

PROGRESSIVE FILTERING FOR FEATURE MATCHING

Xingyu Jiang¹, Jiayi Ma¹, Jun Chen²

¹Wuhan University, Wuhan, China, ²China University of Geosciences, Wuhan, China

jiangx.y@whu.edu.cn, jyma2010@gmail.com, chenjun71983@163.com

ABSTRACT

In this paper, we propose a simple yet efficient method termed as *Progressive Filtering for Feature Matching*, which is able to establish accurate correspondences between two images of common or similar scenes. Our algorithm first grids the correspondence space and calculates a typical motion vector for each cell, and then removes false matches by checking the consistency between each putative match and the typical motion vector in the corresponding cell, which is achieved by a convolution operation. By refining the typical motion vector in an iterative manner, we further introduce a progressive matching strategy based on the coarse-to-fine theory to promote the matching accuracy gradually. The density estimation is utilized to address the island samples and accelerate the convergence of the mismatch removal procedure. In addition, our method is quite efficient where the gridding strategy enables it to achieve linear time complexity. Extensive experiments on several representative real images involving different types of geometric transformations demonstrate the superiority of our approach over the state-of-the-art.

Index Terms— Feature matching, filtering, density estimation, progressive, outlier

1. INTRODUCTION

Establishing reliable correspondences between two image feature sets is a fundamental problem in computer vision, and it has been widely used in a wide range of applications including 3D reconstruction, SLAM, visual homing, panoramic image mosaic, image registration and fusion [1, 2, 3, 4, 5]. These tasks typically desire a robust and efficient matching strategy to seek as many correct correspondences as possible while keeping mismatches to a minimum. However, due to the combinatorial nature, matching N points to another N points would create a total of $O(N!)$ computational complexity. To simplify the matching procedure, existing methods commonly construct a set of putative matches firstly based on the similarity of local image descriptors associated with the feature points, and then filter out false matches from the constructed putative set using additional geometrical constraints

[6, 7, 8]. In this paper, we focus on mismatch removal from a given putative correspondence set.

During the past decades, a number of methods have been investigated to address the mismatch removal problem. The most representative perhaps is random sample consensus (RANSAC) [9] and its variants [10, 11]. Following a hypothesize-and-verify strategy, these methods aim to find a smallest consistent inlier set to fit a given geometric model and estimate a pre-defined transformation by resampling randomly. However, these methods would degrade sharply and even fail when the geometric transformation between two images is not parametric, *e.g.* non-rigid. To deal with this limitation, several non-parametric model based techniques have been developed, such as identifying correspondence function (ICF) [12] and manifold regularization-based robust point matching (MR-RPM) [13]. These methods typically interpolate a mapping function between two images based on a slow-and-smooth prior. They have shown promising performance on addressing the deformable matching problem, but the smoothness prior will be violated if the image scene contains large discontinuous motion, *e.g.* wide baseline image pairs. Graph matching is another alternative for feature matching, which usually constructs an affinity matrix and formulates the matching problem as a quadratic assignment problem to seek the maximum inlier set [14]. Representatives include mode seeking [15, 16], graph shift (GS) [17], graduated consistency regularization [18], *etc.* Nevertheless, graph matching methods suffer from high computational cost, which is not applicable to large-scale matching problem. Recently, some new approaches based on neighborhood consistency preservation are also studied, such as grid-based motion statistics (GMS) [3], locality preserving matching (LPM) [1, 19], and deep learning method [20]. These methods are quite efficient for fast matching, but cannot work well when the putative set involves a large number of outliers.

In this paper, we propose a simple yet efficient method, named *progressive filtering for feature matching* (PFFM), for robust feature matching using a grid-based strategy and filtering in signal denoising. In particular, we first grid the correspondence space and calculates a typical motion vector for each cell, then the mismatches are identified by checking the consistency between each putative match and the corresponding typical motion vector under a convolution operation in lin-

This work was supported by the National Natural Science Foundation of China under Grant 61773295 and Grant 61503288.

ear time complexity. To improve the matching accuracy, we introduce a progressive matching strategy to iteratively refine the typical motion vectors. In addition, the density estimation is adopted to address the island samples and accelerate the convergency of the mismatch removal procedure. The qualitative and quantitative results demonstrate that our PFFM can achieve consistently better matching accuracy compared to the state-of-the-art competitors.

Our major contributions include: i) The proposed method does not require a pre-defined transformation model as many existing methods do, which is more generally and can deal with the matching problem undergoing a wide range of geometrical transformations. ii) From a novel perspective, we convert the putative matches into a motion matrix and handle the outliers with a convolution operation, which provide a guide to address the feature matching problem using deep learning techniques in future. iii) The gridding strategy enables our method to achieve linear time complexity, and our method can fulfil the matching problem in dozens of milliseconds even the putative set contains thousands of matches. This is beneficial for addressing real-time tasks.

2. METHOD

To establish feature matches, we firstly construct a set of putative matches by comparing the similarity of SIFT feature descriptors [21], and then the matching task boils down to rejecting the false matches from the given putative set.

2.1. Problem Formulation

Given two images I and I' of the same or similar scene, suppose we have obtained a set of N putative matches $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ extracted from them, where $\mathbf{x}_i = (u, v)^T$ and $\mathbf{y}_i = (u', v')^T$ are the pixel coordinates (*i.e.*, extracted feature points) of I and I' , respectively. Let $\mathbf{m}_i = \mathbf{y}_i - \mathbf{x}_i$ denote the motion vector of match $(\mathbf{x}_i, \mathbf{y}_i)$, then we can transform the putative set into $\mathcal{S}' = \{(\mathbf{x}_i, \mathbf{m}_i)\}_{i=1}^N$ and our goal is to find reliable matches from the putative set by utilizing the geometrical consistency among true matches¹ [1, 3].

Referring to image filtering for denoising, when given a noisy image, it is to considering the pixels in a local area (determined by the convolution kernel size) comprehensively, such as mean or median operation, to restore the pixel intensity and filter the gaussian or salt noise. From this point, it is feasible to calculate the average motion vector on the potential true matches in a small region, and reject the false matches by checking the deviation between each putative motion vector and the average motion vector based on the consistency.

To this end, we divide each dimension of feature points $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$ into n_c non-overlapping parts equally, and ob-

¹The geometrical consistency denotes that the correct matches should have similar motion behavior, as least in local neighborhoods, while the false matches are usually randomly distributed.

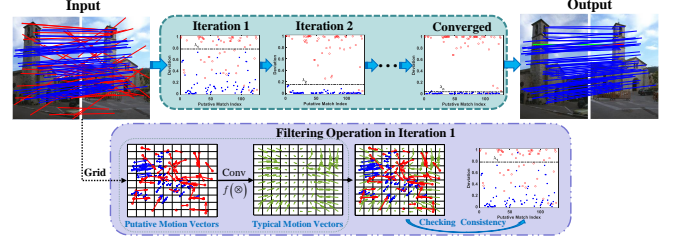


Fig. 1. The proposed progressive filtering framework for robust feature matching. Blue: inlier; red: outlier.

tain $G = n_c \times n_c$ cells. Accordingly, the putative set \mathcal{S}' can be divided into G parts with $\mathcal{X} = \{\mathcal{C}_{j,k}\}_{j,k=1}^{n_c}$ as the gridding putative motion vectors, as shown in Fig. 1. We denote $\bar{\mathbf{M}}$ as the average motion matrix, where $\bar{\mathbf{M}}_{j,k} = \bar{\mathbf{m}}_{j,k}$ is the average motion vector in the (j, k) -th cell determined by

$$\bar{\mathbf{m}}_{j,k} = \begin{cases} \frac{1}{|\mathcal{C}_{j,k}|} \sum_{i|\mathbf{x}_i \in \mathcal{C}_{j,k}} \mathbf{m}_i, & \text{if } |\mathcal{C}_{j,k}| > 0, \\ \mathbf{0}, & \text{if } |\mathcal{C}_{j,k}| = 0. \end{cases} \quad (1)$$

We calculate the deviation as $\mathcal{E} = \{\mathbf{e}_i = \mathbf{m}_i - \bar{\mathbf{m}}_{j,k} | \forall i, \mathbf{x}_i \in \mathcal{C}_{j,k}\}_{i=1}^N$. Without loss of generality, we can an assumption that $\mathbf{e}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$, where \mathbf{I} is a $D \times D$ unit matrix with $D = 2$ being the dimension of feature points. Therefore, it is desirable to utilize the mean filtering for mismatch removal.

Nevertheless, there are still two limitations in the aforementioned strategy. On the one hand, if the putative match is island in one cell, *i.e.*, $\mathbf{x}_i \in \mathcal{C}_{j,k}$ and $|\mathcal{C}_{j,k}| = 1$ then $\bar{\mathbf{m}}_{j,k} = \mathbf{m}_i$, resulting in the deviation is zero for both true and false matches. On the other hand, the average operation in a single cell ignores the connection between neighboring cells. When the putative set contains a large number of outliers (which often occurs in the feature matching problem), the consistency based on single cell will be badly degraded. To address these two problems, in the following we propose two strategies including the density estimation of each sample and the convolution of motion vectors.

2.2. Density Estimation

As aforementioned, there would exist some island matches that affect a lot. To address this issue, we firstly regard $(\mathbf{x}_i, \mathbf{m}_i)$ as a sample from the putative set \mathcal{S}' with dimension D_0 (*e.g.* $D_0 = 4$). Then we divide these samples into $n_0^{D_0}$ non-overlapping cells with a small value of n_0 (*e.g.*, $n_0 = 5$) and select inliers roughly by the density of each high dimension cell, which is defined as

$$S(n) = \frac{C(n) - f^{D_0} N}{\sqrt{f^{D_0} (1 - f^{D_0}) N}}, \quad (2)$$

where $C(n)$ is the sample numbers located in the n -th high dimension cell with $n = 1, \dots, n_0^{D_0}$, $f = \frac{1}{n_0}$ is the probability of each sample to locate in a divided part along one dimension of the sample, $f^{D_0} N$ is the expectation of putative

match number in each cell, $\sqrt{f^{D_0}(1 - f^{D_0})N}$ is its standard deviation. $S(n)$ measures the density degree of cell n , and we set a threshold τ to reject outliers. Here τ with a default value 2 controls the deviation level from expectation of objects.

2.3. Convolution Operation

In order to utilize the interaction among neighboring cells, we consider the local $n_k \times n_k$ cells comprehensively according to the convolution theory. The convolution $f(\otimes)$ of putative motion vectors is defined as

$$f(\otimes) : \widetilde{\mathbf{M}} = \frac{(\mathbf{W} \cdot \widetilde{\mathbf{M}}) \otimes \kappa}{\mathbf{W} \otimes \kappa + \varepsilon}, \quad (3)$$

where $\widetilde{\mathbf{M}}$ is the generated $n_c \times n_c \times D$ matrix after convolution, with $\widetilde{\mathbf{M}}_{j,k} = \widetilde{\mathbf{m}}_{j,k}$ denoting the *typical motion vector* of cell (j, k) , \mathbf{W} is a count matrix with $\mathbf{W}_{j,k} = |\mathcal{C}_{j,k}|$, and κ is a Gaussian kernel distance matrix of size $n_k \times n_k$ ($n_k = 3$ in this paper), where we design it as

$$\kappa = \frac{\exp\{-\mathbf{D}\}}{\sum_{i=1}^{n_k} \sum_{j=1}^{n_k} \exp\{-\mathbf{D}_{i,j}\}}, \quad \mathbf{D} = \begin{bmatrix} \sqrt{2} & 1 & \sqrt{2} \\ 1 & 0 & 1 \\ \sqrt{2} & 1 & \sqrt{2} \end{bmatrix}. \quad (4)$$

The numerator in Eq. (3) is used for weight compensating to preserve the scale of convolution results, where ε is a infinitesimal positive number in case there exists 0 of $\mathbf{W} \otimes \kappa$.

After the convolution, we obtain the typical motion vector of each cell, as shown by the green color vector located in the centre of each cell in Fig. 1. Then we define the deviation between \mathbf{m}_i and the corresponding $\widetilde{\mathbf{m}}_{j,k}$ as

$$d_i = 1 - \exp\left\{-\frac{\|\mathbf{m}_i - \widetilde{\mathbf{m}}_{j,k}\|^2}{\beta^2}\right\}, \quad \forall i, \mathbf{x}_i \in \mathcal{C}_{j,k}, \quad (5)$$

where β is used for determining the width of the range of interaction between two motion vectors, and we empirically set $\beta^2 = 0.08$. Thus the inlier set \mathcal{I}^* can be detected by comparing the deviation and a given threshold λ :

$$\mathcal{I}^* = \{i \mid d_i \leq \lambda\}. \quad (6)$$

2.4. Progressive Filtering

From Fig. 1, we see that the inliers and outliers are not so separable by a certain threshold (as shown in Iteration 1), and we can only filter a part of false matches by λ_1 . In fact, if the typical motion vectors are constructed only by inliers, then the deviation of inliers and outliers will almost tend to 0 and 1, respectively. Nevertheless, the ground truth inliers are not available in advance. To solve this dilemma, we propose an iteration strategy to remove outliers progressively. It iteratively refines the typical motion vector and anneals the threshold λ based on the coarse-to-fine theory. The inlier set is approximated with the results of each iteration until convergence.

As shown in the first line of Fig. 1, the deviation margin between inliers and outliers has been distinctly enlarged as the iteration proceeds. We name our method as progressive filtering for feature matching and summarize it in Alg. 1.

Algorithm 1: The PFFM Algorithm

Input: Putative set $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, parameters n_c, γ, τ

Output: Inlier set \mathcal{I}^*

- 1 Initialize the inlier set \mathcal{I} and parameter λ ;
 - 2 Convert \mathcal{S} into \mathcal{S}' and gridding;
 - 3 Filter out distinct outliers by density estimation by Eq. (2);
 - 4 *Iteration:*
 - 5 Calculate matrix $\widetilde{\mathbf{M}}$ and \mathbf{W} ;
 - 6 Construct matrix $\widetilde{\mathbf{M}}$ using Eq. (3);
 - 7 Calculate the deviations using Eq. (5);
 - 8 Determine \mathcal{I}^* using Eq. (6);
 - 9 Annealing: $\lambda \leftarrow \gamma\lambda$;
 - 10 *Until convergence;*
 - 11 Return \mathcal{I}^* .
-

2.5. Computational Complexity

To obtain the gridding putative set, we need to calculate the quotients of \mathcal{X} over the divided interval, thus, the time cost of initialization, putative set converting and gridding in Lines 1 and 2 of Alg. 1 is $O(N)$. In Lines 3, the density is estimated in each high dimension cell with the time complexity $O(N)$. The average motion vectors and count matrix are calculated in each cell and each match is only used once, which costs $O(N)$ time complexity. As for the convolution operation, it depends on the cell number and the kernel size, which has time complexity close to $O(n_k^2 \times n_c^2)$. Moreover, calculating the deviations and determining the inlier set using Eqs. (5) and (6) in Lines 7 and 8 cost $O(N)$ complexity. Our PFFM can converge in very few iterations (typically 3 times), therefore, the total time complexity of our PFFM is about $O(n_k^2 \times n_c^2 + N)$. The space complexity of our algorithm is $O(N)$ due to the memory requirement for only storing the putative set and the deviation. Generally, n_k and n_c are constants and both much smaller than N , thus both the time and space complexities of our method can be simply written as $O(N)$. That is to say, our PFFM has linear complexity, which is significant for addressing large-scale or real-time problems.

2.6. Implementation Details

There may exist multiple putative matches sharing a common feature point, *i.e.*, $\mathbf{x}_i = \mathbf{x}_j, i \neq j$, which would influence the performance. We initialize these putative matches as outliers. To eliminate the influence of the coordinate scale of feature points, we normalize them ranging from 0 to 1. We empirically set the default value of gridding size n_c and the iteration time as 10 and 5, which works sufficient well for addressing the feature matching problem. Our PFFM uses an iterative strategy to filter the outliers progressively, which is similar to deterministic annealing. Thus, we set the threshold λ to a large value in the first, then decrease it gradually with respect to iteration, the annealing rate is controlled by parameter γ . We initialize $\lambda = 0.8$ and set $\gamma = 0.25$ throughout this paper.

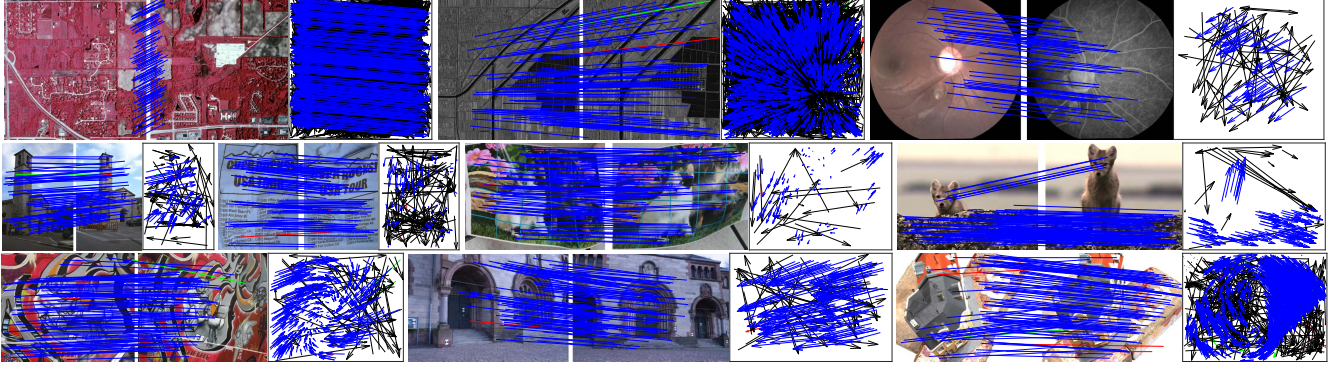


Fig. 2. Feature matching results of our PFFM on 10 typical image pairs involving different types of transformations (blue = true positive, black = true negative, green = false negative, red = false positive). For visibility, in the image pairs, at most 100 randomly selected matches are shown, and we do not show the true negatives.

Table 1. Precision (P), Recall (R), F-score (F) and Run time (T) of RANSAC [9], ICF [12], GS [17], GMS [3], MR-RPM [13], LPM [1], and our PFFM on the 10 image pairs shown in Fig. 2. The Average values are shown at the bottom. Red: the best.

Data	RANSAC				ICF				GS				GMS				MR-RPM				LPM				PFFM (Ours)			
	P (%)	R (%)	F	T (ms)	P (%)	R (%)	F	T (ms)	P (%)	R (%)	F	T (ms)	P (%)	R (%)	F	T (ms)	P (%)	R (%)	F	T (ms)	P (%)	R (%)	F	T (ms)	P (%)	R (%)	F	T (ms)
RS01	100.0	100.0	1.00	1.12e4	12.74	100.0	0.23	4.83e3	100.0	72.37	0.84	3.50e3	96.43	94.74	0.96	2.04e0	10.59	100.0	0.19	5.15e2	81.04	95.61	0.88	2.62e1	100.0	100.0	1.00	7.77e1
RS02	96.72	100.0	0.98	1.96e2	100.0	82.79	0.91	1.66e1	99.54	75.29	0.86	9.67e1	83.33	71.43	0.77	1.13e0	44.36	100.0	0.61	4.72e2	96.65	87.82	0.92	1.73e0	99.29	98.95	0.99	1.25e2
Retina	100.0	100.0	1.00	3.65e2	73.13	100.0	0.84	2.67e3	94.23	100.0	0.97	6.88e3	96.34	86.42	0.91	1.70e0	100.0	91.84	0.96	1.15e1	94.23	100.0	0.97	1.22e1	100.0	100.0	1.00	6.16e1
Church	95.16	83.10	0.89	1.49e2	93.75	63.38	0.76	1.93e1	95.83	97.18	0.97	1.00e2	86.76	83.10	0.85	0.86e0	100.0	80.28	0.89	1.28e1	82.50	92.96	0.87	1.21e0	98.59	98.59	0.99	4.40e1
Tshirt	96.39	80.81	0.88	1.00e3	78.26	90.91	0.84	4.61e1	93.06	67.68	0.78	2.60e2	79.21	80.81	0.80	1.02e0	97.98	97.98	0.98	1.89e1	87.76	86.87	0.87	5.47e0	98.02	100.0	0.99	6.35e1
Dogcat	100.0	97.85	0.99	1.02e1	92.19	63.44	0.75	1.66e1	97.70	91.40	0.94	5.65e2	93.18	88.17	0.91	0.88e0	100.0	100.0	1.00	6.53e0	97.89	100.0	0.99	0.81e0	98.94	100.0	0.99	4.83e1
Fox	98.10	91.15	0.94	4.58e1	98.57	61.06	0.75	1.83e1	100.0	90.27	0.95	1.31e3	96.49	97.35	0.97	0.84e0	97.12	89.38	0.93	8.27e0	94.87	98.23	0.97	0.83e0	100.0	100.0	1.00	6.16e1
Graft	99.74	100.0	1.00	1.45e1	100.0	14.02	0.25	1.86e2	99.66	77.51	0.87	2.42e3	98.07	93.92	0.96	1.23e0	100.0	96.30	0.98	2.11e1	98.14	97.88	0.98	2.76e0	99.21	99.47	0.99	8.67e1
Herjesu	99.09	86.51	0.92	1.19e2	98.55	53.97	0.70	3.07e1	99.07	84.13	0.91	2.32e2	87.07	80.16	0.83	1.87e0	98.29	91.27	0.95	1.49e1	96.88	98.41	0.98	1.13e0	99.21	100.0	1.00	6.33e1
House	98.66	82.20	0.90	8.31e1	100.0	60.67	0.76	1.91e3	100.0	58.71	0.74	2.70e4	95.81	93.76	0.95	1.61e0	97.63	96.18	0.97	2.23e2	94.48	98.97	0.97	8.96e0	97.27	99.44	0.98	1.49e2
Average	98.38	92.16	0.95	1.32e3	84.72	69.02	0.68	9.74e2	97.91	81.45	0.88	4.24e3	91.27	86.98	0.89	1.32e0	84.60	94.32	0.85	1.30e2	92.44	95.68	0.94	6.10e0	99.05	99.65	0.99	7.80e1

3. EXPERIMENTAL RESULTS

We evaluate the performance of PFFM on 10 typical image pairs, which undergo different types of geometric transformations including affine, non-rigid, epipolar geometry, *etc.* The numbers of putative matches in the 10 test pairs are 2152, 1982, 101, 126, 226, 113, 135, 442, 184 and 1367, respectively, and the inlier ratios are about 68.48%, 43.09%, 48.51%, 56.35%, 43.81%, 82.30%, 83.70%, 85.52%, 68.48% and 78.49%, respectively, with the ground truth established by manually checking of each putative match. We use 6 state-of-the-art methods for comparison, including RANSAC [9], ICF [12], GS [17], GMS [3], MR-RPM [13], and LPM [1]. All the experiments are conducted on a desktop with 4.0 GHZ Intel Core i7-6700K CPU, 8GB memory, and Matlab code.

We qualitatively show the matching results of PFFM in Fig. 2. From the results, we see that just a few matches are misjudged even when the outliers are dominated or image pairs suffer from great rotation, scale change, or non-rigid deformation. Further, we provide a quantitative comparison of different methods on the test data. The matching performance is characterized by precision (P), recall (R), F-score (F)², and run time. The detailed results are reported in Table 1. From

² P is defined as the ration of the identified inlier number and the preserved match number, R is defined as the ratio of identified inlier number and the whole inlier number, and F is defined as the ratio of $2PR$ and $P + R$.

the results, we can see that our method can almost consistently achieve the best matching accuracy, such as the precision, recall and F-score. Besides, although the runtime of our PFFM is not the best, it can also achieve comparable results, which requires merely dozens of milliseconds for mismatch removal from thousands of putative matches. This demonstrates the generality and effectiveness of our PFFM to deal with feature matching problem over the state-of-the-art.

4. CONCLUSIONS

In this study, we propose a new feature matching method based on progressive filtering. We first divide the putative set into non-overlapping cells and calculate the average motion vector of each cell, and then we estimate the density and preserve some reliable cells. Finally, we construct the typical motion vectors using convolution operation like mean filtering and detect outliers by checking the deviations between the putative motion vector and its corresponding typical motion vector, and an iterative strategy is proposed to filter out the outliers progressively. Our method can converge in a few iterations, and the gridding strategy enables it to achieve linear time complexity. The qualitative and quantitative results demonstrate that our method is efficient for addressing various types of matching problems with high accuracy.

5. REFERENCES

- [1] Jiayi Ma, Ji Zhao, Junjun Jiang, Huabing Zhou, and Xiaojie Guo, “Locality preserving matching,” *Int. J. Comput. Vis.*, 2018.
- [2] Wen-Yan Lin, Fan Wang, Ming-Ming Cheng, Sai-Kit Yeung, Philip HS Torr, Minh N Do, and Jiangbo Lu, “Code: Coherence based decision boundaries for feature correspondence,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 34–47, 2018.
- [3] Jiawang Bian, Wen-Yan Lin, Yasuyuki Matsushita, Sai-Kit Yeung, Tan-Dat Nguyen, and Ming-Ming Cheng, “Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence,” in *CVPR*, 2017, pp. 2828–2837.
- [4] Jiayi Ma, Ji Zhao, Jinwen Tian, Zhuowen Tu, and Alan L Yuille, “Robust estimation of nonrigid transformation for point set registration,” in *CVPR*, 2013, pp. 2147–2154.
- [5] Josef Maier, Martin Humenberger, Markus Murschitz, Oliver Zendel, and Markus Vincze, “Guided matching based on statistical optical flow for fast and robust correspondence analysis,” in *ECCV*, 2016, pp. 101–117.
- [6] Yang Lin, Zhouchen Lin, and Hongbin Zha, “The shape interaction matrix-based affine invariant mismatch removal for partial-duplicate image search,” *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 561–573, 2017.
- [7] G. Wang, Z. Wang, Y. Chen, Q. Zhou, and W. Zhao, “Context-aware gaussian fields for non-rigid point set registration,” in *CVPR*, 2016, pp. 5811–5819.
- [8] Yuan-Ting Hu, Yen-Yu Lin, Hsin-Yi Chen, Kuang-Jui Hsu, and Bing-Yu Chen, “Matching images with multiple descriptors: An unsupervised approach for locally adaptive descriptor selection,” *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5995–6010, 2015.
- [9] Martin A Fischler and Robert C Bolles, “Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [10] Philip HS Torr and Andrew Zisserman, “MLESAC: A new robust estimator with application to estimating image geometry,” *Comput. Vis. Image Understand.*, vol. 78, no. 1, pp. 138–156, 2000.
- [11] Ondrej Chum and Jiri Matas, “Matching with PROSAC - progressive sample consensus,” in *CVPR*, 2005, pp. 220–226.
- [12] Xiangru Li and Zhanyi Hu, “Rejecting mismatches by correspondence function,” *Int. J. Comput. Vis.*, vol. 89, no. 1, pp. 1–17, 2010.
- [13] Jiayi Ma, Ji Zhao, Junjun Jiang, and Huabing Zhou, “Non-rigid point set registration with robust transformation estimation under manifold regularization,” in *AAAI*, 2017, pp. 4218–4224.
- [14] Marius Leordeanu and Martial Hebert, “A spectral technique for correspondence problems using pairwise constraints,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2005, pp. 1482–1489.
- [15] Minsu Cho and Kyoung Mu Lee, “Mode-seeking on graphs via random walks,” in *CVPR*, 2012, pp. 606–613.
- [16] Chao Wang, Lei Wang, and Lingqiao Liu, “Progressive mode-seeking on graphs for sparse feature matching,” in *ECCV*, 2014, pp. 788–802.
- [17] Hairong Liu and Shuicheng Yan, “Common visual pattern discovery via spatially coherent correspondence,” in *CVPR*, 2010, pp. 1609–1616.
- [18] Junchi Yan, Minsu Cho, Hongyuan Zha, Xiaokang Yang, and Stephen M Chu, “Multi-graph matching via affinity optimization with graduated consistency regularization,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 6, pp. 1228–1242, 2016.
- [19] Junjun Jiang, Qing Ma, Tao Lu, Zhongyuan Wang, and Jiayi Ma, “Feature matching based on top k rank similarity,” in *ICASSP*, 2018, pp. 2316–2320.
- [20] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua, “Learning to find good correspondences,” in *CVPR*, 2018, pp. 2666–2674.
- [21] David G Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.