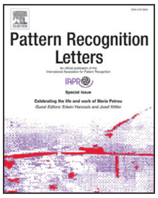
Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec



Scalable object instance recognition based on keygraph matching

Estephan Dazzi[a,*], Teofilo de Campos[b], Adrian Hilton[c], Roberto M. Cesar - Jr.[a]

[a] *Instituto de Matemática e Estatística - Universidade de São Paulo, São Paulo, Brazil*
[b] *Universidade de Brasília - DF, Brazil*
[c] *Center for Vision, Speech and Signal Processing - University of Surrey, Guildford, UK*

ARTICLE INFO

Article history:
Available online xxx

MSC:
41A05
41A10
65D05
65D17

Keywords:
Local invariant features
Semi-local graph matching
Graph topological properties

ABSTRACT

We propose a generalisation of the local feature matching framework, where keypoints are replaced by k-keygraphs, *i.e.*, isomorphic directed attributed graphs of cardinality k whose vertices are keypoints. Keygraphs have structural and topological properties which are discriminative and efficient to compute, based on graph edge length and orientation as well as vertex scale and orientation. Keypoint matching is performed based on descriptor similarity. Next, 2-keygraphs are calculated; as a result, the number of incorrect keypoint matches reduced in 75% (while the correct keypoint matches were preserved). Then, 3-keygraphs are calculated, followed by 4-keygraphs; this yielded a significant reduction of 99% in the number of remaining incorrect keypoint matches. The stage that finds 2-keygraphs has a computational cost equal to a small fraction of the cost of the keypoint matching stage, while the stages that find 3-keygraphs or 4-keygraphs have a negligible cost. In the final stage, RANSAC finds object poses represented as affine transformations mapping images. Our experiments concern large-scale object instance recognition subject to occlusion, background clutter and appearance changes. By using 4-keygraphs, RANSAC needed 1% of the iterations in comparison with 2-keygraphs or simple keypoints. As a result, using 4-keygraphs provided a better efficiency as well as allowed a larger number of initial keypoints matches to be established, which increased performance.

1. Introduction

Many problems in computer vision involve finding correspondences of robust invariant local features (*i.e.*, keypoints), for both 2D images and 3D point clouds generated using depth images. For example, object instance recognition applied to visual search, augmented reality or object manipulation by robots. In such contexts, a query image is matched against a (possibly large) set of model images by individually matching each query feature against the set of model features. Such recognition strategy based on local feature matching provides three main advantages. First, by employing indexing techniques, it is possible to efficiently compare the query features against the model features. Second, local feature matching is effective against problems caused by occlusions and background clutter, leading to a good performance in "real-world" object detection. Third, it is possible to obtain geometrically precise detections, since fine object structures are matched.

The extraction of local features involves two stages: detection and description. In the detection stage, keypoints presenting rich local information are identified. In the description stage, the method assigns to each keypoint local shape information (*e.g.*, scale and orientation) as well as a descriptor representing local visual content. After keypoint extraction, the next stage finds correspondences between keypoints representing the same parts in different images subject to large changes in viewpoint, scale and appearance. Traditionally, correspondences are established between individual keypoints based on descriptor similarity only, *e.g.*, the original SIFT approach of Lowe [21] or the method of Hsiao et al. [15]. Other authors employed spatial information in keypoint neighbourhoods in order to improve the overall quality of keypoint matching. For instance, the approach of Li et al. [19] which matches keypoint pairs or the SCRAMSAC method of Sattler et al. [29] that examines consistency in keypoint neighbourhoods.

We propose a generalisation of the traditional keypoint framework, by replacing keypoints for *keygraphs, i.e.*, isomorphic directed attributed graphs whose vertices are keypoints, in order to explore structural and topological properties. A keygraph constitutes a semi-local descriptor that maintains robustness to geometric deformations provided by its vertex (keypoint) features. Each image is represented by a set of keygraphs and correspondences are established between keygraphs. We refer to a keygraph with k vertices as a "k-keygraph". In this paper, we consider 2-keygraphs, 3-keygraphs and 4-keygraphs.

[*] Corresponding author.
E-mail address: estephan.dazzi@ime.usp.br (E. Dazzi).

https://doi.org/10.1016/j.patrec.2017.10.038
0167-8655/© 2017 Published by Elsevier B.V.

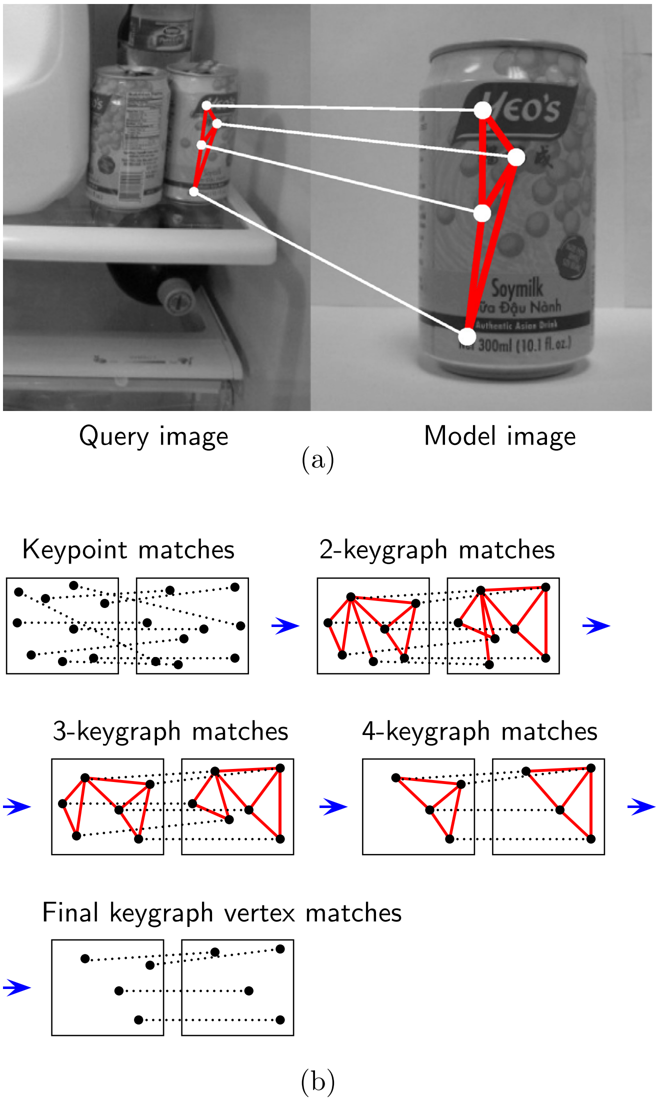


Fig. 1. **(a)** A match of 4-keygraphs between a query image (left) and a model image (right). **(b)** Keygraph matching pipeline. Starting from keypoint matches, keygraphs of cardinality k are defined based on keygraphs of cardinality $k-1$. Finally, RANSAC uses the matches of vertices of 4-keygraphs.

Fig. 1 illustrates the keygraph matching pipeline. Correspondences are established between query image keypoints and model image keypoints based on descriptor similarity; this yields a large number of both correct and incorrect keypoint matches. Next, matches of keygraphs of cardinality k are calculated based on keygraphs of cardinality $k-1$. The computational cost of the stage that obtains 2-keygraphs can be limited to a small fraction (say, 20%) of the cost of the keypoint matching stage, while the stages that obtain 3-keygraphs or 4-keygraphs have a *negligible cost*. As a result of the keygraph matching phase, the vast majority of incorrect initial keypoint matches are filtered out while the correct ones are preserved. Finally, RANSAC is applied to find object poses represented as affine transformations mapping images. By using matches of 4-keygraph vertices, RANSAC requires very few iterations, since a large fraction of correct matches is available.

This paper presents two fundamental contributions. First, we propose structural and topological properties of keygraphs with two, three or four vertices. Keygraph properties involve graph edge length and orientation as well as vertex scale and orientation, being fundamental to the success of the proposed approach. Second, we introduce an efficient method to calculate keygraph matches. The proposed method filters out the vast majority of incorrect keypoint matches, which provides two main advantages: (1) efficiency, since RANSAC needs to perform few iterations and (2) increased recognition performance, as the strategy allows a large number of keypoints matches to be established in the initial stage which results in a large number of correct keypoint matches.

Our experiments consider an object instance recognition problem in which a query image is matched against a large dataset of model objects. Query images are subject to realistic occlusions and illumination, viewpoint transformations. In this paper, we use keygraphs whose vertices are SIFT keypoints, since SIFT is widely known and has a good matching performance, as shown by Lowe [21]. As experimentally evaluated by Lowe [21], SIFT features are robust against problems caused by occlusions, background clutter and illumination changes. And, since transforming keypoint matches into keygraph matches does not eliminate correct keypoint matches, the proposed method also presents robustness against occlusion, background clutter and illumination changes. The model dataset is composed of 10^5 images, which generated a large set of 10^9 SIFT keypoints.

Experiments showed that obtaining 2-keygraphs filtered out 75% of the incorrect initial keypoint matches (while correct keypoint matches were not eliminated); next, obtaining 3-keygraphs reduced in 99% the number of incorrect remaining keypoint matches; then, obtaining 4-keygraphs filtered out a moderate fraction of remaining incorrect matches. As a result, RANSAC required very few iterations. In contrast, if simpler 2-keygraphs were used, RANSAC needed to perform two orders of magnitude more iterations, leading to a total computational at least 25% larger than the cost of the method based on 4-keygraphs. An even worse result was obtained by using simple “1-keygraphs” (*i.e.*, the initial keypoint matches). We also present results for the SCRAMSAC method of Sattler et al. [29]. SCRAMSAC has a similar computational cost as the proposed method while being based on a different use of spatial properties – namely, consistency in keypoint neighbourhoods. As shown by experiments, the proposed method achieved superior results than SCRAMSAC.

The remainder of this paper is organised as follows. Section 2 discusses related works in the literature. Section 3 presents mathematical definitions regarding keygraph matching. Section 4 describes an efficient implementation of the concepts presented in Section 3. Section 5 presents experimental evaluation. Finally, Section 6 draws conclusions.

2. Related work

An advantage gained from using local features is a natural geometric precision in detection, since fine object structures are matched. This property is fundamental in tasks such as Structure-from-Motion, wide-baseline stereo, augmented reality or object manipulation by robots, as discussed by Loncomilla [20]. In the present paper, we consider the problem of large-scale object instance recognition. Since our goal is introducing the keygraphs method, we employ a relatively simple, hand-crafted descriptor (SIFT), instead of using CNN-based descriptors that are tailored for image matching applications, such as the descriptors proposed by Dong and Soatto [7] and Han et al. [10]. Moo Yi et al. [23] employ a CNN in order to assign a canonical orientation to a keypoint; in contrast, SIFT uses the dominant orientation. Buoncompagni et al. [2] propose an efficient method to rank and select keypoints based on their saliency.

In order to achieve a better performance, local features conveying complementary information should be employed. Tombari et al. [30] proposed a keypoint detector and descriptor that operates in textureless regions. The result of the keypoint detection stage is a sparse keypoint set extracted from visually rich regions,

Please cite this article as: E. Dazzi et al., Scalable object instance recognition based on keygraph matching, Pattern Recognition Letters (2017), https://doi.org/10.1016/j.patrec.2017.10.038

which yields accurate keypoint matches. A different strategy relies on extracting a dense keypoint set, *e.g.*, one keypoint from each pixel. Choy et al. [5] employ CNN's in order to learn a feature space where dense local descriptors are compared.

Keypoints can be extracted from 3D point clouds created using depth images generated by sensors such as LIDAR. 3D keypoint detectors and descriptors are evaluated by Tombari et al. [31] and Guo et al. [9], respectively. The Fast Point Feature Histograms (FPFH) proposed by Rusu et al. [28] calculates a 3D keypoint descriptor as a histogram of angles between surface normals, measured in the surrounding keypoint neighbourhood. Kim and Hilton [18] improve the performance of FPFH by separately employing neighbourhoods of different sizes and then combining the matching outputs. Kim et al. [17] propose a framework for registration of visual data acquired from various 2D and 3D sensing modalities.

After the keypoint matching phase, a subset of matches agreeing on an object pose instantiation is found; in this context, the Random Sample Consensus (RANSAC) method proposed by Fischler and Bolles [8] is a standard solution. However, if a large fraction of incorrect matches is used, RANSAC needs to perform a large number of iterations, leading to a prohibitive computational cost. In order to filter out incorrect matches aiming to enable the use of RANSAC, in the approach of Pang et al. [25], a candidate keypoint match (p, q) is filtered out in case the local geometric structure of p is different from the one of q; such local geometric structure is calculated using an optimization process that reconstructs the keypoint from its three neighbours in the image. The approach of da Camara Neto and Campos [3] obtains a coarse global registration between a pair of images, which constrains the keypoint correspondence space; however, in case the fraction of correct keypoint matches is low, the estimation of coarse global registration is likely to fail.

Sattler et al. [29] proposed the Spatial Consensus RANSAC (SCRAMSAC) method. For each tentative keypoint match (p, q), a minimum fraction of the keypoints in a neighbourhood of p is required to match keypoints in a neighbourhood of q otherwise the match is filtered out. Such an approach presents two main drawbacks in comparison to ours. First, a large keypoint neighbourhood is employed in order to decide whether a candidate keypoint match is valid, thus making the approach susceptible to the fraction of correct keypoint matches. In contrast, the proposed method considers small keypoint neighbourhoods (up to four keypoints). Second, SCRAMSAC employs limited spatial information (keypoint distance only), while the proposed method considers orientation, scale and position. The computational cost of SCRAMSAC is similar to the cost of the proposed method: both methods are quadratic in the number of keypoint matches.

Previous work investigated the idea of finding correspondences of small keypoint sets instead of correspondences of individual keypoints. In particular, using matches of *keypoint pairs* demonstrated good results. The method of Carneiro and Jepson [4] checks for consistency in changes in keypoint scale and orientation as well as changes in length and orientation of a vector connecting each keypoint pair. Li et al. [19] additionally employ a Hough transform in order to filter out a number of keypoint matches. Hao et al. [12] detect 3D object models (created using Structure-from-Motion) in 2D query images; a candidate match of 2D-3D keypoint pairs is evaluated by back-projecting the 2D positions into the camera coordinates and then checking whether both 3D distances are similar[1]. propose a method with a linear cost in the number of matches that explores simple pairwise relations. Zhang et al. [33] employ the co-occurrence statistics of visual words within some local image regions.

Instead of using keypoint pairs, previous work used matches of *keypoint triples* or *quadruples*, which allows exploring richer structural information. Zitnick et al. [35] extract all keypoint triples from query and model images, which are then mapped to a canonical space where keypoint descriptors are calculated; then, keypoint triples are matched based on descriptor similarity. Kalantidis et al. [16] use Delaunay triangulations in order to select a subset of keypoint triples from both query and model images. Hao et al. [11] detect 3D object models in 2D images; Delaunay triangulations generate keypoint triples from the models, and a valid match presents consistent changes in keypoint scales and distances of 2D projections of 3D points. Hinterstoisser et al. [14] extract keypoint quadruples and quintuples from the model images, which can be used to instantiate a 3D object model onto a 2D query image. Hashimoto and Cesar [13] introduced the concept of keygraphs: all possible keygraphs with three vertices are extracted from a model image, and Fourier coefficients of keygraph edges are used as local descriptors; then, during matching time, a Delaunay triangulation generates keypoint triples from a query image.

A drawback of the discussed methods based on matches of keypoint triples or quadruples is relying on storing pre-calculated structures of model keypoints in working memory, which presents a scalability issue. Our previous approach ([6]) avoids that scalability problem: Delaunay triangulations are employed in order to select several keypoint triples in the *query* image. Next, each query keypoint triple (*i.e.*, 3-keygraph) is matched against a model image in case all its three constituent keypoint matches exist and the candidate 3-keygraph match satisfies the keygraph properties. Unfortunatelly, it is possible that a set of three correct keypoint matches do not form a 3-keygraph match, which would occur in case those three keypoint matches are not selected by a Delaunay triangulation for composing the same keypoint triple in the query image. In the present paper, we introduce an efficient strategy that matches *all possible* keypoint triples or quadruples during the matching phase, thus preserving correct keypoint matches.

A different graph-based approach for image matching models an image as a global graph whose vertices are keypoints. As shown by Zhang et al. [34], graph matching techniques can be used to find similar images. McAuley et al. [22] consider that graph similarity encompasses three aspects: keypoint descriptors, distances of keypoint pairs and inner angles of keypoint triples. Park et al. [26] also propose a method that is based on similarity of inner angles of keypoint triples. In contrast, our method relies on efficient comparisons of changes occurring in individual graph edges and/or vertices.

3. Definitions

Table 1 summarises the symbols and important concepts adopted in this paper.

3.1. Keypoints

A *keypoint* is a locally distinct feature. Each keypoint p is detected using SIFT and assigned a scale σ_p, an orientation θ_p, a position $\mathbf{x}_p = (x_p, y_p)$ and a descriptor. Let $\mathbb{P}_\mathcal{I}$ be the set of keypoints extracted from an image $\mathcal{I}$; each keypoint $p \in \mathbb{P}_\mathcal{I}$ is assigned an unique, random integer label $\mathcal{L}(p)$ ranging from 1 to $|\mathbb{P}_\mathcal{I}|$.

A *keypoint match* is a pair $\iota = (p, q)$, where keypoint p is in a *query image* $\mathcal{I}_Q$ and keypoint q is in a *model image* $\mathcal{I}_M$. A set $\mathcal{M}_{1\text{v}}$ has the initial keypoint matches between a pair of images.

Each keypoint match (p_i, q_i) is associated to a *change in keypoint scale*,

$$\Delta\sigma_i = \frac{\sigma_{q_i}}{\sigma_{p_i}} , \qquad (1)$$

Table 1
Summary of symbols and concepts adopted in the paper.

Symbols	Description
$\mathcal{I}_Q$, $\mathcal{I}_M$	Query image $\mathcal{I}_Q$ and model image $\mathcal{I}_M$.
p , σ_p , θ_p , $\mathbf{x}_p$	Keypoint p with scale σ_p, orientation θ_p and 2D position $\mathbf{x}_p$.
$\mathcal{L}(p)$	Keypoint label in an image.
$\mathbb{P}_{\mathcal{I}}$	Set of keypoints extracted from an image $\mathcal{I}$.
E , l_E , Θ_E	Keygraph edge E with length l_E and orientation Θ_E.
$l_{\min}$, $l_{\max}$	Minimum $l_{\min}$ and maximum $l_{\max}$ allowed edge length in a query image.
$\mathbb{E}_{\mathcal{I}_Q}$	Set of keygraph edges in a query image $\mathcal{I}_Q$.
$G = (\mathcal{V}_G, \mathcal{E}_G)$	Keygraph G with vertex set $\mathcal{V}_G$ and edge set $\mathcal{E}_G$.
G_{2v}, G_{3v}, G_{4v}	2-keygraph, 3-keygraph and 4-keygraph.
$\iota = (p, q)$	Keypoint match ι between keypoints p and q.
$\mu = (G, H, f)$	Match of keygraphs G and H with bijection mapping vertices f.
$\Delta\sigma$, Δl	Change in keypoint scale $\Delta\sigma$ and change in keygraph edge length Δl.
$\nabla_\Phi(\cdot,\cdot)$	Dissimilarity of changes in keypoint scale and/or edge length.
$\Delta\theta$, $\Delta\Theta$	Change in keypoint orientation $\Delta\theta$ and change in keygraph edge orientation $\Delta\Theta$.
$\nabla_\alpha(\cdot\,,\,\cdot)$	Dissimilarity of changes in keypoint orientation and/or edge orientation.
$\mathcal{M}_{1v}$	Set of initial keypoint matches between a pair of images.
$\mathcal{M}_{2v}, \mathcal{M}_{3v}, \mathcal{M}_{4v}$	Set of matches of 2-keygraphs $\mathcal{M}_{2v}$, 3-keygraphs $\mathcal{M}_{3v}$ and 4-keygraphs $\mathcal{M}_{4v}$.
$\mathcal{N}_{4v}$	Set of matches of vertices of 4-keygraphs.

and a *change in keypoint orientation*, measured as a (signed) difference between a pair of 2D orientations (*i.e.*, a 2D angle):

$$\Delta\theta_i = \theta_{q_i} - \theta_{p_i} \ . \tag{2}$$

3.2. Keygraphs

A *keygraph* $G = (\mathcal{V}_G, \mathcal{E}_G)$ is a directed attributed graph whose vertices are keypoints, with vertex set $\mathcal{V}_G$ (composed of keypoints in the same image) and graph edge set $\mathcal{E}_G$.

A *keygraph match* is a triple $\mu = (G, H, f)$, where $G = (\mathcal{V}_G, \mathcal{E}_G)$ is a keygraph in a query image and $H = (\mathcal{V}_H, \mathcal{E}_H)$ is a keygraph in a model image (with G and H being isomorphic), and $f : \mathcal{V}_G \to \mathcal{V}_H$ is a bijection mapping $\mathcal{V}_G$ and $\mathcal{V}_H$.

3.2.1. Keygraph edges

A *keygraph edge* is defined as an ordered pair $E_{ij} = \langle p_i, p_j \rangle$, where p_i and p_j are keypoints in the same image. In case E_{ij} is an edge in a query image, the keypoint labels are such that that $\mathcal{L}(p_i) < \mathcal{L}(p_j)$; that is, we arbitrarily determine an edge's direction to be such that the edge leaves the keypoint with the smaller label and enters the keypoint with the larger label (*i.e.*, from p_1 to p_2). Determining edge direction is a necessary step in order to assign edge orientation, which is obtained as the regular angle of a 2D vector with the horizontal axis (as illustrated in Fig. 3-c). In a 2D image, a vector $\mathbf{v}_{ij} = \mathbf{x}_{p_j} - \mathbf{x}_{p_i}$ is associated to E_{ij}; the length $l_{E_{ij}} = |\mathbf{v}_{ij}|$ and the orientation $\Theta_{E_{ij}}$ of $\mathbf{v}_{ij}$ are assigned to edge E_{ij}.

A match between edges $E_{ij} = \langle p_i, p_j \rangle$ and $F_{ij} = \langle q_i, q_j \rangle$ with a bijection $f = \{(p_i, q_i), (p_j, q_j)\}$ has a *change in edge length*:

$$\Delta l_{ij} = \frac{l_{F_{ij}}}{l_{E_{ij}}} \ , \tag{3}$$

and a *change in edge orientation*, measured as a (signed) difference between a pair of 2D orientations (*i.e.*, a 2D angle):

$$\Delta\Theta_{ij} = \Theta_{F_{ij}} - \Theta_{E_{ij}} \ . \tag{4}$$

A set $\mathbb{E}_Q$ of *keygraph edges in a query image* $\mathcal{I}_Q$ is composed of every keypoint pair $\langle p_i, p_j \rangle$ in $\mathcal{I}_Q$ whose associated keygraph edge E_{ij} has a length lying between a minimum and a maximum allowed values, $l_{\min} \le l_{E_{ij}} \le l_{\max}$, and whose keypoint scales σ_{p_i}, σ_{p_j} differ in at most one octave, $0.5 \le \sigma_{p_i}/\sigma_{p_j} \le 2.0$. If a keygraph edge with length l in a model image is mapped to a query image in which a change in scale s has occurred, its length becomes sl. Thus, the minimum edge length in a query image $l_{\min}$ is related to how much the area of an object in a query image can be reduced in comparison to the area of that object in a model image. As for the maximum edge length in a query image $l_{\max}$, it should be set sufficiently large to occupy the image of the considered object in a query image but not unnecessarily large to generate unnecessary computational cost.

3.2.2. Dissimilarity between changes in keygraph attributes

A keygraph match is associated to changes in keypoint scale $\Delta\sigma_i$ (Eq. (1)) and changes in edge length Δl_{ij} (Eq. (3)). Let $\Delta\phi$, $\Delta\phi'$ be a pair of changes in keypoint scale and/or edge length; a *dissimilarity*$\nabla_\phi(\Delta\phi,\ \Delta\phi')$ between them is measured as a ratio. We define a pair of changes $\Delta\phi$, $\Delta\phi'$ to be *similar* if the largest one is at most twice the smaller one:[1]

$$0.5 \le \frac{\Delta\phi}{\Delta\phi'} \le 2.0. \tag{5}$$

A keygraph match is also associated to changes in keypoint orientation $\Delta\theta_i$ (Eq. (2)) and changes in edge orientation $\Delta\Theta_{ij}$ (Eq. (4)). Let $\Delta\alpha$, $\Delta\alpha'$ be a pair of changes in keypoint orientation and/or edge orientation; a *dissimilarity* $\nabla_\alpha(\Delta\alpha,\ \Delta\alpha')$ between them is measured as an angle. We define a pair of changes $\Delta\alpha$, $\Delta\alpha'$ to be *similar* if the absolute value of the smaller angle between them is at most 60°:[2]

$$\arccos(\cos(\Delta\alpha - \Delta\alpha')) \le 60^\circ. \tag{6}$$

3.2.3. 2-Keygraphs

A *2-keygraph* $G_{2v} = (\mathcal{V}_{G_{2v}}, \mathcal{E}_{G_{2v}})$ has two vertices and one edge. Given a set $\mathbb{E}_Q$ of keygraph edges in a query image $\mathcal{I}_Q$ (Section 3.2.1), a set $\mathcal{M}_{2v}$ of 2-keygraph matches between $\mathcal{I}_Q$ and a model image $\mathcal{I}_M$ is obtained by matching each edge $E \in \mathbb{E}_Q$ against $\mathcal{I}_M$. Let triple $\mu_{2v} = (G_{2v}, H_{2v}, f_{2v})$ represent a candidate 2-keygraph match, where the query image keygraph G_{2v} has edge $E_{12} = \langle p_1, p_2 \rangle$ and the bijection mapping vertices is $f_{2v} = \{(p_1, q_1), (p_2, q_2)\}$ (with the keypoint matches in f_{2v} being determined in the initial keypoint matching stage). This candidate 2-keygraph match is established if it presents similar changes in attributes: the changes in keypoint scale $\Delta\sigma_1$, $\Delta\sigma_2$ and the change in edge length Δl_{12} must be pairwise similar (Eq. (5)), which yields $\binom{3}{2} = 3$ pairwise comparisons that must be satisfied (Fig. 2-Left). Similarly, the changes in keypoint orientation $\Delta\theta_1$, $\Delta\theta_2$ and the change in edge orientation $\Delta\Theta_{12}$ must be pairwise similar as well (Eq. (6)), which yields additional $\binom{3}{2} = 3$ comparisons that must be satisfied.

3.2.4. 3-Keygraphs

A *3-keygraph* $G_{3v} = (\mathcal{V}_{G_{3v}}, \mathcal{E}_{G_{3v}})$ is composed of three vertices and three edges. A set $\mathcal{M}_{3v}$ of 3-keygraph matches between a pair

[1] In an image subjected to a zoom of factor κ, each keygraph edge length or SIFT scale changes in the same factor κ. Sattler et al. [29] showed that in an image subjected to a change in viewing angle of ψ degrees, a unit circle becomes an ellipse whose longer and shorter axes have length 1 and $\cos\psi$, respectively. For successful matching, this change in viewing angle must be below 60°, since SIFT features lose reliability when $\psi > 60^\circ$, as shown by Lowe [21]. When $\psi = 60^\circ$, the length of the transformed ellipse's shorter axe divided by the original circle's diameter is $\cos 60^\circ/1 = 0.5$. Based on preliminary experiments, we used this value for the parameter, thus making keygraph matching invariant to changes in viewing angle of at most 60°.

[2] Rotating an image by θ degrees changes the orientation of every keypoint and keygraph edge in θ degrees. Under a moderate change in viewing angle, not every keypoint and edge rotates in the same θ degrees, although very distinct changes in orientation cannot occur. We allow a maximum difference in rotations of 60°, according to preliminary experiments.

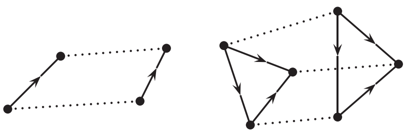


Fig. 2. Structural and topological properties of keygraphs. Left: each 2-keygraph match is associated to *one* change in edge and *two* changes in vertices, yielding $\binom{1+2}{2} = 3$ pairwise comparisons of changes in vertices and/or edge. Middle: each 3-keygraph match is associated to *three* changes in edges and *three* changes in vertices, yielding $\binom{3+3}{2} = 15$ comparisons of changes in vertices and/or edges. Right: each 4-keygraph match is formed of a pair of 3-keygraphs sharing an edge.

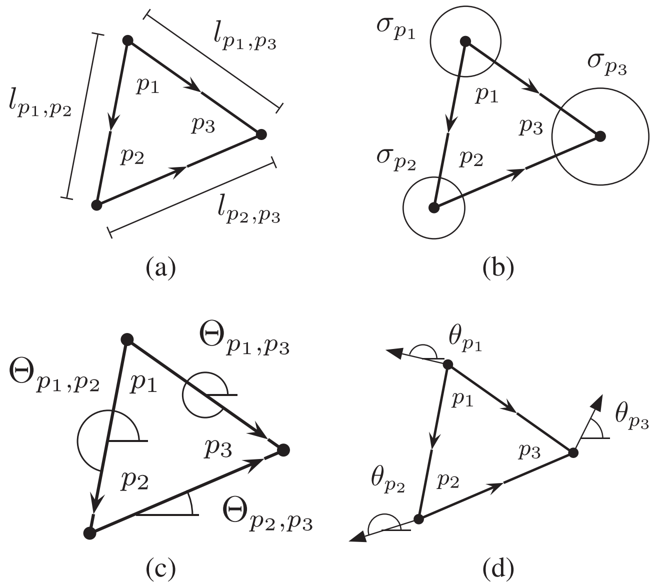


Fig. 3. Structural attributes of 3-keygraphs. **(a)** Edge length. **(b)** Vertex scale. **(c)** Edge orientation. **(d)** Vertex orientation.

of images contains all combinations of three 2-keygraph matches yielding a valid 3-keygraph match. Let $\mu_{3\mathrm{v}} = (G_{3\mathrm{v}}, H_{3\mathrm{v}}, f_{3\mathrm{v}})$ represent a candidate 3-keygraph match, where the query image keygraph $G_{3\mathrm{v}}$ has edges $\mathcal{V}_{G_{3\mathrm{v}}} = \{\langle p_1, p_2\rangle, \langle p_1, p_3\rangle, \langle p_2, p_3\rangle\}$ and the bijection mapping vertices is $f_{3\mathrm{v}} = \{(p_1, q_1), (p_2, q_2), (p_3, q_3)\}$. In order for this candidate 3-keygraph match to be established, similar changes in keygraph attributes are required: the changes in keypoint scale $\Delta\sigma_1$, $\Delta\sigma_2$, $\Delta\sigma_3$ and the changes in edge length Δl_{12}, Δl_{13}, Δl_{23} must be pairwise similar (Eq. (5)), which yields $\binom{6}{2} = 15$ pairwise comparisons that must be satisfied (Fig. 2-Middle). Similarly, the changes in keypoint orientation $\Delta\theta_1$, $\Delta\theta_2$, $\Delta\theta_3$ and the changes in edge orientation $\Delta\Theta_{12}$, $\Delta\Theta_{13}$, $\Delta\Theta_{23}$ must be pairwise similar as well (Eq. 6), which yields additional $\binom{6}{2} = 15$ pairwise comparisons that must be satisfied. Fig. 3 shows 3-keygraph attributes: keypoint scale, orientation and edge length, orientation. Fig. 4 illustrates the structural evaluation of matches of 3-keygraphs.

3.2.5. 4-Keygraphs

A *4-keygraph* $G_{4\mathrm{v}} = (\mathcal{V}_{G_{4\mathrm{v}}}, \mathcal{E}_{G_{4\mathrm{v}}})$ has four vertices and five edges. A set $\mathcal{M}_{4\mathrm{v}}$ of 4-keygraph matches between a pair of images is composed of all combinations of two 3-keygraphs sharing a 2-keygraph (Fig. 2-Right). Both constituent 3-keygraph matches are consistent w.r.t. the changes in the shared edge and vertices. Obtaining 4-keygraphs eliminates disconnected 3-keygraphs as well as 3-keygraphs sharing one vertex only.

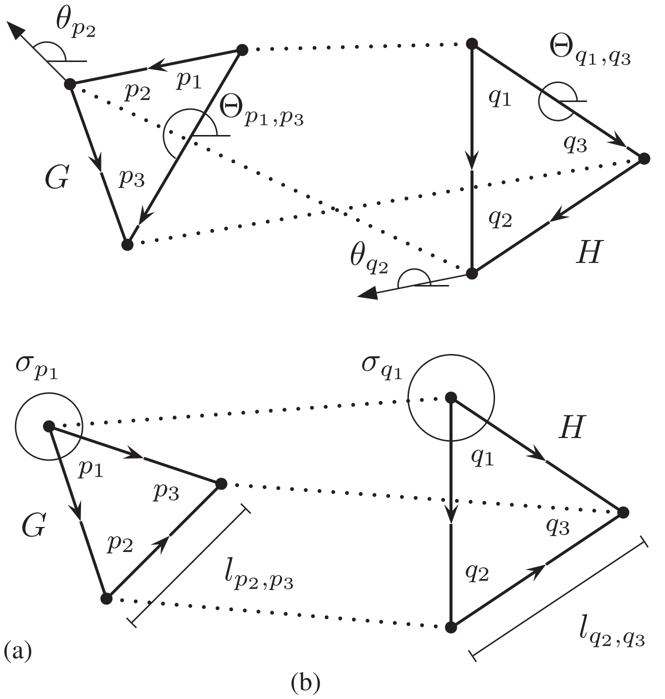


Fig. 4. Consistency in changes of 3-keygraph attributes. **(a)** Consistency among a change in vertex scale (changing from σ_{p_1} to σ_{q_1}) and a change in edge length (from l_{p_2,p_3} to l_{q_2,q_3}): $0.5 \le \frac{\sigma_{q_1}/\sigma_{p_1}}{l_{q_2,q_3}/l_{p_2,p_3}} \le 2.0$. **(f)** Consistency among a change in vertex orientation (from θ_{p_2} to θ_{q_2}) and a change in edge orientation (from Θ_{p_1,p_3} to Θ_{q_1,q_3}): $\arccos(\cos((\theta_{q_2} - \theta_{p_2}) - (\Theta_{q_1,q_3} - \Theta_{p_1,p_3}))) \le 60^\circ$.

3.3. Final matches of keygraph vertices

Given a set $\mathcal{M}_{4\mathrm{v}}$ of 4-keygraph matches between a pair of images, the set $\mathcal{N}_{4\mathrm{v}}$ of 4-keygraph vertex matches is

$$\mathcal{N}_{4\mathrm{v}} = \{(p, q) : (p, q) \in f_{4\mathrm{v}} \text{ and } (G_{4\mathrm{v}}, H_{4\mathrm{v}}, f_{4\mathrm{v}}) \in \mathcal{M}_{4\mathrm{v}}\}. \quad (7)$$

4. Methodology and implementation

The proposed method is composed of a learning phase and a matching phase. During the learning phase, keypoints are extracted from all model images and then indexed in the descriptor space. During the matching phase, keypoints are extracted from a query image $\mathcal{I}_Q$; then, object matching between $\mathcal{I}_Q$ and the dataset of model images follows five stages: keypoint matching and then matching of 2-keygraphs, 3-keygraphs and 4-keygraphs, followed by RANSAC using matches of 4-keygraph vertices. This pipeline is illustrated in Fig. 1-b.

4.1. First stage: keypoint matching

In order to index model keypoint descriptors, we use a modified version of the hierarchical K-means tree proposed by Muja and Lowe [24]. Given all model keypoints, K-means splits the descriptor space, recursively; a region with less than K descriptors then becomes a leaf node. When a query image keypoint p traverses a tree, in an intermediate tree level, p's descriptor is assigned to the nearest cluster center. In a leaf node, the distance from p's descriptor to the cluster center is calculated and represents the similarity between p and each one of the keypoints in this leaf; then, p restarts the traversal from the next most similar cluster mean (in an intermediate level). When p examines a total of L stored descriptors, the traversal stops. As a result, p establishes up to *one* keypoint match with *each* model image $\mathcal{I}_{M_n}$ (where n stands for the n-th model image in the dataset): during

Please cite this article as: E. Dazzi et al., Scalable object instance recognition based on keygraph matching, Pattern Recognition Letters (2017), https://doi.org/10.1016/j.patrec.2017.10.038

the traversal, if p is compared to more than one keypoint from the same model image, only the match with the highest similarity between descriptors is maintained. We also performed experiments employing at most two keypoint matches per model image, instead of one; however, using 4-keygraphs, this approach provided only a small gain in performance, thus we opted to use only one match since it allowed a more fair comparison of the keygraphs method with methods based on simple keypoints.

4.2. Second stage: matching of 2-keygraphs

Let $\mathcal{M}_{1v}$ be a set of initial keypoint matches between a query image $\mathcal{I}_Q$ and a model image $\mathcal{I}_{M_n}$. Each pair of keypoint matches in $\mathcal{M}_{1v}$ is checked for constituting a match of 2-keygraphs (Section 3.2.3). This generates a set $\mathcal{M}_{2v}$ of 2-keygraph matches between images $\mathcal{I}_Q$ and $\mathcal{I}_{M_n}$. The set $\mathcal{M}_{2v}$ is implemented as a (sparse) matrix of lists T_{2v}, with one list $L_{p,\,q}$ for each keypoint match $\iota = (p, q)$ in the set of matches of vertices of 2-keygraphs $\mathcal{N}_{2v} = \{(p, q) : (p, q) \in f_{2v}$ and $(G_{2v}, H_{2v}, f_{2v}) \in \mathcal{M}_{2v}\}$. A list $L_{p,\,q}$ contains all elements (p', q') such that there is an established match of 2-keygraphs whose edge in the query image is $\langle p, p'\rangle$ and whose bijection mapping vertices is $\{(p, q), (p', q')\}$. Each list $L_{p,\,q}$ stores its elements in sorted order, with an element (p', q') being assigned a sort key value consisting of a pair $(\mathcal{L}(p'), \mathcal{L}(q'))$.

For each 2-keygraph match that is established (*i.e.*, that satisfies the 2-keygraph properties shown in Section 3.2.3), the calculated changes in edge and vertices are stored in the matrix T_{2v} as well, in order to be employed in the next stage.

4.3. Third stage: matching of 3-keygraphs

Let $\mu_{2v} = (G_{2v}, H_{2v}, f_{2v})$ be a match of 2-keygraphs in $\mathcal{M}_{2v}$ with edge in the query image $E_{12} = \langle p_1, p_2\rangle$ and bijection mapping vertices $f_{2v} = \{(p_1, q_1), (p_2, q_2)\}$. In a matrix of lists T_{2v}, let lists L_{p_1,q_1} and L_{p_2,q_2} be associated to keypoint matches (p_1, q_1) and (p_2, q_2), respectively (Section 4.2). Since each list stores its elements in sorted order, finding elements which are common to both lists has a *linear complexity* in the total number of elements. Each element (p_3, q_3) which is present in both lists L_{p_1,q_1} and L_{p_2,q_2} yields a candidate match of 3-keygraphs $\mu_{3v} = (G_{3v}, H_{3v}, f_{3v})$ with edges in the query image $\mathcal{E}_{G_{3v}} = \{\langle p_1, p_2\rangle, \langle p_1, p_3\rangle, \langle p_2, p_3\rangle\}$ and bijection mapping vertices $f_{3v} = \{(p_1, q_1), (p_2, q_2), (p_3, q_3)\}$. Then, the method checks whether this 3-keygraph match is valid (Section 3.2.4). Since edge changes $\Delta\Theta_{ij}$, Δl_{ij} and vertex changes $\Delta\theta_i$, $\Delta\sigma_i$ were previously calculated and stored, the current stage only needs to evaluate the pairwise dissimilarities (Eqs. (5) and (6)). Also, since it is known that the changes involved in each individual 2-keygraph match are pairwise similar, they do not need to be re-evaluated; *e.g.*, it is known that the changes in edge length and vertex scale $\{\Delta l_{12}, \Delta\sigma_1, \Delta\sigma_2\}$ are pairwise similar.

Each 2-keygraph match in $\mathcal{M}_{2v}$ is checked for being involved in 3-keygraph matches, as illustrated in Fig. 5.

A set $\mathcal{M}_{3v}$ of 3-keygraph matches between a pair of images is implemented as a (sparse) matrix of lists T_{3v}. In T_{3v}, list $L^{p_1,q_1}_{p_2,q_2}$, which is associated to a pair of keypoint matches $\{(p_1, q_1), (p_2, q_2)\}$, contains all elements (p', q') such that there is a 3-keygraph match whose bijection mapping vertices is $f = \{(p_1, q_1), (p_2, q_2), (p', q')\}$; thus, in case the 3-keygraph match associated with f is established, three elements are inserted into matrix T_{3v}: an element (p', q') inserted into list $L^{p_1,q_1}_{p_2,q_2}$, (p_2, q_2) inserted into $L^{p_1,q_1}_{p',q'}$ and (p_1, q_1) inserted into $L^{p_2,q_2}_{p',q'}$.

4.4. Fourth stage: matching of 4-keygraphs

Let $\mu_{3v} = (G_{3v}, H_{3v}, f_{3v})$ be a 3-keygraph match in $\mathcal{M}_{3v}$ with edges in the query image $\mathcal{E}_{G_{3v}} = \{\langle p_1, p_2\rangle, \langle p_1, p_3\rangle, \langle p_2, p_3\rangle\}$ and bijection mapping vertices $f_{3v} = \{(p_1, q_1), (p_2, q_2), (p_3, q_3)\}$. The method verifies whether there are other 3-keygraph matches sharing an edge with μ_{3v} by individually checking each constituent 2-keygraph match. That is, for the edge $E_{12} = \langle p_1, p_2\rangle$, the method checks in the matrix of lists T_{3v} whether list $L^{p_1,q_1}_{p_2,q_2}$ has any inserted element; for each keypoint match (p', q') in list $L^{p_1,q_1}_{p_2,q_2}$, if $\mathcal{L}(p') > \mathcal{L}(p_3)$, then a match of 4-keygraphs $\mu_{4v} = (G_{4v}, H_{4v}, f_{4v})$ is established, where keygraph G_{4v} in the query image is the union of both 3-keygraphs, with edges $\mathcal{E}_{G_{4v}} = \mathcal{E}_{G_{3v}} \cup \{\langle p_2, p'\rangle, \langle p_3, p'\rangle\}$ and bijection mapping vertices $f_{4v} = f_{3v} \cup \{(p', q')\}$. Similarly, for the edge $E_{13} = \langle p_1, p_3\rangle$, list $L^{p_1,q_1}_{p_3,q_3}$ is verified, while, for the edge $E_{23} = \langle p_2, p_3\rangle$, list $L^{p_2,q_2}_{p_3,q_3}$ is verified.

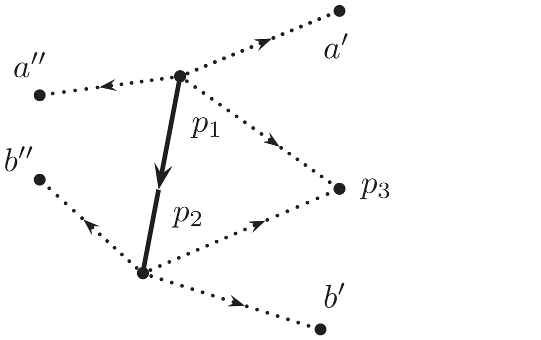


Fig. 5. Transforming 2-keygraphs into 3-keygraphs. In this Figure, each match of 2-keygraphs is represented by its edge in the query image. Given a 2-keygraph G_{2v} with edge $\langle p_1, p_2\rangle$, the method searches for pairs of edges leaving vertices p_1 and p_2 and entering in a same vertex. Figure shows that p_3 is such a common vertex; then, the candidate match of 3-keygraphs represented in this Figure by the vertex set $\{p_1, p_2, p_3\}$ is checked for being valid. Figure also shows that edges $\langle p_1, a'\rangle$, $\langle p_1, a''\rangle$, $\langle p_2, b'\rangle$, $\langle p_2, b''\rangle$ do not form 3-keygraphs.

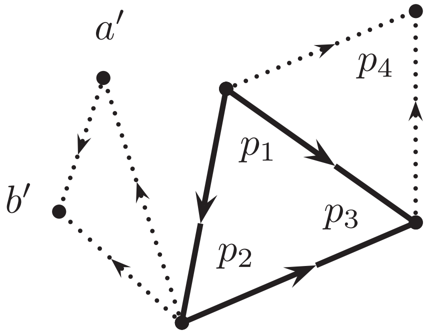


Fig. 6. Transforming 3-keygraphs into 4-keygraphs. In this Figure, each match of 3-keygraphs is represented by its edges in the query image. Given a 3-keygraph G_{3v} with edges $\{\langle p_1, p_2\rangle, \langle p_1, p_3\rangle, \langle p_2, p_3\rangle\}$, each 3-keygraph sharing an edge with G_{3v} generates a match of 4-keygraphs. Figure shows that a 4-keygraph G_{4v} with vertices $\{p_1, p_2, p_3, p_4\}$ is established. Figure also shows that a 3-keygraph G'_{3v} with vertices $\{a', b', p_2\}$ does not form a 4-keygraph together with G_{3v}, since G_{3v} and G'_{3v} do not share edges.

Each established 3-keygraph match is checked for being involved in 4-keygraph matches, as illustrated in Fig. 6.

4.5. Fifth stage: pose estimation using RANSAC

Let set $\mathcal{N}^Q_{M_n}$ contain matches of 4-keygraph vertices between a query image $\mathcal{I}_Q$ and a model image $\mathcal{I}_{M_n}$ (Eq. (7)). Then, RANSAC finds object poses represented as affine transformations mapping images. The total number of pose evaluations, with the whole dataset of model images, is set in advance. One iteration (*i.e.*, pose evaluation) of RANSAC proceeds as follows. Let set $\mathcal{N}^Q = \mathcal{N}^Q_{M_1} \cup \cdots \cup \mathcal{N}^Q_{M_N}$ contain all established keypoint matches between the query image $\mathcal{I}_Q$ and the whole set of N model images. Then, one

Please cite this article as: E. Dazzi et al., Scalable object instance recognition based on keygraph matching, Pattern Recognition Letters (2017), https://doi.org/10.1016/j.patrec.2017.10.038

keypoint match $\iota = (p, q)$ is randomly selected from $\mathcal{N}^Q$; let keypoint q belong to a model image $\mathcal{I}_{M_n}$. Next, two additional keypoint matches are randomly selected from the subset $\mathcal{N}^Q_{M_n}$ which contains matches with the model image $\mathcal{I}_{M_n}$ only. The three keypoint matches generate an affine transformation mapping images $\mathcal{I}_Q$ and $\mathcal{I}_{M_n}$, whose confidence is estimated as the number of inliers.

The next, final stage deals with multiple detections. First, the candidate affine transformations are sorted based on confidence. The best solution is returned and its model image ground-truth segmentation is projected onto the query image by using the recovered affine transformation. Then, the next best solution recounts its agreeing keypoint matches, now discarding matches lying inside the projection of any previously returned solution; in case there are remaining keypoint matches, this solution is then returned and projected onto the query image. This process continues as long as there are remaining candidate solutions.

4.6. Algorithmic complexity

If there are n keypoint matches between a pair of images, $O(n^2)$ candidate matches of 2-keygraphs are evaluated; this yields n_{2v} 2-keygraph matches, with a maximum of d_{2v} edges leaving a same vertex (in the query image). Next, $O(n_{2v} \cdot d_{2v})$ candidate 3-keygraph matches are evaluated; this yields n_{3v} 3-keygraph matches, with a maximum of d_{3v} 3-keygraphs sharing a same edge (in the query image). Next, $O(n_{3v} \cdot d_{3v})$ candidate 4-keygraph matches are evaluated. In terms of computational complexity, the worst-case occurs when a query image is identical to a model image: all candidate 2-keygraph matches would be established, yielding $O(n \cdot d_{2v})$ 2-keygraph matches, $O(n \cdot {d_{2v}}^2)$ 3-keygraph matches and $O(n \cdot {d_{2v}}^3)$ 4-keygraph matches, where $d_{2v} = O(n)$. Thus, an effective approach to control the combinatorial complexity limits the number d_{2v} of edges leaving a vertex as well as the number d_{3v} of 3-keygraphs sharing an edge *i.e.*, makes $d_{2v} = O(1)$ and $d_{3v} = O(1)$. As a result, the cost of the stage that finds 2-keygraphs dominates the cost of the following stages.

5. Experiments and results

We consider an object instance recognition problem where images are subject to realistic viewpoint, scale and appearance changes, as well as occlusion and background clutter. We employed the CMU10 dataset made available by Hsiao et al. [15]. This dataset contains ten types of model objects, for a total of 250 model images with resolution 640 × 480 or 1600 × 1200 pixels. There are 500 query images with resolution 640 × 480 pixels. The dataset collected by Hsiao et al. [15] considers a natural setting, consisting of common household objects in real, cluttered environments under different lighting conditions, occlusions and viewpoints (examples of images are presented in Figs. 1 and 9). Ten different objects are considered: clam chowder can, diet coke can, juice box, orange juice carton, pot roast soup, rice pilaf box, rice tuscan box, soy milk can, soy milk carton and tomato soup can. Some objects present relatively few visual features (*e.g.*, diet coke can and soy milk can) while other objects present a larger number of features (*e.g.*, clam chowder can and rice pilaf box). We decided to average out the final accuracy measure over all the ten object classes in order to better focus on the main result of the present paper, namely, the gain in performance and efficiency provided by using k-keygrahs with $k = 3$ or $k = 4$ in comparison to using $k = 1$ or $k = 2$ (we experimentally observed that all the ten considered object categories presented such gains in performance and efficiency). In order to simulate a large-scale scenario, we also considered an additional set of 10^5 "distractor" model images, obtained from the Oxford Buildings dataset introduced by Philbin et al. [27]. Their model keypoints are indexed together with the true model keypoints from the CMU10 dataset. The distractor images consider a very large range of different visual contexts, effectively simulating a "real-world" scenario in which many distinct model objects are considered. This yielded a total of 10^9 SIFT keypoints which were indexed in a hierarchical K-means tree using $K = 16$. The VLFeat library ([32]) was used for keypoint extraction.

For each detected object in a query image, we used the recovered affine transformation to project the model image's ground-truth segmentation onto the query image, which yields a region A; a detection is correct if $(A \cap A_{\text{gt}})/(A \cup A_{\text{gt}}) > 0.4$, where A_{gt} is the ground-truth in the query image.[3] For each of the ten model objects, we plotted a precision/recall curve and then calculated the area underneath the curve; this is denoted as *Average Precision* (AP). The average AP over all ten AP values summarises the results; we denote this average value as "*AP*".

Object matching can be divided into three stages: keypoint matching, keypoint match filtering and RANSAC which finds affine transformations. We evaluated five strategies for the match filtering stage. (1) Not filtering out keypoint matches, *i.e.*, RANSAC uses the initial keypoint matches; this is similar to a traditional approach that relies only on descriptor similarity and does not consider spatial information, such as the original SIFT method of Lowe [21] which uses simple, individual keypoints. (2) Using 2-keygraphs, which presents similarities with the method of Li et al. [19] that finds matches of keypoint pairs. (3) Using 3-keygraphs. (4) Using 4-keygraphs. And (5) using the SCRAMSAC method of Sattler et al. [29].

The maximum and minimum allowed keygraph edge length in a query image was set as $l_{\max} = 256$ and $l_{\min} = 8$ pixels, for images of resolution 640 × 480; those values were set empirically, following the discussion in Section 3.2.1. RANSAC considers a keypoint match as correct if the distance, in the query image, between the true keypoint position (x, y) and the mapped model keypoint's position (x', y') is lower than three pixels.

During tree traversal, each query keypoint p investigated $L = 4000$ model keypoints, leading p to establish up to 4000 initial keypoint matches (with at most one match with each model image). Next, each query keypoint retained only its $N < L$ matches with highest similarity between descriptors.

Fig. 7 presents the AP per maximum number N of initial matches of a query keypoint. Limiting N is useful in order to control the quadratic complexity of the keygraph matching phase. Experiments show that using a small N generated few initial matches in total. In this case, simple keypoints achieved a similar AP as keygraphs; this demonstrates that *transforming keypoints into keygraphs did not eliminate a significant number of correct keypoint matches*. Simple keypoints benefited moderately from using a larger number of initial matches: the AP improved from .39 to .44 when N increased from 10 to 10^2 (however, a large number of RANSAC iterations, $R = 10^6$, was used). Then, as N increased even more, so did the number of initial matches and the fraction of incorrect correspondences; in this case, simple keypoints achieved a poor performance, as a consequence of an infeasible number

[3] In matching methods based on 3D object models (*e.g.*, the method of [15]), training keypoints obtained from multiple viewpoints can be combined to generate a putative object pose. This enables the projected 3D object to have a larger area than in case matching is performed by using a single 2D warped training image that only considers a frontal view of the object (which is the case of the present paper). Therefore, a threshold of 0.5 on the overlap criterion (intersection divided by union of segmentation masks) is too tight for image-to-image matching methods in Hsiao et al. [15]'s dataset. Based on preliminary experiments, we found that setting the threshold as 0.4 allows a more fair comparison of the investigated methods. As an illustration, Figs. 9-d and 9- shows cases in which keypoints correctly matched the query images but, because of the way the training segmentation is projected, their overlap criterion fell under the 0.5 threshold.

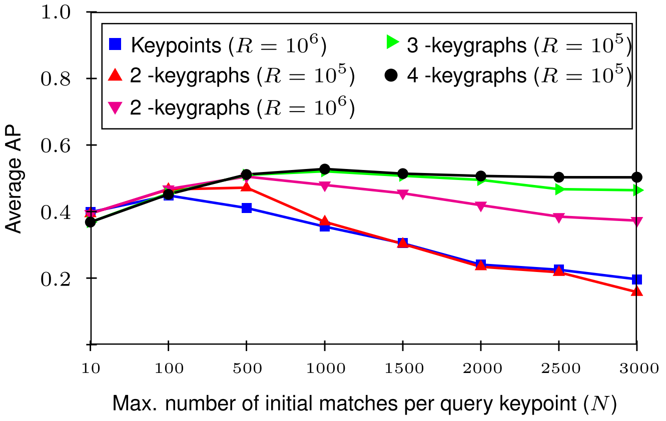


Fig. 7. Average AP (*i.e.*, average value of the area underneath the precision-recall curve over the ten object classes) versus value of the parameter N which sets the maximum number of initial matches of a query keypoint. The parameter R sets the number of RANSAC iterations.

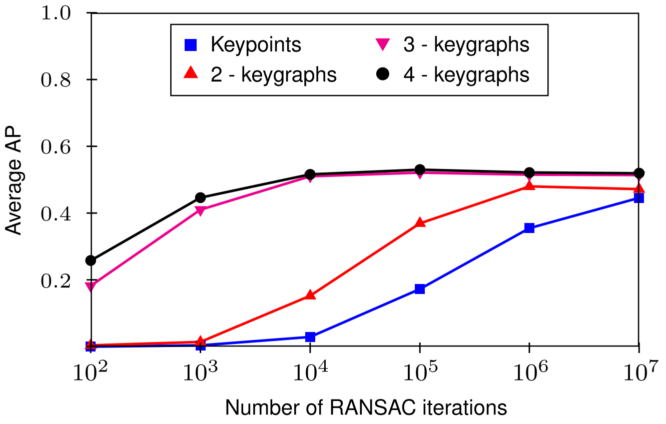


Fig. 8. Average AP versus number R of RANSAC iterations, using $N = 10^3$ initial matches per query keypoint.

of RANSAC iterations becoming necessary. In contrast, 4-keygraphs benefited from using a larger number of initial matches: the AP improved from .37 to .53 when N increased from 10 to 10^3. This demonstrates that using a larger N did yield a larger number of correct keypoint matches, even though those additional matches had a lower descriptor similarity than the matches obtained by using a small N only. Interestingly, 2-keygraphs achieved a consistently lower AP than 3- or 4-keygraphs, even when much more RANSAC iterations were used; this is a consequence of 2-keygraphs not filtering out as many incorrect matches as 3- or 4-keygraphs do. 2-keygraphs achieved its best performance (an AP of .50) by using a moderate number of initial matches, $N = 500$, and a large number of RANSAC iterations, $R = 10^6$.

Fig. 8 shows the estimated AP per number R of RANSAC iterations. We used a moderately large number of initial matches ($N = 10^3$); nevertheless, the stage which finds 2-keygraphs presented a small computational cost in comparison to the keypoint matching stage. 4-keygraphs achieved a high AP even using very few RANSAC iterations. 3-keygraphs performed slightly worse than 4-keygraphs when very few RANSAC iterations were employed. In case of 2-keygraphs, much more RANSAC iterations were necessary; for instance, 2-keygraphs achieved an AP of .36 and .47 for $R = 10^5$ and $R = 10^6$, respectively, while 4-keygraphs achieved .51 and .53 for $R = 10^4$ and $R = 10^5$, respectively.

We evaluated the computational cost of a single-thread C implementation of the object recognition pipeline. By using 4-keygraphs and $N = 10^3$, each query image required approximately eight seconds (not considering SIFT feature extraction). The stage which transforms the initial keypoint matches into 2-keygraphs required approximately 20% of the total cost. Next, obtaining 3-keygraphs and 4-keygraphs had a negligible cost (less than 1%). Then, RANSAC, using $R = 10^5$ iterations, required 1% of the total time. On the other hand, by using 2-keygraphs, employing $R = 10^6$ RANSAC iterations required a significant 20% of the total time; thus, the method based on 2-keygraphs required a total computational cost approximately 25% superior than the methods based on 3- or 4-keygraphs, due to the cost of RANSAC. In case of the method based on simple keypoints, using $R = 10^6$ led RANSAC to require 45% of the total computational cost. Since setting $R = 10^6$ led RANSAC to require a significant percentage of the total time, the fact that 4-keygraphs use $R = 10^5$ is an important advantage in comparison to 2-keygraphs which use $R = 10^6$. In order for simple keypoint to achieve a good performance, an infeasible number of RANSAC iterations, $R = 10^7$, was necessary.

Table 2
Average AP and average number of keypoint matches between a query image and each model image (before RANSAC), using $N = 1000$ initial matches per query keypoint. The method based on 2-keygraphs, which employ $R = 10^6$ RANSAC iterations, require a total computational cost approximately 25% larger than the methods based on 3- or 4-keygraphs.

Method	Avg. number of keypoint matches	AP
Keypoints, $R = 10^6$	44.7	.35
2-keygraphs, $R = 10^6$	10.1	.47
3-keygraphs, $R = 10^5$	0.12	.52
4-keygraphs, $R = 10^5$	0.06	.53
SCRAMSAC, $R = 10^6$	1.5	.48

We also evaluated the SCRAMSAC method proposed by [29]. We set the SCRAMSAC parameter $r = 128$ pixels which provided better results than using the value $r = 7$ that is suggested by the authors (*i.e.*, we considered larger keypoint neighbourhoods). We set $N = 10^3$ initial keypoint matches. By using $R = 10^5$ and $R = 10^6$ RANSAC iterations, SCRAMSAC achieved an average AP of .44 and .48, respectively. In comparison, 4-keygraphs presented a superior performance, with an AP of .53 using $R = 10^5$.

Table 2 shows the average number of keypoint matches between a query image and a model image, before RANSAC (using $N = 10^3$ initial matches). A model image with less than four keypoint matches contributed with zero to the averaged value, since RANSAC can not instantiate an affine transformation in such an image. The stage which finds 2-keygraphs yielded a reduction of 75% in the total number of incorrect keypoint matches. The next stage, which finds 3-keygraphs, yielded a reduction of 99% in the number of remaining incorrect keypoint matches. Next, finding 4-keygraphs filtered out a moderate fraction of incorrect matches. In case of SCRAMSAC, it filtered out a larger number of incorrect keypoint matches than the 2-keygraphs method. However, 4-keygraphs achieved a significantly better performance than SCRAMSAC.

Fig. 9 presents examples of matches between query and model images. Fig. 9-a shows 2-keygraph matches; these were transformed into 4-keygraph matches, as shown in Fig. 9-b. From Fig. 9-a to 9-b, two incorrect 2-keygraph matches were filtered out. Figs. 9-c and 9-d show the same query image as before, but considering a different model image, presenting a larger number of established keypoint matches: Fig. 9-c shows 4-keygraph matches, while Fig. 9-d shows the keypoint matches after RANSAC as well as the affine transformation mapping images. Fig. 9-e presents incorrect 3-keygraph matches that were eliminated when 4-keygraphs were obtained. Fig. 9-f shows another example of keypoint matches after RANSAC and the calculated affine transformation.

Please cite this article as: E. Dazzi et al., Scalable object instance recognition based on keygraph matching, Pattern Recognition Letters (2017), https://doi.org/10.1016/j.patrec.2017.10.038

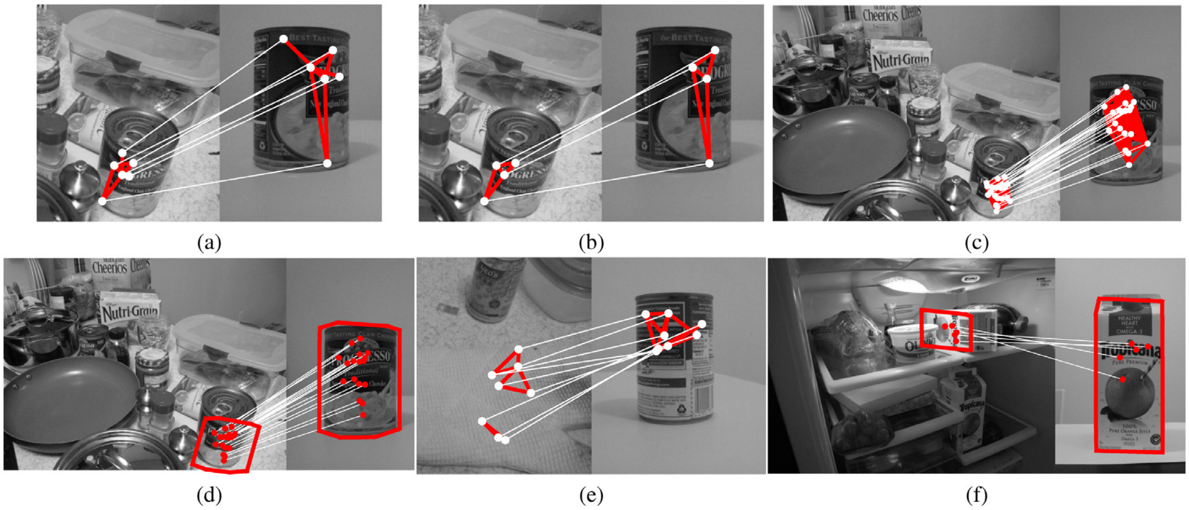


Fig. 9. Matches between query (left) and model (right) images. Keygraph matches before RANSAC: (9-a) 2-keygraphs; (9-b) and (9-c) 4-keygraphs; (9-e) 3-keygraphs. (9-d) and (9-f) show final keypoint matches (after RANSAC) and the calculated affine transformations.

Table 3
AP results obtained on the dataset of [6].

Method	AP
Ratio test of Lowe [21]	.53
Multiple triangulations of Dazzi et al. [6]	.55
4-keygraphs, proposed in this paper	.68

We compared 4-keygraphs with our previous method ([6]), which finds matches of keypoint triples that are generated by using several Delaunay triangulations. A valid keypoint triple is required to satisfy similar spatial properties as 3-keygraphs. Such an approach is particularly effective in a context where a large number of initial keypoint matches is established, since they are filtered out by using triangulations. In this experiment, the original dataset adopted by Dazzi et al. [6] is used. It is composed of the 250 images of the CMU10 dataset employing a K-means tree with $K = 16$. During tree traversal, a query image keypoint is compared against $L = 50$ stored model keypoints. In case of 4-keygraphs, a maximum of $N = 5$ initial keypoint matches is retained for each query keypoint, while the method of Dazzi et al. [6] uses $N = 50$. We also investigated the performance of the "ratio test" proposed in Lowe [21], in which each query keypoint matches a single model keypoint. In this experiment, $R = 10^3$ RANSAC iterations were used, which was sufficiently large for all methods. Table 3 shows the results. Lowe's ratio test and Dazzi et al. [6]'s method achieved an AP of .53 and .55, respectively, while 4-keygraphs achieved a significantly higher AP (.68). This result demonstrates that the methods of Dazzi et al. [6] and Lowe's ratio test did eliminate correct keypoint matches, since they achieved a lower AP than 4-keygraphs.

6. Conclusion

Methods based on local feature matching establish keypoint correspondences relying on descriptor similarity. In object instance recognition, establishing matches between query keypoints and different model images yields a large number of correct keypoint matches, which improves recognition performance. However, this produces a large fraction of incorrect correspondences, which hinders the performance of RANSAC-like pose estimation. In order to avoid this problem, we proposed a method that filters a large number of incorrect keypoint matches (while the correct ones are preserved), thus enabling RANSAC to be used. The proposed method transforms keypoint matches into matches of k-keygraphs. Each valid keygraph match satifies semi-local affine constraints which are efficiently evaluated.

Keygraphs of cardinality k are defined based on keygraphs of cardinality $k - 1$. Keypoint matches are transformed into 2-keygraphs, which involves calculating changes in length, orientation and scale. In our experiments, obtaining 2-keygraphs reduced 75% of the incorrect keypoint matches; this operation had a small computational cost in comparison to the cost of keypoint matching. Next, 3-keygraphs are obtained at a negligible computational cost, which yielded a reduction of 99% of the remaining incorrect keypoint matches. As a consequence, the method based on 3-keygraphs achieved a high performance even using 1% of the RANSAC iterations in comparison with the method based on 2-keygraphs. Our experiments also showed that using all the initial keypoint matches, as well as SCRAMSAC, performed worse than using 3-keygraphs.

We proposed 4-keygraphs, that are generated from a pair of 3-keygraphs sharing an edge. Such method to obtain 4-keygraphs can be employed to find keygraphs of cardinality larger than four. However, since a valid match of k-keygraphs requires k correct keypoint matches, one drawback associated to using $k > 4$ is a reduced probability of detecting small or occluded objects, which present few keypoint matches. In this paper, we used 4-keygraphs, that provided the best results.

The keygraphs method has few intrinsic parameters. The two most important ones are the thresholds on attribute changes (scale/length and orientation). The values were selected in order to allow a large range of viewpoint change between images.

In this paper, we employed SIFT features, chosen mostly due to their popularity. Keygraph matching uses scale, orientation and position computed by the keypoint detector. SIFT could be readily replaced by other descriptors which generate scale, orientation and position information with a similar precision as SIFT. Examples of methods proposed more recently than SIFT are SURF (Speeded-Up Robust Features) and ORB (Oriented FAST corner detector and Rotated BRIEF features). The main improvement that they brought is in terms of computational efficiency, while the improvement in terms of performance is not as substantial. CNN-based local descriptors have been proposed more recently and could also replace SIFT, as long as they are designed to produce local scale and orientation information. However, CNN-based descriptors require substantially more computational power than SIFT, SURF and ORB. It

Please cite this article as: E. Dazzi et al., Scalable object instance recognition based on keygraph matching, Pattern Recognition Letters (2017), https://doi.org/10.1016/j.patrec.2017.10.038

is worth noting that the proposed keygraphs method tolerates an elevated amount of noise in the estimations of scale, orientation and position of keypoints, being robust to wide variations in viewpoint. Its focus is on the structure of small sets of local features rather the quality of individual matches. A benchmark evaluation of descriptors is therefore beyond the scope of this paper and constitutes a suggestion for future work.

Another direction for future work is an extension of our keygraphs method for 3D point clouds, using 3D keypoint detectors and descriptors for applications such as 3D object retrieval and 3D scene understanding.

Acknowledgment

The authors are grateful to FAPESP grant #2011/50761-2, #14/50769-1, 2015/22308-2, CNPq grant #502472/2014-6, CAPES, NAP eScience - PRP - USP.

References

[1] Y. Avrithis, G. Tolias, Hough pyramid matching: speeded-up geometry re-ranking for large scale image retrieval, IJCV 107 (2014) 1–19.
[2] S. Buoncompagni, D. Maio, D. Maltoni, S. Papi, Saliency-based keypoint selection for fast object detection and matching, Pattern Recognit. Lett. 62 (2015) 32–40.
[3] V.F. da Camara Neto, M.F. M. Campos, On the improvement of image feature matching under perspective transformations, IEEE, 2010. 23rd Conference on Graphics, Patterns and Images SIBGRAPI 2010.
[4] G. Carneiro, A.D. Jepson, Flexible spatial models for grouping local image features, in: ICCV and Pattern Recognition, 2004, pp. II–747.
[5] C.B. Choy, J. Gwak, S. Savarese, M. Chandraker, Universal correspondence network, in: Advances in Neural Inf. Proc. Systems, 2016, pp. 2406–2414.
[6] E. Dazzi, T. de Campos, A. Hilton, R.M. Cesar-Jr., Efficient object recognition using sampling of keypoint triples and keygraph structure, in: XXIX Conference on Graphics, Patterns and Images SIBGRAPI 2016, 2016.
[7] J. Dong, S. Soatto, Domain-size pooling in local descriptors: DSP-SIFT, in: CVPR, 2015, pp. 5097–5106.
[8] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (1981) 381–395.
[9] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, N.M. Kwok, A comprehensive performance evaluation of 3D local feature descriptors, IJCV 116 (2016) 66–89.
[10] X. Han, T. Leung, Y. Jia, R. Sukthankar, A.C. Berg, Matchnet: unifying feature and metric learning for patch-based matching, in: CVPR, 2015, pp. 3279–3286.
[11] Q. Hao, R. Cai, Z. Li, L. Zhang, Y. Pang, F. Wu, 3D visual phrases for landmark recognition, in: CVPR, 2012, pp. 3594–3601.
[12] Q. Hao, R. Cai, Z. Li, L. Zhang, Y. Pang, F. Wu, Y. Rui, Efficient 2D-to-3D correspondence filtering for scalable 3D object recognition, in: CVPR, 2013, pp. 899–906.
[13] M. Hashimoto, R.M. Cesar Jr, Object detection by keygraph classification, in: Graph-Based Representations in Pattern Recognition, Springer, 2009, pp. 223–232.
[14] S. Hinterstoisser, S. Benhimane, N. Navab, N3m: natural 3D markers for real–time object detection and pose estimation, in: ICCV, 2007, pp. 1–7.
[15] E. Hsiao, A. Collet, M. Hebert, Making specific features less discriminative to improve point-based 3D object recognition, in: CVPR, 2010, pp. 2653–2660.
[16] Y. Kalantidis, L.G. Pueyo, M. Trevisiol, R. van Zwol, Y. Avrithis, Scalable triangulation-based logo recognition, in: Proceedings of the ACM International Conference on Multimedia Retrieval, 2011.
[17] H. Kim, A. Evans, J. Blat, A. Hilton, Multi-modal visual data registration and web-based visualisation, IEEE Trans. Circuits Syst. Video Technol. (2016).
[18] H. Kim, A. Hilton, Hybrid 3D feature description and matching for multi-modal data registration, in: ICIP, 2014, pp. 3493–3497.
[19] X. Li, M. Larson, A. Hanjalic, Pairwise geometric matching for large-scale object retrieval, in: CVPR, 2015, pp. 5153–5161.
[20] P. Loncomilla, Object recognition using local invariant features for robotic applications: a survey, Pattern Recognit. (2016).
[21] D.G. Lowe, Distinctive image features from scale-invariant keypoints, IJCV 60 (2004) 91–110.
[22] J.J. McAuley, T. De Campos, T.S. Caetano, Unified graph matching in euclidean spaces, in: CVPR, 2010, pp. 1871–1878.
[23] K. Moo Yi, Y. Verdie, P. Fua, V. Lepetit, Learning to assign orientations to feature points, in: CVPR, 2016, pp. 107–116.
[24] M. Muja, D.G. Lowe, Scalable nearest neighbor algorithms for high dimensional data, PAMI 36 (2014) 2227–2240.
[25] S. Pang, J. Xue, Q. Tian, N. Zheng, Exploiting local linear geometric structure for identifying correct matches, CVIU 128 (2014) 51–64.
[26] S. Park, S.K. Park, M. Hebert, Fast and scalable approximate spectral matching for higher order graph matching, PAMI 36 (2014) 479–492.
[27] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Object retrieval with large vocabularies and fast spatial matching, in: CVPR, 2007, pp. 1–8.
[28] R.B. Rusu, N. Blodow, M. Beetz, Fast point feature histograms (FPFH) for 3D registration, in: IEEE International Conference on Robotics and Automation, 2009, pp. 3212–3217.
[29] T. Sattler, B. Leibe, L. Kobbelt, SCRAMSAC: improving RANSAC's efficiency with a spatial consistency filter, in: ICCV, 2009, pp. 2090–2097.
[30] F. Tombari, A. Franchi, L. Di Stefano, Bold features to detect texture-less objects, in: ICCV, 2013, pp. 1265–1272.
[31] F. Tombari, S. Salti, L. Di Stefano, Performance evaluation of 3D keypoint detectors, IJCV 102 (2013) 198–220.
[32] A. Vedaldi, B. Fulkerson, VLFeat: An open and portable library of computer vision algorithms, 2008. http://www.vlfeat.org/.
[33] Y. Zhang, Z. Jia, T. Chen, Image retrieval with geometry-preserving visual phrases, in: CVPR, 2011, pp. 809–816.
[34] Z. Zhang, Q.S.J. McAuley, W.W.Y. Zhang, A. van den Hengel, Pairwise matching through max-weight bipartite belief propagation, CVPR, 2016.
[35] C.L. Zitnick, J. Sun, R. Szeliski, S. Winder, Object instance recognition using triplets of feature symbols, Technical Report, Microsoft Research, 2007.

Please cite this article as: E. Dazzi et al., Scalable object instance recognition based on keygraph matching, Pattern Recognition Letters (2017), https://doi.org/10.1016/j.patrec.2017.10.038