G2MF-WA: Geometric multi-model fitting with weakly annotated data

Chao Zhang\(^1\) (✉), Xuequan Lu\(^2\), Katsuya Hotta\(^1\), and Xi Yang\(^3\)

© The Author(s) 2020.

Abstract In this paper we address the problem of geometric multi-model fitting using a few weakly annotated data points, which has been little studied so far. In weak annotating (WA), most manual annotations are supposed to be correct yet inevitably mixed with incorrect ones. Such WA data can naturally arise through interaction in various tasks. For example, in the case of homography estimation, one can easily annotate points on the same plane or object with a single label by observing the image. Motivated by this, we propose a novel method to make full use of WA data to boost multi-model fitting performance. Specifically, a graph for model proposal sampling is first constructed using the WA data, given the prior that WA data annotated with the same weak label has a high probability of belonging to the same model. By incorporating this prior knowledge into the calculation of edge probabilities, vertices (i.e., data points) lying on or near the latent model are likely to be associated and further form a subset or cluster for effective proposal generation. Having generated proposals, α-expansion is used for labeling, and our method in return updates the proposals. This procedure works in an iterative way. Extensive experiments validate our method and show that it produces noticeably better results than state-of-the-art techniques in most cases.

Keywords geometric multi-model fitting; weak annotation; multi-homography detection; two-view motion segmentation

1 Introduction

Geometric model fitting aims to fit a model to data which contains both inliers and outliers. A well-known approach is RANSAC \(^1\), the main idea of which is to generate a number of random model proposals and select as the best solution the one which includes the largest inlier set based on an inlier threshold. The geometric multi-model fitting task further assumes that the input data requires multiple models. Multi-model fitting algorithms have to optimize the global solution, rather than taking a greedy strategy to maximize inliers for single models like RANSAC. To evaluate the numerous possible solutions, a common approach is to design an energy function \(^2–5\), such that an approximate solution can be achieved by energy minimization (or maximization) by balancing geometric errors (data fidelity) and regularity of inlier clusters (e.g., smoothness, complexity). Although finding the optimal solution is NP-hard \(^4\), α-expansion \(^2\) provides a powerful alternative which can find solutions with guaranteed approximation bounds over a given set of model proposals. However, the quality of the solution and convergence largely depend on the quality of the proposals, which greatly influence the overall efficiency and effectiveness.

Most methods attempt to improve the quality of model proposals by sampling “clean” subsets of data points from the input data. We, however, claim that weakly annotated data (WA data), which has so far been little considered in multi-model fitting tasks, can be exploited to improve the quality of the proposals and improve fitting performance. In Fig. 1, we show an example of weak annotations in a scenario of motion segmentation. There are four objects in the image with independent motions as can
be seen from a pair of two-view images (i.e., a two-view motion segmentation problem; only one view is shown here). Due to camera shake, the movement of feature points on each object may also involve camera motion, biasing inliers and making it hard to distinguish outlier points on objects by observation. Nevertheless, a human annotator can at least tell that points on the same object should either belong to the outlier model or a specific motion model for that object. We refer to such imprecise annotations as weak annotations (WA) in this work (Fig. 1(b)). To take advantage of WA data, two priors are observed and exploited: (i) data points with the same weak label have a high probability of belonging to the same model, and (ii) data points with different weak labels have a low probability of belonging to the same model, other than the outlier model.

The main technical contribution of this paper is showing how to independently construct a proposal sampling graph with only WA data, separately from the adjacency graph of the $\alpha$-expansion [2]. Inspired by the random clustering approach [6], the sampling graph probabilistically forms subsets or clusters used to generate model proposals controlled by edge probabilities. By incorporating the prior assumptions mentioned above into the update procedure for the edge probabilities, proposals with high quality can be generated, thereby leading to good fitting performance. Extensive experiments validate our approach, and show that it often outperforms state-of-the-art methods, in terms of accuracy and runtime.

2 Related work

In this section, we first review two popular categories of techniques for multi-model fitting. Then, we consider efficient proposal generators, which are important to both of the above fitting techniques and closely related to our study.

2.1 Greedy methods

RANSAC [1] and its variants [7–9] belong to a category of methods which aim to estimate the parameters of a single model by greedily choosing the largest number of inliers (i.e., maximum consensus). The main philosophy is to iterate the following two steps: (i) generating “good” proposals based on proposal-verification, (ii) refining the proposals by maximum consensus. Since RANSAC is efficient when fitting a single model, many researchers have worked on extending it to multi-model cases [10–12]. In Refs. [10, 11], standard RANSAC is used sequentially, the model with maximum consensus in the current round being removed in the next round. However, in Ref. [12], the authors claim that a parallel approach is more stable than a sequential one for dealing with multiple models. Other common greedy methods are considered in Refs. [13–15]. The authors of Ref. [15] solve the multi-model fitting problem in terms of set coverage. In Refs. [13, 14], data points lying on or near the same model are considered to share similar preferences (a vector of proposals is sorted according to residual). This is an important property for grouping points into the same model, which is also taken into account for edge probability calculations in our work.

2.2 Energy-based methods

Optimization frameworks have become the predominant approach to solving the fitting problem in recent years. Energy-minimization based methods [3–5, 16] design a global energy function (objective function) to evaluate solutions, and the optimal solution the one with minimum energy value. The energy function is composed of different terms such as a data fidelity term [4], smoothness term [4], and label term [3]. In Ref. [16], multi-model fitting of geometric structures is formulated in terms of quadratic programming, in which data fidelity and similarity of associated data are balanced. Most energy-based methods follow a two-stage strategy: (i) generating a large number of proposals using random subsets of data,
and (ii) evaluating the quality of each proposal using likelihood functions [17]. Proposals with large likelihood values are sampled and used for labeling. For more on multi-model fitting methods, interested readers can refer to the survey in Ref. [18].

2.3 Proposal generation

Both of the above categories of methods require high-quality proposals to achieve low fitting errors or a good convergence rate. In particular, in the case of a large data set, it is computationally impractical to exhaustively evaluate each possible model proposal with full data. Furthermore, the correct number of models is usually unknown in real-world tasks. Such challenges motivate the design of fitting algorithms which discretize the sampling space using subsets of the data and generate proposals by fitting each subset. Although the proposals can be updated iteratively in a propose-and-refine fashion, different initialization of proposals can cause convergence to differing results. The generation of high-quality proposals at an early stage, an important issue in computer vision from the general perspective of robust fitting [19], is crucial for high-quality final labeling results for both greedy and energy-based methods.

Instead of full random initialization, many works improve the quality of proposals by utilizing information from inliers [20], or certain meta-information (e.g., keypoint matching score) [7, 21, 22] or sparsity priors [23]. The main factors that affect the quality of proposals include the inlier rate of each subset, and the sizes of the subsets: a large subset with a high inlier rate provides a high-quality proposal. The issues here are that small subsets with high inlier rates may amplify noise [19] while large subsets with low inlier rates may decrease the efficiency of proposal sampling [19] and lead to an exponential growth in computational cost. Pham et al. [6] overcame this problem by using the Swendsen–Wang method [24] to improve the efficiency of proposal sampling.

In this paper, we show how to generate high-quality proposals from WA data, which is a new approach. Note that a proposal generated from an outlier-free sample is not guaranteed to be consistent with all inliers in practice, which makes the problem challenging even with the aid of weakly annotated data. We show how to resolve this issue in the next section.

3 Our approach

3.1 Preliminaries

In order to achieve good multi-model fitting results, we aim to produce effective subsets for generating model proposals with high inlier rates under the guidance of partially and inaccurately labeled data (i.e., weak annotations). The multi-model fitting problem can be formulated mathematically as follows. We are given an input data set \( X = \{x_i\}_{i=1}^N \), which contains outliers and weakly annotated data \( \hat{X} = \{\hat{x}_a\}_{a=1}^Z \), which has arisen from multiple unknown models \( M = \{m_k\}_{k=1}^K \) which need to be estimated (\( m_1 \) is the outlier model, and \( K \) is also unknown).

Each \( x_i \) is assigned to a certain \( m_k \) by labeling, the labels being denoted by \( L = \{l_i\}_{i=1}^N \), and indicating that \( x_i \) is assigned to a certain model in \( M \). From a perspective of energy minimization, this problem can be solved by minimizing the following global energy function:

\[
E(X,M,L) = D(X,M,L) + S(X,L) + O(M)
\]

where the data term \( D \) is usually a distance or error metric to evaluate the data fidelity arising from the labeling result. In this paper, residuals in the form of Sampson distances [25] are used; larger \( D \) indicates larger errors in assigned labels. The smoothness term \( S \) is based on the prior assumption that spatially close neighbors have the same label with a high probability. Neighbors are determined by a neighborhood system (e.g., Delaunay triangulation), with weights on edges indicating how likely it is that two data points come from the same model. A larger \( S \) indicates worse local smoothness. The complexity term \( O \) penalizes the complexity (e.g., number of models) of the solution determined by the whole optimization task. Solution exploration by minimizing \( E \) has been shown to be effective in many works [4, 6]. We aim to explore the solution more effectively and efficiently with the help of the WA data, which can be readily produced interactively by a human operator, as feature points belonging to different models are usually visually distinguishable as points in images belonging to different objects, structures, etc.

3.2 Proposal sampling graph with WA data

The solution quality of \( \min_{M,L} E(X,M,L) \) is closely related to the quality of the proposals generated
by the data subsets sampled from \( \mathcal{X} \). We build a sampling graph \( \hat{G} = (v, e) \) from \( \hat{X} \), and separate from the adjacency graph \( G \) built from \( \mathcal{X} \). See for example \( \hat{G} = (v, e) \) in Fig. 2 for a specific neighbor system. In our implementation, Delaunay triangulation is used to construct the neighbor system as suggested by Ref. [3]. \( d_{ij} \) can be treated as a “switch” to turn on or off the connection between vertices, with the probability determined by the corresponding \( w_{ij} \). A certain sample of \( \{d_{ij}\} \) links to a clustering result of \( \hat{X} \), and each subset is used to calculate the model proposal \( \theta_g \) depending on the task setting. For example, in the case of a multi-homography detection task, the homography proposals can be found by the direct linear transformation (DLT) method [25] as long as the number of points in each subspace is four or above. \( w_{ij} \) objectively indicates how likely it is that a pair of points \((\hat{x}_i, \hat{x}_j)\) belong to the same model. In unsupervised situations, a common idea is to assume that the preferences of inliers from the same model are correlated over a set of so-far-generated proposals [6, 22]. Specifically, let \( \mathcal{H} = \{\theta_g\}_{g=1}^G \) be the set of proposals generated during iteration, and the residuals of \( \hat{x}_u \in \hat{X} \) with respect to each proposal \( \theta \) in \( \mathcal{H} \) form a vector

\[
r^{x_u} = (r^{x_u}_1, r^{x_u}_2, \ldots, r^{x_u}_G)
\]

This can be viewed as a preference vector quantified by residuals. By sorting \( r^{x_u} \) in ascending order and omitting the elements after the \( h \)-th place, the preference permutation can be represented as

\[
p^{x_u} = (p^{x_u}_1, p^{x_u}_2, \ldots, p^{x_u}_h)
\]

where each element in \( p^{x_u} \) is a proposal in \( \mathcal{H} \). Then \( w_{ij} \) can be updated by the correlation [6] between \( p^{x_u} \) and \( p^{x_v} \) in an online fashion using

\[
w_{ij} = \frac{|p^{x_u} \cap p^{x_v}|}{h}
\]

The main drawback of Eq. (4) is that the confidence of \( w_{ij} \) grows as \( G \) becomes larger. At the beginning of any iterative algorithm, \( w_{ij} \) can have low confidence, which hinders the whole algorithm from converging to the correct solution. Our aim is to utilize the prior knowledge brought by the WA data to make the \( w_{ij} \) more confident: WA data with the same weak label has a high probability of being assigned assigned to the same model and vice versa. By incorporating this property, Eq. (4) can be reformulated as a weighted function:

\[
w_{ij} = \lambda |p^{x_i} \cap p^{x_j}|/h + (1 - \lambda) \Pr(\hat{x}_i, \hat{x}_j)
\]

where

\[
\Pr(\hat{x}_i, \hat{x}_j) = \begin{cases} 1-\sigma, & \hat{x}_i, \hat{x}_j \ have \ the \ same \ weak \ label \\ \sigma, & \text{otherwise} \end{cases}
\]

\( \Pr(\hat{x}_i, \hat{x}_j) \) is a Bernoulli prior distribution, and \( \lambda, \sigma \in [0, 1] \). The prior distribution could also be learnt using more complex distribution models such as Gaussian mixture models [26], but we have empirically found that a Bernoulli distribution works well in our experiments.

### 3.3 Proposal sampling and labeling with WA data

Having introduced the proposal sampling graph, updating of proposals and of labeling results can be realized by alternately sampling \( d = \{d_{ij}\} \) and optimizing \( \mathcal{L} \) within a random clustering framework [6], which solves \( \min_{d, \mathcal{L}} E \) instead of \( \min_{\mathcal{M}, \mathcal{L}} E \). Note that sampling of \( d \) and optimization of \( \mathcal{L} \) are respectively conducted with the two graph models \( \hat{G} \) and \( G \) in our method. \( G \) is built with \( \mathcal{X} \) for \( \alpha \)-expansion [2], as illustrated in Fig. 3.

The two steps can be summarized as follows:

**Step (1) \( \mathbb{P}(d|\hat{G}) \).** Sample \( d \) with the current labeling of WA data \( (\hat{L} \in \mathcal{L} \ on \ \hat{G}) \):

- \( \mathbb{P}(d_{ij} = 1|\hat{l}_i = \hat{l}_j) := w_{ij} \)
- \( \mathbb{P}(d_{ij} = 1|\hat{l}_i \neq \hat{l}_j) := 0 \)

**Step (2) \( \mathbb{P}(\mathcal{L}|d) \).** Optimize \( \mathcal{L} \) by minimizing Eq. (1) with the current \( d \):

- Generate a proposal according to the sampled \( d \) on \( \hat{G} \)
- Update \( \mathcal{L} \) by \( \alpha \)-expansion, taking the new proposal into account.

---

**Fig. 2** Proposal sampling graph. Vertices represent WA data points \( \hat{x}_i \). They are divided into two independent subsets (blue/yellow), according to the connectivity of dotted edges \( d_{ij} \). \( d_{ij} \) are binary “bonds” in a random cluster model. Here, \( d_{25}, d_{35}, d_{47} \) have weights of zero and others have weights of one. \( w_{ij} \) denotes the edge probability between \( i \)-th and \( j \)-th vertices, used to probabilistically determine the values of \( \{d_{ij}\} \). Clusters induced by \( \{d_{ij}\} \) form model proposals.
The complexity term \( O \) in Eq. (1) is not involved in \( \alpha \)-expansion, as our algorithm does not follow the two-stage strategy [4, 16, 27] of generating a huge number of random proposals and conducting labeling based on the proposals. In our method, one proposal is generated and probabilistically included or excluded using a framework of simulated annealing, so does not suffer from the complexity problem. The smooth term \( S \) in Eq. (1) follows the Potts model [2] and is defined as \( \sum_{\langle i,j \rangle \in G} c_{ij}s_{ij} \), where

\[
s_{ij} = \begin{cases} 
1, & l_i = l_j \\
0, & l_i \neq l_j 
\end{cases}
\] (7)

The smooth prior \( c_{ij} \) can be defined using a spatial prior, since closer points in the neighbor system are more likely to belong to the same model. For simplicity, we set \( c_{ij} \) as a fixed constant that only penalizes discontinuity for each edge.

**Step (1)** generates clusters of WA data, with each cluster indicating a model proposal. **Step (2)** uses the proposals to perform labeling, and the labeling result in turn encourages **Step (1)** to connect those WA points having the same label. Obviously, this is a chicken-and-egg problem as the calculation in each of the two steps depends on the result of the other step. Good labeling improves clustering and vice versa. An iterative algorithm is a realistic solution in this situation. We modify the simulated annealing approach in Ref. [6] to further involve an optional subjective prior limitation by introducing a flag variable \( f \in \{0, 1\} \) to indicate whether the number of weak labels equals the number of embedded models (apart from the outlier model) \( (f = 1) \) or not \( (f = 0) \). The whole procedure is listed in Algorithm 1.

To demonstrate the improvement on convergence, we compare our method with SA-RCM [6] in Fig. 4. As the simulated annealing iterations are meta-heuristic, we use 100 runs for each method with different random seeds. We observe from Fig. 4 that G2MF-WA (our method) converges faster with lower segmentation errors than SA-RCM in most trials. It is easy to see that, with the aid of WA data, G2MF-WA generally achieves convergence in less than 0.5

![Fig. 3 Constructing a neighbor system by Delaunay triangulation. Each point \( x_i \) is in \( \mathcal{X} \). For high-dimensional data (e.g., feature-point pairs in 4D in homography detection), the concatenated data is projected onto the first two principal axes extracted by PCA, and the neighbor system is constructed in this 2D plane. Distant edges are removed.](Image)

**Algorithm 1** Geometric multi-model fitting with weak annotations (G2MF-WA)

**Input:** Data set \( \mathcal{X} \), including WA data \( \mathcal{X} \). Proposal set \( \Theta = \emptyset \), proposal pool \( \mathcal{H} = \emptyset \), \( \lambda = 0.1 \), \( \sigma = 0.1 \), \( h = 10 \), \( c_{ij} = 0.1 \), initial temperature \( T \), \( f \in \{0, 1\} \), number of ground-truth labels nlabels if \( f = 1 \)

**Output:** Estimated models \( \Theta \) and labels \( \mathcal{L} \)

1. Construct \( \mathcal{G} \) and \( \mathcal{G} \) from \( \mathcal{X} \) and \( \mathcal{X} \) respectively
2. if \( f = 0 \) then
3. Sample randn \( \in [0, 1] \) uniformly
4. if \( \text{randn} < 0.5 \) then
5. Update \( w \) using Eq. (5) in \( \mathcal{G} \) and sample a \( d \) (Section 3.2)
6. Update proposal set \( \Theta \) by adding the new proposal \( \theta_b \) generated from \( d \), \( \Theta' = \Theta \cup \theta_b \) (Section 3.3)
7. Add \( \theta_b \) to \( \mathcal{H} \)
8. else
9. Randomly remove one proposal \( \theta_b \) from \( \Theta \): \( \Theta' = \Theta \setminus \{ \theta_b \} \)
10. end if
11. else if \( f = 1 \) then
12. if \( |\Theta| < \text{nlabels} \) then
13. Run steps 5-7
14. else
15. Run step 9
16. end if
17. end if
18. if \( |\mathcal{H}| > 100 \) then
19. \( h = 0.1|\mathcal{H}| \)
20. end if
21. Estimate \( \mathcal{L}' \) with \( \Theta' \) via \( \alpha \)-expansion on \( \mathcal{G} \) (Section 3.3)
22. if \( E(\mathcal{X}, \Theta', \mathcal{L}') < E(\mathcal{X}, \Theta, \mathcal{L}) \) then
23. \( \Theta = \Theta' \), \( \mathcal{L} = \mathcal{L}' \)
24. else
25. Run step 23 with probability \( \exp ((E(\mathcal{X}, \Theta, \mathcal{L}) - E(\mathcal{X}, \Theta', \mathcal{L}'))/T) \)
26. \( T := 0.99T \) and repeat from step 2 until \( T \approx 0 \)
27. end if
Fig. 4 Convergence analysis on the dataset Hartley from Ref. [28]. Dotted lines denote convergence curves for different random seeds.

seconds, while SA-RCM still has not converged after 1.5 seconds in some trials.

4 Experimental results

We first explain the experimental setup, including the techniques used as a basis for comparison, and the parameter settings. We then consider applications of our approach: multi-homography detection, two-view motion segmentation, and augmented reality.

4.1 Experimental setup

We have compared our method (G2MF-WA) with two state-of-the-art methods, PEARL [4] and SA-RCM [6]. Note that the comparisons are not performed with the same inputs, as we utilize additional weakly annotated data (which can be easily obtained). The purpose of the comparisons is to demonstrate that the weak annotations can help achieve better fitting results. Older methods [16, 27, 29], which have been shown to be less accurate [6], are not included. Parameters of all methods were carefully tuned based on the authors’ implementations for best performance. The settings of each method are explained as follows.

PEARL [3]. As a typical two-stage method, PEARL generates a large proposal set $\Theta$ at once, followed by energy minimization (Eq. 1) with a complexity term. This differs from SA-RCM and G2MF-WA which expand $\Theta$ sequentially from an empty set. The complexity term is formulated using label costs, and counts the number of unique labels in $L$ to penalize complex solutions. Minimization is realized by running $\alpha$-expansion iteratively, and the optimum solution after each iteration $L'$ corresponds to an optimum $\Theta' \in \Theta$. $\Theta'$ is then refined with the labeling results $L'$ and $\Theta$ is replaced by $\Theta'$. Iterations are repeated until convergence. It is obvious that the number and quality of proposals in the initial $\Theta$ significantly affect the final result.

We set $|\Theta| = 1000$ to ensure accuracy. The minimum number of iterations is set to 10 and the maximum number to 20: PEARL often converges within a few iterations.

SA-RCM [6]. To control convergence in a practical way, the minimum number of iterations is set to 500 and the maximum number to 5000. Iteration is terminated when the energy changes little between iterations. Unlike G2MF-WA, SA-RCM conducts proposal sampling and $\alpha$-expansion in the same adjacency graph over all data points, which has a high computational cost from Eq. (4) when the sizes of $\mathcal{X}$ and $\mathcal{G}$ are large.

G2MF-WA (our method). G2MF-WA uses a similar simulated annealing framework to SA-RCM. The hyperparameters are shown in Algorithm 1 and the number of iterations is set as for SA-RCM. We generate simulated weak annotations instead of real manual annotations to facilitate annotation controllability and enable large-scale comparisons. The simulated weak annotations consist of two types of data: (i) $N_g$ points are randomly selected from each ground-truth label (except the outlier), and (ii) $N_o$ points from the ground-truth outlier are selected and assigned other ground-truth labels randomly. Different weak annotation settings were evaluated as below:

- G2MF-WA-A: $N_g = 5, N_o = 5, |\hat{\mathcal{X}}| = 10$
- G2MF-WA-B: $N_g = 10, N_o = 10, |\hat{\mathcal{X}}| = 20$
- G2MF-WA-C: $N_g = 10, N_o = 5, |\hat{\mathcal{X}}| = 15$

The segmentation error [6] based on the ground-truth labeling is used as the evaluation criteria; it can be calculated for all methods. All experiments were conducted on an off-the-shelf PC with an Intel i7 CPU (3.6 GHz) and 32 GB RAM.

4.2 Application 1: multi-homography detection

Given two views of a scene, a number of feature points can be extracted from these two images and matched by feature matching techniques. Matched points can be further related by a $3 \times 3$ homography matrix if the points lie on the same planar structure. The goal of multi-homography detection is to recover such homography matrices from a set of matches. Each model corresponds to one homography and incorrect matches correspond to the outlier model in our fitting algorithm. The DLT algorithm [30], which requires at least four matches, is employed for model generation, and the residual error is
calculated by Sampson distance. The full H part of the AdelaideRMF dataset [28] is used in this experiment for fairness purposes. Examples and statistical results are shown in Fig. 5 and Table 1, respectively. In Table 1, colored cells represent top-3 cells in each row (same dataset, different methods), in terms of median error and average processing time. According to the total number of colored cells in each column (summarized in the last row), it is clear that G2MF-WA-A converges fastest and G2MF-WA-B and G2MF-WA-C achieve lowest segmentation error. The processing time includes both sampling

![Fig. 5](image)

Table 1 Median results over 100 trials on the multi-homography detection task with full AdelaideRMF dataset (H part). Darker colors represent lower errors (%) and runtime (in seconds), denoted by fuchsia and cyan, respectively. The top-3 cells in each row are colored

| Method         | PEARL [4] | SA-RCM [6] | G2MF-WA-A | G2MF-WA-B | G2MF-WA-C |
|----------------|-----------|------------|-----------|-----------|-----------|
| Dataset (#labels) | Error | Time | Error | Time | Error | Time | Error | Time | Error | Time |
| barrsmith (3) | 10.37 | 1.37 | 10.37 | 1.35 | 10.37 | 1.05 | 1.66 | 1.52 | 1.66 | 1.02 |
| bonhall (7) | 6.09 | 2.88 | 8.80 | 4.11 | 8.57 | 3.75 | 9.08 | 3.95 | 8.80 | 3.93 |
| bonython (2) | 1.52 | 1.24 | 2.27 | 1.05 | 1.52 | 0.81 | 1.52 | 1.01 | 1.52 | 0.84 |
| elderhalla (3) | 20.56 | 1.40 | 6.54 | 1.02 | 7.01 | 0.92 | 5.61 | 0.97 | 6.07 | 0.95 |
| elderhallb (4) | 21.57 | 1.43 | 6.27 | 0.95 | 5.88 | 1.01 | 5.10 | 1.11 | 5.29 | 1.08 |
| hartley (3) | 2.19 | 1.66 | 1.88 | 1.28 | 2.19 | 1.08 | 2.19 | 1.16 | 2.19 | 1.15 |
| johnsona (5) | 7.51 | 1.61 | 3.49 | 1.27 | 7.51 | 1.31 | 3.22 | 1.46 | 3.22 | 1.42 |
| johnsonb (8) | 14.10 | 2.13 | 9.86 | 2.16 | 14.33 | 2.37 | 9.40 | 2.54 | 10.02 | 2.57 |
| ladyssycon (3) | 4.64 | 1.33 | 5.06 | 0.93 | 4.64 | 0.96 | 4.64 | 0.99 | 4.64 | 1.01 |
| library (3) | 3.26 | 1.30 | 3.26 | 0.88 | 2.79 | 0.90 | 2.79 | 0.94 | 2.79 | 0.96 |
| mcl (6) | 10.90 | 4.73 | 4.30 | 9.11 | 5.36 | 7.71 | 4.51 | 8.13 | 4.87 | 8.22 |
| mc3 (7) | 29.92 | 4.80 | 4.69 | 9.22 | 3.88 | 8.38 | 2.93 | 8.42 | 2.62 | 8.51 |
| napiera (3) | 17.22 | 1.58 | 18.05 | 1.40 | 12.58 | 1.09 | 13.25 | 1.12 | 13.25 | 1.20 |
| napierb (4) | 17.37 | 1.44 | 18.92 | 1.00 | 20.08 | 1.07 | 15.44 | 1.12 | 17.95 | 1.15 |
| neem (4) | 7.47 | 1.41 | 5.39 | 0.89 | 6.43 | 0.99 | 4.98 | 1.04 | 4.56 | 1.06 |
| nese (3) | 0.79 | 1.38 | 0.79 | 0.86 | 0.79 | 0.92 | 0.79 | 1.45 | 0.79 | 1.01 |
| oldclassicswing (3) | 1.06 | 1.65 | 1.06 | 1.16 | 1.06 | 1.16 | 1.06 | 1.23 | 1.06 | 1.29 |
| physics (2) | 19.81 | 1.03 | 26.42 | 0.88 | 19.81 | 0.72 | 23.58 | 0.77 | 27.36 | 0.75 |
| raglan (12) | 17.73 | 6.07 | 45.48 | 15.17 | 42.46 | 14.79 | 10.23 | 15.03 | 15.00 | 15.08 |
| sene (3) | 1.20 | 1.39 | 1.20 | 0.91 | 1.20 | 0.91 | 1.20 | 1.13 | 1.60 | 0.99 |
| sene (3) | 32.17 | 5.45 | 4.32 | 10.18 | 4.70 | 8.37 | 2.50 | 8.80 | 2.40 | 8.73 |
| unionhouse (6) | 38.15 | 5.25 | 4.17 | 10.19 | 2.74 | 8.59 | 2.54 | 8.72 | 2.54 | 8.72 |

#colored cells in each column | 14 | 7 | 17 | 11 | 15 | 22 | 21 | 15 | 21 | 14
and optimization time. In Fig. 5, according to the segmentation error in parentheses, we see that whether the optional assumption holds true (1st row) or not (2nd row), the WA data clearly contributes to improving the performance.

4.3 Application 2: two-view motion segmentation

Given two views of a scene and feature point matches, the goal of two-view motion segmentation is to estimate simultaneously motion models modeled by $3 \times 3$ fundamental matrices and the labeling. Points in a match are supposed to undergo the same motion (usually on the same object or background). Outliers correspond to incorrect matches. The full F part of the AdelaideRMF dataset [28] was employed for fairness. Examples and statistical results are shown in Fig. 6 and Table 2, respectively. The DLT algorithm, which requires at least eight matches, is adopted for model generation, and the residual error is calculated.

![Fig. 6 Examples of two-view motion segmentation. Above: cubetoy shows the situation when the number of weak labels in (d) equals the number of ground-truth labels in (a) without considering the outlier label (red circle in (a)), i.e., 2 labels in (d) and 3 labels in (a) including the outlier. Below: game shows the situation when only one label exists. In (d) and (i), WA data points are shown.](image)

| Table 2 | Median results over 100 trials on the two-view motion segmentation task with full AdelaideRMF dataset (F part). Darker colors represent lower errors (%) and runtime (in seconds), denoted by fuchsia and cyan, respectively. Top-3 cells in each row are colored |
|---------|---------------------------------------------------------------------------------|
| Dataset (#labels) | PEARL [4] | SA-RCM [6] | G2MF-WA-A | G2MF-WA-B | G2MF-WA-C |
| biscuit (2) | 0.30 1.01 0.00 1.36 | — — 0.61 1.07 0.61 1.05 |
| biscuitbook (3) | 1.47 1.04 2.93 1.04 | — — 1.17 1.17 2.35 1.12 |
| biscuitbookbox (4) | 1.93 0.98 2.70 0.89 | — — 1.93 1.07 1.93 1.01 |
| boardgame (4) | 16.49 1.00 16.85 0.95 | — — 16.13 1.15 17.20 1.09 |
| book (2) | 1.07 0.87 0.53 0.97 | — — 0.53 0.83 0.53 0.78 |
| breadcartoychips (5) | 29.11 0.95 6.96 0.92 | — — 13.71 1.04 8.86 1.00 |
| breadcube (3) | 5.58 0.96 23.55 0.83 | — — 7.64 0.85 3.72 0.83 |
| breadcubechips (4) | 14.78 0.93 7.39 0.85 | — — 9.57 0.99 7.39 0.94 |
| breadtoy (3) | 3.82 1.01 3.65 0.97 | — — 3.82 1.05 3.30 1.00 |
| breadtoyocar (4) | 31.33 0.86 9.04 0.76 | — — 18.37 0.92 9.34 0.87 |
| carchipscube (4) | 15.15 0.87 12.12 0.73 | — — 3.64 0.88 2.42 0.85 |
| cube (2) | 1.66 1.00 10.93 1.18 | — — 2.65 1.00 1.32 0.97 |
| cubebreadcartoychips (5) | 21.41 1.08 8.10 1.13 | — — 7.80 1.25 9.02 1.20 |
| cubechips (3) | 7.75 0.99 3.52 0.94 | — — 5.63 1.05 4.05 0.99 |
| cubetoy (3) | 7.23 0.95 5.62 0.82 | — — 5.22 0.97 4.02 0.90 |
| dinobooks (4) | 20.83 1.09 17.08 1.24 | — — 16.25 1.25 12.50 1.20 |
| game (2) | 1.72 0.90 4.72 1.05 | — — 3.43 0.87 0.86 0.83 |
| gamebiscuit (3) | 1.52 1.04 6.10 1.01 | — — 1.22 1.16 0.61 1.05 |
| toycubeocar (4) | 23.50 0.89 13.00 0.73 | — — 13.00 0.92 14.00 0.89 |
| #colored cells in each column | 12 | 19 | 14 | 16 | — | — | 19 | 8 | 18 | 19 |
using Sampson distance. Results are unavailable for G2MF-WA-A in Table 2, as it is impossible to sample correct proposals when \(N_g = 5\), as \(5 < 8\). We can clearly observe from Table 2 that G2MF-WA-B achieves the lowest error but performs the least efficiently, while G2MF-WA-C is fastest and is also competitive in achieving low error.

Unlike Ref. [6] that only used part benchmark data, we evaluate full benchmarks in both first and second applications.

Although the difference between the number of ground-truth labels and the number of WA labels is supposed to affect the final labeling result, our method can still detect the homography or fundamental matrices accurately (see e.g., the second row in Fig. 5). One potential reason could be the robustness of the \(\alpha\)-expansion algorithm to the initial estimate [4]. Also, although the WA data imposes priors on the edge probabilities between vertices in the sample graph, which allows the algorithm to generate proposals close to the intent of the annotator, the randomness of proposal generation is still included to ensure proposal diversity. This could also contribute to the above finding.

4.4 Application 3: planar augmented reality application

We further show a real-world application involving planar augmented reality in Fig. 7, which must insert multiple prepared images onto the planar structures in the scene. Here, visual success closely depends on the accuracy of detection of the planar surfaces in the scene. Our algorithm is designed to improve this accuracy with the help of additional weak annotations interactively provided by users.

5 Conclusions

In this paper, we have presented a multi-model fitting method which uses weakly annotated data. The main contribution is to take advantage of the prior knowledge brought by the weakly annotated data, and incorporate it into the calculation of edge probabilities in the proposal sampling graph, for effective model proposal generation and further labeling. Extensive experiments demonstrate that our method typically outperforms state-of-the-art methods in terms of both accuracy and runtime.

Despite the effectiveness of our method, it still has a few limitations. Since the model proposals are explored heuristically, fitting performance can depend on random seeding, and the segmentation error can grow while the algorithm remains stuck in a local optimum. One potential way to solve these issues is to increase the number of iterations of the simulated annealing algorithm. In future, we would like to design an interactive annotation interface and embed it within the proposed framework. We also plan to improve the usability of our algorithm by further reducing the user effort.

Acknowledgements

Chao Zhang is supported in part by JSPS KAKENHI Grant JP18K17823. Xuequan Lu is supported in part by Deakin CY01-251301-F003-PJ03906-PG00447.

References

[1] Fischler, M. A.; Bolles, R. C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* Vol. 24, No. 6, 381–395, 1981.

[2] Boykov, Y.; Veksler, O.; Zabih, R. Fast approximate energy minimization via graph cuts. In: Proceedings of the 7th IEEE International Conference on Computer Vision, 377–384, 1999.

[3] Delong, A.; Osokin, A.; Isack, H. N.; Boykov, Y. Fast approximate energy minimization with label costs. *International Journal of Computer Vision* Vol. 96, No. 1, 1–27, 2012.

[4] Isack, H.; Boykov, Y. Energy-based geometric multi-model fitting. *International Journal of Computer Vision* Vol. 97, No. 2, 123–147, 2012.
[5] Amayo, P.; Pinies, P.; Paz, L. M.; Newman, P. Geometric multi-model fitting with a convex relaxation algorithm. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8138–8146, 2018.

[6] Pham, T. T.; Chin, T. J.; Yu, J.; Suter, D. The random cluster model for robust geometric fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 36, No. 8, 1658–1671, 2014.

[7] Chum, O.; Matas, J. Matching with PROSAC: Progressive sample consensus. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1, 220–226, 2005.

[8] Nistér, D. Preemptive RANSAC for live structure and motion estimation. *Machine Vision and Applications* Vol. 16, No. 5, 321–329, 2005.

[9] Torr, P. H. S. Geometric motion segmentation and model selection. *Philosophical Transactions of the Royal Society of London Series A: Mathematical, Physical and Engineering Sciences* Vol. 356, No. 1740, 1321–1340, 1998.

[10] Vincent, E.; Laganieré, R. Detecting planar homographies in an image pair. In: Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis, 182–187, 2001.

[11] Zuliani, M.; Kenney, C. S.; Manjunath, B. S. The multiRANSAC algorithm and its application to detect planar homographies. In: Proceedings of the IEEE International Conference on Image Processing, Vol. 3, 131–1340, 1998.

[12] Toldo, R.; Fusiello, A. Robust multiple structures estimation with J-linkage. In: *Computer Vision – ECCV 2008. Lecture Notes in Computer Science*, Vol. 5302. Forsyth, D.; Torr, P.; Zisserman, A. Eds. Springer Berlin Heidelberg, 537–547, 2008.

[13] Magri, L.; Fusiello, A. T-linkage: A continuous relaxation of J-linkage for multi-model fitting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3954–3961, 2014.

[14] Magri, L.; Fusiello, A. Multiple models fitting as a set coverage problem. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3318–3326, 2016.
[28] Wong, H. S.; Chin, T. J.; Yu, J.; Suter, D. Dynamic and hierarchical multi-structure geometric model fitting. In: Proceedings of the International Conference on Computer Vision, 1044–1051, 2011.

[29] Pham, T. T.; Chin, T.-J.; Yu, J.; Suter, D. Simultaneous sampling and multi-structure fitting with adaptive reversible jump MCMC. In: Proceedings of the Advances in Neural Information Processing Systems 24, 540–548, 2011.

[30] Abdel-Aziz, Y. I. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. In: Proceedings of the ASP Symposium on Close-Range Photogrammetry, 1–18, 1971.

Chao Zhang received his Ph.D. degree from Iwate University (Japan) in 2017. He is now a full-time assistant professor at the Faculty of Engineering, University of Fukui (Japan). His research interests include computer vision and graphics, mainly focusing on feature matching and vision-based optimization problems. He is a member of the IEEE Computer Society, IEEE Signal Processing Society, ACM, and IEICE.

Xuequan Lu is a lecturer (assistant professor) at Deakin University, Australia. He spent more than two years as a research fellow in Singapore. Prior to that, he earned his Ph.D. degree from Zhejiang University (China) in 2016. His research interests lie mainly in visual computing, in areas such as geometry modeling, processing and analysis, animation and simulation, 2D data processing and analysis. More information can be found at http://www.xuequanlu.com.

Katsuya Hotta received his B.E. degree in 2017 and is now pursuing a Ph.D. degree at the University of Fukui, Japan. His current research focuses primarily on computer vision, mainly in subspace clustering and visual tracking.

Xi Yang is currently a project assistant professor in the Graduate School of Information Science and Technology at The University of Tokyo. He received his B.E. degree from the College of Information Engineering at Northwest A&F University in 2012. He received his M.E. and D.E. degrees from the Graduate School of Engineering, Iwate University. His research interests include geometric processing, visualization, and deep learning.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made.

The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

Other papers from this open access journal are available free of charge from http://www.springer.com/journal/41095. To submit a manuscript, please go to https://www.editorialmanager.com/cvmj.