

Fast Rejecting Mismatches Using Pair-wise Similarity

Deheng Qian, Xu Yang and Hong Qiao*

Abstract—Point matching problem seeks the optimal correspondences between two sets of points. However, the matching result often includes some mismatches that decrease the matching precision. In this paper, we propose a fast algorithm to reject mismatches using pair-wise similarity. The intuition of our algorithm is that the matches should be similar with their neighboring matches due to local consistency. Our algorithm consists of two steps. In the first step, the algorithm eliminates mismatches at the cost of rejecting some correct matches to obtain a refined matching result with a high precision. In the second step, the algorithm regains the correct matches rejected in the first step to improve the recall of the final matching result. The time complexity of the algorithm is $O(n^2)$, which is asymptotically faster than conventional algorithms that reject mismatches. We demonstrate the effectiveness of the proposed algorithm by multiple experiments over widely used datasets.

I. INTRODUCTION

PPOINT matching is a fundamental problem in the field of robotic vision. Its applications broadly include image mosaicking, 3D reconstruction and SLAM. The most common way to obtain a matching result is to match points while minimizing the dissimilarities of their unary features. In spite of its efficiency, this approach does not guarantee a good matching result.

Consequently, post-processing algorithms are proposed to refine the initial matching result. However, the conventional algorithms are often not efficient enough due to their high computational complexities. In this paper, we propose a new algorithm which runs in time $O(n^2)$ and is asymptotically faster than conventional algorithms.

Our algorithm works under two assumptions. The first one is that, correct matches are similar to their neighbor correct ones. In the meanwhile, incorrect matches are generally arbitrary and do not necessarily be similar with each other. The second assumption is that, in the initial matching result, there are more correct matches than the incorrect ones. We will see that the second assumption would be unnecessary by modifying the proposed algorithm.

Under these assumptions, the algorithm proceeds by two steps. In the first step, we compare every match with its neighbor matches to check whether there is a pair of matches that their similarity is lower than a predefined threshold. If there is such a pair, under the first assumption, we know that there is at least one incorrect match in this pair. Hence, we delete both matches of the pair, resulting in that the precision of the remaining matches is improved. Recursively, we would finally obtain a matching result with a high matching precision. Every match in the result is

similar to its neighbor matches. However, in the first step, we would delete many correct matches. In order to improve the recall of our final result, the algorithm regains these deleted correct matches in its second step. We compare every deleted match with its nearest matches that remained after the first step to see that whether it is similar with these remained matches. If so, the deleted match can be regained as a correct match. Consequently, a final matching result is obtained. Most incorrect matches are deleted in the first step at the cost of deleting some correct matches. These deleted correct matches are then be regained in the second step.

As a straightforward extension of the above algorithm, we can check every k -tuple in the first step to see whether there is a k -tuple where all matches in it is not similar with any other match in the same tuple. If so, we can delete all the k matches since there is at most one correct match in it, under our first assumption. As a consequent, the second assumption may be relaxed.

We evaluate the proposed algorithm by multiple experiments and demonstrate its effectiveness. Compared with conventional methods, our algorithm works asymptotically faster and obtains competitive results.

The remainder of this paper is organized as follows: Section II discusses some related studies. Section III details our algorithm. Section IV describes our experiments. Section V presents our conclusions.

II. RELATED WORK

In this section, we only discuss the most related studies here. In order to obtain matching result with high precision, rejecting mismatches is studied. RANSAC is a classical method which estimates parameters of a predefined model and checks which model fits the matching result the best [1]. All matches that do not fit the best model are rejected. A variety of this method are also widely used. ICF estimates a pair of correspondence functions to model the transformation between different point sets [2]. Matches that do not coherent with the functions are rejected. VFC estimate the transformation between point sets by a non-parameter model [3]. After that, we can judge each match depending on the estimated model. However, all the above mentioned methods have issues on their computation complexities. Specifically, the worst time complexity of RANSAC is exponential. The worst time complexities of the ICF and VFC are both $O(n^3)$.

Note that, it only costs $O(n^2)$ to match points depending on the unary features of points. Hence, it is reasonable to require the postprocessing method, which rejects mismatches, run in time no more than $O(n^2)$. In this paper, we propose a method rejecting mismatches effectively, which runs in time $O(n^2)$.

Deheng Qian, Xu Yang and Hong Qiao are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. (email: hong.qiao@ia.ac.cn)

III. METHODOLOGY

In this section, we elaborate the proposed algorithm. We first present the general problem formulation. Then we introduce respectively two steps in the algorithm, by which mismatches are rejected while correct matches are maintained.

A. Problem Formulation

Let $P = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$ and $Q = \{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_m\}$ be the template point set and the scene point set respectively. Without loss of generality, we assume $n \leq m$. Further, we assume that both the point sets have zero mean and their standard deviations are 1. Let $C = \{c_1, c_2, \dots, c_n\}$ be the set of matches between P and Q , where c_i is the i -th correspondence.

B. Rejecting incorrect Matches

We first elaborate the pre-processing steps of P . While the processing of Q is similar. We first calculate the distance between every pair of points in P . We calculate

$$D_{ij}^P = \text{dist}(\mathbf{p}_i, \mathbf{p}_j), \quad (1)$$

where D_{ij}^P is the distance between the i -th point and the j -th point in P . Based on D^P , the neighbor points of each point can be defined. For example, we divide all the points into d subsets according to their distances to a point \mathbf{p}_i . All the points in the first subset are the closest points to \mathbf{p}_i . The distances between the points and \mathbf{p}_i are smaller than ϵ_1 . While in the last subset, the distances are larger than ϵ_{d-1} . Note $\epsilon_a < \epsilon_b$, if and only if $a < b$. As a result, all the points are divided into d subsets. We call \mathbf{p}_i and \mathbf{p}_j a pair of a -th degree neighbors if and only if \mathbf{p}_j is in the j -th subset, which is denoted by $a = N(\mathbf{p}_i, \mathbf{p}_j)$.

Based on the above definitions, we define the neighboring relations between two matches. Given two matches $(\mathbf{p}_i, \mathbf{q}_k)$, $(\mathbf{p}_j, \mathbf{q}_l)$, we say they are b -th degree neighbors if and only if $b = \min(N(\mathbf{p}_i, \mathbf{p}_j), N(\mathbf{q}_k, \mathbf{q}_l))$.

We assume that correct matches should be similar to each other if they are neighbors. Hence, if two matches are not similar while they are neighbors, there is at least one of them is an incorrect match.

We also assume that there are more correct matches in the result than the incorrect ones. Hence, we can safely reject the neighbor matches that are not similar with each other. By this process, we reject more incorrect matches than correct ones. Hence, the precision of the remaining matches is improved.

Specifically, given a pair of matches which are a -th degree neighbors, we check whether their dissimilarity is larger than a predefined threshold δ_a . If so, we reject them both. Otherwise, we maintain them and check another pair of matches until all pairs are similar enough according to their degrees of neighbors.

As a straightforward extension of the above algorithm, we select k neighboring matches at a time. If the k matches are not similar with each other, there are at least $k - 1$ of them are incorrect ones. If the precision of the initial matching

is higher than $1/k$, we can safely reject the k matches to improve the matching precision of the remaining matches.

C. Regaining Correct Matches

Note that in the process of rejecting incorrect matches, we would also reject some correct matches. Hence, the recall of the resulting matches might be low. In order to improve the recall of the final result, we try to regain the rejected correct matches.

After the first step, the resulting matches are supposed to have a high precision. Hence, we check every rejected match that if it is similar to the remaining matches. If so, than the rejected match is vary likely to be correct. Hence, we can regain it into our final matching result.

Specifically, given a rejected match $(\mathbf{p}_i, \mathbf{q}_j)$, we find all remained matches that are closest to $(\mathbf{p}_i, \mathbf{q}_j)$, and check whether there is any of them that is similar to $(\mathbf{p}_i, \mathbf{q}_j)$. If the similarity between $(\mathbf{p}_i, \mathbf{q}_j)$ and its most similar match is larger than a predefined threshold δ'_a , the rejected match $(\mathbf{p}_i, \mathbf{q}_j)$ is regained.

By this step, most of the correct matches that are rejected in the first place can be regained. Hence, the final matching would have a high precision and a high recall.

IV. EXPERIMENTS

To evaluate the improvement achieved by our algorithm, we test our algorithm on two datasets.

A. House and Hotel Sequences Dataset

In our first experiment, we assess our matching algorithm on the CMU house and hotel dataset. This dataset includes 111 successive frames of a house. Each image contains 30 manually labeled feature points with matching ground truth. The tested image pairs are created using two images separated by 10 images. In order to evaluate the improvement achieved by our algorithm, we first match all the points according to their ground truth, then randomly perturb some portion of the matching. The perturbed percentage increases from 10 to 60 by step size 10.

TABLE I
MATCHING RESULTS ON THE HOUSE DATASET.

Perturbed portion (%)	10	20	30	40	50	60
Precision (%)	99.9	99.8	99.8	99.9	99.4	96.3
Recall (%)	97.3	97.6	97.1	98.6	98.6	98.1

As shown in Tab. IV-A, the matching precision is able to be improved significantly by our algorithm. In the meanwhile, the improvement matching has a high recall. Interestingly, when the perturbed percentage increases to 60, our algorithm still performs well. Besides, in this dataset, our algorithm takes on average 40ms to process an initial matching including 30 matches, which demonstrates its efficiency.

B. Google Earth Pictures Dataset

In the second dataset, we manually label the matching between two pairs of pictures sampled from the Google earth. As shown in Figs. IV-B, IV-B, IV-B, IV-B, our algorithm successfully refined the initial matchings.

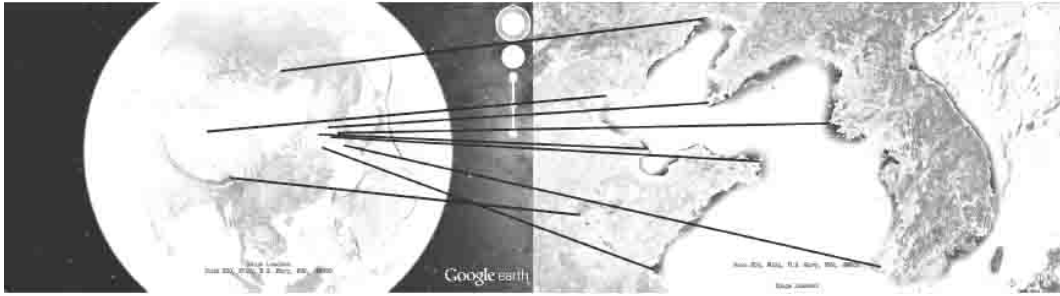


Fig. 1. The initial matching of the first pair of pictures.

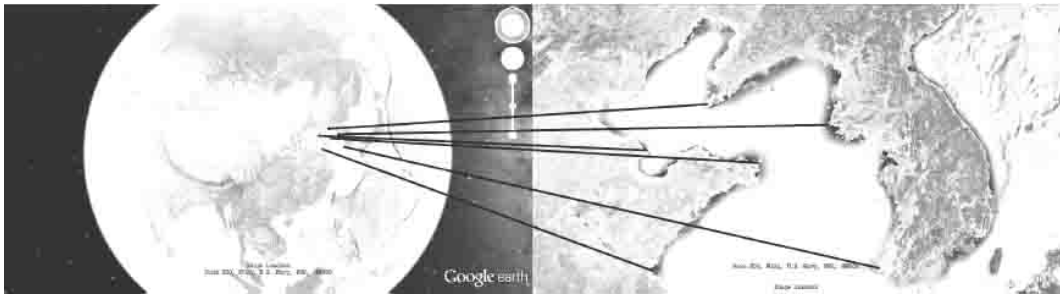


Fig. 2. The refined matching of the first pair of pictures.

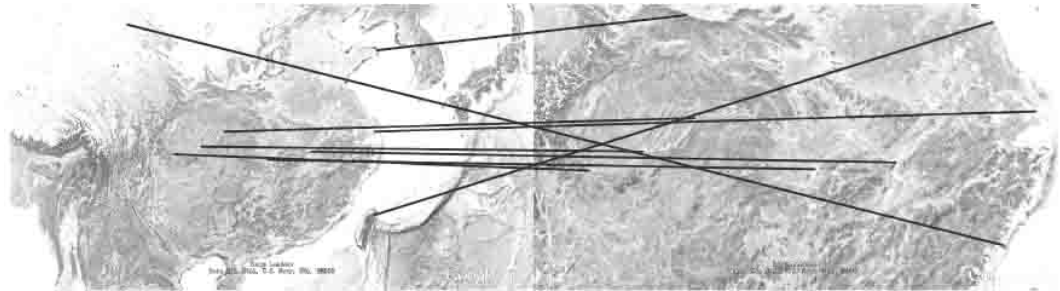


Fig. 3. The initial matching of the second pair of pictures.

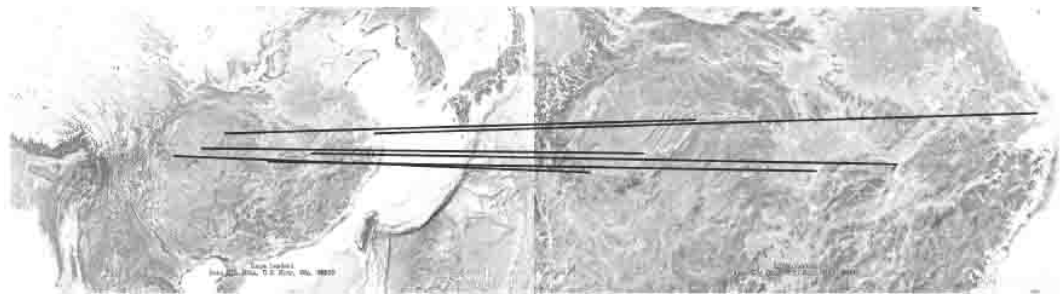


Fig. 4. The refined matching of the second pair of pictures.

V. CONCLUSIONS

In this paper, we presented a novel method to refine the matching result between two point clouds. Our method proceeds by two steps. In the first step, most incorrect matches, as well as some correct ones, are removed. The resulting matching is of high precision. In the second step, the removed correct matches are regained to obtain a final matching with a high recall. Our experiments show that the proposed method is effective and efficiency.

In the future, we would like to improve this algorithm with a more sophisticated probabilistic point of view.

ACKNOWLEDGMENT

The authors would like to thank Yuren Zhang for useful discussions.

REFERENCES

- [1] Raguram, Rahul and Frahm, Jan-Michael and Pollefeys, Marc, "A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus," *Eur. Conf. Comput. Vis.*, pp. 500-513, 2008
- [2] Li, Xiangru and Hu, Zhanyi, "Rejecting mismatches by correspondence function," *Int. J. Comput. Vision*, vol. 89, pp. 1-17, 2010.
- [3] Zhao, Ji and Ma, Jiayi and Tian, Jinwen and Ma, Jie and Zhang, Dazhi, "A robust method for vector field learning with application to mismatch removing," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2977-2984, 2010.