Scale-Invariant Proximity Graph for Fast Probabilistic Object Recognition

Jerome Revaud
Université de Lyon, CNRS,
INSA-Lyon, LIRIS, UMR5205,
F-69621, France
jerome.revaud@liris.cnrs.fr

Guillaume Lavoué
Université de Lyon, CNRS,
INSA-Lyon, LIRIS, UMR5205,
F-69621, France
guillaume.lavoue@liris.cnrs.fr

Ariki Yasuo
CS-17
Kobe University
Japan
ariki@kobe-u.ac.jp

Atilla Baskurt
Université de Lyon, CNRS,
INSA-Lyon, LIRIS, UMR5205,
F-69621, France
atilla.baskurt@liris.cnrs.fr

ABSTRACT

A pseudo-hierarchical graph matching procedure dedicated to object recognition is presented in this paper. From a single model image, a graph is built by extracting invariant local features and linking them according to a so-called proximity rule. The resulting graph presents several interesting properties including invariance to scale, robustness to various distortions and empirical linearity of the number of edges with respect to the number of nodes. The matching process is made hierarchical in order to increase both speed and detection performances. It relies on progressively incorporating the smaller model features as the hierarchy level increases. As a result, even a matching between graphs containing thousands of nodes is very fast (a few milliseconds). Experiments demonstrates that the method outperforms state-of-the-art specific object detectors in terms of precision-recall measures and detection time.

Categories and Subject Descriptors

I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*Object recognition*; I.4.10 [**Image Processing and Computer Vision**]: Image Representation—*Hierarchical*; I.5.1 [**Pattern Recognition**]: Models—*Structural*; G.2.2 [**Discrete Mathematics**]: Graph Theory—*Graph labeling*

General Terms

Object recognition; graph

CIVR '10, July 5-7, Xi'an China

Keywords

Specific object recognition; Hierarchical graph matching; invariant local features

1. INTRODUCTION

Object recognition has been a very active topic in the past 30 years. Whereas the perfect *class* object recognition system is yet to be invented, *specific* object recognition has received a decreasing attention since the apparition of *keypoints,* whose most famous avatar is probably SIFT [12]. Indeed, recognition methods using keypoints present numerous advantages: they are invariant to translation, scale, rotation and occlusion without significant increase in complexity; thanks to the high descriptive power of the keypoints, any training is quite unnecessary; they are close to real-time; and finally they are simple to carry out.

However, extracting keypoints is one thing but detecting the full object is another. We can roughly distinguish between two classes of methods doing so: (1) methods using a global transform, and (2) methods derived from graph matching (i.e. using local transforms). So far, methods using a global transform have been thoroughly investigated and have shown quite convincing results [11, 12, 16, 14, 15]. Yet, there are several problems with using a global transform. First of them, the choice of the transform: since most detection systems operate in the real world, a 3D projective transform should be used by default, however the large number of required parameters is discouraging both for RANSAC-derived methods [16, 14] and for approaches inspired from the Hough transform [12]. In the first case, RANSAC would have to iterate $O(n^k)$ times, where n is the number of matches and k the number of matches necessary for estimating the transform parameters, while in the second, the space of parameters must have k dimensions. Since it is either not feasible or too slow for large values of k, most approaches use a simplified transform (e.g. an affine transform) to approximate the reality while preserving a decent processing time. Another problem of the global methods is their inability to handle non-rigid distortions, such as what can happen to a magazine or to a face. For those reasons,

using a global transform is not entirely satisfying.

On the other hand, graph matching seems to be a straightforward way to object detection. Indeed, after having extracted some salient keypoints, both model object and scene can be represented as graphs. Moreover, graph matching operates at a local scale by comparing pairs of nodes or pairs of edges, thus avoiding the need of a global transform while in the same time giving more flexibility to the model [9] and increasing performances [6] if distant features are disconnected. In sum, the only problem with graph matching is that it is NP-hard. While recent researches have focused on global approximations (graph-cuts [17], tensor-based [7]), the timing performances remain disappointing for large graphs. In comparison, the historically older relaxation methods perform faster and stay competitive in practice [10, 13, 17], although no theoretical guarantee ensures their convergence. Since we focus here on a subclass of problems where we can rely on scale-invariant local features, it is possible to further increase the matching speed by using the additional information provided by the features, that is, their orientation and scale. In this paper and on the contrary of RANSAC-derived approaches for instance, we take full advantage of both information, but more especially of the scale, to build a hierarchical matching system relying on probabilistic relaxation.

Indeed, hierarchies have been shown to be an efficient way of reducing the computational burden by spreading the spatial constraints over several scale levels, which in addition improves the robustness to intra-class variability [8, 18]. In particular, Wilson and Hancock have developed a graph matching procedure with hierarchical relaxation [18] based on a bottom-up process that compares pairs of so-called *super-cliques* between the model and scene graphs. However, the comparison between super-cliques somehow unnecessarily increases the complexity with respect to a simpler pairwise edge comparison as done by Christmas et al. [4] for instance. This last work describes another probabilistic relaxation and was also adapted to a pseudo-hierarchical graph matching in a related paper [3]. It presents several advantages: the framework is minimalist and simple to use; it is implicitly designed for subgraph isomorphism; it is robust to noise; and the algorithm was empirically shown to converge faster than most of the other existing methods (typically it takes less than 5 iterations). Unfortunately, the hierarchy used was simplistic and difficult to generalize. Initially developed for an application of road-segments matching, the authors used a two-pass procedure to overcome the large number of segments with respect to the computational power available at that time. In the first pass, they simplified the model graph by removing every segment smaller than a threshold. Then, they executed the matching process with this reduced graph and, based on the result, they estimated the parameters of the similarity transform that matched the model on the scene. Finally, they incorporated all model segments in the second pass - eliminating every hypothesis not compliant with the global transform - and ran the relaxation once again to get the final result. This simple approach succeeded in this case, however it has many drawbacks: the searched object has to be present only once in the scene in order to correctly estimate the global transform; the number of levels used (i.e. 2) is minimal and immutable; the requirement of a global transform itself makes lose the very interest of the method, i.e. feed the algorithm with only simple local information and let the algorithm aggregate them to decide upon the global result. Although we used the same probabilistic framework, our method is perfectly suited for multi-object detection and extends the hierarchy to an arbitrary number of levels without the need of a global transform. Our matching procedure is however not hierarchical in the strict sense of the word: in hierarchical strategies, nodes of the lower levels of abstraction are contracted into single nodes of higher levels. Instead, we base our strategy on gradually incorporating the model features in the matching process as the hierarchy level increases, while in the same time the scene graph is pruned for compensation.

The rest of the paper is organized as follows: we begin by briefly presenting the original theory of Christmas et al. [4]. Then, we introduce the notion of proximity graph in section 3. The pseudo-hierarchical matching procedure is described in detail in section 4. Finally, we demonstrate the effectiveness of the method in section 5 and conclude in section 6.

2. PROBABILISTIC RELAXATION

In this section, we sum up for the reader the probabilistic framework developed by Christmas et al. in [4]. Given two complete graphs G^m and G^s (respectively, the model and scene graphs), the aim of the matching is to find the best mapping between each model and scene node. In our formalism, $G = (V, E, X)$ where E represents the set of edges, V is the set of vertices and X the set of their associated unary measurements (in our case, a SIFT descriptor). The eventuality of subgraph isomorphism is dealt with by adding the null node $v_0 \in V^m$ to the model graph ; in other words extraneous nodes in the test image are simply tagged with the null-label.

As in similar works, the method needs two kinds of probabilistic measures to estimate the likelihood of matches between each scene node and each model node: (a) the probability of a node-to-node assignment $p(u_\alpha \leftarrow v_i | x_\alpha)$ using unary attributes only ($u_\alpha \in V^s$, $x_\alpha \in X^s$ and $v_i \in V^m$), and (b) an edge compatibility function that describes the local affinity between two presumed matches:

$$p(e_{\alpha\beta} | u_\alpha \leftarrow v_i, u_\beta \leftarrow v_j) \tag{1}$$

with $e_{\alpha\beta} \in E^s$. After having initialized the probabilities using measure (a), the relaxation iterates until convergence of the system according to the following update rule:

$$p^{(n+1)}(u_\alpha \leftarrow v_i) = \frac{p^{(n)}(u_\alpha \leftarrow v_i) Q^{(n)}(u_\alpha \leftarrow v_i)}{\sum_{v_j \in V^m} p^{(n)}(u_\alpha \leftarrow v_j) Q^{(n)}(u_\alpha \leftarrow v_j)} \tag{2}$$

where

$$Q^{(n)}(u_\alpha \leftarrow v_i) = \prod_{u_\beta \in V^s \setminus u_\alpha} \sum_{v_j \in V^m} p^{(n)}(u_\beta \leftarrow v_j) p(e_{\alpha\beta} | u_\alpha \leftarrow v_i, u_\beta \leftarrow v_j). \tag{3}$$

For further details, we refer the reader to the original paper [4].

3. SCALE-INVARIANT PROXIMITY GRAPH

Although Christmas et al. [4] had formulated the matching problem using complete graphs (i.e. $\forall i \neq j,\ v_i, v_j \in V \times V \Rightarrow e_{ij} \in E$) in their original paper, it is usually not feasible computationally speaking. Indeed, it would involve the calculation of $O(|V^m|^2\,|V^s|^2)$ edge probabilities where each set easily contains hundreds or thousands of features in realistic condition of use as well as in our experiments. A critical point for our system is thus to be able to relax the spatial constraints between distant features. Surprisingly, it remains compatible with the relaxation mechanism of [4] provided that we force the density function to worth zero when the model or scene edge does not exist:

$$\begin{cases} \forall e_{ij} \notin E^m, & p(e_{\alpha\beta}|u_\alpha \leftarrow v_i, u_\beta \leftarrow v_j) = 0 \\ \forall e_{\alpha\beta} \notin E^s, & p(e_{\alpha\beta}|u_\alpha \leftarrow v_i, u_\beta \leftarrow v_j) = 0 \end{cases} \quad (4)$$

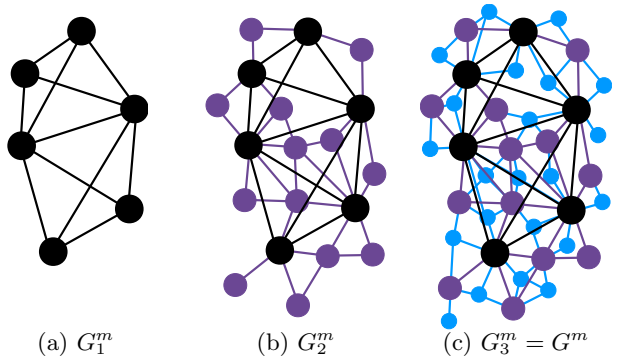
Theoretically speaking, this constraint may sound counter-intuitive since in the event where $(u_\alpha,\ u_\beta)$ or $(v_i,\ v_j)$ are not connected, it looks like inducing that the two matches $u_\alpha \leftarrow v_i$ and $u_\beta \leftarrow v_j$ are necessary false. In reality however, it results in that the features can only receive an influence from their direct neighbors (but indirect influence between distant features still occurs after several steps of diffusion of the probabilities).

Thus, we simply define the *proximity graph* as a graph in which distant features are not connected. Formally, we restrain the set of edges to:

$$E = \left\{ e_{ij} \,\middle|\, \forall i,j\ \frac{\|\mathbf{p}_i - \mathbf{p}_j\|}{\sqrt{\sigma_i \sigma_j}} < \chi \right\} \quad (5)$$

where $\mathbf{p} = (p_x, p_y)$ denotes a keypoint position, σ its scale and χ is a constant. This definition yields several interesting properties with respect to our application:

- The graph topology is independent of the scale, i.e. both model and scene graph structures are invariant to the size of the object in the image. This comes from the fact that the distance between keypoints is normalized by their scale in eq. (5).
- Each graph edge stands for a stable connection. Indeed, from the "perspective" of a keypoint, the noise on the relative position of the other keypoints increases with their distance in the pyramidal scale-space (i.e. bigger points appear closer).
- The proximity graph substantially reduces the computational burden while in the same time improving the detection performances (section 5).
- From above, the graph displays a hierarchically centralized structure (Figure 1.(c)): the biggest the patch, the more connections it has. It is interesting since the density of features is uniform in the pyramidal scale-space of the image. In other words, the high proportion of smaller keypoints compensates for the previous property. Experimentally, we found out that the number of edges is linear with the number of vertices instead of squared with a complete graph.
- No planarity constraint is imposed. Contrary to a classical Delaunay triangulation [18, 2], our graph is unaffected by node disappearing due to noise.



(a) G_1^m (b) G_2^m (c) $G_3^m = G^m$

Figure 1: Model graph decomposition (here, 3 levels). Smaller features are incorporated as the level increases.

4. PSEUDO-HIERARCHICAL MATCHING

Globally, the graph matching is processed in a top-down manner that starts with the coarsest scale and ends with the finest (contrary to true hierarchical approaches). For each scale level, a probabilistic relaxation is run to find out the best possible mapping between a subset of the model graph and the gradually pruned scene graph. Thanks to this restriction, our method is very fast. The complete algorithm is detailed in algorithm 1 but we now detail the different steps.

4.1 Model Graph Decomposition

To begin with, we decompose the model graph into a set of subgraph $\{G_l^m\}_{l=1}^L$ based on the scale of the keypoints. For each level l, only the features whose scale is superior to a threshold s_l are retained (for the null feature, $\sigma_0 = \infty$ by convention). More specifically, the thresholds are defined such that the coarsest one is equal to a fraction $\rho \in [0, 1]$ of the model object radius w_{obj}, and the finest one to the minimal possible feature scale σ_{min}:

$$s_l = \sigma_{min} \left(\frac{\rho . w_{obj}}{\sigma_{min}} \right)^{\frac{L-l}{L-1}}$$

Hence, $G_l^m = (V^{m,l}, E^{m,l}, X^{m,l})$ with $V^{m,l} = \{\forall i\, v_i^m |\ \sigma_i^m > s_l\}$ (and so on for $E^{m,l}$ and $X^{m,l}$). An example of such a decomposition is given in Figure 1. Note that the graph topology does not change across the levels, i.e. $\forall l < l'$, $E^{m,l} \subseteq E^{m,l'} \subseteq E^m$. We call this decomposition a pseudo-hierarchical structure since, in the coarser levels, no "super-node" or whatsoever is intended to represent a subset of nodes of inferior levels. Instead, smaller nodes are simply appearing in the graph when the level increases. Our approach is thus easy to handle and do not need any specific addition to the mathematical background of [4]. Finally, note that only the model graph is decomposed ; since we do not know in advance the scale of the model object possibly present in the scene graph, it would be impossible to apply the same procedure to this graph.

4.2 Association Graph

As in other graph matching papers [13, 17], we introduce the notion of association graph to describe the discrete space of match hypothesis between the model and scene nodes.

Although the original approach of Christmas could be implemented by updating a single vector of $|V^m|$ probabilities for each scene node at each iteration, we adopted a more flexible architecture to deal with the different operations required by our optimization.

Formally, the association graph $A = (V^A, E^A, X^A, Y^A)$ represents the candidate hypothesis examined during the matching along with their relations of mutual influence. Here, V^A is the set of hypothesis, $X^A = \left\{p^{(n)}\right\}$ the corresponding probabilities estimated at the iteration n, E^A the set of edges and Y^A their associated weight from eq. (1). An illustration of such a graph is given in Figure 2.(a). In the following of this paper, we will refer to an hypothesis $h_{i\alpha} \in V^A$ as a couple of model and scene nodes $h_{i\alpha} = (v_i, u_\alpha)$ and to a null hypothesis as $h_{0\alpha} = (v_0, u_\alpha)$.

Before explaining how to build V^A and E^A from the model and scene graphs, we now describe a set of operations, common to all hierarchy levels, executed on the association graph before, during and after the relaxation process:

4.2.1 Optimizing the relaxation

Prior to the relaxation, we reorganize for each hypothesis in V^A the list of its edges by grouping together the edges connected to neighboring hypothesis having the same model node. This way, the succession of a product and a sum in formula (3) can be evaluated in $O(n_{edge})$ time complexity instead of $O(n_{edge}^2)$ with a naive implementation.

4.2.2 Dynamic Pruning of the graph

To further increase performances, the association graph is lightened at each relaxation iteration by ejecting the hypothesis for which the associated scene node corresponds to the null node with a minimum confidence level (typically, more than 99.9%).

4.2.3 Extraction of the detections

Finally, after completion of the relaxation process, the association graph is processed to extract detections. Firstly, we apply the MAP rule to every scene node, that is we eliminate every non-maximum hypothesis in terms of the posterior probability. Also, every null hypothesis is deleted as well. What remains is a set of connected components $\{C_k = \{h_{i\alpha}\}\}$, each of them representing a single detection in the scene image. Note that the model subgraph $C_k^m = \{v_i\}$ and scene subgraph $C_k^s = \{u_\alpha\}$ derived from C_k are also connected in their respective graph thanks to the construction procedure of the association graph (eq. (4), see next section).

4.3 Matching Initialization

The coarser subgraph G_1^m is used for the initial matching with the scene. Since this graph contains a small number of features, the matching is almost instantaneous. We detail here the required operations:

4.3.1 Hypothesis generation

Before the relaxation process, the unary nodes attributes are used to set the initial probabilities:

$$\begin{aligned} p^{(0)}(u_\alpha \leftarrow v_i) &= p(u_\alpha \leftarrow v_i|\mathbf{x}_\alpha) \\ &= \frac{p(\mathbf{x}_\alpha|u_\alpha \leftarrow v_i)p(u_\alpha \leftarrow v_i)}{\sum_{v_j \in V^m} p(\mathbf{x}_\alpha|u_\alpha \leftarrow v_j)p(u_\alpha \leftarrow v_j)} \end{aligned}$$

with $p(u_\alpha \leftarrow v_i) = \textit{constant}$ since we have no mean of estimating this prior, and:

$$p(\mathbf{x}_\alpha|u_\alpha \leftarrow v_i) = \begin{cases} \phi_i(\mathbf{x}_\alpha) & \text{if } \phi_i(\mathbf{x}_\alpha) > \varepsilon_1, \\ 0 & \text{else.} \end{cases} \quad (6)$$

In the case where $p^{(0)}(u_\alpha \leftarrow v_i)$ is null, then the hypothesis is not considered. We assumed that the measurement noise on the SIFT descriptors follows a Gaussian distribution, that is $\phi_i(\mathbf{x}_\alpha) = \mathcal{N}(x_\alpha; x_i, \Sigma)$ with uniform variance. Moreover, if v_i is the null node, then we set $p(\mathbf{x}_\alpha|u_\alpha \leftarrow v_0) = \eta_1$ (see section 5.1 for how to set ε_1 and η_1).

4.3.2 Edge generation

Looking at eq. (2), one can realize that two hypothesis only needs to be connected if their edge compatibility is not null. Since we already force the compatibility to be null for every pair of hypothesis whose corresponding nodes are not linked in the model graph or in the scene graph by definition of eq. (4), it is sufficient to simply iterate on every model edge e_{ij} and scene edge $e_{\alpha\beta}$, each time connecting the hypothesis $h_{i\alpha}$ and $h_{j\beta}$ (note that the null node is connected to every other nodes in the model graph, including itself), in order to fully initialize E^A.

Practically, the edge compatibility $p(e_{\alpha\beta}|u_\alpha \leftarrow v_i, u_\beta \leftarrow v_j) = y_{i\alpha,j\beta} \in Y^A$ is estimated by extracting 4 locally invariant features from e_{ij} and $e_{\alpha\beta}$:

- the normalized edge length $e_{\alpha\beta}^{(1)} = \|\mathbf{p}_\alpha - \mathbf{p}_\beta\| / (\sigma_\alpha + \sigma_\beta)$,
- the normalized angle $e_{\alpha\beta}^{(2)} = \theta_{\alpha\beta} - \theta_\alpha$,
- the normalized scale difference $e_{\alpha\beta}^{(3)} = |\sigma_\alpha - \sigma_\beta| / max(\sigma_\alpha, \sigma_\beta)$ and
- the angle difference $e_{\alpha\beta}^{(4)} = \theta_\alpha - \theta_\beta$

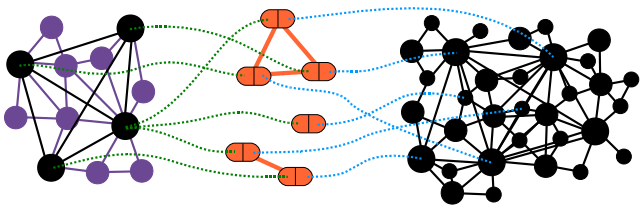
where θ denotes the orientation of a keypoint or an edge. We assumed four independent Gaussian distributions with respect to the model edge descriptors $\left\{e_{ij}^{(n)}\right\}_{n=1}^4$ to calculate the final compatibility. Again, if the result is inferior to a constant threshold ε_2, the edge is ignored, and when the model edge contains the null node, the result is worth η_2 (again, see section 5.1).

4.4 Updating the association graph

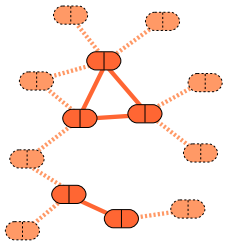
After the first relaxation using G_1^m, we get a set of connected components, each one corresponding to a localized detection in the scene image. Most of those components only contains a single match, i.e. one scene keypoint descriptor was similar to a model one but no consistent point was found in the neighborhood. We consider those detections as insufficient and eliminate them.

Then, the rest of the update algorithm consists in iteratively refining the model (i.e. adding smaller model features) and expanding the connected components in the scene graph (i.e. trying to add neighbors). The expansion step is itself divided in two steps: first, to add new hypothesis involving neighbors of detected nodes (Figure 2.(b)) and, second, to connect the new hypothesis between them (Figure 2.(c)).

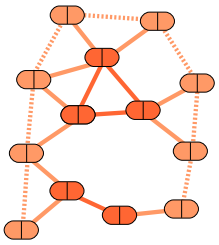
Since the sizes of the considered sets of neighbors are very small with respect to the scene graph, updating the association graph is very fast. Moreover, wrong detections do not



(a) example of an initial matching (coarse). Note that only the model graph is decomposed.



(b) Adding new hypothesis.



(c) Linking them.

Figure 2: (a) Illustration of the association graph (orange nodes) between the model graph (left) and the scene graph (right). (b), (c): Update algorithm (see text for details).

grow over time and hence the matching time do not explode. The final complete matching procedure is summarized in algorithm 1.

5. EXPERIMENTS

5.1 Parameter training

Independent parameters.

The probabilistic framework of Christmas et al. [4] does not require hyper-parameters (contrary to RANSAC, for instance). However, we have to learn instead the constant ε_1, ε_2, η_1 and η_2 during a pseudo training stage independent of the model images.

Practically, we tuned the threshold ε_1 (eq. (6)) so as to eliminate 99% of the candidate hypothesis. It is rather generous, since it virtually amounts to building a visual dictionary of only $1/1\% = 100$ words. For that purpose, we extracted a large number of SIFT descriptors in natural images and performed random pairwise comparisons. Then, η_1 was fixed to the expected value of formula (6) when two random descriptors are used, since it corresponds to a comparison between a known descriptor and an unknown one (the null node).

To fix the value of η_2, we assumed a uniform distribution over the ranges of the four invariants (section 4.3.2), respectively 2, 2π, 1 and 2π, so that $\eta_2 = 1/(8\pi^2)$. We then arbitrarily fixed the threshold ε_2 to $\eta_2/10$.

Finally, the number of relaxation iterations R was set to 2 without observing noticeable loss of performances, proof that the relaxation process converges excessively fast.

Model-dependent parameters.

The parameter χ controls the trade-off between a densely connected proximity graph and a high detection speed. As a consequence, we set this parameter to its minimal value

---

Algorithm 1 Full pseudo-hierarchical matching procedure.

Initialization (level $l = 1$):

1. For each $v_i \in V_1^m$ and for each $u_\alpha \in V^s$:
   Try to generate an hypothesis $h_{i\alpha}$ (section 4.3.1).
2. For each $e_{ij} \in E_1^m$ and for each $e_{\alpha\beta} \in E^s$:
   if $h_{i\alpha} \in V^A$ and $h_{j\beta} \in V^A$: Try to generate an edge between them (section 4.3.2).

Update: For each $l \in [2..L]$:

1. Initialize T to an empty list.
2. For each $h_{i\alpha} \in V^A$:
   Sort $h_{i\alpha}$'s list of edges (section 4.2.1)
3. Repeat R times (number of relaxation iterations):
   - Run one iteration of relaxation (eq. (2)).
   - Prune the association graph (section 4.2.2).
4. Apply MAP and extract the set of connected components $\{C_k\}_{k=1}^C$ (section 4.2.3).
5. if $l = L$: **exit** and return the set of $\{C_k\}$.
6. For each connected component C_k, $k \in C$ (section 4.4):
   Compute the set of neighboring scene nodes $N_k^s = \{u_\beta \in V^s | u_\alpha \in C_k^s, u_\beta \notin C_k^s, e_{\alpha\beta} \in E^s\}$.
   For each $u_\beta \in N_k^s$ and for each $v_j \in V_l^m$:
   - Try to generate a new hypothesis $h_{j\beta}$.
   - If successful: connect $h_{j\beta}$ with C_k and add $h_{j\beta}$ to T.
7. For each hypothesis $h_{j\beta} \in T$ (section 4.4):
   For each v_k neighbor of v_j:
   For each u_γ neighbor of u_β:
   If $h_{k\gamma} \in T$: add an edge between $h_{j\beta}$ and $h_{k\gamma}$

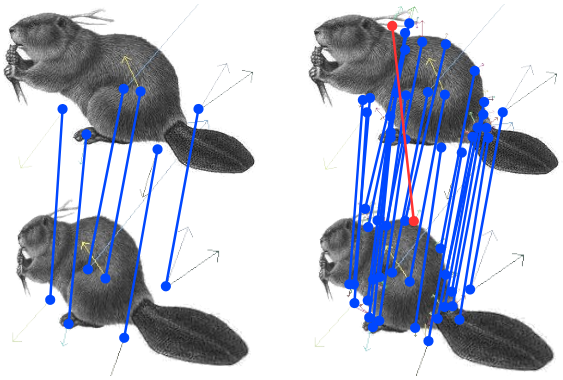


Figure 3: Robustness to a projective transform: matching results using complete graphs (left, 7 matches) and proximity graph (right, 46/47 correct matches).

provided that the model features are sufficiently connected (i.e. $|E^m|/|V^m| \approx 8$). In most cases, a value of $\chi = 1$ produces good results when σ corresponds to the radius of a SIFT patch. The influence of ρ and L is studied in the following experiments.

5.2 Robustness to distortions

In order to validate the alleged property of the proximity graph to withstand non-rigid matching, we now compare the mappings obtained using complete graphs and proximity graphs. To begin with, we present in Figure 3 the matching results between two identical pictures, one of them undergoing a projective distortion (note how the beaver's head is smaller in the bottom row). Only 7 keypoints are matched using complete graphs whereas 46 correct pairs are found using proximity graphs (the same parameters are used in both cases).

Then, robustness to 3D viewpoint change is presented through the CMU hotel dataset [1]. We matched pairs of images separated by a number of frames ranging from $\Delta = 20$ to $\Delta = 80$ using SIFT keypoints. Note that SIFT itself is not affine invariant, making the task even more difficult. Since we do not control the keypoint generation process (i.e. contrary to manual landmarks as in [2]), it is difficult to quantitatively assess the quality of the result, but the results in Figure 4 show that the proposed method succeeds in connecting together the keypoints present on the different facades despite an important viewpoint change.

5.3 Comparison with existing methods

Since our method is straddling two domains (namely, graph matching and object detection), it is difficult to compare with existing graph matching procedures. Indeed, our algorithm requires the existence for each node of a scale and an orientation - in addition to their spatial position and descriptor. Moreover, our model and scene graphs must have a specific structure (i.e. a proximity graph). Unfortunately, those conditions are not fulfilled in standard benchmarks, as for example in the hotel dataset [1, 2] (manual landmarking makes the node scale *per se* unavailable). Instead, we compared against some more traditional object detection methods from the state-of-the-art:

- a baseline RANSAC [11] (with an homography)

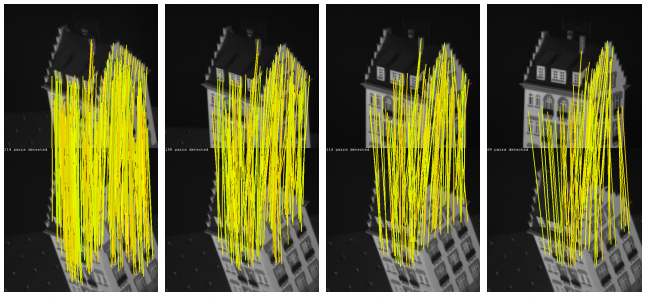


Figure 4: Matching results between pairs of images separated by Δ frames from the hotel dataset [2] (SIFT keypoints are used instead of manual landmarks). The proposed matching remains robust to important 3D viewpoint changes.

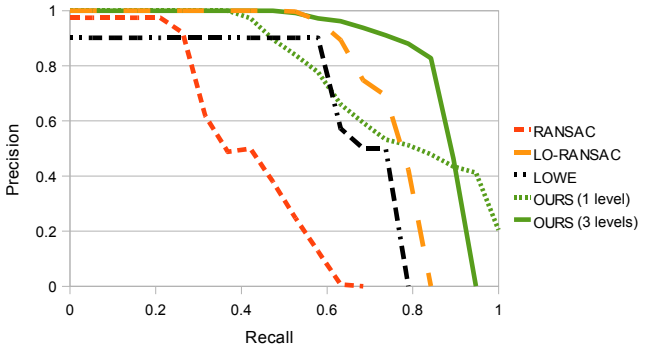


Figure 5: Precision-Recall curves (see text for details).

- Locally Optimized RANSAC (LO-RANSAC) [5, 14] (2D similarity followed by an homography)
- Lowe's method [12] (voting followed by an affine transform)

5.3.1 *Evaluation dataset*

We manually shot two videos with a standard SONY handycam (720x480 px). Since it was taken under realistic viewing conditions for an indoor robot, the videos naturally contains a variety of noises including movement blur, video interlace, poor lightning. The videos were sampled to obtain a set of 400 images (1160 vertices per scene graph on average). Two objects were used to benchmark our method (Figure 6), each of them appearing about 200 times in the test dataset. One close-up image of each object was used to build our model (respectively 225 and 1093 vertices in the model graphs) and to learn the other methods.

5.3.2 *Experimental Results*

Results are presented in Figure 5 in terms of precision-recall (P-R) curves. Precision and recall are defined as N_c/N_d and N_c/N_g, respectively, with N_c the number of correct detections, N_d the total number of detections and N_g the number of ground truth boxes (the higher is the curve, the better). We used a threshold on the cardinality of the connected components (i.e. count of matches) to generate

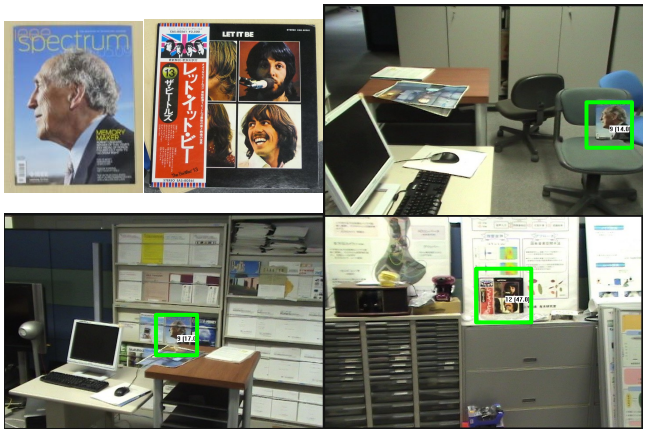


Figure 6: Model objects (top-left) and sample detections (threshold set for 95% precision).

the curves for our method. Some detection examples are shown in Figure 6.

Globally, the proposed method outperforms the others. We mainly explain this fact by our hierarchical procedure and by the distance used between keypoints. Firstly, most keypoints on a model image are very small and blur, hence being quite unspecific. Since our method starts the matching from the bigger keypoints (i.e. the most specific) and progressively adds smaller keypoints, we are less concerned by this issue. A clear evidence of this fact is the difference of precision between the hierarchical method (3 levels) and the same method without hierarchy (1 level contains all the model features) in Figure 5. Secondly, we used an absolute distance between SIFT descriptors whereas the other methods use an approximate relative distance (ratio nearest neighbor/second nearest) computed with a k-d tree. In noisy condition, an absolute distance is more robust although it generates more pairwise hypothesis.

Influence of the parameters L and ρ.

We also investigated the effect in terms of detection performances of the number of levels in the hierarchy L and of the initial scale threshold ρ. We varied in turn ρ and L, each time fixing the alternate parameter to its optimal value. Results are summarized in Figure 7 in terms of P=R measure and average detection time.

Interestingly, the maximal detection performances are reached for intermediate values of ρ and L, namely $[0.2, 0.3]$ for ρ and $[3, 6]$ for L. This corresponds in the first case to a minimum patch size of about 25% of the model size or equivalently, about 6% of the model area. Note that for higher values of ρ not enough features remain in G_1^m and the detection becomes ineffective (hence the P=R measure is null for $\rho \geq 0.35$). Inversely, the detection time logically explodes for lower values of ρ. The number of levels clearly does not have a great importance as long as $L \geq 3$, so setting $L = 3$ seems the best choice since the detection time linearly increases with L. To sum up, hierarchies with more than 3 levels outperform two-pass approaches like [3] without significant increase of processing time.

5.3.3 *Timing Performance*

We compared the average processing time with different

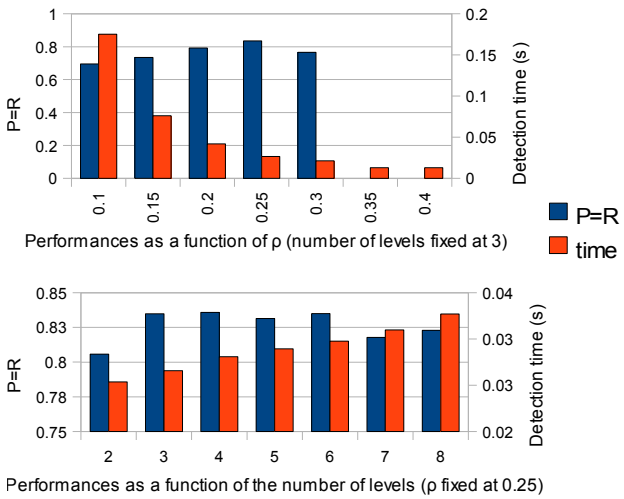


Figure 7: Influence on the performances of the first level threshold ρ (top) and the number of levels L (bottom).

Table 1: Average processing times to detect the two model objects (i.e. one 225-nodes model graph and one 1093-nodes model graph against a scene graph of 1160 nodes on average) for different levels of optimization.

Methods	*time (s)*
Complete graph	100,000
Proximity graph	2.58
Pseudo-hierarchy	0.027
RANSAC	0.118
LO-RANSAC	0.077
Lowe's method	0.128

levels of optimization:

- with complete graphs as in [4],
- with proximity graphs ($L = 1$),
- with proximity graphs and pseudo-hierarchy ($L = 3$).

The results are summarized in Table 1. As can be observed, there is a difference of 5 orders of magnitude between the first and the the second option[1], and again another difference of 2 orders of magnitude between the second and the third one. All in all, the detection speed of the original work [4] was improved by a factor 10^6. Moreover, our method appears to be very competitive compared to the state-of-the-art detectors. This results is however partially due to the fact that the k-d tree used in those methods to index keypoint descriptors is almost empty (only 2 model images), thus generating more matches than it normally would in a realistic case of use.

[1]This result was extrapolated from the number of vertices and edges in the graphs. For reference, a matching between two complete graphs of 163 and 120 vertices takes about 13 s.

6. CONCLUSION

We demonstrated that a pseudo-hierarchical relaxation can be efficient in terms of both computation time and detection performances. It outperforms several state-of-the-art methods in terms of precision-recall curves, and the detection time was reduced by several order of magnitudes compared to the original approach thanks to the proximity graph and to a novel multi-level matching procedure. Although the proposed method is not pure graph matching strictly speaking (since we have put some restrictions on the graph structure), it proved to be very effective and usable in practice contrary to many classical graph matching approaches where the experiments are conducted under heavy simplifications (less than 100 nodes per graph, Delaunay triangulation to reduce as much as possible the number of edges, manual landmarks, and so on). Finally, our system is not yet scalable (the complexity is linear with the number of model objects), but we believe that these results are very encouraging and we are looking forward improving this issue and extending to class object recognition in next works.

7. REFERENCES

[1] CMU 'hotel' dataset: http://vasc.ri.cmu.edu/idb/html/motion/hotel/index.html.

[2] T. S. Caetano, J. J. McAuley, L. Cheng, Q. V. Le, and A. J. Smola. Learning graph matching. In *ICCV*, 2007.

[3] W. J. Christmas, J. Kittler, and M. Petrou. Matching of road segments using probabilistic relaxation: Reducing the computational requirements. In *Sensing, Imaging and Vision for control and guidance of aerospace vehicles, volume SPIE 2220*, pages 169–179, 1994.

[4] W. J. Christmas, J. Kittler, and M. Petrou. Structural matching in computer vision using probabilistic relaxation. *PAMI*, 17:749–764, 1995.

[5] O. Chum, J. Matas, and J. Kittler. Locally optimized ransac. *Pattern Recognition*, pages 236–243, 2003.

[6] O. Chum, M. Perdoch, and J. Matas. Geometric min-hashing: Finding a (thick) needle in a haystack. In *CVPR*, pages 17–24, 2009.

[7] O. Duchenne, F. Bach, I. Kweon, and J. Ponce. A tensor-based algorithm for high-order graph matching. In *CVPR*, 2009.

[8] B. Epshtein and S. Ullman. Feature hierarchies for object classification. In *ICCV*, pages 220–227, 2005.

[9] V. Ferrari, T. Tuytelaars, and L. Van Gool. Simultaneous object recognition and segmentation by image exploration. In *ECCV*, 2004.

[10] B. Fischer, M. Sauren, M. O. Güld, and T. M. Deserno. Scene analysis with structural prototypes for content-based image retrieval in medicine. In J. M. Reinhardt and J. P. W. Pluim, editors, *Medical Imaging 2008: Image Processing*, volume 6914. SPIE, 2008.

[11] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[13] S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *ICDE*, 2002.

[14] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, pages 1–8, 2007.

[15] J. Revaud, G. Lavoué, Y. Ariki, and A. Baskurt. Fast and cheap object recognition by linear combination of views. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 194–201, New York, NY, USA, 2007. ACM.

[16] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *IJCV*, 66(3):231–259, 2006.

[17] L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models and global optimization. In *ECCV*, pages 596–609, 2008.

[18] R. C. Wilson and E. R. Hancock. Graph matching with hierarchical discrete relaxation. *PRL*, 20(10):1041–1052, 1999.