Hierarchical Multi-Affine (HMA) algorithm for fast and accurate feature matching in minimally-invasive surgical images

Gustavo A. Puerto-Souza, and Gian Luca Mariottini

Abstract—The ability to find similar features between two views of the same scene (feature matching) is essential in robotics and computer vision. We are here interested in the robotic-assisted minimally-invasive surgical scenario, for which feature matching can be used to recover tracked features after prolonged occlusions, strong illumination changes, image clutter, or fast camera motion. In this paper we introduce the Hierarchical Multi-Affine (HMA) feature-matching algorithm, which improves over the existing methods by finding a larger number of image correspondences, at an increased speed and with a higher accuracy and robustness. Extensive experimental results are presented that compare HMA against existing methods, over a large surgical-image dataset and over several types of detected features.

I. INTRODUCTION

Feature matching consists of finding image similarities between two views of the same scene.

Differently from feature tracking [1]–[3], feature matching does not require any assumption about the sequential nature of the two images (e.g., consecutive frames of a video), and can then be used to recover corresponding features after a prolonged occlusion, a fast camera motion, or a strong illumination change.

Due to their generality, feature-matching algorithms are essential in many robotics and vision applications [4]–[9]. In particular, feature matching is important in robotic-assisted minimally-invasive surgical (MIS) scenarios, e.g., for tissue-shape recovery [10]–[12], camera tracking [13], [14], or for recovering lost features used as anchors for augmented reality [15], [16]. Figure 1(a)-(c) shows a MIS image sequence in which feature matching is used to recover lost tracked features after a prolonged occlusion and organ movement. These points are then used to re-initialize an augmented-reality display of a kidney (Fig. 1(d)).

Feature matching algorithms have been proposed in the recent years [17]–[19] that make limited (or no) assumptions about the two given images, or about the observed scene.

In this paper, we present a novel feature-matching algorithm that improves over the existing methods, by finding a larger number of image correspondences at an increased speed, and with a higher accuracy, and robustness to image clutter. Our method, called Hierarchical Multi Affine (HMA), robustly matches clusters of image features on the observed objects’ surface. HMA adopts hierarchical k-means [20] to iteratively cluster the matching features according to their similarity-space representation. For each of these spatially-distributed clusters, an affine transformation is estimated, to further prune the outliers and to capture features along the entire object surface.

Experimental results are presented over a large and varied MIS image dataset (extracted from six 3-hours-long surgical interventions) and in the case of several detected features (SIFT, ASIFT, and SURF). We show that HMA outperforms the existing state-of-the-art methods in terms of speed, inlier-detection rate, and accuracy.

A. Related work

The first step in feature matching consists of extracting salient features from the two images of the same scene. Several algorithms have been proposed to detect features (SIFT [17], SURF [21], ASIFT [22], etc.) along with their local descriptors, each one with different invariant properties.

After this feature-extraction part, a feature-matching phase follows, which in general consists of two phases; first, an appearance-based matching, in which the local appearance of each feature is used to determine a set of candidate (or initial) matches. The work in [23] compares potential matches based only on their local appearance. Recently, several methods have been introduced [17]–[19] that prune from ambiguities the candidate matches by using an ad-
ditional geometrical-validation step, which leverages the spatial arrangement of features.

The algorithm in [17] matches first SIFT features according to the Euclidean distance between descriptors. Then, a geometric phase is enforced to detect a predominant set of matches (inliers) that satisfy a unique affine-transformation constraint [24]. While this method can reliably discard many wrong matches, however only a limited number of those belonging to a predominant planar surface will be detected.

The above issue was solved in [18], where multiple local-affine mappings are used to detect a larger number of matches over the entire non-planar object surface. However, due to its high computational time, this method can only be used for off-line data processing.

The work in [19] uses a dissimilarity function that measures the matches similarity based on both a geometrical and an appearance constraint. Finally, an agglomerative step generates clusters of matches by iteratively merging similar ones (according to the dissimilarity function). A lack of this algorithm is in the high number of parameters required for the dissimilarity function. In addition, choosing these parameters is difficult because of their unclear physical interpretation.

The HMA algorithm presented in this paper improves over the existing algorithm as described in what follows:

- HMA is fast because of the adoption of hierarchical k-means in the clustering phase. In particular, when a large number of features are detected (e.g., when using HD images or extracting ASIFT features), HMA is almost two-orders-of-magnitude faster than [19].
- HMA is accurate: a large percentage of wrong matches is automatically removed. Furthermore, the cluster of affine transformations guarantees accurate mapping over the entire surface.
- HMA is easy to use: only few threshold values need to be specified by the user, which have an intuitive physical interpretation.

The paper is organized as follows: Sect. II describes the feature matching problem and introduces the basic notation. Sect. III describes the HMA algorithm, while Sect. IV reports the results of our extensive experimental evaluation and the comparison of HMA with the state-of-the-art feature matching methods. Finally, in Sect. V we draw the conclusions and illustrate possible future extensions.

II. OVERVIEW OF THE FEATURE MATCHING PROBLEM

A. Keypoints and Descriptors: Notation

Consider a pair of images, \( I^t \) (training) and \( I^q \) (query), and their two corresponding sets of detected image features \( F^t \) and \( F^q \), respectively. For a generic image \( I \), an image feature (e.g., SIFT) is defined as, \( f \triangleq (k, d) \in F \), consisting of a keypoint, \( k \), and a local descriptor, \( d \). The keypoint \( k \triangleq [x, \sigma, \theta]^T \) contains the feature pixel position, \( x = [x, y]^T \), the scale, \( \sigma \), and the orientation, \( \theta \). The descriptor \( d \) is a vector containing information about the local appearance around the feature position.

Based on the above notation, the feature-matching problem can be defined as retrieving the set of similar feature pairs (correspondences) from all the features extracted in both images. Such a (generic) match \( m \) is defined as \( m \triangleq (f^t, f^q) \), between features in \( F^t \) and \( F^q \).

B. Initial Appearance-based Feature Matching

In feature matching, a set \( M \) of candidate matches is initially determined by means of appearance-based criteria (e.g., the Nearest-Neighbor distance Ratio (NNR)).

In NNR, the best training feature candidate to match the \( i \)-th feature \( f^t_i = (k^t_i, d^t_i) \) with the closest descriptor-distance, i.e.,

\[ j(i) = \arg \min_{j \in \mathcal{F}^q - \{j(i)\}} \|d^t_i - d^q_{j(i)}\|_2, \]

and the smallest ratio between the closest and the second-closest descriptor distances, defined as,

\[ \text{NNR}(i) \triangleq \frac{\|d^t_i - d^q_{j(i)}\|_2}{\min_{j \in \mathcal{F}^q - \{j(i)\}} \|d^t_i - d^q_j\|_2}. \]

This ratio has to be less than some threshold \( \tau_{\text{NNR}} \) which allows to discard many wrong matches that could arise, e.g., due to ambiguity with the background. Typically, a value for the ratio-threshold is \( \tau_{\text{NNR}} \geq 0.8 \) [17].

However, due to image similarities, local appearance only will generate a high percentage of outliers (i.e., wrong matches). As a consequence, the matches obtained from this appearance-based phase are unreliable and might not be sufficient in several applications, such as tracking recovery (see [25] for a performance comparison of appearance-based criteria).

C. Similarity Transformation from Keypoints

Corresponding features in both images can be related by a similarity transformation, which consists of a rotation, translation and scale. In particular, the two matching keypoints \( (k^t_i, k^q_j) \) of a generic match \( m \), can be used to compute such a geometric (rigid) image motion, which models the rotation and translation for a pair of image points \( [u^t, v^t]^T \in \mathcal{T}^t \) and \( [u^q, v^q]^T \in \mathcal{T}^q \), as follows,

\[
\begin{align*}
[u^q & ] \ = \ \delta x_i \cos(\delta \theta_i) - \delta y_i \sin(\delta \theta_i) \\
[v^q & ] \ = \ \delta x_i \sin(\delta \theta_i) + \delta y_i \cos(\delta \theta_i) \\
R(\delta \theta_i) \] & \ = \ \begin{bmatrix}
\delta x_i & \delta y_i \\
\delta y_i & - \delta x_i 
\end{bmatrix} \\
\end{align*}
\]

where the scale \( \delta \sigma_i \triangleq \sigma^q_i / \sigma^t_i \), the 2-D rotation angle \( \delta \theta_i \triangleq \theta^q_i - \theta^t_i \), and the translation \( [\delta x_i, \delta y_i]^T \) are fast to compute from the matched-keypoint parameters.

A match \( m \) is considered to agree with a generic transformation \( \mathbf{A} \), if the symmetric re-projection error of mapping its keypoints is below some threshold \( \tau \), i.e.,

\[
e_s(x^t_i, x^q_i; \mathbf{A}) = 0.5 \|x^t_i - Ax^t_i\| + 0.5 \|x^q_i - A^{-1}x^q_i\| < \tau
\]
III. THE HMA ALGORITHM

A. Overview

Given a set of (appearance-based) candidate matches, \( \mathcal{M} \), (see Sect. II-B), the goal of HMA is to compute (i) a set of outlier-free matches, \( \mathcal{M}^* \), and (ii) a set of spatially-distributed image transformations, \( \mathcal{A}^* = \{ \mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \} \). Each of these transformations has the capability to map features lying on different portions of the training image, to corresponding features in the query image. As illustrated in Fig. 1, our algorithm can then be used to recover any image feature (not necessarily SIFT but also corners) that were lost in \( \mathcal{T}^q \) due to occlusions or rapid camera motion.

HMA achieves these tasks by clustering the initial matches according to a spatial and an affine constraints. We adopt a hierarchical k-means strategy [20] to iteratively cluster all the matching features, as illustrated in Fig. 2. The hierarchical structure of our algorithm, and its corresponding tree structure, present several advantages:

- **Speed**: our divide-and-conquer expansion of the tree can be stopped until a desired accuracy is achieved.
- **Robustness**: the quantization of the k-means clustering will iteratively correct at each level of the tree, by reassigning matches to more appropriate clusters.
- **Accuracy**: The multiple ATs locally adapt more precisely to the non-planar regions of the observed scene than a single global-affine mapping.

Fig. 3 shows all the phases of the HMA algorithm: the clustering phase partitions each node’s matches into \( k \) clusters, \( \mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k \). The affine estimation phase, for each cluster, estimates an affine transformation, \( \mathcal{A}_i \), the sets of inliers, \( \mathcal{M}_i \), and the outliers \( \mathcal{O}_i \). A correction phase is used to reassign matches to more adequate clusters and, if necessary, to obtain an updated affine mapping, \( \mathcal{A}_i' \) (and new sets \( \mathcal{M}_i' \) and \( \mathcal{O}_i' \)). Each cluster defines a new child node as \( \mathcal{N}_{j(i)} \), where \( j(\cdot) \) is the function that determines the index of the cluster \( i \) in the tree structure. Finally, a criteria based on the matches’ reprojection error \( e_a \) is employed to determine if these new nodes require further clustering.

These three phases are repeated until each branch reaches a leaf node. Then, the set of outlier-free correspondences, \( \mathcal{M}^* \), and the set of transformations, \( \mathcal{A}^* \), are extracted from the leaf nodes that satisfy a threshold over the minimal number of inliers, i.e., \( |\mathcal{M}_l| \geq \tau_e \). Once a leaf is reached, a final validation step is used to recover those outliers that might be potential inliers for other nodes.

B. Clustering phase

In this phase the input matches are partitioned into \( k \) subsets by using k-means over an extended feature space, consisting of both the matches’ similarity-transformation parameters, as well as their keypoint position in the query image. The resulting clusters \( \mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k \), are illustrated in the example of Fig. 4. Note that each cluster represents a new branch of the tree at the current level.

1) **Similarity-space clustering**: For all the matches in each node, the four similarity transformation parameters \( [\delta_x, \delta_y, \delta \sigma, \delta \theta]^T \) are first computed (c.f., Sect. II-C). Clustering in this similarity space is the key, since it will result in clusters with close similarity parameters (representing indeed an initial approximation for the AT).
From our experiments, we noticed that this step does not always ensure that the clusters are disjoint, which would be ideal in order to define a unique affine transformation features for that portion of the scene. For this reason, we apply k-means to an extended (six-dimensional) similarity space given by $[x^q, y^q, \delta x, \delta y, \delta \sigma, \delta \theta]^T$.

We also noticed that, in earlier levels of the tree, the translation parameters $[\delta x, \delta y]$ are more descriptive to isolate outliers. On the other hand, in latter levels of the tree, the position parameters $[x^q, y^q]$ become more important than the others, since $[\delta x, \delta y]$ tends to be less informative. Finally, we noticed that $\delta \sigma$ and $\delta \theta$ are not very informative and can be very sensitive to changes of view, in particular when the observed object is non-planar.

Due to the above observations, a scaling vector $W$ was chosen to adaptively weigh the relative importance of those similarity parameters before applying k-means. Our strategy varies $W$ depending on the values of the AT residuals. As from what discussed above, $W$ will put more importance on the translation (than on the other parameters) when the cluster's AT is not adequate to model the matches (e.g., due to the large number of outliers such as in the initial levels of the tree). When the number of outliers is reduced, $W$ will reduce its importance on the translation parameters, and increase its values for the position (thus enforcing spatial contiguity of the clusters).

The weights $W$ are computed by interpolating between two given weigh values $W_i$ (initial), and $W_f$ (final), as follows: $W = W_i + (W_f - W_i)\alpha(\bar{r})$, where $\bar{r}$ represent the average of all the symmetric reprojection errors, and $\alpha$ is a decreasing function with image in the interval $[0, 1]$, defined as,

$$
\alpha(\bar{r}) = \begin{cases} 
1 & \text{if } \bar{r} \leq \tau_r \\
1 - erf\left(\frac{\bar{r} - \tau_r}{\sqrt{2} \sigma_w}\right) & \text{otherwise}
\end{cases},
$$

where $\tau_r$ is the mapping maximum error threshold, and $\sigma_w$ is a parameter which smooths the shape of $\alpha$. In Fig. 5 we can see an example of the plot of the $\alpha$-function.

Finally, these weights are used to scale the dataset $Q = \{x^q, y^q, \delta x, \delta y, \delta \sigma, \delta \theta\}W$, which is then clustered by k-means.

In order to eliminate from each node subsets of isolated matches with large residuals\footnote{These matches could negatively affect subsequent clustering phases.}, we adopt a non-parametric outlier-detection technique [26]. In particular, a threshold $\tau_o$ is computed from the statistics of the symmetric reprojection errors, in order to discard matches with an error larger than $\tau_o$. This threshold is computed as $\tau_o \triangleq q3 + \gamma(q3 - q1)$, where $q_i$ is the $i$-th quartile of the error over all the matches, and $\gamma$ is a factor\footnote{$\gamma$ has been fixed to 3/2}.

**C. Affine-estimation phase**

For each cluster, the keypoints $(k^i, k^q)$ are used in the affine-transformation (AT) estimation phase to compute $A_i$. RANSAC is used, for a given pixel error threshold $\tau_{R}$, to provide a robust estimate of $A_i$, as well as to obtain the inliers $M_i$. The affine model $A_i$ maps a generic image point $[u^i, v^i]^T \in I^i$ (in the corresponding cluster $C_i$) to $[\hat{u}^q, \hat{v}^q]^T \in I^q$, according to:

$$
\begin{bmatrix} \hat{u}^q \\ \hat{v}^q \end{bmatrix} = \begin{bmatrix} m_1 & m_2 & u^i \\ m_3 & m_4 & v^i \\ m_5 & m_6 & 1 \end{bmatrix}.
$$

(3)

The parameters in $A_i$ represent the translation, rotation, and shear associated with the AT. The vector $m = [m_1, m_2, \ldots, m_6]$ can be estimated at each RANSAC iteration by randomly picking triplets of matches and by using them to solve for $m$ in a least-squares sense from (3). Fig. 6 shows an example of the AT and the set of inliers. The colored boxes indicate how each region is mapped between images. As observed, these regions nicely capture keypoints according to the slope of their local surface.

In the correction phase the outliers $O_i$ are reassigned to other, more appropriate) clusters and, if necessary, the ATs

\[2\text{This six-dimensional dataset is normalized (each column vector will have norm one) and scaled (with respect to each column).}\]

\[3\text{In fact, spatial contiguity of the clusters has higher priority}.\]
are recomputed with the new set of features. In particular, Linear Discriminant Analysis (LDA) [27] is used to separate features belonging to all the known classes (clusters) in the query image, thus guaranteeing non overlap among the final clusters.

E. Stop Criteria

In order to stop the expansion of a \( i \)-th node, and to deem it as a leaf node, HMA examines the number of inliers (consensus), \(|M_i|\), and the ratio of inliers, \( \rho_i \triangleq |M_i|/(|M_i| + |O_i|)\). In particular, a node is deemed as a leaf when \(|M_i| < \tau_c \) or \( \rho_i > \tau_p \). Note that these thresholds have indeed a very intuitive interpretation and were fixed for our experiments (c.f., Sect. IV).

IV. EXPERIMENTAL RESULTS AND DISCUSSION

We validated the performance of the HMA algorithm in two scenarios: (i) a fully controlled in-lab experiment, and (ii) a real MIS scenario, composed of almost 100 images extracted from 6 surgical interventions. For each image, we measured the inlier ratio\(^6\) that is descriptive of the detection power of our algorithm, the mapping accuracy over a set of manually-labelled ground-truth correspondences (per image), and the computational time. The computational time is measured in seconds of timer CPU\(^7\) required by each run of each algorithm.

In addition, we compared HMA against other recent algorithms: Lowe’s [17], AMA [18], and Cho’s [19], when using either SIFT [17], SURF [21], or ASIFT [22] features.

Our implementation of HMA used the following parameters: \( \tau_R = 5 \) [pixels], \( W_i = [0.2, 0.2, 0.25, 0.25, 0.05, 0.05]^T \), \( W_f = [0.5, 0.5, 0, 0, 0, 0]^T \).

In order to provide a fair comparison between the aforementioned algorithms, we used the same sets of initial matches\(^8\). We consider a threshold of 5 pixels for AMA and Lowe, while a cutoff value of 25 in the linkage function was chosen for Cho.

A. In-lab experiment

In this experiment a pair of images with resolution 640 × 480 was used. The initial matches are 354, 330 and 4903 for SIFT, SURF, and ASIFT, respectively. The ground-truth corresponding corners are 41 manually-selected matches.

Table I provides detailed statistical results of the algorithms’ performance for all the features. The second, fifth and eighth columns show the mean and standard deviation of the symmetric reprojection error, the third, sixth and ninth indicate the computational time (in seconds), and the forth, seventh and tenth contain the percentage of inliers.

Discussion: From the results in Table I we observe that HMA achieves better results than AMA in terms of speed (approximately 11, 19, and 22 times faster for SIFT, SURF, and ASIFT, respectively). While HMA requires slightly higher computational time when compared to Cho (in SIFT and SURF), however HMA obtains a reduced reprojection error. Furthermore, when using ASIFT features (in general more than 4000 are detected per image), HMA requires significant less computational time when compared to Cho.

B. Surgical-images dataset

Each laparoscopic image has a resolution of 480 × 704 pixels, and the set of ground-truth correspondences contains an average of 20 points. Table II summarizes the results obtained by the four feature-matching algorithms.

Discussion: HMA achieves comparable accuracy than AMA, but with a significantly lower computational effort. In fact, HMA is approximately 21 times faster than AMA for both SIFT and SURF, and 66 times for ASIFT. Furthermore, HMA demonstrated a smaller reprojection error than Cho’s, with a comparable time (note Cho’s large standard deviation) for SIFT and SURF. As from the previous results, HMA shows a dramatic low computation time (and low accuracy cost) with a high number of ASIFT matches. Note that HMA has an inlier ratio comparable with other methods. Finally, Fig. 7 shows the boxplots of the average reprojection errors. HMA achieves high stability over all the experiment’s image pairs. These results suggest that HMA formulation achieves a good balance between speed and accuracy.

![Symm. reproj. err. (SIFT)](image1)

![Symm. reproj. err. (SURF)](image2)

![Symm. reproj. err. (ASIFT)](image3)

Fig. 7. Boxplots of the reprojection errors.

Fig. 8 and Fig. 9 show qualitative examples of the matching performance of the four algorithms for SIFT and ASIFT. Note that Lowe is not able to fully adapt to the non-planar surface in both cases, while AMA, Cho, and HMA obtain similar results (covering most of the organ’s surface).

V. CONCLUSIONS

In this work we have been presented a novel feature-matching algorithm, HMA, and applied to robotic-assisted minimally-invasive surgery (MIS). The MIS scenario is challenging because laparoscopic images have a lot of clutter, non-planar structures, changes in viewpoint and organ deformations. HMA improves over existing methods by achieving a lower computational time, and a higher accuracy and robustness to identify many matches. This comparison was
made over a very large (annotated) MIS-image dataset, and over three most-popular feature descriptors (SIFT, SURF, and ASIFT). As future work we plan to include fusion of descriptors and to further improve HMA computational time.

TABLE I

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Lowe</td>
<td>22.90 ± 24.67</td>
<td>20.77 ± 22.15</td>
<td>19.68 ± 20.38</td>
<td>0.38</td>
<td>0.40</td>
<td>0.44</td>
<td>27.7%</td>
<td>34.6%</td>
<td>34.3%</td>
</tr>
<tr>
<td>AMA</td>
<td>4.37 ± 6.18</td>
<td>4.45 ± 7.38</td>
<td>2.48 ± 2.76</td>
<td>7.98</td>
<td>13.00</td>
<td>133.47</td>
<td>38.3%</td>
<td>69.4%</td>
<td>76.2%</td>
</tr>
<tr>
<td>Cho</td>
<td>6.33 ± 8.89</td>
<td>6.92 ± 6.83</td>
<td>9.37 ± 23.29</td>
<td>0.58</td>
<td>0.56</td>
<td>935.18</td>
<td>39.9%</td>
<td>66.6%</td>
<td>40.6%</td>
</tr>
<tr>
<td>HMA</td>
<td>4.30 ± 4.75</td>
<td>4.44 ± 6.66</td>
<td>3.00 ± 2.84</td>
<td>0.73</td>
<td>0.67</td>
<td>595</td>
<td>39.3%</td>
<td>66.7%</td>
<td>73.0%</td>
</tr>
</tbody>
</table>

TABLE II

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Lowe</td>
<td>4.66 ± 4.24</td>
<td>4.06 ± 3.64</td>
<td>4.77 ± 4.25</td>
<td>0.37 ± 0.16</td>
<td>0.36 ± 0.06</td>
<td>1.83 ± 1.57</td>
<td>34%</td>
<td>28%</td>
<td>35%</td>
</tr>
<tr>
<td>AMA</td>
<td>2.49 ± 2.42</td>
<td>3.11 ± 3.57</td>
<td>2.57 ± 2.85</td>
<td>5.74 ± 4.46</td>
<td>11.16 ± 5.55</td>
<td>162.43 ± 136.88</td>
<td>40%</td>
<td>34%</td>
<td>46%</td>
</tr>
<tr>
<td>Cho</td>
<td>3.56 ± 3.35</td>
<td>4.44 ± 5.06</td>
<td>7.56 ± 11.16</td>
<td>0.31 ± 0.12</td>
<td>0.41 ± 0.17</td>
<td>116.10 ± 162.39</td>
<td>39%</td>
<td>39%</td>
<td>31%</td>
</tr>
<tr>
<td>HMA</td>
<td>2.84 ± 2.64</td>
<td>3.70 ± 3.45</td>
<td>3.62 ± 3.55</td>
<td>0.27 ± 0.1</td>
<td>0.51 ± 0.13</td>
<td>2.44 ± 1.11</td>
<td>39%</td>
<td>32%</td>
<td>50%</td>
</tr>
</tbody>
</table>

REFERENCES


