

FOMP: A Novel Preprocessing Technique to Speed-Up the Outlier Removal from Matched Points

Jonathan S. Ramos*, Carolina Y. V. Watanabe[†] and Agma. J. M. Traina*

**Institute of Mathematics and Computer Sciences - ICMC/USP, São Carlos - SP, 13566-590, Brazil*

[†]*Computer Science Department, UNIR, Porto Velho, RO, 76808-659, Brazil*

jonathan@usp.br, carolina@unir.br and agma@icmc.usp.br

Abstract—Image matching plays a major role in many applications, including pattern recognition and biomedical imaging. It encompasses three steps: 1) interest point selection; 2) feature extraction from each interest point; 3) features point matching. For steps 1 and 2, traditional interest point detectors/extractors have worked well. However, for step 3 even a few points incorrectly matched (outliers), might lead to an undesirable result. State-of-the-art consensus algorithms present a high time cost as the number of outlier increases. Aimed at overcoming this problem, we present FOMP, a novel preprocessing approach, that reduces the amount of outliers in the initial set of matched points by filtering out the vertices that present a higher difference among their edges in a complete graph representation of the points. The precision of traditional methods is kept, while the time is speed up in 50%. The approach removes, in average, more than 65% of outliers, while keeping over 98% of the inliers.

Keywords-Feature Point Matching; Outliers Removal; Filtering; Graph-based Approach.

I. INTRODUCTION

Image matching aims at finding corresponding elements or regions in pairs of images. It plays a major role in Computer Vision and it can be used to integrate/fuse images from the same scene, taken at different times or by different devices with distinct resolution. Image matching becomes more important when it is not possible to take a picture of the whole scene (with high resolution), such as in microscopical images [1] and in satellite/remote sense images [2], and it is necessary to build the full scene representation. Nevertheless, a few other applications also take advantage of image matching/fusion, such as medical imaging [3], image restoration [4], forensic document examination [5], content-based image retrieval [6].

The most common approach to match a pair of images is the feature-based methods, which encompasses three steps [7]: (1) interest point detection, in which algorithms such as Scale-Invariant Feature Transform (SIFT) [8] and Speed-Up Robust Features (SURF) [9] have been widely used; (2) feature extraction from each point, which takes into account the relationship among neighboring pixels of the interest point, for example, gradient orientations; (3) matching of extracted features from each interest point between pairs of images using a distance metric, for instance,

Euclidean Distance. The main drawback of feature based methods is that a few interest points might be incorrectly matched (outliers). In image matching/mosaicing, it is very import to have a precise adjustment of the points (inliers), since outliers lead to an incorrect adjustment.

An approach to detect outliers in matched points is the RANdom SAMple Consensus (RANSAC) method [10], which iteratively tries to minimize the least-square error between pairs of points from two images. RANSAC chooses random points on each iteration to estimate parameters of a mathematical model. The iteration that yields the maximum amount of inliers is chosen as the answer.

Graph-based methods have been used as well. The matched points from each image are considered as vertices and the spatial distance among vertices are explored. For instance, in [11] is presented the Graph Transformation Matching (GTM) algorithm that iteratively removes points with the highest residual difference among two median k -nearest neighbor (k - nn) graphs. The algorithm stops at reaching a null residual matrix. Based on the GTM algorithm, in [12] is proposed a weighted version (WGTM) that uses angular distances between neighboring features, which are less sensitive to rotation and scale, according to [12]. In [13] is presented a Spatial Order Constraints Bilateral-neighbor Vote (SOCBV), which uses Spatial Order Constraint (SOC) and k - nn density estimation to remove outliers, presenting a higher performance than WGTM.

GTM, WGTM and SOCBV algorithms eliminate one outlier at a time, which increase the processing time as the number of outliers grow. Aimed at minimizing this processing time, we propose a preprocessing approach, hereafter called *Filtering out Outliers from Matched Points* (FOMP), which reduces the amount of outliers in the set of matched points. Our approach filters out outliers based on the difference among edges in the complete graph representation of the matched points on each image. The experimental results show that the processing time is reduced, in average, by half. Moreover, our approach removes, in average, more than 65% of outliers, while keeping over 98% of the inliers.

The remainder of this paper is organized as follows. Section II presents our proposed method. In Section III the experimental results and discussion are shown. Finally,

Section IV summarizes the conclusions.

II. PROPOSED METHOD

Initially, we filter out matched points by using the edges differences between the two complete graphs representation of the two sets of matched points. A vertex that is an outlier, normally presents edges with different weights between the two graphs. However, as an edge connected to an outlier might as well be connected to an inlier, we summarize the differences for all edges connected to a vertex. Therefore, the vertex that presents the highest edge difference is removed at each iteration. Our approach, hereafter called *Filtering out Outliers from Matched Points* (FOMP), is divided in three steps, as summarized in Algorithm 1 and detailed as follows.

Let the interest points from the first image be $P = \{p_1, \dots, p_n\}$, where $p_i = (x, y)$ represents an interest point coordinates and n is the total amount of points. Likewise, for the second image, $P' = \{p'_1, \dots, p'_n\}$, where p'_i is matched to point p_i . The task is to find a subset of P and P' that presents a spatial consensus (geometrical consistency), i.e., both sets of points must present a similar shape even if there are changes of rotation, scale and affine transforms between them. Let's summarize the algorithm steps as follows.

Step 1: A complete weighted graph representation is created for each set of points, $G_p(V_p, E_p)$, where $V_p = \{v_1, \dots, v_n\}$ represents coordinate points in P . Likewise for the points from the second image, $G_{p'}(V_{p'}, E_{p'})$, where $V_{p'} = \{v'_1, \dots, v'_n\}$ represents coordinate points in P' . Each graph is then represented by an adjacency matrix W , with size $n \times n$, for the first graph, and W' , for the second one. The sets of edges, E_p and $E_{p'}$, are weighted, so that the connection from vertex v_i to vertex v_j , or vice-versa, is measured by the Euclidean distance between them, such as:

$$W(i, j) := \|v_i - v_j\|, \forall i, j, 1 \leq i, j \leq n \quad (1)$$

Step 2: In order to minimize the influence of outliers due to the images scale, all edges weights are normalized by its own mean:

$$W_\mu := \frac{W}{\text{mean}(W(i, j))} \quad (2)$$

Step 3: The differences among corresponding edges in the two graphs are summed up and normalized by the number of vertices:

$$D(i) := \frac{1}{n} \sum_{j=1}^n |W_\mu(i, j) - W'_\mu(i, j)|, 1 \leq i \leq n \quad (3)$$

The maximum value of D is calculated:

$$d_{max} := \max_{i=1 \dots n} D(i) \quad (4)$$

If d_{max} is greater than a threshold α , remove vertex corresponding to the d_{max} value from W and W' and go back

to step 2. This process continues until d_{max} is not greater than α . The value of α is set to 0.5 (found empirically) for all the results presented in this paper. In Figure 1 is shown an example of the FOMP method applied to a set of matched points, such that $P = \{\{0, 0\}, \{0, 4\}, \{4, 4\}, \{4, 0\}, \{2, 5\}\}$ and $P' = \{\{0, 0\}, \{0, 4\}, \{4, 4\}, \{4, 0\}, \{2, 1\}\}$. Note that p_5 (v_5) is an outlier.

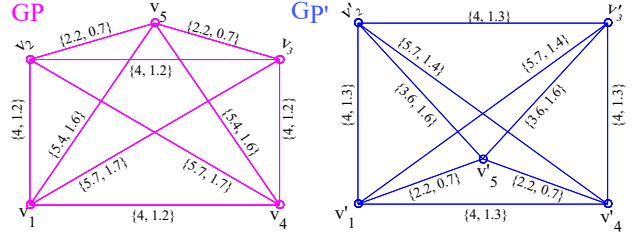


Figure 1. Applying FOMP method on a set of matched points. The mean edge values for both graphs are $\mu = 3.4$ (G_P) and $\mu = 3.1$ ($G_{P'}$). The two values shown in each edge represents the Euclidean distance (Equation 1) and the normalized value (Equation 2), respectively. For this example, $D = \{0.25, 0.18, 0.18, 0.25, 0.55\}$ (Equation 3). Therefore, by applying Equation 4, v_5 and v'_5 are filtered out.

Algorithm 1 Filtering out Outliers from Matched Points (FOMP): Pre-processing step.

```

1: function FOMP( $P, P', \alpha$ )
2:    $W \leftarrow \|p_i - p_j\|, \forall i, j, 1 \leq i, j \leq n$ 
3:    $W' \leftarrow \|p'_i - p'_j\|, \forall i, j, 1 \leq i, j \leq n$ 
4:    $flag \leftarrow true$ 
5:   while  $flag$  do
6:      $W_\mu \leftarrow \frac{W}{\text{mean}(W(i, j))}, \forall i, j$ 
7:      $W'_\mu \leftarrow \frac{W'}{\text{mean}(W'(i, j))}, \forall i, j$ 
8:      $D(i) \leftarrow \frac{1}{n} \sum_{j=1}^n |W_\mu(i, j) - W'_\mu(i, j)|, \forall i.$ 
9:      $d_{max} \leftarrow \max_{i=1 \dots n} D(i)$ 
10:    if  $d_{max} < \alpha$  then
11:       $flag \leftarrow false$ 
12:    else
13:      remove vertices regarding to  $d_{max}$ 
14:      from  $W$  and  $W'$ 
15:    end if
16:  end while
17: end function

```

A. Time Complexity

Analyzing Algorithm 1, we can see that creating a complete graph (lines 2 and 3) takes $O(n^2)$, since the Euclidean distance from all points to all points are calculated. Normalizing each graph weights by its own mean (lines 6 and 7) depends on the amount of edges on the network, $((n * (n - 1))/2)$, which leads to $O(n^2)$. Summing up the edges differences (line 8) takes $O(n^2)$ as well. Finding

the maximum value of D takes $O(n)$. The while condition iterations depends on the α value. However, let's assume the worst case, in which the algorithm iterates n times, filtering a pair of matched points at a time. Therefore, the algorithm has a time complexity of $O(1^2+2^2+3^2+\dots+n^2) = O(n^3)$. Even though this is a considerable high time complexity, most consensus algorithms such as GTM, WGTM and SOCBV eventually already calculate complete graphs in order to generate k -nn graphs. The GTM algorithm has a time complexity of $O(n^3 \log(n))$, WGTM and SOCBV have a time complexity of $O(n^2 \log(n))$. However, in our case the complete graphs calculated by FOMP can be reused inside those algorithms, reducing the processing time at the end. Besides, FOMP only has a few simple steps. The most time consuming step in this algorithm is the complete graph calculation.

III. EXPERIMENTS

In order to evaluate the performance of FOMP, we compared the methods RANSAC [10], GTM [11], WGTM [12] and SOCBV [14] with and without FOMP preprocessing. We used RANSAC implementation available in [15]. GTM and WGTM algorithm were generously provided by [12] and SOCBV algorithm by [14]. We analyzed the running time and the following measures:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (5)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (6)$$

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (7)$$

where:

- True positive (TP): inliers detected as inlier;
- False positive (FP): outlier identified as inlier;
- True negative (TN): outlier detected as outlier;
- False negative (FN): inlier identified as outlier.

The precision (Eq. 5) is the percentage of relevant inliers (TP) correctly detected. The recall (Eq. 6) is the percentage of inliers correctly detected among all inliers. The specificity (Eq. 7) is the percentage of outliers correctly detected among all outliers.

We divided the experiments in seven parts: First we analyzed the precision and recall with and without FOMP preprocessing. The methods in which FOMP preprocessing was applied is preceded by F in its name, for instance, F-GTM implies that FOMP was applied before the GTM method. We analyze if there are changes in precision and recall with and without FOMP under a few variations, such as camera movements/zoom, matching ambiguities, deformable objects (dataset provided by [11]) and affine transformations (dataset available in [16]). Then, we compared the running time for these four experiments. We

also analyzed the individual running time, recall and the number of iterations of FOMP. Furthermore, we analyzed the percentage of remaining inliers (recall) and the percentage of filtered outliers (specificity) by employing FOMP algorithm. Finally, we discuss and analyze the α parameter in the FOMP algorithm.

For all experiments, SURF feature points were extracted and matched for each image pair. Even though references [11], [14] have used SIFT for their experiments, we choose SURF, which is faster than SIFT [16]. All outliers were removed manually and we randomly selected sixty inliers, which is the same number used in the experiments of [11], [12], [14]. We added outliers percentage randomly in a controlled manner, so that no randomly generated outlier is equivalent to an inlier. The number of outliers added is generated according to Equation 8 [12].

$$\text{outliers} = \frac{\text{outlier percent} \times \text{inliers}}{100 - \text{outlier percent}} \quad (8)$$

where *inliers* represents the total number of inliers and *outliers* is the total number of outliers to be added. The outlier percentage starts from 5%, incrementing by 10%, up to 95%. We redo this 20 times for each outlier percentage, totalizing $n \times 20 \times 10$ runs for each experiment, where n is the total number of pairs of images in each dataset.

The experiments were performed on a desktop with 2.67 GHz Intel(R) Core(TM) i7 CPU, 6 GB memory. We used the parameters settings defined by the authors of each method. The k -value for GTM is set to 5, WGTM and SOCBV k -value is set to 20. The stop condition for WGTM (ϵ) is set to 0.001. $\beta = 0.2$ for SOCBV. For RANSAC, the distance threshold for determining the outliers is set to 0.1.

A. Camera movements

This dataset is composed of 10 pairs of images, which presents variations of translation, rotation and scale. These images have a resolution of 640×480 pixels. Figures 2(a)-(b) show the precision and recall comparison with/without FOMP preprocessing. The precision and recall are calculated using Equations 5 and 6, respectively. For the precision, all methods presented similar results from 5% to 85% of outliers. At 95% of outliers, SOCBV and WGTM presented better precision than F-SOCBV and F-WGTM, respectively. On the other hand, F-RANSAC performed better, while F-GTM and GTM present similar results. Analyzing the recall, all methods presented, in general, similar results for all percentage of outliers, leaving out F-GTM with better result.

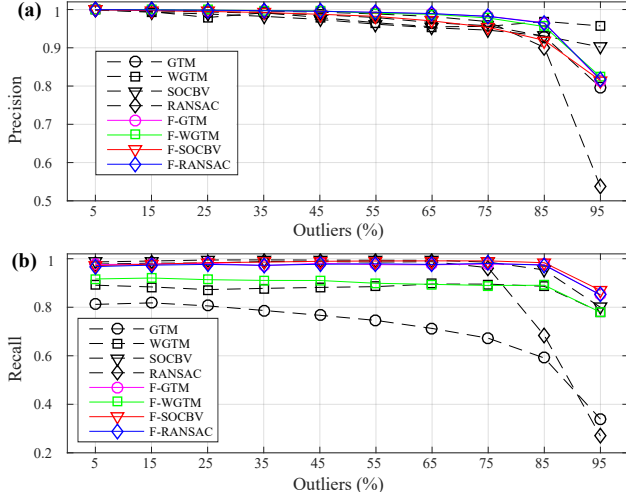


Figure 2. Comparison of precision and recall values for the four methods with/without FOMP preprocessing in the camera movements dataset, considering the number of outliers manually added.

Figure 3 shows a single pair of images, considering 85% of outliers in the initial set of matched points, comparing the methods with/without FOMP preprocessing. Note that, for F-GTM and F-RANSAC, a larger number of inliers (TP) is detected. WGTM detected more inliers than F-WGTM. However, the amount of FP are smaller for F-WGTM. Therefore, in this example, F-WGTM has a better precision. For SOCBV, the amount of TP is slightly bigger than F-SOCBV. Nonetheless, the amount of FP is bigger as well for F-SOCBV, which can be compensated by the running time of F-SOCBV (gain of 71%). Moreover, there is a considerable running time gain for F-GTM and F-WGTM as well (in average, gain of 73%). On the other hand, F-RANSAC did not present a gain considering the running time. In contrast, the amount of true positives is drastically changed for F-RANSAC. In addition, the running time difference is not too big, RANSAC is only 0.11 seconds faster than F-RANSAC.

B. Matching ambiguities

This dataset is composed of 10 pairs of images, which presents repetitive patterns leading to ambiguities in the point matching due to duplicity of the matching features. These images have an average resolution of 640×480 pixels.

Figures 4(a)-(b) show the precision and recall values for this dataset, respectively. Again, all methods presented similar precision from 5% to 85% of outliers. At 95% of outliers, GTM, WGTM and SOCBV showed better precision than F-GTM, F-WGTM and F-SOCBV, respectively. Inversely, F-RANSAC showed higher precision. For the recall, F-GTM performed considerable better for all outliers percentage, while kept similar results for the other methods from 5% to 75% of outliers. At 85% to 95% of outliers, RANSAC presented a considerable drop on the recall. At 95% of recall, all methods performed better with FOMP.

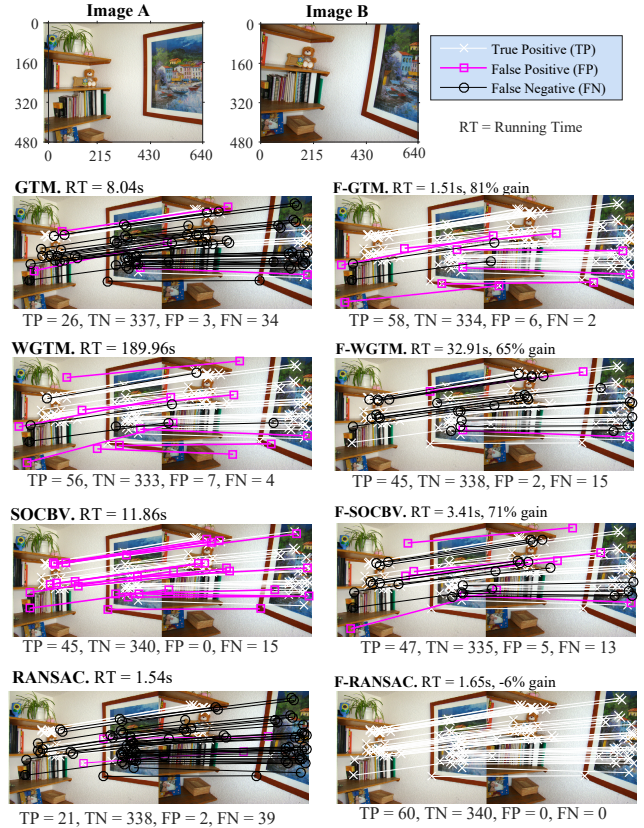


Figure 3. Image matching results for two images from the camera movement dataset (85% of outliers in the initial set of matched points).

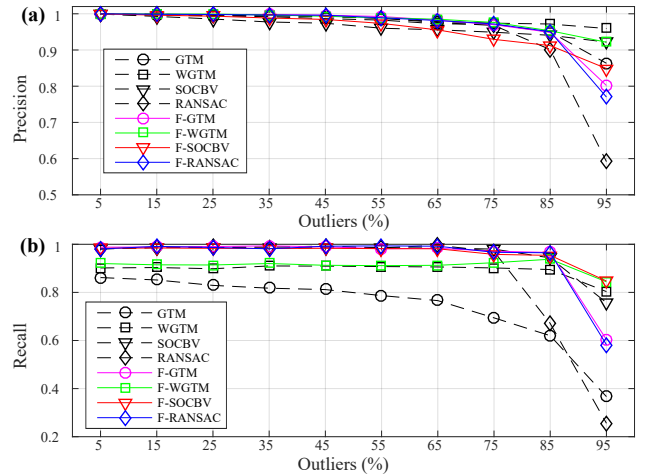


Figure 4. Comparison of precision and recall values for the four methods with/without FOMP preprocessing in the matching ambiguities dataset, considering the number of outliers manually added.

Figure 5 shows a single pair of images, considering 85% of outliers in the initial set of matched points, comparing the methods with/without FOMP. Again, the amount of TP for F-GTM and F-RANSAC is considerable greater. Besides, the

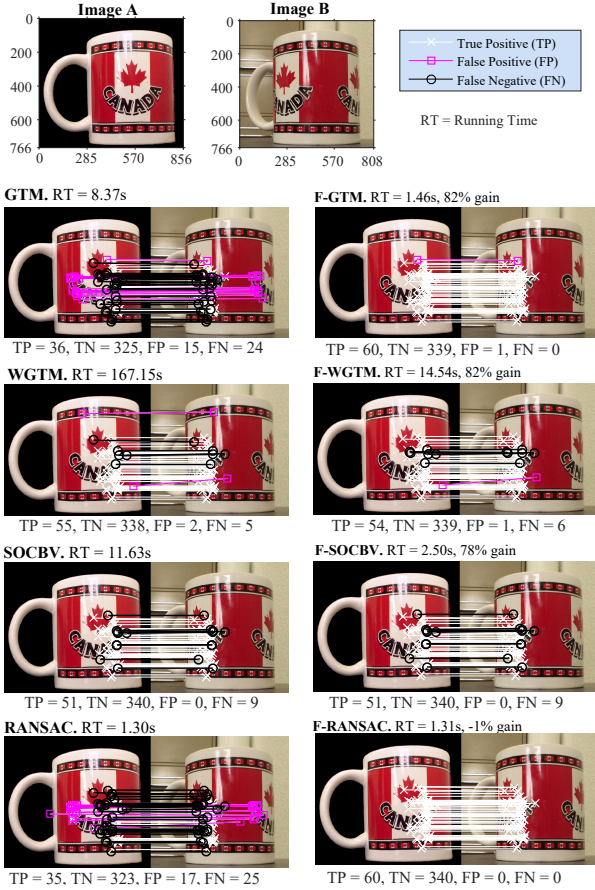


Figure 5. Image matching results for two images from the matching ambiguities dataset (85% of outliers in the initial set of matched points).

amount of FP is also considerable better, leading to a higher precision. For WGTM and SOCBV, the amount of TP and FP is quite identical to F-WGTM and F-SOCBV. However, there is a noticeable processing time gain (in average 80%) with FOMP. There is a running time gain of up to 82% for F-GTM as well. RANSAC presented a really small running time difference from F-RANSAC, 0.01 seconds.

C. Deformable objects

This dataset is composed of 10 pairs of images, which presents deformable objects, such as papers or plastic wrappers. These images have a resolution of 640×480 pixels. In Figures 6(a)-(b) are shown the precision and recall comparison for this dataset, respectively. Once again, from 5% to 85% outliers, there was no significant precision difference among the methods. Notwithstanding, at 95%, the precision is only better for F-RANSAC. Analyzing the recall, F-GTM performed undoubtedly better than GTM for all percentage of outliers. For the other methods, there was no significantly difference from 5% to 85% of outliers. At 95% of outliers, all methods presented a better recall with FOMP.

Figure 7 shows the results for a single pair of images,

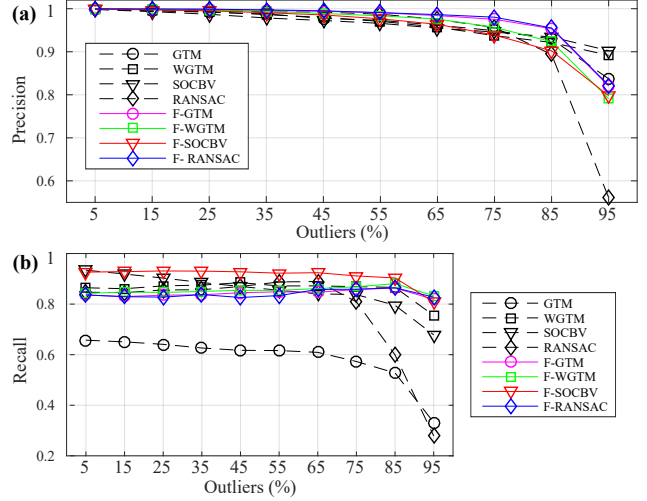


Figure 6. Comparison of precision and recall values for the four methods with/without FOMP preprocessing in the deformable object dataset, considering the number of outliers manually added.

considering 85% of outliers in the initial set of matched points, comparing the methods with/without FOMP. For this example, the number of TP is higher with FOMP. While the amount of FP is smaller for F-WGTM and F-SOCBV, this number is slightly bigger for GTM and RANSAC. The running time is incredible reduced for F-GTM, F-WGTM and F-SOCBV, achieving in average, 72% of gain.

D. Affine transformations

This dataset is composed of 5 pairs of images, which presents affine variations due to viewpoint changes (graffiti dataset [16]). These images have a resolution of 800×640 pixels. Figures 8(a)-(b) show the precision and recall comparison for this dataset, respectively. Analyzing the precision, all methods presented similar results from 5% to 25% outliers. From 35% to 85% outliers, GTM, SOCBV and RANSAC presented similar results compared to F-GTM, F-SOCBV and F-RANSAC, respectively. WGTM presented slightly better results without FOMP. However, at 95% of outliers, all methods presented better precision without FOMP. For the recall, F-GTM performed a lot better than GTM, while WGTM performed better than F-WGTM. Comparing F-RANSAC to RANSAC and F-SOCBV to SOCBV, there were similar values of recall from 5% to 95% outliers. At 95% outliers, RANSAC performed better than F-RANSAC. It is important to highlight that at 95% of outliers, the matching process suffers anyway.

Figure 9 shows an example of a single pair of images, considering 85% of outliers in the initial set of matched points, comparing the methods with/without the FOMP preprocessing. Note that, F-GTM, F-SOCBV and F-RANSAC presented a higher number of TP and a lower number of FP than GTM, SOCBV and RANSAC, respectively.

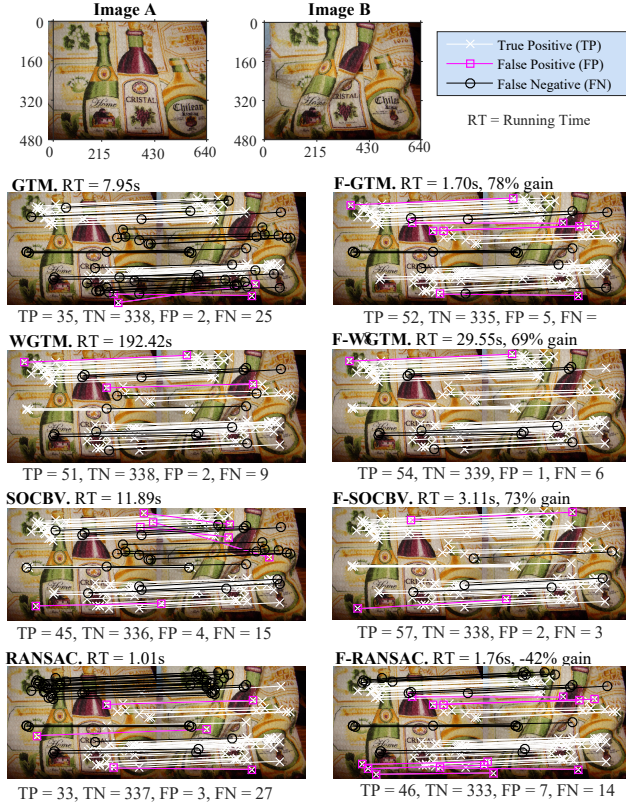


Figure 7. Image matching results for two images from the non-rigid dataset (85% of outliers in the initial set of matched points).

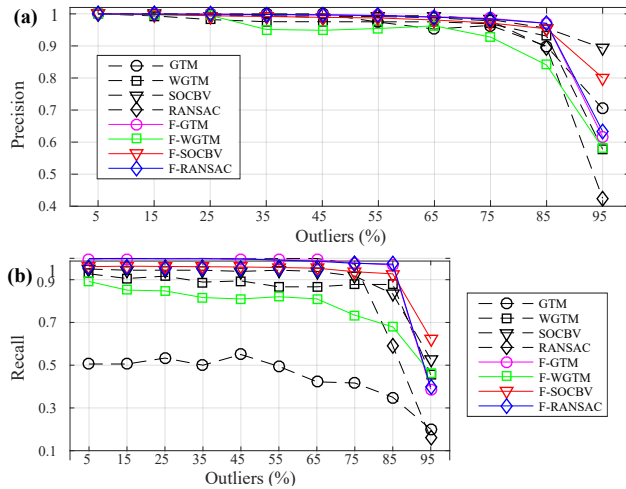


Figure 8. Comparison of precision and recall values for the four methods with/without FOMP preprocessing in the affine variations dataset, considering the number of outliers manually added.

For WGTM the number of TP is the same of F-WGTM, while the number of FP is slightly smaller. The running time gain is considerable better for F-GTM, F-WGTM and F-SOCBV algorithms, in average 77% of gain. In contrast, for F-RANSAC, there was no running time gain. However, the

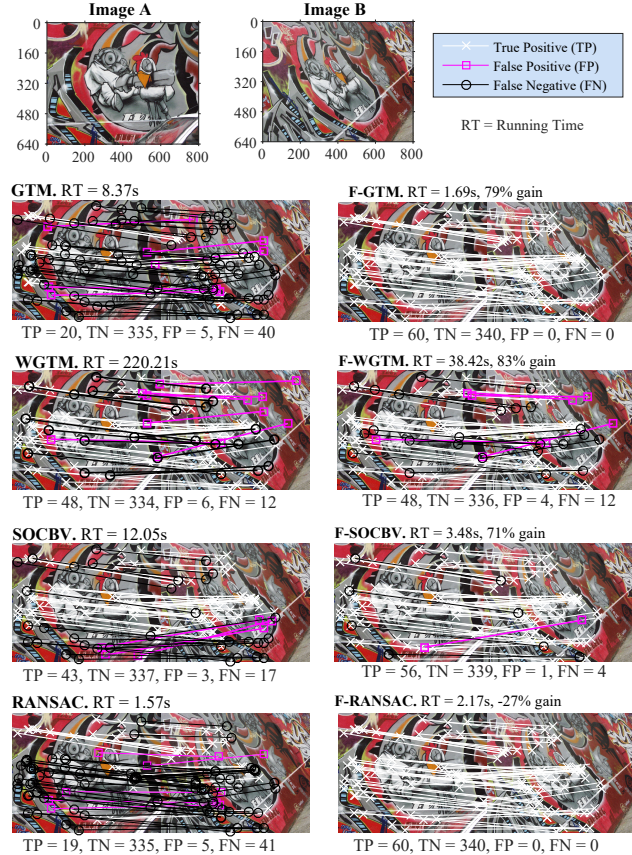


Figure 9. Image matching results for two images from the affine transformation dataset (85% of outliers in the initial set of matched points).

number of TP and FP is much better, giving 100% precision and recall values.

E. Running time

Table I summarizes the average running time gain percentage of all experiments regarding outliers. Note that, FOMP considerable reduced the processing time for GTM, WGTM and SOCBV for all outliers percentages. The average running time gain for F-GTM and F-SOCBV is about 52%, while F-WGTM presented the best running time gain, 98% with a small standard deviation, when compared to the other methods. The running time is faster for RANSAC compared to F-RANSAC, since F-RANSAC does not take advantage of already calculated complete graphs. However, as the experiments have shown in the previous sections, the number of inliers detected with F-RANSAC is considerable larger, which means that the precision and recall of F-RANSAC is improved.

F. FOMP alone analysis

In order to analyze the behavior of FOMP apart from the consensus analysis, we used the measures of recall, specificity, running time and the number of iterations. Table II

Table I
COMPARISON OF AVERAGE RUNNING TIME. THE TIME IS EXPRESSED IN SECONDS AND \pm REPRESENTS THE STANDARD DEVIATION

	GTM	F-GTM	WGTM	F-WGTM	SOCBV	F-SOCBV	RANSAC	F-RANSAC
5	0.03	0.025	1.44	0.080	0.04	0.054	0.007	0.025
15	0.05	0.029	2.12	0.089	0.09	0.080	0.010	0.033
25	0.07	0.042	3.23	0.097	0.15	0.100	0.015	0.038
35	0.10	0.046	4.97	0.107	0.24	0.123	0.023	0.049
45	0.17	0.065	7.68	0.119	0.40	0.171	0.044	0.065
55	0.28	0.095	12.5	0.158	0.68	0.260	0.086	0.100
65	0.53	0.163	21.2	0.210	1.21	0.428	0.124	0.164
75	1.24	0.386	38.4	0.358	2.41	0.846	0.258	0.411
85	4.82	1.423	85.7	0.965	7.53	2.342	0.988	1.400
95	131	15.78	616	16.36	141	31.97	1.33	15.75
Gain	58% \pm 19		98% \pm 1.46		46% \pm 30		-48% \pm 23	

summarizes these four measures for the four datasets (camera movements, matching ambiguities, deformable objects and affine variations). Note that, the recall is equal or higher

Table II
AVERAGE RUNNING TIME, NUMBER OF ITERATIONS, SPECIFICITY AND RECALL FOR ALL EXPERIMENTS USING ONLY FOMP FILTERING. \pm REPRESENTS THE STANDARD DEVIATION

	Running Time (s)	Number of Iterations	Specificity	Recall
5	0.016 \pm 0.01	3 \pm 1	0.69 \pm 0.34	1.00 \pm 0
15	0.019 \pm 0.01	7 \pm 3	0.70 \pm 0.22	1.00 \pm 0
25	0.025 \pm 0.01	12 \pm 5	0.70 \pm 0.18	0.99 \pm 0.003
35	0.037 \pm 0.02	18 \pm 8	0.67 \pm 0.15	0.99 \pm 0.003
45	0.057 \pm 0.03	26 \pm 12	0.67 \pm 0.14	0.99 \pm 0.004
55	0.100 \pm 0.06	38 \pm 18	0.66 \pm 0.11	0.99 \pm 0.006
65	0.171 \pm 0.10	56 \pm 26	0.64 \pm 0.11	0.99 \pm 0.026
75	0.289 \pm 0.17	89 \pm 42	0.62 \pm 0.09	0.98 \pm 0.064
85	0.882 \pm 0.54	162 \pm 75	0.61 \pm 0.07	0.97 \pm 0.081
95	16.83 \pm 10.5	510 \pm 231	0.57 \pm 0.03	0.90 \pm 0.182
Mean	1.84 \pm 6.01	92.3 \pm 166	0.65 \pm 0.17	0.98 \pm 0.07

than 90% for all outliers percentages, which means FOMP preprocessing kept almost all inliers from the initial set of matched points. Furthermore, for small outliers percentage (5% and 15%), FOMP keeps all inliers while filtering out (specificity), in average, 70% of outliers. Analyzing the specificity, in average, FOMP removed 65% of the outliers in the initial set of matched points, which means FOMP filtered out more than half of the outliers. For the running time, from 5% to 85% of outliers, the algorithm runs quite fast, taking less than 1 second.

To analyze the effect of changing the inliers amount in the initial set of matched points, we varied the amount of inliers, starting from 10, incremented in steps of 10, up to 60. Figures 10(a)-(b) show the average recall and specificity for the four datasets. Note that the recall and specificity differences are really small over 20 inliers. When the amount

of inliers is 10, the recall drops over 45% outliers and the specificity assumes the lowest values. Therefore, we assume our filtering approach works better with an amount of inliers higher than or equal to 20.

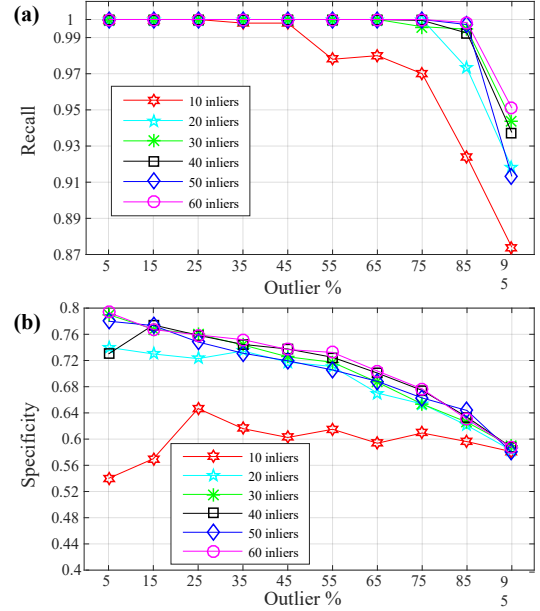


Figure 10. Incrementing the number of inliers. The plots show the average values for all experiments, which sums up to 35 pairs of images, totalizing 7,000 runs for each inlier amount.

G. Discussion and Analysis regarding the α parameter in the FOMP algorithm

The α parameter can be set according to the application goal. For this purpose, we have conducted experiments on each one of the four datasets described in this paper. We varied the α value, starting from 0.1, incrementing in steps of 0.1 until 1. Figures 11(a)-(d) show the amount of remaining inliers (recall) and the number of filtered out (specificity) outliers for each dataset. Note that there is a trade-off between recall and specificity. The larger the value of α , the lower the specificity and the higher the recall. The lower the value of α , the higher the recall and the lower the specificity.

If there is previous knowledge about the images, α can be set so that the recall is less affected. As a rule of thumb, for images with differences of camera movements, ambiguities and non-rigid transformations, low values of α would work fine. However, for images with affine transformations, a value larger than 0.6 would be a better choice, since this value keeps a better balance between precision and recall.

If the goal is to filter out outliers, but at the same time keep as much inliers as possible, the best choice would be $\alpha = 1$. On the other hand, if the goal is to remove the maximum amount of outliers as possible, while keeping at least 50% of the inliers, the best choice would be $\alpha = 0.1$.

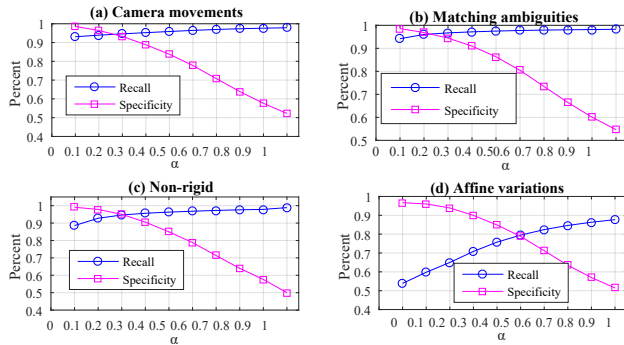


Figure 11. FOMP: Recall and specificity for variations in α value.

IV. CONCLUSION

In this paper, we proposed FOMP, which is a preprocessing filter algorithm that can be used before applying consensus algorithms, such as, GTM, WGTM, SOCBV and RANSAC. FOMP filters out outliers by eliminating points that present higher edges differences based on two complete graphs. The stop criteria depends on the α value, which can be set accordingly to the application goal if there is previous knowledge about the image. Small values of α removes a greater percentage of outliers, while it also removes a few inliers. Larger values of α removes less outliers, maintaining a larger number of inliers. The experimental results showed that by applying FOMP, the precision for consensus algorithms are alike most of the times and the recall, in general, is a lot better by applying FOMP. Analyzing the FOMP filter itself, it kept over 98% of inliers and removed over 65% of the outliers. Moreover, the running time gain is considerable better by employing FOMP before consensus' algorithm, speeding up the process by 50% of the time.

ACKNOWLEDGMENT

The authors would like to thank M. Izadi for providing the GTM and WGTM algorithm and image dataset and F. Meng for providing the SOCBV algorithm. We also would like to thank CNPq, CAPES and FAPESP Brazilian funding agencies for the financial support. Finally, we would like to thank the Image and Database Group¹ (GBDI) for providing the machine to run our experiments.

REFERENCES

[1] K. Loewke, D. Camarillo, W. Piyawattanametha, M. Mandella, C. Contag, S. Thrun, and J. Salisbury, "In Vivo micro-image mosaicing," *Biomedical Engineering, IEEE Transactions on*, vol. 58, no. 1, pp. 159–171, Jan 2011.

[2] C.-L. Chien and W.-H. Tsai, "Image fusion with no gamut problem by improved nonlinear IHS transforms for remote sensing," *IEEE T. Geoscience and Remote Sensing*, vol. 52, no. 1, pp. 651–663, 2014.

[3] D. Ji, R. Yang, L. Zhang, B. Wang, and X. Chen, "The research of medical microscopic image mosaic based on the algorithm of SURF," in *Wavelet Active Media Technology and Information Processing, 2013 10th International Computer Conference on*, Dec 2013, pp. 16–20.

[4] A. Mora, J. Fonseca, and R. Ribeiro, "Real-time image recovery using temporal image fusion," in *Fuzzy Systems (FUZZ), 2013 IEEE International Conference on*, July 2013, pp. 1–5.

[5] M. Fontani, T. Bianchi, A. De Rosa, A. Piva, and M. Barni, "A framework for decision fusion in image forensics based on dempster-shafer theory of evidence," *Information Forensics and Security, IEEE Transactions on*, vol. 8, no. 4, pp. 593–607, April 2013.

[6] X. Benavent, A. Garcia-Serrano, R. Granados, J. Benavent, and E. de Ves, "Multimedia information retrieval based on late semantic fusion approaches: Experiments on a wikipedia image collection," *Multimedia, IEEE Transactions on*, vol. 15, no. 8, pp. 2009–2021, Dec 2013.

[7] T. Collins, P. Mesejo, and A. Bartoli, "An analysis of errors in graph-based keypoint matching and proposed solutions," in *European Conference on Computer Vision (ECCV)*, Springer, Zurich, Switzerland: Springer, 09/2014 2014.

[8] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ser. ICCV '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 1150–1157.

[9] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded Up Robust Features," in *In ECCV*, 2006, pp. 404–417.

[10] 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.

[11] W. Aguilar, Y. Frauel, F. Escolano, M. E. Martinez-Perez, A. Espinosa-Romero, and M. A. Lozano, "A robust graph transformation matching for non-rigid registration," *Image and Vision Computing*, vol. 27, no. 7, pp. 897 – 910, 2009, 7th IAPR-TC15 Workshop on Graph-based Representations (Gbr 2007).

[12] M. Izadi and P. Saeedi, "Robust weighted graph transformation matching for rigid and nonrigid image registration," *Image Processing, IEEE Transactions on*, vol. 21, no. 10, pp. 4369–4382, Oct 2012.

[13] Z. Liu, J. An, and Y. Jing, "A Simple and Robust Feature Point Matching Algorithm Based on Restricted Spatial Order Constraints for Aerial Image Registration," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, pp. 514–527, Feb. 2012.

[14] F. Meng, X. Li, and J. Pei, "A feature point matching based on spatial order constraints bilateral-neighbor vote," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4160–4171, Nov 2015.

[15] P. D. Kovesei, "MATLAB and Octave functions for computer vision and image processing," available from: <<http://www.peterkovesei.com/matlabfn/>>.

[16] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool, "A comparison of affine region detectors," *Int. J. Comput. Vision*, vol. 65, no. 1-2, pp. 43–72, Nov. 2005.

¹<http://www.gbdi.icmc.usp.br/>