Improved Object Matching Using Structural Relations

Estephan Dazzi[1,2,⋆], Teofilo de Campos[2], and Roberto M. Cesar Jr.[1]

[1] Instituto de Matemática e Estatística - IME,
Universidade de São Paulo - USP, São Paulo, Brasil
[2] CVSSP, University of Surrey, Guildford, GU2 7XH, UK
{estephan.dazzi,roberto.cesar}@vision.ime.usp.br,
t.decampos@st-annes.oxon.org

Abstract. This paper presents a method for object matching that uses local graphs called keygraphs instead of simple keypoints. A novel method to compare keygraphs was proposed in order to exploit their local structural information, producing better local matches. This speeds up an object matching pipeline, particularly using RANSAC, because each keygraph match contains enough information to produce a pose hypothesis, significantly reducing the number of local matches required for object matching and pose estimation. The experimental results show that a higher accuracy was achieved with this approach.

Keywords: Local feature matching, SIFT, hierarchical k-means tree, RANSAC, graph-based structural information.

1 Introduction

An important problem in computer vision is object recognition through matching, which consists in localising objects in test images and estimating their 3D pose. Solutions to this problem are useful in different application domains, such as robotics, medical images analysis and augmented reality. One of the most succesful approaches for this problem involves establishing correspondences between interest points (keypoints) in test and training images; next, a pose estimation algorithm is used, e.g. based on RANSAC, which operates by removing outliers that do not conform with global pose parameters. In this paper, we present a novel method that is capable of providing more accurate solutions besides being computationally cheaper. Our method is generic, in the sense that it can be used with any keypoint extractor method; we validate it using SIFT features [1].

In keypoint-based object recognition, point-to-point correspondences are obtained by matching discriminative features and reducing the set of matches in a

⋆ We would like to thank FAPESP (grant 2011/50761-2), CNPq, CAPES (process 14745/13-5) and NAP eScience - PRP - USP for the support. During most of the production of this paper, T. de Campos had been working in Neil Lawrence's group at the DCS, University of Sheffield, 211 Portobello, Sheffield, S1 4DP, UK.

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 444–453, 2014.

post-processing step. For instance, the ratio test [1] compares the distances to the first and the second nearest neighbor and only establishes a match if the former is significantly smaller than the latter. However, usually there are locations on manmade objects that have similar local appearances, thus discriminative matching may prevent features with similar descriptors from being matched, which becomes a problem particularly when matching keypoints coming from different images.

In our work, initially, we also produce matches solely based on photometric information, but we allow a large number of matches to be established. Then, aiming to eliminate most of the incorrect matches, we use *structural information* within the images; this is done by establishing matches between small sets of keypoints, which we treat as graphs. In this way, our method produces a better set of keypoint matches, even when there are locations on the objects with similar appearances. Not only those matches have a high probability of being correct, but this also benefits the next stage, based on RANSAC, which ends up using a small set of graph correspondences.

Differently from previous approaches that model an image as a global graph and then proceed by employing graph matching methods, such as the work of Sirmacek and Unsalan [2] which uses SIFT keypoints as graph vertices or the work of McAuley and Caetano [3], our approach is local: we decompose the scene and pattern into collections of local graphs and perform only local graph matching, leaving the global matching to the RANSAC procedure. We built upon insights from the work of Morimitsu et al. [4], which focused on fast object detection using a single training image. For that, they used a graph edge descriptor based on Fourier transform and explicitly stored several structures obtained from the training image, which are matched to similar structures found in the test image. In the present paper, we focus on an object recognition task in which there are several images per training object and also many objects stored. We use a more discriminative keypoint extractor (SIFT), and since it is not computationally feasible to explicitly store structures found in the (many) training images, we develop a strategy based on quickly evaluating, during execution time, different aspects of strucures within test and training images.

2 Methodology

The first step of our object recognition process involves extracting SIFT keypoints from all the training images. We use the ground-truth segmentation to eliminate keypoints that are not on the object. We store all the training keypoints in a global indexing structure, which allows to quickly find the approximate nearest neighbors of a query (test) keypoint. We chose to use the hierarchical k-means tree proposed by Muja and Lowe [5] due to its efficiency. For each SIFT keypoint extracted from a training image we store its normalized descriptor (a 128-D feature vector), its scale, its orientation, an identifier of its source image and its x, y position in that image.

Matching of a test object is done following a pipeline of three stages. First, photometric information is used: each SIFT descriptor of the test image runs through the hierarchical k-means tree, producing many matches to the keypoints of the training images. In the second stage, most of the incorrect keypoint matches are eliminated using structural information within images. The strategy consists in substituting the matches previously established between one-to-one keypoints by matches established between small sets of keypoints, i.e., graphs, called *keygraphs* [4]. A keygraph is a graph whose vertices are keypoints, and whose edges carry structural information about its keypoints. The third stage of the matching process consists in using a modified RANSAC (Random Sample Consensus) algorithm, which employs matches established between keygraphs.

2.1 Keypoint Matching

SIFT keypoints are often located very close to each other and this can lead to poor pose estimation results with minimal sets. We select a maximal subset $\mathcal{S}$ of keypoints in the test image such that the distance, in pixels, between any two keypoints in $\mathcal{S}$ is above a threshold d_{pix}; we use $d_{pix} = 10$ pixels.

After selecting the set $\mathcal{S}$ of keypoints in the test image, we match them to the keypoints of the training images, which are stored in a hierarchical k-means tree. We let each test keypoint establish a match with at most *two* keypoints of *each* training image. In order to establish a match between keypoints, it is necessary that the Euclidean distance between their (normalized) SIFT descriptors is below a threshold t; we set t with a relatively high value, as the next stage eliminates possibly incorrect matches. If a test keypoint can establish more than two matches with a same training image, only the two closest matches are kept.

2.2 Keygraph Matching

A *keygraph* is defined as a graph $G = (V, E)$, where the vertex set V is composed of keypoints, and E is the set of graph edges. All the keypoints in a keygraph are present in the same image. Every keygraph has the same number of vertices, κ, and it consists in an *oriented circuit in the clockwise direction*, $G = (v_1, v_2, \ldots, v_\kappa)$.

Each keygraph in the test image can establish matches with keygraphs in every training image. Let $G = (v_1, v_2, \ldots, v_\kappa)$ and $H = (w_1, w_2, \ldots, w_\kappa)$ be keygraphs in a test and in a training image, respectively. The existence of a match between G and H, denoted as $\mathcal{M} = (G, H)$, implies κ matches between the keypoints (vertices) of G and H. For instance, (G, H) may imply the set of keypoints matches $\mathcal{M} = \{(v_1, w_1), (v_2, w_2), \ldots, (v_\kappa, w_\kappa)\}$, i.e., it implies the occurrence of κ matches between pairs of keypoints.

Obtaining Keygraphs in the Test Image. We begin with the subset $\mathcal{S}$ of keypoints in the test image and execute the Delaunay Triangulation, generating a set of triangles, i.e. we use keygraphs with $\kappa = 3$ vertices, $G = (v_1, v_2, v_3)$, represented as triangles whose edges are oriented in the clockwise direction.

Obtaining Keygraphs in the Training Images The keygraphs in the training images are not obtained using the Delaunay Triangulation. Instead, we first calculate the potential keygraph matches that may occur from the test image to each training image. Then we analyse which of those potential keygraph matches imply a valid keygraph in the training image.

Let $G = (v_1, v_2, v_3)$ be a keygraph in a test image, obtained using the Delauney Triangulation. For each training image, we verify whether G establishes keygraph matches with that image. As an illustration, consider the case in which every keypoint of G, v_1, v_2 and v_3, establishes two matches with keypoints of a same training image; then there are *eight* different possible matches between G and keygraphs of that training image: choose one of the two matches of v_1 *and* choose one of the two matches of v_2 *and* choose one of the two matches of v_3. Considering that the keypoint matches are (v_1, w_1), (v_1, w_2), (v_2, w_3), (v_2, w_4), (v_3, w_5) and (v_3, w_6), at most eight sets of keypoint matches (i.e. keygraph matches) can be established:

$$\mathcal{M}_1 = \{(v_1, w_1), (v_2, w_3), (v_3, w_5)\},\ \mathcal{M}_2 = \{(v_1, w_1), (v_2, w_3), (v_3, w_6)\},$$
$$\mathcal{M}_3 = \{(v_1, w_1), (v_2, w_4), (v_3, w_5)\},\ \mathcal{M}_4 = \{(v_1, w_1), (v_2, w_4), (v_3, w_6)\},$$
$$\mathcal{M}_5 = \{(v_1, w_2), (v_2, w_3), (v_3, w_5)\},\ \mathcal{M}_6 = \{(v_1, w_2), (v_2, w_3), (v_3, w_6)\},$$
$$\mathcal{M}_7 = \{(v_1, w_2), (v_2, w_4), (v_3, w_5)\} \text{ and } \mathcal{M}_8 = \{(v_1, w_2), (v_2, w_4), (v_3, w_6)\}.$$

Each one of those keygraph matches requires the existence of a specific keygraph in the training image; for instance, $\mathcal{M}_1 = \{(v_1, w_1), (v_2, w_3), (v_3, w_5)\}$ requires $H_1 = (w_1, w_3, w_5)$ in the training image. As we assume that mirroring is not a possible distortion of the test image, the circuit of a keygraph H in a training image must be oriented in the clockwise direction; if it is oriented in the counter-clockwise direction then H and the tentative keygraph match involving H are not accepted[1].The set of possible keygraph matches $\mathcal{M}_1$, $\mathcal{M}_2$, …, $\mathcal{M}_8$ established by a test keygraph $G = (v_1, v_2, v_3)$ with a training image can also be reduced if v_1, v_2 or v_3 establish fewer keypoint matches with that image; naturally, this set becomes empty if v_1, v_2 or v_3 does not establish any match or none of the implied circuits is in the clockwise direction.

Discarding Keygraph Matches Using Structural Relations. After obtaining a set of (at most eight) tentative keygraph matches between a test keygraph G and keygraphs in a training image, we use five additional tests aiming to eliminate incorrect keygraph matches. This is very effective: the total number of keygraph matches is reduced in orders of magnitude. The tests are based on photometric and structural information within the keygraphs.

To illustrate the tests, let $G = (v_1, v_2, v_3)$ be a keygraph in the test image and $\mathcal{M} = \{(v_1, w_1), (v_2, w_2), (v_3, w_3)\}$ be a tentative keygraph match implied by G in a training image, such that $\mathcal{M}$ requires the existence of the keygraph $H = (w_1, w_2, w_3)$ in that training image.

[1] We assume that training objects are convex or that there is, in the training set, at least one viewpoint of the object where this assumption is valid.

The first test is based on the edges of a training keygraph. In $H = (w_1, w_2, w_3)$, there are three edges: $e^H_{1,2} = (w_1, w_2)$, $e^H_{2,3} = (w_2, w_3)$ and $e^H_{3,1} = (w_3, w_1)$, whose lengths in pixels in the training image are denoted as, respectively, $|e^H_{1,2}|$, $|e^H_{2,3}|$ and $|e^H_{3,1}|$ (an edge is a straight line connecting two vertices). This first test verifies whether the edges length respect a minimum and a maximum value: we use 10 and 100 pixels, i.e. this test verifies whether $10 \leq |e^H_{i,j}| \leq 100$. As the keygraphs in the test image in general have edges with a length equal to or slightly greater than $d_{pix} = 10$ pixels, this test allows the objects to appear in the test image considerably smaller than the (large) object in the training image.

The second test is based on the ratio of edges length in corresponding keygraphs. Considering the tentative match $\mathcal{M}$ between the keygraphs $G = (v_1, v_2, v_3)$ and $H = (w_1, w_2, w_3)$, the three ratios between the length of corresponding edges are $r_{ij} = |e^G_{i,j}|/|e^H_{i,j}|$. This test verifies whether the larger ratio, r_{ij}, is at most twice the smaller one, r_{kl}, i.e. $r_{ij} \leq 2r_{kl}$. This test still allows the occurrence of a large variation between the viewpoints of the object in the test and the training images, but since many training images are taken around an object, a very drastic viewpoint change is not allowed to occur.

The third test is based on the ratio of the scale of corresponding SIFT keypoints. The motivation of this third test is similar to that of the second one. In the test keygraph $G = (v_1, v_2, v_3)$, consider that the scale of the SIFT keypoint v_1 is s^G_1 and similarly we have the scales s^G_2 for v_2 and s^G_3. In a similar way, for the training keygraph $H = (w_1, w_2, w_3)$ we have the scales s^H_1, s^H_2 and s^H_3. Thus the three ratios between the scale of corresponding keypoints are $r_1 = s^G_1/s^H_1$, $r_2 = s^G_2/s^H_2$ and $r_3 = s^G_3/s^H_3$. Similarly to the second test, this third test verifies whether the larger ratio is at most twice the smaller one.

The fourth test is based on both edges and scales. It uses results calculated in the second and the third tests: the ratios between edges length r_{12}, r_{23} and r_{31} and the ratios between scales r_1, r_2 and r_3. Ideally, the value $E = r_{12} + r_{23} + r_{31}$ would be similar to the value $S = r_1 + r_2 + r_3$, as the change in the object size and viewpoint from the training image to the test image should impact similarly the edges length and the SIFT scale. However, as imprecisions can occur, we let the values E and S differ: this fourth test verifies whether the larger value is at most 50% greater than the smaller value, i.e., if $E > S$ this test verifies whether $E \leq 1.5S$ and if $S > E$ it verifies whether $S \leq 1.5E$.

The fifth test uses the orientation (angle) of SIFT keypoints. One of the three pairs of matched keypoints is selected, and the variation of angle between the test and the training keypoints is calculated; then, for the other two keypoint pairs, this variation is applied and it is verified whether the resulting angle is within a margin of error of 45 degrees from the original SIFT orientation. The test suceeds if both keypoint pairs agree with the angle variation implied by the first keypoint pair. If the test fails using a keypoint pair to calcule the angle variation, it can be evaluated again using the other two keypoint pairs to calculate the angle variation: it must suceed for at least one of the three pairs. For example, in the tentative match $\mathcal{M}$ of keygraphs $G = (v_1, v_2, v_3)$ and $H = (w_1, w_2, w_3)$, the pair (v_1, w_1) is used to calculate the angle variation. The angle of v_1 is 0° and the angle of w_1 is 30°, i.e. from v_1 to w_1 occurs an increasing of 30°. Now, this angle variation (+30°) is verified with the other two pairs of keypoints. The angles of v_2 and w_2 are, respectively, 40° and 80°; applying the variation of +30°, we obtain that the angle of w_2 should be 70° (40° plus 30°), which is within the margin of error of 45°, as the true orientation of w_2, 80°, is just 10° above the 70° implied by the first keypoint pair. A similar verification is made by applying the variation of +30° to the pair (v_3, w_3). The whole evaluation can also be made using the angle variation from v_2 to w_2 or the angle variation from v_3 to w_3. We use a large margin of error of 45 degrees which allows the occurrence of imprecisions but still avoids the establishment of absurd keygraph matches.

Figure 2.2 illustrates the establishment of keygraph correspondences.

2.3 Third Stage: RANSAC on Keygraphs

One keygraph match generates $\kappa = 3$ keypoint matches. In the experiments in this paper, we use an affine transformation to instantiate an object, thus *one* keygraph match is necessary to instantiate an affine transformation. Compared to the tradicional RANSAC approach, which would require the random selection of three independent keypoint matches, the keygraph method requires the verification of a smaller number of poses.

Let $\mathcal{G}$ be the set of all keygraph matches between the test image and a training image, $\mathcal{G} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_{|\mathcal{G}|}\}$, in which $\mathcal{M}_i$ is a set of three keypoint matches; thus the set $\mathcal{P}$ of *keypoint* matches between those images is $\mathcal{P} = \mathcal{M}_1 \cup \mathcal{M}_2 \cup \ldots \cup \mathcal{M}_{|\mathcal{G}|}$. To evaluate the quality of an affine transformation which instantiates, in the test image, the object present in that training image, we count the number of keypoint matches that agree with it: for each keypoint in the training image, let x, y be its position in the test image as established by the keypoint match, and let x', y' be its position in the test image as predicted by the affine transformation under evaluation. If the distance between x, y and x', y' is below three pixels, we consider that this keypoint match agrees with the transformation. If at least six keypoint matches agree with a transformation (i.e. the three matches used to instantiate it plus three other matches), we consider that a correct pose of the object is found, and the algorithm returns this affine transformation. If more than one solution is found for a test image, the algorithm returns the one with more matches agreeing with it.

3 Experiments and Results

In our experiments we use a challenging object recognition dataset which contains ten different types of common household objects. For each object type, there are 25 training images taken around the object and 50 test images in which the object appears in a cluttered, realistic scene (in half of them there is one object instance, in the other half, two instances). This dataset was produced and made available by Hsiao et al. [6]. The authors evaluated it in a 3D object recognition task, in which a 3D model was created for each training object.

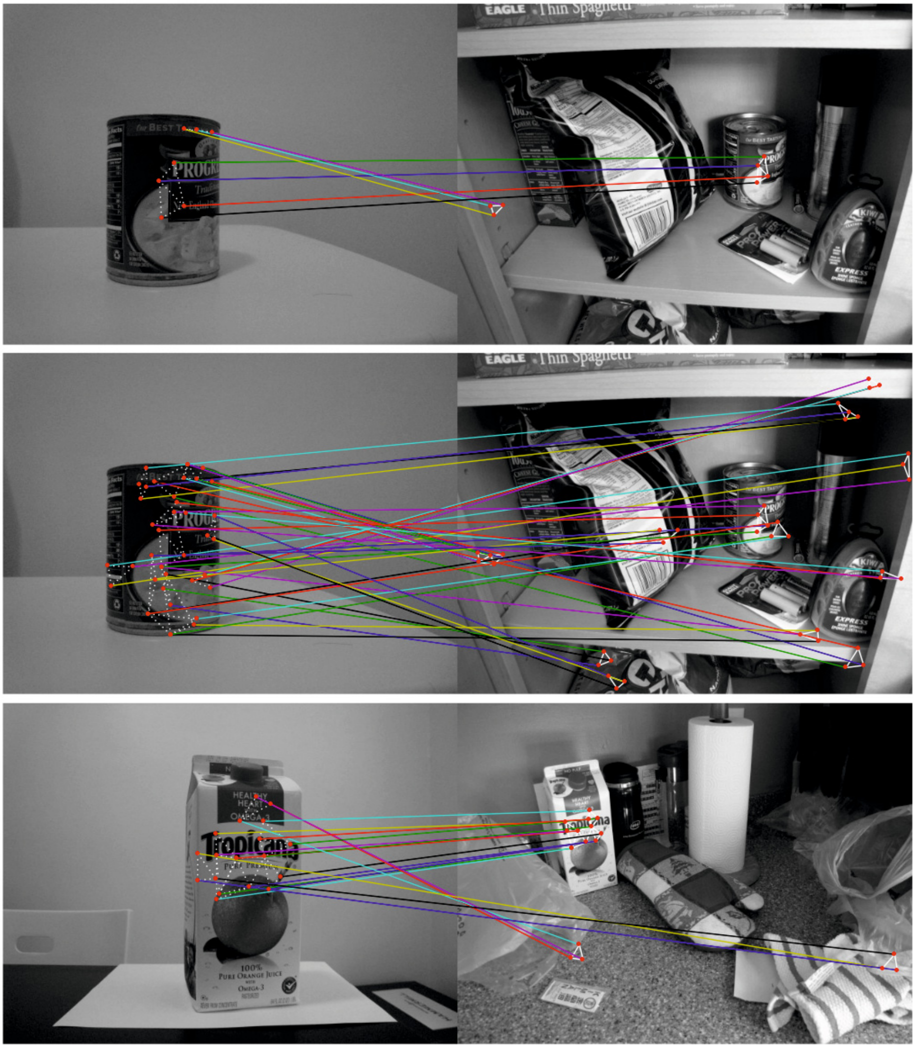

Fig. 1. Keygraph matches established between training (left side) and test (right side) images. Top: "clam chowder can" object: after using the five structural tests, three keygraph matches remain, where two of them are correct. Middle: for the same pair of images of the previous example, we show the keygraph matches that remain after using only four of the five tests; the fifth test, which uses SIFT angle, is not used. It can be seen that the number of (wrong) keygraph matches increases significantly. Bottom: "orange juice carton" object: after using the five tests, we obtain more correct keygraph matches than incorrect ones.

In the present paper, we follow a simpler approach, in which we recognize objects by using an affine transformation between a test and a training image.

We compare the keygraph method proposed in this paper to the ratio test approach [1]. We run each SIFT descriptor v of the test image through the hierarchical k-means tree [5] and obtain the nearest neighbor of v, which is at a distance d_1 from v, and the second nearest neighbor of v that is of a *different* object type than the first nearest neighbor, which is at a distance d_2 from v; a match is established between v and its nearest neighbor if $d_1/d_2 \leq 0.8$ [1]. Then, for every training image for which there are at least four keypoint matches to the test image, we use an exhaustive version of RANSAC, evaluating every possible combination of three keypoint matches to instantiate an affine transformation and then count the number of keypoint matches that agree with this pose. We consider that an instance of an object is found when at least four keypoint matches agree with a transformation (i.e. the three matches used to instantiate it plus one match). We use only four matches for the ratio test method, while for the keygraph method we use six matches (as explained in section 2.3), because the former usually produces fewer keypoints matches than the latter.

We use the same hierarchical k-means tree (with $k = 16$) for both methods, keygraph and ratio test. A query (test) keypoint is compared to a total of 4000 training keypoints stored in the tree leaves; the use of a smaller number of comparisons lowers the accurary of both methods. On average, a training image is described by 1080 SIFT keypoints (using the ground-truth segmentation) and a test image is described by 1070 keypoints (selected to compose the maximal subset $\mathcal{S}$ of keypoints). SIFT descriptors are normalized for zero mean and unit standard deviation; this normalization is useful because we use a threshold $t = 14$ for establishing keypoint matches in the first stage of the keygraph method.

We also compare our keygraph approach to the modified ratio test proposed by Hsiao et al. [6]. Aiming to establish more keypoint matches, the authors proposed to use the regular ratio test in conjunction with a modified ratio test which establishes discriminative matches with *clusters* of keypoints, such that establishing a match with a cluster produces matches to all the training keypoints in that cluster. For that, we create two additional hierarchical k-means trees (with $k = 16$ and $k = 32$). For each test keypoint, for each additional tree we verify whether that test keypoint establishes a discriminative match with a cluster composed of original training keypoints (i.e. a cluster that stores tree leaves); a discriminative match is established if $d_1/d_2 \leq 0.8$, in which d_1 is the distance to the closest cluster and d_2 is the distance to the second nearest cluster. We also use the traditional ratio test (using the original tree with $k = 16$), and employ all the established keypoint matches. Since the modified ratio test generates more keypoint matches then the ratio test, we consider that an instance of an object is found when at least six keypoint matches agree with a transformation, similarly to the keygraph method (for the ratio test, we consider that a pose is found when four keypoint matches agree it).

Table 1 summarizes the results obtained. When a pose is found, we manually verified it by checking if the correct viewpoint of the object was projected in

Table 1. Percentage of test images for which a correct object was found (and for which a wrong object was found), i.e. true positives (and false positives), for the original ratio test [1], the modified ratio test [6] and the keygraph method.

Object type	Ratio test (Lowe [1])	Modified ratio test (Hsiao et al. [6])	Keygraph method (this paper)
Clam chowder can	14% (4%)	22% (4%)	46% (2%)
Soy milk can	2% (12%)	2% (10%)	8% (6%)
Tomato soup can	14%	10% (4%)	36%
Orange juice carton	54% (4%)	58% (2%)	72%
Soy milk carton	44% (8%)	46% (4%)	54% (4%)
Diet coke can	0% (2%)	2% (6%)	2%
Pot roast soup	10% (2%)	6% (4%)	36%
Juice box	26% (10%)	32% (12%)	42% (6%)
Rice pilaf box	64%	62% (2%)	74% (2%)
Rice tuscan box	68% (4%)	58%	62% (2%)

the test image. For the test images with two object instances, we consider that finding just one of them is a correct solution.

Our method performs significantly better than the ratio test and the modified ratio test. In the matching stage (before RANSAC), the keygraph method established an average of 2.8 keygraph matches (7.4 keypoint matches) between a test image and each training image, while the modified ratio test, on average, established only 1.7 keypoint matches between a test and each training image; this number could be increased by using additional k-means trees in the modified ratio test, but we observed that this also increased the false positive rate.

Before the RANSAC stage, the computational time demanded by the ratio test method and the keygraph method is similar, as the time spent to establish keypoint matches through the hierarchical k-means tree is largely dominant in comparison to the next stage of keygraph matching.

4 Conclusion

In this paper we described a method for object matching based on keygraphs, rather than keypoints, i.e., objects are matched using sets of triangles, where each vertex is a keypoint detected and described using SIFT. In the first step, keypoints are matched using a hierarchical k-means tree. Delaunay triangulation generates keygraphs in the test image and the matched keypoints generate keygraphs in the training images. We proposed to use five triangle features in order to evaluate the match between keygraphs, removing a significant number of false matches before running RANSAC to select inliers to compute an affine transformation between training and test images. Our method achieved a significantly higher accuracy than two state-of-the-art methods, the ratio test [1] and the modified ratio test [6]. Furthermore, the number of keygraph matches is small. On average, 2.8 keygraph matches are established between a test image and each training image. The quality of these matches is high, i.e., a small number of false matches occur. Besides, each keygraph match carries enough information to instantiate a pose hypothesis using an affine transformation. On the other hand, the ratio test method requires at least three keypoint matches.

As future work, we plan to use our method for 3D object recognition and pose estimation as in [6], which uses a structure-from-motion algorithm to create a 3D model of each training object. We believe that our approach is especially suited for this 3D setting. Only two keygraph matches between a test image and (possibly different) training images generate six keypoint matches, which is a good minimal set to generate a 3D pose [7]. This is an important advantage in comparison to a method that solely uses keypoints, which requires the selection of six keypoint matches to instantiate a 3D pose [7] or the selection of four keypoints matches with the use of an algorithm such as EPnP [8]. We expect that the keygraph method will demand a smaller number of 3D pose evaluations.

Another future work involves the use of Domain Adaptation (D.A.) techniques, which are useful when the training data is different from the test data (e.g. [9]). Such a domain change can occur due to variations in the object viewpoint, camera parameters, illumination change, motion etc. We expect that the use of D.A. will improve the first stage of our method (keypoint matching), as this stage is, essentially, a classification task through the hierarchical k-means tree. We also suggest the use of structured learning methods in order to optimize the weight of features used for graph matching, as done in [10].

References

1. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV (2004)
2. Sirmacek, B., Unsalan, C.: Urban-area and building detection using SIFT keypoints and graph theory. IEEE Trans. on Geoscience and Remote Sensing (2009)
3. McAuley, J.J., Caetano, T.S.: Fast matching of large point sets under occlusions. Pattern Recognition (2012)
4. Morimitsu, H., Hashimoto, M., Pimentel, R.B., Cesar Jr., R.M., Hirata Jr., R.: Keygraphs for sign detection in indoor environments by mobile phones. In: Jiang, X., Ferrer, M., Torsello, A. (eds.) GbRPR 2011. LNCS, vol. 6658, pp. 315–324. Springer, Heidelberg (2011)
5. Muja, M., Lowe, D.G.: Fast approximate nearest neighbors with automatic algorithm configuration. In: VISSAPP (2009)
6. Hsiao, E., Collet, A., Hebert, M.: Making specific features less discriminative to improve point-based 3D object recognition. In: CVPR (2010)
7. Szeliski, R.: Computer vision: algorithms and applications. Springer (2010)
8. Lepetit, V., Moreno-Noguer, F., Fua, P.: EPnP: An accurate O(n) solution to the PnP problem. IJCV (2009)
9. Ni, J., Qiu, Q., Chellappa, R.: Subspace interpolation via dictionary learning for unsupervised domain adaptation. In: CVPR (2013)
10. McAuley, J.J., Campos, T.d., Caetano, T.S.: Unified graph matching in Euclidean spaces. In: CVPR (2010)