fitting hierarchical models is complex, an instance can be supported by different classes, e.g. $\mathbf{F}$ by k planes or 7 points; or a rectangle may be supported by line segments as well as points. Definition of all the cost functions and the optimization procedure is beyond the scope of this work.

Acknowledgement. The authors were supported by the Czech Science Foundation Project GACR P103/12/G084.

References

1. Hough, P.V.C.: Method and means for recognizing complex patterns (1962)
2. Illingworth, J., Kittler, J.: A survey of the hough transform. Comput. Vis. Graph. Image Process. **44**(1), 87–116 (1988)
3. Guil, N., Zapata, E.L.: Lower order circle and ellipse hough transform. Pattern Recognit. **30**(10), 1729–1744 (1997)
4. Matas, J., Galambos, C., Kittler, J.: Robust detection of lines using the progressive probabilistic hough transform. Comput. Vis. Image Underst. **78**(1), 119–137 (2000)
5. Rosin, P.L.: Ellipse fitting by accumulating five-point fits. Pattern Recognit. Lett. **14**(8), 661–669 (1993)
6. Xu, L., Oja, E., Kultanen, P.: A new curve detection method: randomized hough transform (rht). Pattern Recognit. Lett. **11**(5), 331–338 (1990)
7. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
8. Vincent, E., Laganiére, R.: Detecting planar homographies in an image pair. In: International Symposium on Image and Signal Processing and Analysis (2001)
9. Kanazawa, Y., Kawakami, H.: Detection of planar regions with uncalibrated stereo using distributions of feature points. In: British Machine Vision Conference (2004)
10. Zuliani, M., Kenney, C.S., Manjunath, B.: The multiransac algorithm and its application to detect planar homographies. In: ICIP, IEEE (2005)

[4] The source code and the datasets are available on https://github.com/danini/multi-x.

11. Isack, H., Boykov, Y.: Energy-based geometric multi-model fitting. Int. J. Comput. Vis. **97**(2), 123–147 (2012)
12. Magri, L., Fusiello, A.: T-Linkage: a continuous relaxation of J-Linkage for multi-model fitting. In: Conference on Computer Vision and Pattern Recognition (2014)
13. Magri, L., Fusiello, A.: Robust multiple model fitting with preference analysis and low-rank approximation. In: British Machine Vision Conference (2015)
14. Magri, L., Fusiello, A.: Multiple model fitting as a set coverage problem. In: Conference on Computer Vision and Pattern Recognition (2016)
15. Toldo, R., Fusiello, A.: Robust multiple structures estimation with j-linkage. In: European Conference on Computer Vision (2008)
16. Delong, A., Gorelick, L., Veksler, O., Boykov, Y.: Minimizing energies with hierarchical costs. Int. J. Comput. Vis. **100**(1), 38–58 (2012)
17. Pham, T.T., Chin, T.J., Schindler, K., Suter, D.: Interacting geometric priors for robust multi-model fitting. TIP **23**(10), 4601–4610 (2014)
18. Pham, T.T., Chin, T.J., Yu, J., Suter, D.: The random cluster model for robust geometric fitting. Pattern Anal. Mach. Intell. **36**(8), 1658–1671 (2014)
19. Barath, D., Matas, J., Hajder, L.: Multi-H: efficient recovery of tangent planes in stereo images. In: British Machine Vision Conference (2016)
20. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. Pattern Anal. Mach. Intell. **26**(9), 1124–1137 (2004)
21. Zhang, W., Kŏsecká, J.: Nonparametric estimation of multiple structures with outliers. In: Vidal, René, Heyden, Anders, Ma, Yi (eds.) WDV 2005-2006. LNCS, vol. 4358, pp. 60–74. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-70932-9_5
22. Stricker, M., Leonardis, A.: ExSel++: a general framework to extract parametric models. In: International Conference on Computer Analysis of Images and Patterns (1995)
23. Wang, H., Xiao, G., Yan, Y., Suter, D.: Mode-seeking on hypergraphs for robust geometric model fitting. In: International Conference of Computer Vision (2015)
24. Delong, A., Osokin, A., Isack, H.N., Boykov, Y.: Fast approximate energy minimization with label costs. Int. J. Comput. Vis. **96**(1), 1–27 (2012)
25. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. Pattern Anal. Mach. Intell. **23**(11), 1222–1239 (2001)
26. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. Pattern Anal. Mach. Intell. **25**(5), 281–288 (2002)
27. Nasuto, D., Craddock, J.M.B.R.: NAPSAC: high noise, high dimensional robust estimation - its in the bag (2002)
28. Chum, O., Matas, J.: Matching with PROSAC-progressive sample consensus. In: Conference on Computer Vision and Pattern Recognition, IEEE (2005)
29. Shapira, L., Avidan, S., Shamir, A.: Mode-detection via median-shift. In: International Conference on Computer Vision (2009)
30. Weiszfeld, E.: Sur le point pour lequel la somme des distances de n points donnés est minimum. Tohoku Math. J. **43**, 355–386 (1937)
31. Tukey, J.W.: Mathematics and the picturing of data. In: ICM (1975)
32. Datar, M., Immorlica, N., Indyk, P., Mirrokni, V.S.: Locality-sensitive hashing scheme based on p-stable distributions. In: SoCG (2004)
33. Muja, M., Lowe, D.G.: Fast approximate nearest neighbors with automatic algorithm configuration. In: International Conference on Computer Vision Theory and Applications (2009)

34. Rockafellar, R.T., Wets, R.J.B.: Variational Analysis. Springer, Berlin (2009)
35. Lebeda, K., Matas, J., Chum, O.: Fixing the locally optimized RANSAC. In: British Machine Vision Conference (2012)
36. Georgescu, B., Shimshoni, I., Meer, P.: Mean shift based clustering in high dimensions: a texture classification example. In: International Conference on Computer Vision (2003)
37. Wong, H.S., Chin, T.J., Yu, J., Suter, D.: Dynamic and hierarchical multi-structure geometric model fitting. In: International Conference on Computer Vision (2011)
38. Lazic, N., Givoni, I., Frey, B., Aarabi, P.: Floss: facility location for subspace segmentation. In: International Conference on Computer Vision (2009)
39. Pham, T.T., Chin, T.J., Yu, J., Suter, D.: Simultaneous sampling and multi-structure fitting with adaptive reversible jump mcmc. In: Annual Conference on Neural Information Processing Systems (2011)
40. Chin, T.J., Wang, H., Suter, D.: Robust fitting of multiple structures: the statistical learning approach. In: International Conference on Computer Vision (2009)
41. Liu, H., Yan, S.: Efficient structure detection via random consensus graph. In: Conference on Computer Vision and Pattern Recognition (2012)
42. Tardif, J.P.: Non-iterative approach for fast and accurate vanishing point detection. In: International Conference on Computer Vision (2009)
43. Tron, R., Vidal, R.: A benchmark for the comparison of 3-d motion segmentation algorithms. In: Conference on Computer Vision and Pattern Recognition (2007)
44. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: Conference on Computer Vision and Pattern Recognition (2009)