Guide to Match: Multi-Layer Feature Matching with a Hybrid Gaussian Mixture Model

Kun Sun, *Member, IEEE,* Wenbing Tao, *Member, IEEE,* Yuhua Qian, *Member, IEEE,*

***Abstract*—As a fundamental yet challenging task in computer vision, finding correspondences between two sets of feature points has received extensive attention. Among all the proposed methods, the Gaussian Mixture Model (GMM) based algorithms show their great power in formulating such problems. However, they are vulnerable to large portion of outliers in the extracted feature points. In this paper, a new Hybrid Gaussian Mixture Model (HGMM) combined with a multi-layer matching framework is proposed. Different from existing GMM based methods, HGMM uses a set of seed correspondences to guide the matching procedure. To automatically find seed correspondences, the feature points are divided into multiple layers according to their matching potential. With the help of Locality Sensitive Hashing, this can be done economically and efficiently. Correspondences found in lower layers which contain few outliers will be used as hard constraint when matching features in higher layers where a large portion of outliers exist. Extensive experiments show that the proposed method is efficient and more robust to outliers when images have large viewpoint difference or small scene overlap.**

***Index Terms*—Feature matching, Multi-layer, Hybrid Gaussian Mixture Model.**

I. Introduction

FINDING feature point correspondences between two images is a fundamental task in vision-based tasks [1]–[15]. The typical pipeline includes three main steps: detecting salient keypoints, computing feature descriptors and finding matching relationships between them [16]–[20]. A subsequent mismatch removal method [21]–[26] could be optionally applied to refine the matching result. The Scale Invariant Feature Transform(SIFT) detector and descriptor [27] are widely used for their outstanding performance [28]. The number of correct matches and precision are two most concerned issues and have attracted great attention from researchers in this field. However, this is a non-trivial task because situations such as viewpoint differences, wide baseline, local ambiguity and occluding still pose great challenges for many real applications [29].

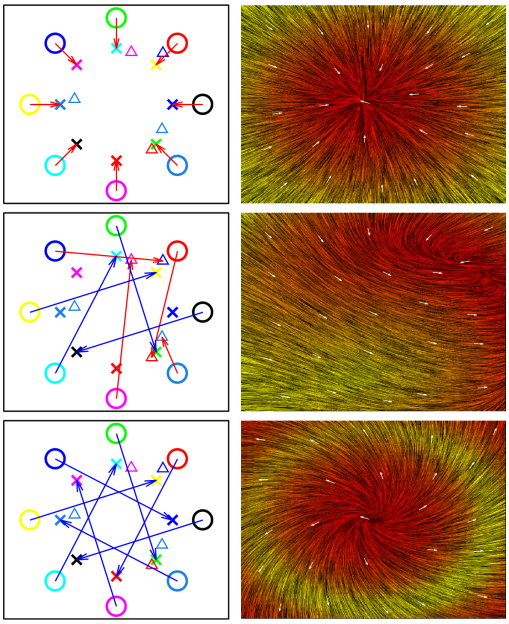

Fig. 1. A toy example showing the challenges for GMM based methods. The reference points are represented by small circles. The target points and outliers are represented by small crosses and triangles. Blue and red arrows on the left indicate correct and false correspondences. The interpolated motion field is visualized on the right. The first two rows are correspondences found by CPD [30] and NGMM [31], respectively. The last row is the result by fixing the four correct matches in the second row as prior when performing NGMM.

This paper focuses on the third step of the matching procedure, *i.e.* how to find good correspondences between two sets of extracted feature points. Traditional methods do this in an individual manner, which processes the feature points one by one by searching its nearest neighbor in the feature space. It is fast but sensitive to local feature ambiguity. Another way to find correspondences is global optimization. The Gaussian Mixture Model (GMM) shows its great power in formulating such problems. It is originally used in aligning two point sets [30], [32]. One point set is treated as the centers of GMM and the other point set is treated as the data. A coherent spatial transformation is then computed to update the position of the GMM until it is best fitted to the data. Despite great success in aligning two point sets, these methods could not

Manuscript received XXXX; revised XXXX. This work is supported by the Fundamental Research Funds for the Central Universities, China University of Geosciences(Wuhan) (No. CUG170675), by the National Natural Science Foundation of China (No. 61802356, No. 61772213), also in part by the Open Project Foundation of Intelligent Information Processing Key Laboratory of Shanxi Province (No. CICIP2018003), by Fund from Science,Technology and Innovation Commission of Shenzhen Municipality Grants (JCYJ20170818165917438) and Wuhan Science and Technology Plan under Grant 2017010201010121. (Corresponding author: Wenbing Tao.)

Kun Sun is with the Hubei Key Laboratory of Intelligent Geo-Information Processing, School of Computer Science, China University of Geosciences(Wuhan), Wuhan 430074, China(e-mail: sunkun@cug.edu.cn).

Wenbing Tao is with the National Key Laboratory of Science and Technology on Multi-spectral Information Processing, School of Automation, Huazhong University of Science and Technology, Wuhan 430074, China. He is also with the Shenzhen Huazhong University of Science and Technology Research Institute, Shenzhen 518057, China(e-mail: wenbingtao@hust.edu.cn).

Yuhua Qian is with the Institute of Big Data Science and Industry, also with Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Shanxi University, Taiyuan, 030006, China(e-mail: jinchengqyh@126.com).

be directly used to match feature points between two images since they only take spatial information into consideration. To tackle this problem, Tao and Sun proposed a method called Non-uniform Gaussian Mixture Model (NGMM) [31]. It fuses feature information by assigning different weights to the GMM components according to feature similarity. The feature similarity provides guidance for the spatial transformation and the spatial coherency resists local feature ambiguity. However, it's still challenging for NGMM when a large portion of outliers exist in the extracted feature points. Here, a feature point is denoted as an outlier if it is not repeatedly detected on the other image or it is not in the overlapping region between two views. When large inter-view geometric or photometric changing exist, the ratio of outliers will further increase (as large as 90% in some cases). Such a high portion of outliers ruins the inherent structure of correct matches, which trapped the optimization into a wrong solution.

A toy example is shown in Fig. 1 to illustrate the challenges. We first synthesizes 8 reference points on a 2D plane (represented by small circles). The colors of these points can be used as feature information. By applying rotation and scaling we get another set of target points (represented by small crosses). We also simulate outliers and feature ambiguity by adding five random points (represented by small triangles) to the target point set. The first row is the correspondences for the reference points found by CPD [30]. Without using the color information, it thought scaling as the most economic way to align two point sets. The correspondences are all wrong. The second row is the result found by NGMM [31]. With the guidance of colors, it recovered both scaling and rotation between two point sets. However, due to outliers and local ambiguity, some false correspondences are included. In real image feature matching applications, more outliers exist in both two point sets and the results become more vulnerable. Despite the false matches in the second row of Fig. 1, the correct matches found by NGMM could offer useful constraints for the other points. If we fix the four correct correspondences as prior knowledge and then recompute the correspondence between the remaining points, all of them could be correctly matched, which is shown in the last row of Fig. 1. This inspired us to find correspondences in a progressive way. That is, we can initialize some correct seed matches from a cleaner subset and let them guide us to find new correspondences in a gradually increasing search space.

In this paper, a new Hybrid Gaussian Mixture Model (HGMM) is proposed to solve the above problem. Different from existing GMM based methods such as CPD and NGMM which try to match all the points at a time, our method performs in a progressive manner. At the very beginning, feature points are divided into different layers. A few seed correspondences are automatically initialized from the first layer and then guide us to find new correspondences in the next layer. This is repeated until the last layer. Each step is a guided matching problem, which can be modeled by a hybrid of binary and continuous weights in the GMM. For feature points belonging to the seed matches, the corresponding GMM weights take binary values from $\{0, 1\}$ since they have fixed relationship. For the remaining feature points, their GMM weights are computed from pairwise feature similarities and regularized to continuous interval $[0, 1]$, as done by NGMM. With the guidance of the seed correspondences, both robustness and convergence speed are increased. To ensure that reliable seed correspondences could be initialized from the first layer without any prior and new matches could be found in the following layers, a well designed layer division method is required. Specifically, we first use the Locality Sensitive Hashing (LSH) algorithm [33] to map all the feature descriptors into binary codes and search for nearest neighbors in the hamming space. This algorithm is chosen because it can approximate feature similarities and matching potential between feature points efficiently. Afterwards, these nearest neighbors are sorted according to their hamming distance in an ascending order and divided into several layers. As a result, points in lower layers have small feature discrepancy, relatively high matching potential and lower outlier ratio. So it's easy to initialize some reliable seed matches in the first layer. Although in higher layers the matching potential decreases and the ratio of outliers increases, the matching performance can be protected by using seed correspondences found in the previous layer as hard constraint. In this way, HGMM encodes both intra-layer constraint and inter-layer constraint in a unified framework. The intra-layer constraint requires that correspondences should not only have similar feature descriptors but also satisfy a global coherent spatial structure. The inter-layer constraint uses correspondences found in a previous layer as prior when finding new correspondences in the next layer.

To summarize, the contribution of this paper lies in the following aspects: (1) A multi-layer feature matching framework is proposed to progressively find correspondences even if the feature points contain a high ratio of outliers. Seed correspondences are initialized in lower layers where the feature points have large matching potential and inlier ratio. They are used as hard constraint in higher layers in which a high ratio of outliers exist. (2) A new Hybrid Gaussian Mixture Model, called HGMM, is proposed. It uses a hybrid of continuous and binary weights for the GMM, which encodes both inter-layer guidance information and intra-layer feature-spatial information. Experiments show that the robustness and convergence speed are increased when compared with traditional GMM based methods such as CPD and NGMM.

The remainder of this paper is organized as follows. A brief review of related work is given in Section II. Section III introduces the proposed multi-layer matching algorithm, including the proposed HGMM and how to divide feature points via hashing acceleration. At last, experiments evaluation and conclusion are given in Section IV and Section V.

II. Related Work

Finding matches between two images is to find correspondences between two sets of keypoints. The most common way is to compute a feature descriptor which encodes the local image content around the keypoint and then search for its nearest neighbor in the feature space [34]. The searching strategy can be brute-force searching for small scale problem

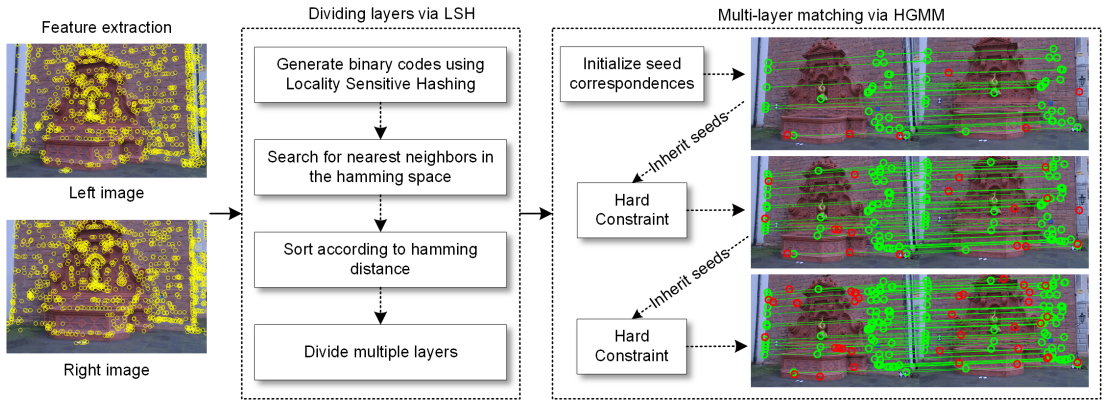

Fig. 2. An overall pipeline of the proposed method in this paper. After extracting SIFT features (yellow circles) from both images, they are mapped to binary codes via locality sensitive hashing, which enables us to efficiently find nearest neighbors in the hamming space. The nearest neighbors are sorted in a distance ascending order and divided into multiple layers. HGMM will initialize a few seed correspondences from the first layer and use them as hard constraint when finding new matches in the next layer. This step repeats until all the layers have be processed. Green lines and circles denote successfully matched feature points in that layer. Red circles indicate feature points that fail to find correspondences in that layer.

or kd-tree based searching for large scale problem. Since SIFT [27] has been proposed about 20 years ago, a variety of feature descriptors [35]–[37] are already off-the-shelf for real application. Recently, the powerful deep neural networks are used by researchers to develop better features [38]–[40]. For better use of such rich features, Lin *et al.* used the homography space as the domain to select a good descriptor locally instead of using one global descriptor [41]. This technique is then combined with a candidate enrichment framework that can find more correspondences in their later work [42]. To increase the speed of nearest neighbor searching, Cheng *et al.* adopted a hashing accelerating method CasHash [33]. The original SIFT features are mapped into the hamming space, in which a small candidate neighborhood set can be efficiently determined by performing bitwise operation. Much time can be saved since the searching space is reduced. Later Bian *et al.* proposed a real-time matching method GMS [43] which induces correspondences from motion smoothness. For fast implementation, motion statistics are computed with the help of shifting grid cells. To match images with large scale difference, Zhou *et al.* [44] divided the keypoints by multiple scale layers and then used Bag of Feature (BoF) [45] to determine the scale factor. Hartmann *et al.* [46] observed that when unmatchable feature points are removed, the matching performance could be improved. Hence they proposed a learning method to judge whether a feature point is matchable or not, but it requires labeled samples to train a complex classifier off-line.

Finding correspondences between point sets can also be accomplished with the spatial information, such as Kernel Correlation (KC) [47]. These methods usually find a spatial transformation to align points. The Gaussian Mixture Model (GMM) is widely used in this field. For example, the CPD algorithm [30] treated one point set as GMM and used the Motion Coherent Function to model the non-rigid transformation. By treating both point sets instead of one as GMMs, the GMMReg method [48] tried to align them via minimizing the L_2 distance between two distributions. The L_2E estimator is proven robust to outliers and later used in [49]. Except for registering two point sets, the correspondences between multiple point sets can also be acquired by spatially aligning them to a global GMM [50], [51]. Inspired by these works, new registration methods based on manifold regularization [52], context constraint [53], [54], or local connectivity constraint [55] are proposed. The Student's t-mixture model (TMM) is used as an alternative of GMM since t-distributions are naturally robust to outliers [56]. In this framework, Nguyen *et al.* [57] pointed that kernel selection is crucial for the registration task. Their method pruned ineffective kernels by adjusting their weights and proved to get better results.

However, finding correspondences with either feature or spatial information only could not receive satisfactory results. On the one hand, local features usually have strong ambiguity, which may result in wrong matches. On the other hand, aligning two point sets may fall into a wrong optimal solution when image content information is entirely abandoned. To address these problems, recent methods take advantage of both information in a unified framework. Torki *et al.* [58] and Hamid *et al.* [59] adopted subspace learning method to compute a new representation, which encodes both feature similarity between two point sets and preserves spatial structure within each point set. Then matching task is transfered from the original feature space into the new subspace. Inspired by this work, Sun *et al.* proposed a matching method based on subspace coherent constraint [60], [61]. After fusing feature and spatial information into a new subspace, a coherent spatial transformation is computed in this subspace to find correspondences between two point sets. Consistent correspondences can also be established via graph matching [62]–[65]. The correspondences between two graphs should not only have similar vertex features but also have consistent local edge structures. Sun *et al.* [31], [66] proposed a non-uniform Gaussian Mixture Model (NGMM) to find correspondences between images. Instead of using uniform weight for each GMM component, NGMM computes these weights according to feature similarities. Such a biased weighting strategy will use feature information to guide the spatial transformation.

Based on GMM, Yang *et al.* proposed to preserve both global and local structure in [67]–[70]. The core idea is to fuse different kinds of global and local features. Based on the feature descriptor information extracted from the images, a dynamic Gaussian component density was designed to recover inlier pairs [71]. Their transformation preserves both local structure and image space curvature.

Despite a lot of methods mentioned above, most of them rely on external constraint such as feature similarity and structure consistency. However, the value of internal guidance provided by the correspondences found earlier is somewhat overlooked. This inspired us to design a progressive matching algorithm in a multi-layer framework, which uses correspondences found in a subset to guide itself to find new correspondences in a larger set.

III. The Proposed Multi-Layer Feature Matching Algorithm

A. Overview

Fig. 2 is the overall pipeline of the propose method. Given an image I_1, a set of SIFT feature points $P = \{p_i\}_{i=1}^{N} = \{(x_i, f_i)|x_i \in \mathbb{R}^2, f_i \in \mathbb{R}^{128}\}_{i=1}^{N}$ are extracted. Each $p_i \in P$ is associated with a 2D spatial coordinate x_i and a 128D feature descriptor f_i. Similarly, the set of SIFT feature points on I_2 are denoted as $Q = \{q_j\}_{j=1}^{M} = \{(y_j, g_j)|y_j \in \mathbb{R}^2, g_j \in \mathbb{R}^{128}\}_{j=1}^{M}$. Usually $M \neq N$. Without loss of generality, for each feature point in P we find its tentative matching feature point in Q, *i.e.* image I_1 is matched to image I_2. Next, the feature descriptors are mapped to binary codes using the LSH method. After sorting all the nearest neighbors according to their hamming distance, the feature points are divided into multiple layers. Finally, we find correspondences with HGMM. It initializes some seed correspondences from the first layer and gradually passes them as hard constraint to the next layer. We will introduce each part separately in the following part.

B. Dividing Feature Points into Multiple Layers with Hashing Mapping

The basic idea of our multi-layer matching algorithm is: we can find some reliable prior matches on a small subset of feature points, and then use them as guidance to find new matches when more feature points are given. This prior knowledge reduces the risk of mismatching and speeds up the optimization. There are two requirements when dividing layers. (1) Inclusion constraint. Feature points that are likely to form tentative matches should be in the same layer, that is, in the same search space. This enables us to traverse the layers once without any cross-layer matching. (2) Order constraint. The matching potential between feature points should be decreasing from lower to higher layers, so that we can initialize reliable matches in the early stage when no prior knowledge is available.

To satisfy these requirements, we use the Locality Sensitive Hashing (LSH) method [33] to transfer the 128D SIFT descriptors to B-bit binary codes by a group of hash functions. Each hash function $h_b(v)$ is a randomly generated hyperplane and returns one bit of the code:

$$h_b(v) \longrightarrow \{0, 1\}, v \in \mathbb{R}^{128}, b = 1, \cdots, B. \tag{1}$$

After obtaining these binary codes, for each p_i we find its nearest neighbor $\widehat{q_j}$ in the hamming space and record the corresponding distance d_i. This could be done economically and efficiently with bitwise operation. In this way, dividing two unstructured feature point sets now changes to dividing a set of feature point pairs, which is more friendly to the first constraint. Since smaller hamming distance indicates higher matching confidence, d_i is an approximation of the matching potential. We sort all the feature point pairs according to d_i in ascending order to meet the second requirement. Ideally, two matching points are close in the feature space and tend to have the same hash codes. However, it is possible that they are separated by one of the random hash functions, which is a small probability event. In order to resist the randomness of LSH, we generate 20 groups of LSH functions and take the mean hamming distance.

Note that the nearest neighbor does not necessarily form a match, but a correct match is most likely to come from a nearest neighbor with small distance. Due to binary relaxation, there might be more than one nearest neighbors which are d_i far away from p_i. These nearest neighbors are treated as matching candidates of p_i. One can set an upper-bound threshold on d_i to discard feature points who present very large feature discrepancy with their nearest neighbors. But we didn't do this because we want to find correspondences as more as possible. All the feature points p_i together with their nearest neighbors are sorted by d_i in an ascending order and then divided into segments. An intuitive way is to put all points with the same hamming distance into a single layer. But this does not work well in practice since the layers might be quite unbalanced in size. In this case, dealing with very large layers is inefficient and becomes a bottleneck. Therefor, we divide each layer into constant size k. The number of layers K is then computed from $K = ceil(N/k)$. A feature point from Q might appear in different layers for the reason that it might be the nearest neighbor for more than one point from P, but we found that this issue can be easily solved by redundancy elimination in the following steps.

C. Finding Matches with the Hybrid Gaussian Mixture Model

After dividing feature points into layers, we start to find correspondences from the first layer. Since points in this layer have the most similar features, it's easy to find some reliable initial correspondences without any prior. These correspondences are inherited to provide guidance when we match new feature points in the next layer. If a feature point is not matched in the current layer, we send it to the next layer and try again. If the attempt still fails, it is probably an outlier and abandoned in the following layers. The above steps are repeated until all the layers have been traversed.

We denote feature points involved in the $L^{th}(1 \leq L \leq K)$ layer as $P_L = \{p_i^L\}_{i=1}^{N_L} = \{(x_i^L, f_i^L)|x_i^L \in \mathbb{R}^2, f_i^L \in \mathbb{R}^{128}\}_{i=1}^{N_L}$ and $Q_L = \{q_j^L\}_{j=1}^{M_L}\{(y_j^L, g_j^L)|y_j^L \in \mathbb{R}^2, g_j^L \in$

$\mathbb{R}^{128}\}_{j=1}^{M_L}$, respectively. Both P_L and Q_L consist of two parts: the first s_L seed matches $S = \{p_i^L, q_i^L\}_{i=1}^{s_L}$ inherited from the previous layer and the remaining wild points to be matched with the guidance of these seeds. The final output of our algorithm is the correspondences found in the last, *i.e.* the K^{th} layer. We can formulate it as the following optimization problem:

$$\begin{aligned} C = &\arg\min_{C_K, \Omega_K} E^K(C_K, \Omega_K; P_K, Q_K, C_{K-1}) \\ &s.t. \quad C_0 = \emptyset, \end{aligned} \tag{2}$$

in which Ω_K contains the parameters of the matching constraints in the K^{th} layer, C_K is the correspondences found in the K^{th} layer and C is the final output of our algorithm. Eq. (2) is a recursive optimization problem, since E^K is conditioned by C_{K-1}, which is the correspondences found in the previous layer. If we want to know the optimal C_K, we need to compute C_{K-1} by optimizing E^{K-1} first. The deepest recursion minimizes E^1, which computes correspondences from the first layer without any guidance, as described by the equality constraint.

Next we derive the specific form of the objective function $E^L(C_L, \Omega_L; P_L, Q_L, C_{L-1})$. Since the seeds have fixed corresponding relationship, each x_i^L can be modeled by a Gaussian distribution centered at y_i^L under a transformation $\boldsymbol{T}_L$. That is,

$$p(x_i^L) = \frac{1}{\sqrt{2\pi}\sigma_L} e^{-\frac{1}{2\sigma_L^2}\|x_i^L - \hat{y}_i^L\|^2}, 1 \le i \le s_L, \tag{3}$$

in which $\hat{y}_i^L = y_i^L + \boldsymbol{T}_L(y_i^L)$ is the new position of y_i^L and $\boldsymbol{T}_L(y_i^L)$ is the spatial displacement vector. While for the wild feature points, the corresponding relationship is unknown. Each x_i^L is then modeled by a Gaussian Mixture Model which treats all the wild feature points in Q_L as centers given a certain transformation $\boldsymbol{T}_L$. So we have:

$$p(x_i^L) = \sum_{j=s_L+1}^{M_L} \frac{\hat{w}_{ij}}{\sqrt{2\pi}\sigma_L} e^{-\frac{1}{2\sigma_L^2}\|x_i^L - \hat{y}_j^L\|^2}, s_L + 1 \le i \le N_L. \tag{4}$$

$\hat{w}_{ij}$ is the weight for each Gaussian component. If no other information is available, $\hat{w}_{ij}$ can be a constant for different j. However, if two feature points are more likely to form a correspondence according to some prior knowledge, the corresponding weight should be assigned a larger value. In this paper, we compute $\hat{w}_{ij}$ from feature similarity:

$$\hat{w}_{ij} = \frac{e^{-\alpha\|f_i^L - g_j^L)\|^2}}{\sum_{j=s_L+1}^{M_L} e^{-\alpha\|f_i^L - g_j^L)\|^2}}, i > s_L, j > s_L. \tag{5}$$

Two feature points tend to haver larger weight if they have similar feature descriptors. In this way, the feature similarity information is incorporated. The parameter σ_L in both Eq. (3) and Eq. (4) is the variance of the Gaussian function. We use asymmetrical Gaussian model with the same σ_L in this paper.

Eq. (3) and Eq. (4) can be combined into a more general formulation, which is:

$$p(x_i^L) = \sum_{j=1}^{M_L} w_{ij} p(x_i^L \left| y_j^L \right.). \tag{6}$$

Here $p(x_i^L \left| y_j^L \right.) = \frac{1}{\sqrt{2\pi}\sigma_L} e^{-\frac{1}{2\sigma_L^2}\|x_i^L - \hat{y}_j^L\|^2}$ is the j^{th} GMM component. x_i^L can be either a seed or a wild feature point in the L^{th} layer. The covariance σ_L is initialized by

$$\sigma_L^2 = \frac{1}{2N_L M_L} \sum_{i=1}^{N_L} \sum_{j=1}^{M_L} \|x_i^L - y_j^L\|^2. \tag{7}$$

In Eq. (6), the weight coefficient w_{ij} is computed from:

$$w_{ij} = \begin{cases} \frac{e^{-\alpha\|f_i^L - g_j^L)\|^2}}{\sum_{j=s_L+1}^{M_L} e^{-\alpha\|f_i^L - g_j^L)\|^2}}, & if \quad i > s_L, j > s_L \\ 1, & if \quad i = j \le s_L \\ 0, & otherwise \end{cases}. \tag{8}$$

If x_i^L is a seed feature point, w_{ij} will be 1 for its matching feature points and 0 for all the other feature points. If x_i^L is a wild feature point, w_{ij} is a continuous value computed from Eq. (5). Since the weights of such a Gassian Mixutre Model are mixture of binary values and continuous values, it is called the Hybrid Gaussian Mixture Model.

Since not all the extracted feature could find its correspondence, another term should be added to Eq. (6) to account for outliers. Suppose such unmatchable feature points are randomly distributed in the image plane, a uniform distribution can be used to model them:

$$p(x_i^L) = \theta \frac{1}{N_L} + (1 - \theta) \sum_{j=1}^{M_L} \omega_{ij} p(x_i^L \left| y_j^L \right.). \tag{9}$$

θ is a harmonic parameters, which takes value 1 if x_i^L is a seed and a positive constant less than 1 (0.7 in this paper) otherwise.

Similar to [31], we use a non-rigid transformation $\boldsymbol{T}_L$, which is more flexible. However, as pointed by [30], an uncontrolled non-rigid transformation may align individual points exactly, but break the structure (or shape) of the whole point set. Therefore, a global regularization term which requires the non-rigid transformation to be smooth and coherent is needed. That is to say, such a transformation will map nearby points to close positions so that the topological structure is similar to that before. The smoothness prior on the transformation is defined as:

$$p(\boldsymbol{T}) = e^{-\frac{\lambda}{2}\|\boldsymbol{T}_L\|_{\mathcal{H}}^2}, \tag{10}$$

in which $\|\cdot\|_{\mathcal{H}}^2$ is the norm of function $\boldsymbol{T}_L$ in the Reproducing Kernel Hilbert Space (RKHS). Smoother transformation will have smaller norm. According to the Motion Coherence Theory (MCT) [72], such a transformation has the form of Gaussian Radial Basis Function (GRBF). The displacement for any position z in the image plane can be computed from the following equation:

$$\boldsymbol{T}_L(z) = \sum_{j=1}^{M_L} \varphi_j G(z, y_j^L). \tag{11}$$

where y_j^L is the j^{th} model point in this layer, φ_j is the GRBF coefficients. $G(z, y_j^L)$ is a kernel computed from:

$$G(z, y_j^L) = e^{-\frac{1}{2\beta}\|z - y_j^L\|^2}. \tag{12}$$

In this paper, we change the position of all the model points by the transformation function, so z is one of the model points rather than a random position. The compact matrix form of Eq. (11) can be expressed as:

$$\boldsymbol{T}_L(\boldsymbol{Y^L}) = \boldsymbol{G\Phi}, \tag{13}$$

where $\boldsymbol{Y^L}$ is the stacked position of all the model points in that layer. $\boldsymbol{G}$ is a constant $M_L \times M_L$ kernel matrix which can be computed from the model points, and $\boldsymbol{\Phi}$ is a unknown $M_L \times 2$ coefficient matrix.

Given Eq. (9) and Eq. (10), the Maximum a Posteriori (MAP) problem under the i.i.d. (independent identically distributed) constraint can be easily derived and it is equivalent to minimizing:

$$E^L(C_L, \Omega_L; P_L, Q_L, C_{L-1}) = -\sum_{i=1}^{N_L} \log(p(x_i^L)) + \frac{\lambda}{2}\|\boldsymbol{T}_L\|_{\mathcal{H}}^2, \tag{14}$$

with $\Omega_L = \{\sigma_L, \boldsymbol{T}_L\}$ as parameters.

Directly optimizing Eq. (14) is difficult. However, it can be solved using the Expectation Maximization (EM) algorithm in an iterative fashion. The EM algorithm alternates between correspondence computing and parameter updating. The E-step estimates the correspondences C_L while fixing the current parameters Ω_L. The matching probability between the i^{th} data point and the j^{th} model point is computed from:

$$p(y_j^L|x_i^L, \sigma_L, \boldsymbol{T}_L) = \frac{\frac{(1-\theta)w_{ij}}{2\pi\sigma^2} e^{-\|x_i^L - (y_j^L + \boldsymbol{T}_L(y_j^L))\|^2/2\sigma_L^2}}{(1-\theta)\sum_{k=1}^{M_L} w_{ik} p(x_i^L|y_k^L) + \theta/N_L}, \tag{15}$$

where σ_L and $\boldsymbol{T}_L$ are the parameters with their current estimation in the L^{th} layer. The matrix $\boldsymbol{P}$ formed by all these pairwise matching probabilities is a fuzzy representation of the correspondences.

The M-step then updates the parameters $\Omega_L = \{\sigma_L, \boldsymbol{T}_L\}$ according to the current correspondences C_L by minimizing:

$$\begin{aligned}\arg\min_{\Omega_L} \log\sigma_L^2 \sum_{i=1}^{N_L}\sum_{j=1}^{M_L} p(y_j^L|x_i^L, \Omega_L) \\ + \frac{1}{2\sigma_L^2}\sum_{i=1}^{N_L}\sum_{j=1}^{M_L} p(y_j^L|x_i^L, \Omega_L)\left\|x_i^L - \left(y_j^L + \boldsymbol{T}_L(y_j^L)\right)\right\|^2 \\ + \frac{\lambda}{2} tr(\boldsymbol{\Phi}^T \boldsymbol{G\Phi}).\end{aligned} \tag{16}$$

Formula (16) is convex with respect to the variables, so the optimal value is the solution which makes the partial derivative zero. The new σ_L can be computed by:

$$\sigma_L^2 = \frac{\sum_{i=1}^{N_L}\sum_{j=1}^{M_L} p(y_j^L|x_i^L, \Omega_L)\left\|x_i^L - (y_j^L + \boldsymbol{T}_L(y_j^L))\right\|^2}{2\sum_{i=1}^{N_L}\sum_{j=1}^{M_L} p(y_j^L|x_i^L, \Omega_L)}. \tag{17}$$

The new $\boldsymbol{\Phi}$ is the solution of

$$(diag(\boldsymbol{P1})\boldsymbol{G} + \lambda\sigma^2)\boldsymbol{\Phi} = \boldsymbol{PX^L} - diag(\boldsymbol{P1})\boldsymbol{Y^L}, \tag{18}$$

where $\boldsymbol{X^L} \in \mathbb{R}^{N_L \times 2}$, $\boldsymbol{Y^L} \in \mathbb{R}^{M_L \times 2}$ are the stacked coordinates of the data and model points in the L^{th} layer. $\mathbf{1}$ is a column vector with all elements equal 1.

Algorithm 1 The Proposed Matching Algorithm

Input: SIFT keypoint sets P and Q.
Output: Correspondences $\boldsymbol{C}$.

Part I: Dividing layers
Generate B LSH functions and map 128D SIFT features to B-bit hash codes
Compute the average hamming distance matrix
Sort candidate feature point pairs in a distance ascending order
Divide them into K layers: $\{P_L\}_{L=1}^K$ and $\{Q_L\}_{L=1}^K$
Part II: Matching with HGMM
Initialize the seeds as $\emptyset$
for $L = 1; L <= K; L++$ **do**
    Build HGMM from (6), (8), (9) with the seeds, unmatched points in the previous layers and points in the current layer
    repeat
        Solving the EM algorithm from (15), (17) and (18)
        Find correspondences and filter with ρ
        Increase ρ
    until $\rho > 0.5$ and the number of correct matches does not change for 3 times
    Update the seeds
    if the increase of seeds is less than 20 **then**
        Break
    end if
end for
Return the seeds as correspondences $\boldsymbol{C}$

After the EM algorithm converges, each data point will take the model point who has the largest matching probability as its correspondence. All the correspondences found in the current layer will be inherited as seeds in the next layer. In order to make sure that the seeds are absolutely correct, we adopt a match-and-filter strategy in each layer. That is, the retained matching feature points are re-matched and filtered by a set of increasing probability threshold ρ, until the current ρ is greater than 0.5 and the number of retained correspondences does not change for three successive filtering. Although we have to match the points multiple times, it is still efficient because most of them are carried out based on the optimal solution of the last step. The final correspondences are denoted as C_L and they are passed to the next layer. We also found that we only get a small increase of correct correspondences in higher layers, but have to spend more time efforts. The trick is not to deal with higher layers if the increase ratio of the total number of correspondences found in two consecutive layers falls below a threshold (20 in this paper). The complete method is summarized in Algorithm 1.

IV. Experiment Results

A. Dataset Description

The proposed feature point matching method is quantitatively tested on three public datasets. The following is a brief introduction of them:

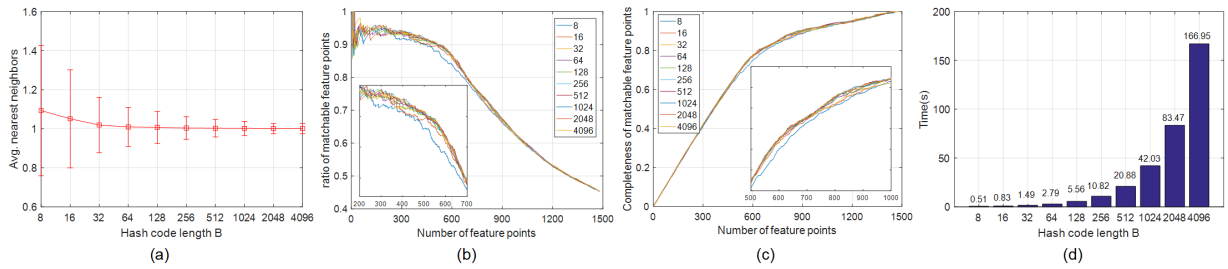

Fig. 3. Comparison with different hash code length B. (a) The mean and standard deviation of the number of nearest neighbors for all the feature points on an image. (b) The ratio of matchable feature points. (c) The completeness of matchable feature points. (d) The time needed for dividing layers.

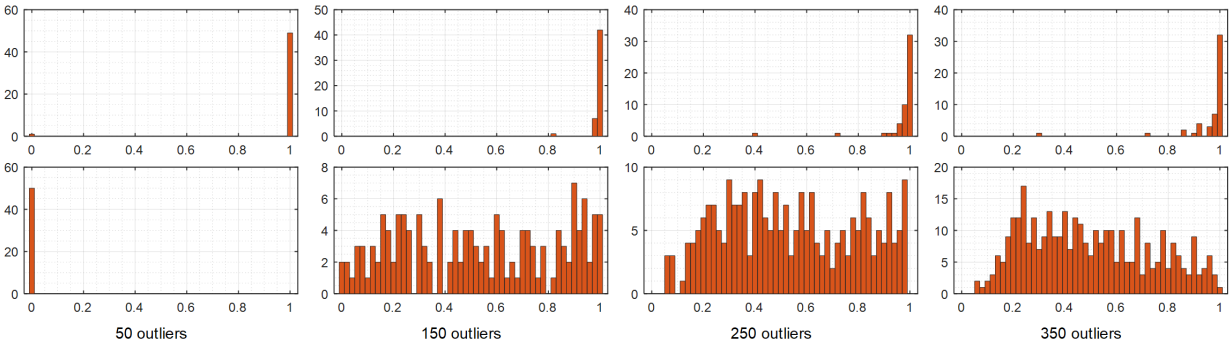

Fig. 4. The distribution of the matching probability after the EM algorithm converges. From left to right: 50 correct correspondences are manually selected and different amount of outliers are added to both sets. From top to bottom are the distribution for correct and false correspondences, respectively.

- VGG dataset [28]. This dataset contains 8 groups of challenging situations such as different viewpoints, rotation, illumination or blurring. Each group contains 6 images and the first image is matched to the other five images(in total 40 image pairs). The homography matrix H between every image pair is provided as ground truth.
- Lebeda dataset [73]. This dataset contains image pairs with large viewpoint differences or very wide baseline. It has two parts: Lebeda-A and Lebeda-B. Lebeda-A contains 16 image pairs with non-planar scenes. The fundamental matrix F between two views is provided as ground truth. Lebeda-B contains 16 image pairs with planar scenes. The homography matrix H between two views is provided as ground truth.
- Panorama dataset [74]. This dataset contains 10 scene groups in total and we use 4 medium-sized subsets: carmel (18 images and 47 pairs), diamondhead (23 images and 59 pairs), fishbowl (13 images and 35 pairs) and halfdome (14 images and 51 pairs). All the image pairs are provided with the homography matrix H as ground truth.
- Non-rigid dataset. We use four image pairs "tiger", "bee", "cushion" and "tshirt" to test the matching result on non-rigid images. The first two image pairs are collected from on-line videos by ourselves. The last two pairs are from [75] and [76]. Since no ground truth are provided, we identify correct correspondences manually.

For quantitative evaluation, we use precision, recall and F-score. In our experiment, a pair of matching points (x, x') will be thought as correct if its homography residual

$$res_H(x, x') = \|x' - Hx\|^2 \tag{19}$$

or its epipolar residual

$$res_F(x, x') = \frac{(x'Fx)^2}{\|Fx\|^2 + \|x'F\|^2} \tag{20}$$

is smaller than 2.0 pixels. Precision is the ratio of true positives in the final matching set. Recall reflects how many correct matches in the input set are contained in the final result. It can be computed from:

$$recall = \frac{TP_O}{TP_I}, \tag{21}$$

where TP_O and TP_I are the number of true positives in the output and input, respectively. However, since the matching relationship between our input point sets is unknown in advance, computing TP_I is not as straightforward as [13] and [15]. Our strategy is to build a large putative set by matching each point to its 2 nearest neighbors in the feature space, and then use the ground-truth to identify true positives. We use the top two nearest neighbors in case of missing possible true positives. After getting recall from Eq. (21), we can compute F-score from:

$$F_{score} = \frac{2 * precision * recall}{precision + recall + \epsilon}, \tag{22}$$

where ϵ is a very small positive value for numerical stability. F-score will be used as an overall evaluation of both precision and recall.

B. Parameters and Settings

To show the impact of different binary code length B, we test on a pair of image from the VGG matching dataset. The extracted feature points are classified into two parts: matchable and unmatchable. If there is at least one feature point on I_2 who is less than 2 pixels from Hx_i, then x_i will be thought as matchable. Otherwise x_i is thought as unmatchable. Note that although this is a rough estimation for matchability, it's enough to reveal the differences with different B. Fig. 3 (a) shows the mean and standard deviation of the number of nearest neighbors for feature points on I_1. Smaller B will lead to more frequent collision. As the candidate matching set becomes larger, the matching results may be affected by the outliers. This is further illustrated in Fig. 3 (b), which shows the ratio of matchable feature points when more and more feature points are added to the multi-layer matching framework. Fig. 3 (c) shows the completeness of matchable feature points as more and more feature points are involved. More matchable feature points will be included in the earlier stage when B is larger. However, longer codes require heavier computational cost, which is shown in Fig. 3 (d). Here computing the pairwise hamming distance takes the majority of the running time. It is implemented with the matlab function *pdist2* and the "hamming" flag. Using C/C++ will reduce the magnitude of time but the trend doesn't change. As a trade-off, we set $B = 256$ throughout this paper.

Ideally, the matching probability should be close to 1 for correct correspondences and close to 0 for false ones. But this may be different when there is noise. Fig. 4 shows how the distribution of matching probability changes when the data contains different amounts of outliers. We first setup two inlier sets by manually selecting 50 correct correspondences. Next, different amounts of unmatchable feature points are randomly added to both inlier sets and the matching probability matrix is computed via EM optimization. Note that in this process there is no prior knowledge on correspondences and which feature point is inlier or outlier is unknown, neither. As we can see from Fig. 4, when outliers are as many as inliers, the matching probability still exhibits good property. When we add three times of outliers, the distribution of correct correspondences moves towards 0 a little bit and the distribution of false correspondences covers the entire interval. This is more obvious when more outliers are added, which makes it difficult to distinguish correct and false correspondences. To this end, we should avoid setting the segment size k too large since the ratio of outliers in the first few layers will increase rapidly. However, k should not be very small. This is because a small layer can not capture enough structure information to build robust matches. In this paper, we empirically set $k = 300$.

Fig. 4 also tells us that using a single probability threshold will either exclude correct or include false correspondences. However, we find that this can be solved by iteratively matching and filtering with a set of increasing thresholds ρ. A small threshold can filter a part of false correspondences while saving almost all correct ones. Since the remaining data contains fewer outliers, the two distributions will separate from each other, which enables us to use a higher threshold to further remove bad results. We empirically sample ρ from $[0.1, 0.5]$ by step 0.1, from $[0.55, 0.9]$ by step 0.05, from $[0.91, 0.99]$ by step 0.01 and from $[0.991, 0.999]$ by step 0.001. Such a non-uniform sampling enables us to quickly remove outliers in a coarse-to-fine manner. In practice, there is no need to traverse all values in ρ. It stops if the current ρ is greater than 0.5 and the number of retained correspondences does not change for three successive filtering. Even though we have to match the points multiple times, it is still efficient because: (1) The data becomes cleaner and the scale of the problem is reduced as outliers are gradually removed. (2) Most of them are carried out based on the optimal solution of the last step. No mismatching removal methods are applied as a post-processing step since the results are satisfactory enough and it's more impartial for the comparison. For the other parameters we set $\theta = 0.7$, $\lambda = 5$, $\beta = 3.5$ and $\alpha = 0.1$ according to the reported work [31].

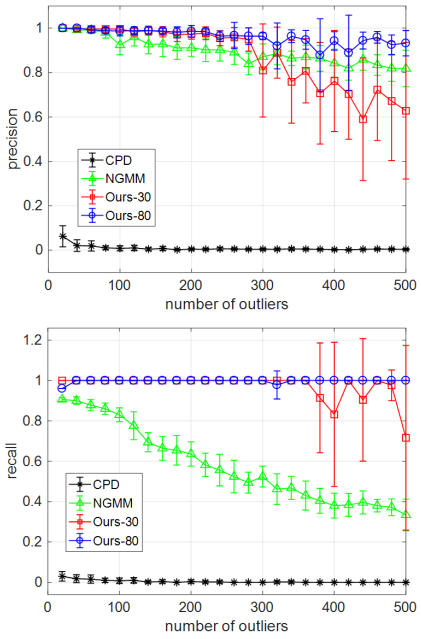

Fig. 5. The error bar curves of precision (left) and recall (right) for different methods when adding outliers to the 50 ground truth matches.

The proposed algorithm is implemented using MATLAB and C-Mex files. All the methods are tested with their default parameters. The experimental platform is a machine with one Intel Xeon E5 2.1GHz CPU, 32GB memory and Ubuntu 16.04 operating system.

C. Evaluation on the VGG Dataset

For the VGG dataset, we first use the image pair (graf1,graf4) to test the robustness of the proposed method. This image pair is challenging due to large distortion. We manually selected 50 feature points on each image so that they form 50 correct ground truth matching pairs. Different amount of outliers (from 20 to 500) are randomly added to both images. The correspondences are computed by CPD [30], NGMM [31] and the proposed method. Two metrics are used to evaluate the performance: precision and recall. Precision is

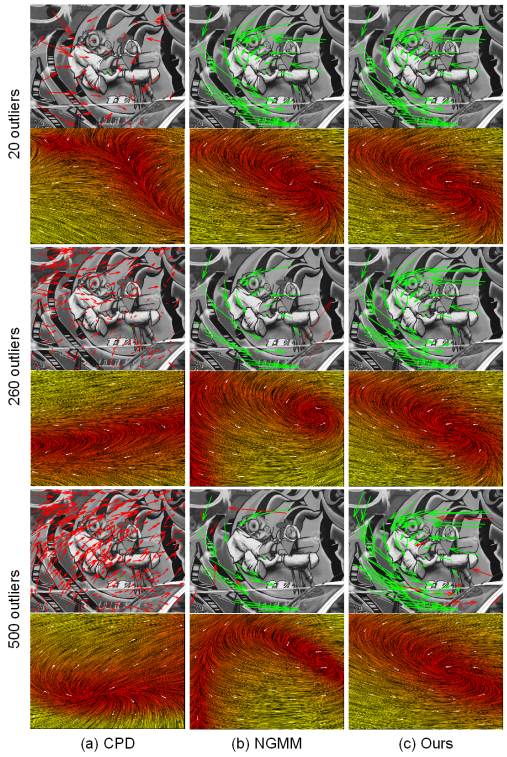

Fig. 6. Visualization of the sparse correspondences (top) and the interpolated dense vector field (bottom) when different amount of outliers are added. Green and red quivers indicate correct and wrong matches, respectively.

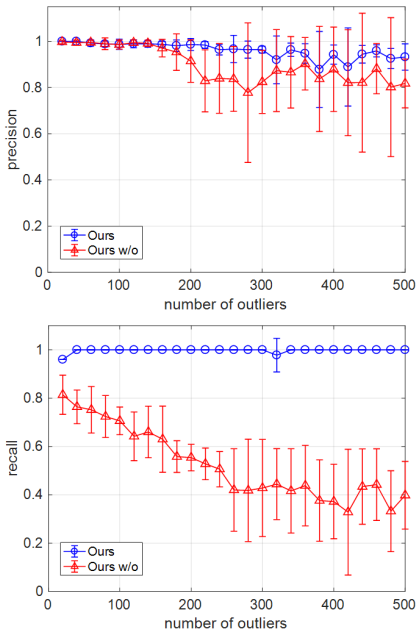

Fig. 7. An ablation study. The proposed method with and without the hard constraint between different layers are in blue and red, respectively. The error bar curves of precision (left) and recall (right) are plotted with the same settings in Fig. 5.

TABLE I
THE QUANTILE STATISTICS FOR THE NUMBER OF LAYERS AND THE AVERAGE NUMBER OF FILTERING WHEN MATCHING EACH PAIR OF IMAGE.

Quantile	10%	25%	50%	75%	90%	100%
#Total Layers	5	5	7	8	11	11
#Added Layers	2	3	5	8	11	11
#Filtering	5.1	5.3	5.7	7	15	26

the ratio of correct matches found by the method and recall is the ratio of ground truth matches that are found. When adding random outliers, we repeat 10 times and compute the mean and variance. The results are shown in Fig. 5. It shows that the CPD algorithm is sensitive to geometric distortion and outliers. The NGMM algorithm performs better. It has a precision over 80% even though the outliers are 10 times more than the inliers. However, its recall drops a lot. For the proposed method we use two different settings: layer size 30 (denoted as Ours-30) and layer size 80 (denoted as Ours-80). Here we do not use $k = 300$ as suggested before because in this experiment we have only 50 inliers. If k is large, the outlier ratio in the first layer would be high and the success rate of finding correct seed correspondences will drop. Both of them work well when the number of outliers is less than 6 times of the inliers. But the precision of Ours-30 drops significantly and its recall fluctuated as more outliers are added. In contrast, Ours-80 achieves the best performance. This is because a small layer does not contain enough structure information to resist outliers. Fig. 6 shows some visualization results. This example shows that the proposed method is more robust to outliers.

With the same settings of Fig. 5, an ablation study of the proposed method is presented in Fig. 7. More specifically, the matching results with and without hard constraint between different layers are compared. When hard constraint is abandoned, there are no binary weights in our GMM and it is equivalent to a multi-layer version of NGMM. As we can see from the figure, without hard constraint the matching result, especially recall, is vulnerable to outliers. On the contrary, by imposing hard constraint between different layers, our algorithm is still robust even if there are 10 times of outliers.

The evaluation results on the whole VGG dataset compared with SIFT [27], RANSAC [24], CPD [30], GMMReg [48], OSM [58], PGM [62], PRGLS [21], NGMM [31] and GMS [43] are presented in Fig. 8. We plot the quantile statistics of precision, recall and F-score. The first four image groups, *i.e. bark*, *boat*, *graf* and *wall* contain relative geometry differences such as large viewpoint changing or rotation, while the last four groups, *i.e. ubc*, *leuven*, *trees* and *bikes* mainly contain image quality but small spatial differences. CPD can find some correct matches on the last four groups, but its precision is low. It almost fails on all the first four groups. GMMReg, however, is the worst on most cases, for the reason that it does not consider feature information. We do not compare with them any more in the following experiments for their bad performance. The GMS algorithm can find the most correspondences for most image pairs. This is because it uses up to 10^4 ORB features instead of $10^2 \sim 10^3$ SIFT features for each view. However, it is sensitive to geometric changing (see the first four image groups). Similar situation happens to OSM, PGM and PRGLS as well. PRGLS is good at finding high precision correspondences when there is no

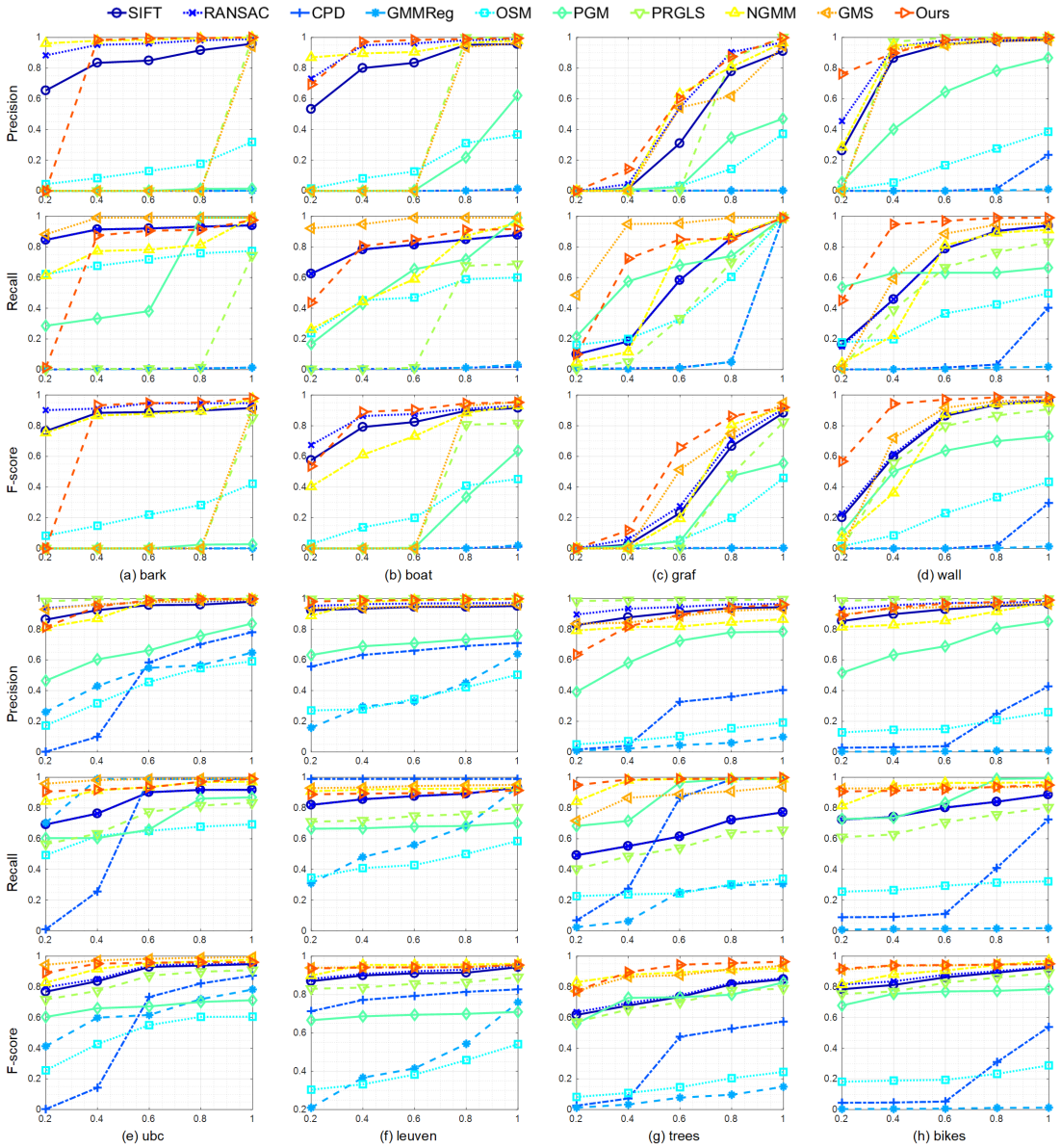

Fig. 8. The results on eight image groups in the VGG matching dataset [28]. For each image group, the quantile statistics (x-axis) for precision, the number of correct matches and the F-score (y-axis) are plotted. Each position (c_x, c_y) on the curve means that there are c_x (percentage) image pairs whose evaluation results are smaller than c_y.

big geometric difference, but easily fails on the opposite case. For most image pairs, OSM and PGM are not as effective as the widely used SIFT algorithm. This is because unmatchable feature points offer misleading structure information when they compute the subspace or the graph. RANSAC improves SIFT matching result effectively. Although it is not the best one, it is efficient and stable on different data. As an overall evaluation, the proposed method has better performance than the others, especially when the geometric difference is large.

Then we analyze the efficiency of the proposed method. Since we have set two stopping conditions on adding new layers and filtering with a higher ρ, it is worth seeing how many layers are actually added and how many times the filtering is carried out. Table I is a quantile statistics for these things. The second and third rows are the total number of layers and the number of added layers. It shows that for about 75% of the image pairs we don't have to process all the feature points. The last row is the average number of filtering for all the layers when matching an image pair. For about 75% of the image pairs, we could get satisfactory results after filtering 7 times. Here ρ reaches 0.6. For the other 20% challenging cases, we need much more filtering and ρ grows as high as 0.99. Fig. 9 shows the comparison with NGMM in the matching time and the number of iterations. As can be seen, the proposed method runs faster and needs fewer iterations than NGMM. There are two reasons for this acceleration. (1) After dividing layers, the original large scale optimization problem is divided into several small scale problems. As pointed by [31], the computational complexity of the matching procedure is $O(M^3) + O(MN)$, in which M and N are

TABLE II
THE MATCHING RESULTS EVALUATED BY THE GIVEN FUNDAMENTAL MATRIX FOR DIFFERENT METHODS ON LEBEDA-A. THE PRECISION, RECALL AND F-SCORE ARE LISTED. IF NO MATCHES ARE FOUND, BOTH PRECISION AND RECALL ARE TREATED AS ZERO. THE BEST F-SCORE IS IN BOLD.

Data	Precision / Recall / F-score					
	OSM [58]	PGM [62]	RANSAC [24]	GMS [43]	NGMM [31]	Ours
booksh	12.7% / 0.61 / 0.211	1.3% / 0.42 / 0.025	84.6% / 0.63 / 0.724	0% / 0 / 0	94.0% / 0.48 / 0.636	95.2% / 0.72 / **0.823**
box	1.0% / 0.08 / 0.018	0.5% / 0.17 / 0.010	27.9% / 0.22 / 0.247	15.3% / 0.005 / 0.010	40.1% / 0.27 / 0.329	42.7% / 0.66 / **0.519**
castle	10.7% / 0.44 / 0.173	24.3% / 0.63 / 0.351	69.5% / 0.35 / 0.472	62.5% / 0.46 / 0.535	68.2% / 0.29 / 0.414	70.4% / 0.67 / **0.691**
corr	51.0% / 0.75 / 0.609	63.0% / 0.65 / 0.641	91.1% / 0.85 / 0.880	81.6% / 0.81 / 0.817	99.3% / 0.90 / **0.946**	99.3% / 0.89 / 0.941
head	4.5% / 0.10 / 0.063	60.9% / 0.51 / 0.558	77.9% / 0.52 / 0.624	73.4% / 0.64 / **0.689**	90.3% / 0.40 / 0.554	87.6% / 0.34 / 0.496
kampa	5.9% / 0.35 / 0.101	25.3% / 0.72 / 0.376	66.6% / 0.46 / 0.549	69.4% / 0.36 / 0.481	79.4% / 0.43 / 0.558	91.4% / 0.51 / **0.655**
Kyoto	0.7% / 0.11 / 0.014	35.9% / 0.55 / 0.437	41.1% / 0.28 / 0.334	70.1% / 0.59 / **0.641**	65.1% / 0.16 / 0.260	58.6% / 0.33 / 0.428
leafs	0.5% / 0.17 / 0.010	9.5% / 0.60 / 0.164	22.3% / 0.21 / **0.220**	0% / 0 / 0	29.0% / 0.06 / 0.102	61.1% / 0.07 / 0.135
plant	6.4% / 0.37 / 0.110	1.4% / 0.35 / 0.028	67.7% / 0.39 / 0.500	63.5% / 0.33 / 0.441	74.6% / 0.37 / 0.495	44.7% / 0.75 / **0.561**
rotunda	1.4% / 0.26 / 0.026	6.6% / 0.79 / 0.122	31.4% / 0.20 / 0.246	0% / 0 / 0	23.5% / 0.05 / 0.089	45.9% / 0.20 / **0.283**
shout	3.5% / 0.34 / 0.064	0.4% / 0.27 / 0.008	48.1% / 0.31 / 0.382	49.5% / 0.16 / 0.244	81.8% / 0.12 / 0.222	86.6% / 0.48 / **0.619**
valbonne	2.8% / 0.32 / 0.052	0.9% / 0.60 / 0.018	37.2% / 0.25 / 0.299	48.8% / 0.15 / 0.237	63.6% / 0.21 / 0.325	53.6% / 0.29 / **0.379**
wall	1.1% / 0.18 / 0.021	36.2% / 0.75 / 0.491	70.2% / 0.39 / 0.507	66.5% / 0.61 / **0.637**	91.1% / 0.17 / 0.293	86.2% / 0.36 / 0.511
zoom	4.3% / 0.37 / 0.077	15.2% / 0.64 / 0.245	66.0% / 0.43 / 0.527	9.3% / 0.01 / 0.018	83.6% / 0.32 / 0.464	93.9% / 0.43 / **0.593**

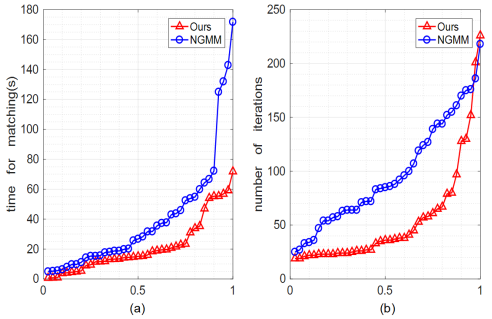

Fig. 9. Efficiency comparison with NGMM. For 40 image pairs we plot the quantile statistics (x-axis) of: (a) the time for matching and (b) the number of EM iterations (y-axis). Each position (c_x, c_y) on the curve means that there are c_x (percentage) image pairs whose evaluation results are smaller than c_y.

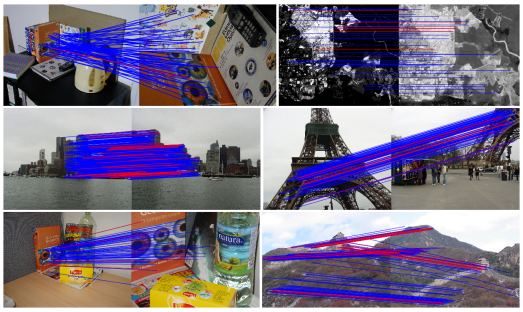

Fig. 10. The matching results of the proposed method on some typical image pairs. Blue and red lines indicate correct and wrong matches, respectively.

the size of the model and data set respectively. The total cost of running several small scale problems is still smaller than running on a single large scale problem if M and N decrease. (2) With the guidance provided by the previous layer in HGMM, fewer iterations are needed before the EM algorithm reaches its optimal solution. This also contributes to a faster convergence in each layer.

D. Evaluation on the Lebeda Dataset

In this part, we test our algorithm on the Lebeda Dataset. This dataset is quite challenging because image pairs are either very wide-baseline (Lebeda-A) or with extremely small scene overlap (Lebeda-B). For this reason, some of the tested methods may fail in the matching task. In our experiment, if four or more methods find less than 8 correct correspondences we will think this image pair as a failure case. Table II shows the evaluation results on Lebeda-A between OSM [58], PGM [62], RANSAC [24], GMS [48], NGMM [31] and the proposed method. PRGLS [21] failed on too many cases so it's results are not shown. The two failure cases *graff* and *wash* are not listed in the table. For each method, we measure the precision, recall and F-score. GMS also returns the most correct matches but its recall is not the highest because its input contains more true positives as well. It achieves the best overall performance on *head*, *Kyoto* and *wall*, but fails on *booksh*, *leafs* and *rotunda*. OSM and PGM manages to find some correct matches, but their precision is poor. Our method achieves the highest overall performance except for *corr*, *head*, *Kyoto*, *leafs* and *wall*. Similarly, the results on Lebeda-B are reported in Table III. Five failure cases *CapitalRegion*, *ExtremeZoom*, *LePoint1*, *LePoint2* and *LePoint3* are not listed. RANSAC doesn't fail on any case in both Table II and Table III. Although it does not achieve the best performance, the results show its stability and effectiveness on a wide range of scenarios. This verifies why it still plays an important role nowadays in real application such as Structure from Motion (SfM). Our method has the highest score for most cases.

Fig. 10 shows the matching results for some typical image pairs in both datasets. We can see that this dataset is very challenging due to large viewpoint differences and small scene overlap. Although our method achieves the best overall performance, there is still large room to improve both precision and the number of correct matches.

E. Evaluation on the Panorama dataset

The panorama dataset contains ten different scenes. It is originally used for image stitching, which combines a set of overlapping images into a larger image with a wider field of view [77]–[79]. Here we use four medium-sized subsets: carmel, diamondhead, fishbowl and halfdome. Fig. 11 shows

TABLE III
THE MATCHING RESULTS EVALUATED BY THE GIVEN HOMOGRAPHY MATRIX FOR DIFFERENT METHODS ON LEBEDA-B. THE PRECISION, RECALL AND F-SCORE ARE LISTED. IF NO MATCHES ARE FOUND, BOTH PRECISION AND RECALL ARE TREATED AS ZERO. THE BEST F-SCORE IS IN BOLD.

Data	Precision / Recall / F-score					
	OSM [58]	PGM [62]	RANSAC [24]	GMS [43]	NGMM [31]	Ours
adam	25.3% / 0.74 / 0.377	18.0% / 0.39 / 0.247	65.2% / 0.88 / 0.749	52.1% / 0.67 / 0.589	62.7% / 0.81 / 0.709	73.6% / 0.89 / **0.806**
boat	1.8% / 0.47 / 0.034	4.8% / 0.57 / 0.088	23.7% / 0.95 / 0.381	21.5% / 0.42 / 0.287	27.2% / 0.58 / 0.372	25.1% / 0.91 / **0.394**
Boston	9.5% / 0.49 / 0.160	53.2% / 0.65 / 0.587	78.2% / 0.84 / 0.811	66.5% / 0.80 / 0.729	84.1% / 0.38 / 0.525	94.4% / 0.96 / **0.953**
BostonLib	1.3% / 0.24 / 0.026	0% / 0 / 0	42.1% / 0.87 / 0.568	66.6% / 0.74 / 0.704	60% / 0.50 / 0.548	70.2% / 0.82 / **0.757**
BruggeSquare	0.8% / 0.24 / 0.016	7.5% / 0.61 / 0.135	9.7% / 0.70 / 0.171	16.4% / 0.53 / **0.251**	10.5% / 0.32 / 0.159	10.9% / 0.85 / 0.193
BruggeTower	4.9% / 0.49 / 0.089	9.7% / 0.54 / 0.165	43.8% / 0.84 / 0.576	54.7% / 0.67 / 0.604	46.3% / 0.71 / 0.563	50.7% / 0.92 / **0.654**
Brussels	2.8% / 0.22 / 0.050	31.9% / 0.69 / 0.438	37.6% / 0.84 / 0.521	31.2% / 0.50 / 0.385	40.3% / 0.46 / 0.432	38.9% / 0.95 / **0.552**
city	12.0% / 0.56 / 0.198	19.3% / 0.50 / 0.280	71.1% / 0.55 / 0.621	60.4% / 0.69 / 0.645	62.0% / 1.00 / **0.765**	86.36% / 0.65 / 0.745
Eiffel	1.47% / 0.27 / 0.027	12.3% / 0.59 / 0.204	55.2% / 0.82 / 0.663	15.4% / 0.56 / 0.242	71.9% / 0.50 / 0.594	92.3% / 0.80 / **0.858**
graf	5.6% / 0.33 / 0.096	13.9% / 0.69 / 0.231	39.8% / 0.69 / 0.505	33.4% / 0.59 / 0.428	33.8% / 0.87 / 0.488	42.5% / 0.84 / **0.565**
WhiteBoard	0.6% / 0.25 / 0.013	8.2% / 0.55 / 0.143	46.6% / 0.82 / 0.595	62.9% / 0.61 / 0.623	78.5% / 0.48 / 0.600	68.5% / 0.89 / **0.777**

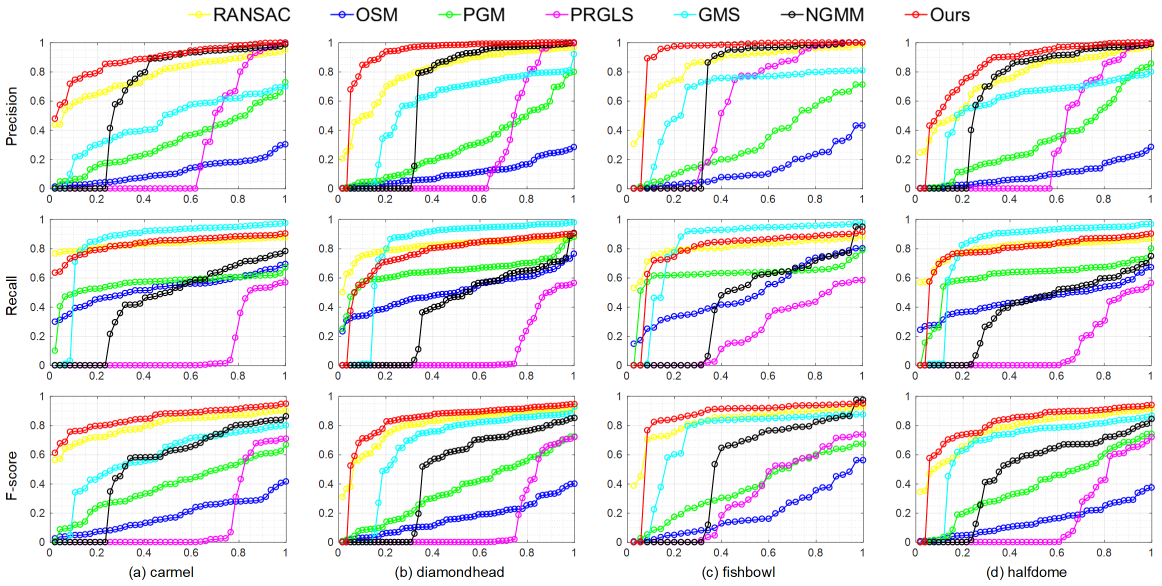

Fig. 11. For all the image pairs in carmel, diamondhead, fishbowl and halfdome, the quantile statistics (x-axis) for precision, recall and F-score (y-axis) are plotted. Each position (c_x, c_y) on the curve means that there are c_x (percentage) image pairs whose evaluation results are smaller than c_y.

the comparison results between RANSAC [24], OSM [58], PGM [62], PRGLS [21], GMS [48], NGMM [31] and the proposed method on these four scenes. We can see that these image pairs are still challenging since none of these methods can achieve 100% precision on all of them. For each scene, PRGLS fails on nearly half of the image pairs since SIFT matching doesn't provide correct initialization as well. OSM and PGM seldom fails, but their precision need improvement. GSM performs better than the above two methods. It finds the most correct matches and is more accurate than OSM and PGM. The precision of RANSAC is greater than OSM, PGM, PRGLS and GMS, but lower than NGMM and the proposed method. It has stable performance across different testing images. Although the proposed method fails on one or two image pairs in the last three scenes, we have higher precision, comparable recall and the best overall performance in general.

We then show the matching and stitching results on some typical image pairs in Fig. 12. In order to generate a panorama image from two views, we estimate the planar homography transformation with the correspondences. Since some wrong correspondences are included in the matching results, we adopt the RANSAC strategy. The better the matching results are, the more accurate the homography transformation will be. As we can see from the results, our method produces good correspondences and aligns pixels in the overlapping region well.

F. *Evaluation on non-rigid images*

We show that our method can deal with images with mild non-rigid deformation. Four image pairs "tiger", "bee", "cushion" and "tshirt" are used in this test. Our method is compared with SIFT and NGMM. Since no ground truth are provided, we identify correct correspondences manually. The precision and number of correct matches are given in Table IV and Table V, respectively. As we can see from the results, our method has similar or higher precision on these images. At the same time, we get more correct matches than the other methods. Fig. 13 shows example matches of the proposed method. Green lines indicate correct matches and

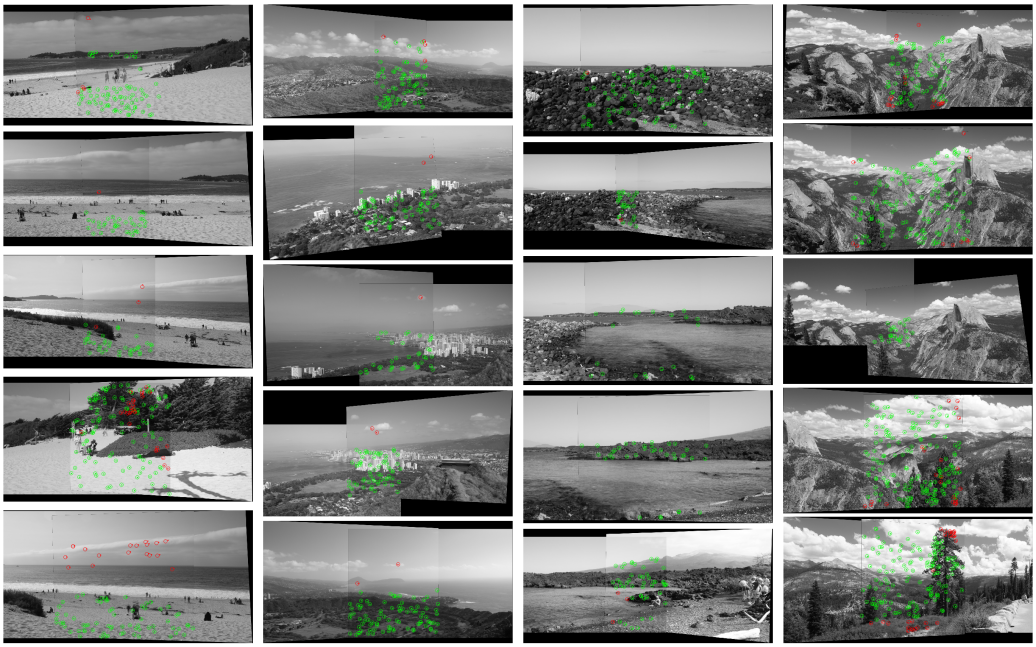

Fig. 12. Matching and stitching results for some typical image pairs in each scene. Green circles and dots indicate correct matches aligned by the homography transformation. Red circles and dots are wrong matches.

TABLE IV
THE MATCHING PRECISION FOR DIFFERENT METHODS ON NON-RIGID IMAGES.

Method	tiger	bee	cushion	tshirt
SIFT	64.19%	63.91%	60.18%	69.49%
NGMM	75.45%	63.00%	57.85%	80.00%
Ours	78.52%	63.96%	62.58%	80.95%

TABLE V
THE NUMBER OF CORRECT MATCHES FOR DIFFERENT METHODS ON NON-RIGID IMAGES.

Method	tiger	bee	cushion	tshirt
SIFT	104	62	65	41
NGMM	83	63	81	40
Ours	128	71	92	51

red lines represent wrong matches. For clarity only a subset of correspondences are plotted.

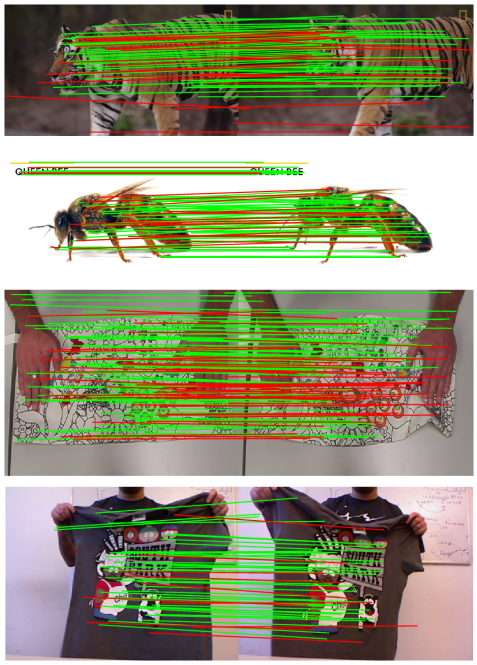

Fig. 13. The matching result of our method on non-rigid images. Green lines indicate correct matches and red lines represent wrong matches. For clarity only a subset of correspondences are plotted.

V. CONCLUSION AND DISCUSSION

In this paper, a novel multi-layer feature matching algorithm is proposed. It divides feature points into different layers according to their matching uncertainties. Correspondences with lower matching uncertainties could be found earlier and then used to guide the search of new correspondences. Such a strategy avoids involving outliers too early and improves the robustness. To achieve this goal, a Hybrid Gaussian Mixture Model, which imposes both intra and inter layer constraints is proposed. Extensive experiments are carried out to test the performance of the proposed method. It has the best overall performance for most cases, especially when images have large viewpoint differences or small scene overlap. Besides, owing to the self-provided guidance, it converges faster than the traditional GMM based methods.

For images with very large geometric or photometric differences, the distances between matching feature points will increase and matching them is a challenging task. In this case,

the seed correspondences found in the first layer may include some outliers, which will further affect the following matching procedure. We found in the experiments that other compared methods failed as well. We have to seek for new technologies to tackle this problem. What's more, it's possible to combine the proposed method with the L_2E estimation, which is more robust to outliers than the EM algorithm used currently. We will plan this as a future work.

REFERENCES

[1] M. Havlena and K. Schindler, "Vocmatch: Efficient multiview correspondence for structure from motion," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 46–60.

[2] K. Sun and W. Tao, "A constrained radial agglomerative clustering algorithm for efficient structure from motion," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 1089–1093, July 2018.

[3] H. Cui, X. Gao, S. Shen, and Z. Hu, "Hsfm: Hybrid structure-from-motion," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 2393–2402.

[4] T. Liu, Y. F. Li, H. Liu, Z. Zhang, and S. Liu, "Risir: Rapid infrared spectral imaging restoration model for industrial material detection in intelligent video systems," *IEEE Transactions on Industrial Informatics*, vol. doi:10.1109/TII.2019.2930463, pp. 1–1, 2019.

[5] L. Magerand and A. D. Bue, "Revisiting projective structure for motion: A robust and efficient incremental solution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2018.

[6] J. Ma, W. Yu, P. Liang, C. Li, and J. Jiang, "Fusiongan: A generative adversarial network for infrared and visible image fusion," *Information Fusion*, vol. 48, pp. 11 – 26, 2019.

[7] B. Yi, X. Shen, H. Liu, Z. Zhang, W. Zhang, S. Liu, and N. Xiong, "Deep matrix factorization with implicit feedback embedding for recommendation system," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 8, pp. 4591–4601, Aug 2019.

[8] G. Lu, L. Nie, S. Sorensen, and C. Kambhamettu, "Large-scale tracking for images with few textures," *IEEE Transactions on Multimedia*, vol. 19, no. 9, pp. 2117–2128, Sep. 2017.

[9] T. Liu, H. Liu, Y. Li, Z. Zhang, and S. Liu, "Efficient blind signal reconstruction with wavelet transforms regularization for educational robot infrared vision sensing," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 1, pp. 384–394, Feb 2019.

[10] X. Qin, J. Shen, X. Mao, X. Li, and Y. Jia, "Structured-patch optimization for dense correspondence," *IEEE Transactions on Multimedia*, vol. 17, no. 3, pp. 295–306, March 2015.

[11] J. Ma, X. Jiang, J. Jiang, and X. Guo, "Robust feature matching using spatial clustering with heavy outliers," *IEEE Transactions on Image Processing*, pp. 1–1, 2019.

[12] T. Liu, H. Liu, Z. Chen, and A. M. Lesgold, "Fast blind instrument function estimation method for industrial infrared spectrometers," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2018.

[13] J. Ma, X. Jiang, J. Jiang, J. Zhao, and X. Guo, "Lmr: Learning a two-class classifier for mismatch removal," *IEEE Transactions on Image Processing*, vol. 28, no. 8, pp. 4045–4059, Aug 2019.

[14] T. Liu, H. Liu, Y. Li, Z. Chen, Z. Zhang, and S. Liu, "Flexible ftir spectral imaging enhancement for industrial robot infrared vision sensing," *IEEE Transactions on Industrial Informatics*, vol. doi:10.1109/TII.2019.2934728, pp. 1–1, 2019.

[15] X. Jiang, J. Jiang, A. Fan, Z. Wang, and J. Ma, "Multiscale locality and rank preservation for robust feature matching of remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6462–6472, Sep. 2019.

[16] J. Yan, J. Wang, H. Zha, X. Yang, and S. Chu, "Consistency-driven alternating optimization for multigraph matching: A unified approach," *IEEE Transactions on Image Processing*, vol. 24, no. 3, pp. 994–1009, March 2015.

[17] J. Chao and E. Steinbach, "Keypoint encoding for improved feature extraction from compressed video at low bitrates," *IEEE Transactions on Multimedia*, vol. 18, no. 1, pp. 25–39, Jan 2016.

[18] J. Yan, M. Cho, H. Zha, X. Yang, and S. M. Chu, "Multi-graph matching via affinity optimization with graduated consistency regularization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 6, pp. 1228–1242, June 2016.

[19] M. Karpushin, G. Valenzise, and F. Dufaux, "Keypoint detection in rgbd images based on an anisotropic scale space," *IEEE Transactions on Multimedia*, vol. 18, no. 9, pp. 1762–1771, Sep. 2016.

[20] J. Yan, C. Li, Y. Li, and G. Cao, "Adaptive discrete hypergraph matching," *IEEE Transactions on Cybernetics*, vol. 48, no. 2, pp. 765–779, Feb 2018.

[21] J. Ma, J. Zhao, and A. L. Yuille, "Non-rigid point set registration by preserving global and local structures," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 53–64, Jan 2016.

[22] J. Ma, J. Zhao, H. Guo, J. Jiang, H. Zhou, and Y. Gao, "Locality preserving matching," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 4492–4498. [Online]. Available: https://doi.org/10.24963/ijcai.2017/627

[23] J. Ma, J. Jiang, H. Zhou, J. Zhao, and X. Guo, "Guided locality preserving feature matching for remote sensing image registration," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–13, 2018.

[24] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981. [Online]. Available: http://doi.acm.org/10.1145/358669.358692

[25] K. Moo Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua, "Learning to find good correspondences," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[26] J. Ma, J. Zhao, J. Jiang, H. Zhou, and X. Guo, "Locality preserving matching," *International Journal of Computer Vision*, Sep 2018. [Online]. Available: https://doi.org/10.1007/s11263-018-1117-z

[27] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[28] K. Mikolajczyk and C. Schmid, "A performance evaluation of localdescriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[29] A. Rana, G. Valenzise, and F. Dufaux, "Learning-based tone mapping operator for efficient image matching," *IEEE Transactions on Multimedia*, vol. 21, no. 1, pp. 256–268, Jan 2019.

[30] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2262–2275, Dec 2010.

[31] W. Tao and K. Sun, "Robust point sets matching by fusing feature and spatial information using nonuniform gaussian mixture models," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3754–3767, Nov 2015.

[32] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding*, vol. 89, no. 2, pp. 114 – 141, 2003, nonrigid Image Registration.

[33] J. Cheng, C. Leng, J. Wu, H. Cui, and H. Lu, "Fast and accurate image matching with cascade hashing for 3d reconstruction," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 1–8.

[34] J. L. Schönberger, H. Hardmeier, T. Sattler, and M. Pollefeys, "Comparative evaluation of hand-crafted and learned local features," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 6959–6968.

[35] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417.

[36] E. Tola, V. Lepetit, and P. Fua, "Daisy: An efficient dense descriptor applied to wide-baseline stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 815–830, May 2010.

[37] D. C. Hauagge and N. Snavely, "Image matching using local symmetry features," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 206–213.

[38] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "Lift: Learned invariant feature transform," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 467–483.

[39] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "Matchnet: Unifying feature and metric learning for patch-based matching," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 3279–3286.

[40] Y. Ono, E. Trulls, P. Fua, and K. M. Yi, "Lf-net: Learning local features from images," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc.,

2018, pp. 6234–6244. [Online]. Available: http://papers.nips.cc/paper/7861-lf-net-learning-local-features-from-images.pdf

[41] Y. T. Hu, Y. Y. Lin, H. Y. Chen, K. J. Hsu, and B. Y. Chen, "Matching images with multiple descriptors: An unsupervised approach for locally adaptive descriptor selection," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5995–6010, Dec 2015.

[42] Y. T. Hu and Y. Y. Lin, "Progressive feature matching with alternate descriptor selection and correspondence enrichment," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 346–354.

[43] J. Bian, W. Y. Lin, Y. Matsushita, S. K. Yeung, T. D. Nguyen, and M. M. Cheng, "Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 2828–2837.

[44] L. Zhou, S. Zhu, T. Shen, J. Wang, T. Fang, and L. Quan, "Progressive large scale-invariant image matching in scale space," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 2381–2390.

[45] Y. Cao, C. Wang, Z. Li, L. Zhang, and L. Zhang, "Spatial-bag-of-features," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 3352–3359.

[46] W. Hartmann, M. Havlena, and K. Schindler, "Predicting matchability," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 9–16.

[47] Y. Tsin and T. Kanade, "A correlation-based approach to robust point set registration," in *Computer Vision - ECCV 2004*, T. Pajdla and J. Matas, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 558–569.

[48] B. Jian and B. C. Vemuri, "Robust point set registration using gaussian mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1633–1645, Aug 2011.

[49] J. Ma, W. Qiu, J. Zhao, Y. Ma, A. L. Yuille, and Z. Tu, "Robust$l_2e$estimation of transformation for non-rigid registration," *IEEE Transactions on Signal Processing*, vol. 63, no. 5, pp. 1115–1129, March 2015.

[50] G. D. Evangelidis, D. Kounades-Bastian, R. Horaud, and E. Z. Psarakis, "A generative model for the joint registration of multiple point sets," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 109–122.

[51] G. D. Evangelidis and R. Horaud, "Joint alignment of multiple point sets with batch and incremental expectation-maximization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1397–1410, June 2018.

[52] J. Ma, J. Zhao, J. Jiang, and H. Zhou, "Non-rigid point set registration with robust transformation estimation under manifold regularization," in *AAAI*, 2017.

[53] G. Wang, Z. Wang, Y. Chen, Q. Zhou, and W. Zhao, "Context-aware gaussian fields for non-rigid point set registration," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 5811–5819.

[54] G. Wang, Q. Zhou, and Y. Chen, "Robust non-rigid point set registration using spatially constrained gaussian fields," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1759–1769, April 2017.

[55] L. Bai, X. Yang, and H. Gao, "Nonrigid point set registration by preserving local connectivity," *IEEE Transactions on Cybernetics*, vol. 48, no. 3, pp. 826–835, March 2018.

[56] N. Ravikumar, A. Gooya, S. Çimen, A. F. Frangi, and Z. A. Taylor, "Group-wise similarity registration of point sets using student's t-mixture model for statistical shape models," *Medical Image Analysis*, vol. 44, pp. 156 – 176, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1361841517301810

[57] T. M. Nguyen and Q. M. J. Wu, "Multiple kernel point set registration," *IEEE Transactions on Medical Imaging*, vol. 35, no. 6, pp. 1381–1394, June 2016.

[58] M. Torki and A. Elgammal, "One-shot multi-set non-rigid feature-spatial matching," in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2010, pp. 3058–3065.

[59] R. Hamid, D. Decoste, and C. J. Lin, "Dense non-rigid point-matching using random projections," in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2013, pp. 2914–2921.

[60] K. Sun, L. Liu, and W. Tao, "Progressive match expansion via coherent subspace constraint," *Information Sciences*, vol. 367-368, pp. 848 – 861, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025516304996

[61] ——, "Image matching via feature fusion and coherent constraint," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 3, pp. 289–293, March 2017.

[62] M. Cho and K. Lee, "Progressive graph matching: Making a move of graphs via probabilistic voting," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2012.

[63] T. Wang, H. Ling, C. Lang, and S. Feng, "Graph matching with adaptive and branching path following," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2017.

[64] B. Jiang, J. Tang, C. Ding, and B. Luo, "Binary constraint preserving graph matching," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 550–557.

[65] F. Zhou and F. D. la Torre, "Factorized graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 9, pp. 1774–1789, Sept 2016.

[66] K. Sun, P. Li, W. Tao, and Y. Tang, "Feature guided biased gaussian mixture model for image matching," *Information Sciences*, vol. 295, pp. 323 – 336, 2015.

[67] Y. Yang, S. H. Ong, and K. W. C. Foong, "A robust global and local mixture distance based non-rigid point set registration," *Pattern Recognition*, vol. 48, no. 1, pp. 156 – 173, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320314002398

[68] K. Yang, A. Pan, Y. Yang, S. Zhang, S. H. Ong, and H. Tang, "Remote sensing image registration using multiple image features," *Remote Sensing*, vol. 9, no. 6, 2017. [Online]. Available: http://www.mdpi.com/2072-4292/9/6/581

[69] S. Zhang, K. Yang, Y. Yang, Y. Luo, and Z. Wei, "Non-rigid point set registration using dual-feature finite mixture model and global-local structural preservation," *Pattern Recognition*, vol. 80, pp. 183 – 195, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S003132031830089X

[70] S. Zhang, K. Yang, Y. Yang, and Y. Luo, "Nonrigid image registration for low-altitude suav images with large viewpoint changes," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 4, pp. 592–596, April 2018.

[71] Z. Yang, Y. Yang, K. Yang, and Z. Wei, "Non-rigid image registration with dynamic gaussian component density and space curvature preservation," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2584–2598, May 2019.

[72] A. L. Yuille and N. M. Grzywacz, "A mathematical analysis of the motion coherence theory," *International Journal of Computer Vision*, vol. 3, no. 2, pp. 155–175, Jun 1989. [Online]. Available: https://doi.org/10.1007/BF00126430

[73] K. Lebeda, J. Matas, and O. Chum, "Fixing the locally optimized ransac," *British Machine Vision Conference*, 2012.

[74] J. Brandt, "Transform coding for fast approximate nearest neighbor search in high dimensions," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 1815–1822.

[75] M. Salzmann, R. Hartley, and P. Fua, "Convex optimization for deformable surface 3-d tracking," in *2007 IEEE 11th International Conference on Computer Vision*, Oct 2007, pp. 1–8.

[76] A. Varol, M. Salzmann, P. Fua, and R. Urtasun, "A constrained latent variable model," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 2248–2255.

[77] F. Zhang and F. Liu, "Parallax-tolerant image stitching," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 3262–3269.

[78] C. Chang, Y. Sato, and Y. Chuang, "Shape-preserving half-projective warps for image stitching," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 3254–3261.

[79] J. Li, Z. Wang, S. Lai, Y. Zhai, and M. Zhang, "Parallax-tolerant image stitching based on robust elastic warping," *IEEE Transactions on Multimedia*, vol. 20, no. 7, pp. 1672–1687, July 2018.