Graphe de proximité invariant en échelle pour la détection rapide d'objets

Jerome Revaud[1] Guillaume Lavoué[1] Ariki Yasuo[2] Atilla Baskurt[1]

[1] Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, F-69621, France
{prenom.nom}@liris.cnrs.fr

[2] CS17, Kobe University, Japon
ariki@kobe-u.ac.jp

Résumé

Une procédure d'appariement de graphes pseudo-hiérarchique dédiée à la reconnaissance d'objets est présentée dans cet article. A partir d'une image modèle, un graphe est construit automatiquement en extrayant des caractéristiques invariantes locales et en les reliant selon une règle dite de proximité. Le graphe résultant présente plusieurs propriétés intéressantes dont l'invariance en échelle, la robustesse à diverses déformations non-rigides et la linéarité du nombre d'arêtes par rapport au nombre de nœuds. Le processus d'appariement est effectué de manière hiérarchique afin d'augmenter la vitesse et les résultats de détection. Par conséquent, même un appariement entre des graphes contenant des milliers de nœuds est très rapide (quelques millisecondes). Des expériences démontrent que la méthode surpasse les détecteurs d'objets spécifiques de l'état de l'art en termes de mesures rappel-précision et de temps de détection.

Mots clefs

Détection d'objets, appariement de graphes, relaxation probabiliste, hiérarchie.

1 Introduction

L'utilisation de points d'intérêt invariants pour la reconnaissance d'objets (e.g. [1]) présente de nombreux avantages : la détection est invariante en translation, en échelle, en rotation et en occultation, cela sans augmentation significative de la complexité grâce à la puissance descriptive élevée des points d'intérêt ; l'apprentissage est inexistant ; ces méthodes sont proches du temps réel ; et enfin elles sont simples à mettre en oeuvre.

Cependant, extraire les points d'intérêt est une chose ; mais détecter l'objet complet en est une autre. On peut grossièrement distinguer deux catégories de méthodes pour ce faire : (1) Les méthodes utilisant une transformation globale, et (2) les méthodes issues de l'appariement de graphe (i.e. utilisant une transformation locale). Jusqu'à présent, les méthodes utilisant une transformation globale ont produit des résultats très convaincants [2, 1, 3, 4]. Idéalement, une transformation projective 3D devrait être systématiquement utilisée, mais le trop grand nombre de paramètres requis conduit souvent à se servir d'une transformation simplifiée (par exemple affine dans [1]) pour approximer la réalité. Un autre problème des transformations globales est leur incapacité à traiter les déformations non rigides, comme par exemple avec un magazine ou un visage. D'un autre côté, l'appariement de graphes apparait comme une solution logique : après avoir extrait quelques points saillants, à la fois l'objet modèle et la scène peuvent être représentés sous forme de graphes. En outre, la comparaison de couples de sommets ou d'arêtes à une échelle locale évite la nécessité d'une transformation globale et donne dans le même temps plus de flexibilité au modèle [5]. En somme, le seul problème avec l'appariement de graphe est qu'il est NP-complet. Malgré tout, les méthodes de relaxation comme [6, 7], historiquement assez vieilles, sont rapides et restent compétitives dans la pratique [8] même si aucune garantie théorique n'assure leur convergence. Puisque nous nous concentrons ici sur une sous-classe de problèmes pour lesquels nous disposons de caractéristiques locales invariantes, la détection peut encore être optimisée grâce à une hiérarchie exploitant les informations complémentaires fournies par les caractéristiques (i.e. leur orientation et leur échelle).

En effet, les hiérarchies se sont révélées être un moyen efficace de réduire la charge de calcul en répartissant les contraintes spatiales sur plusieurs niveaux d'échelle, ce qui améliore en plus la robustesse aux variabilités intra-classe [9]. Christmas et al. [6] ont décrit en 1995 une méthode de relaxation probabiliste qu'ils ont ensuite adapté en une pseudo-hiérarchie dans un article connexe [10]. Elle présente plusieurs avantages : le cadre est minimaliste et simple à utiliser, la méthode est robuste au bruit, et l'algorithme converge rapidement (généralement en moins de 5 itérations). Malheureusement, la hiérarchie mise en oeuvre était simpliste et difficile à généraliser car l'objet recherché ne devait être présent qu'une seule fois dans la scène, le nombre de niveaux utilisés (i.e. 2) était minimal et non-modifiable, et elle nécessitait malgré tout d'utiliser une transformation globale. Même si nous avons utilisé le même cadre théorique, notre méthode est parfaitement adaptée à la détection multi-objets et étend la hiérarchie à un nombre arbitraire de niveaux, sans transformation glo-

bale.
La suite de l'article est organisée comme suit : nous commençons par présenter brièvement la théorie originelle de Christmas et al. [6]. Ensuite, nous introduisons la notion de graphe de proximité dans la section 3. La procédure d'appariement pseudo-hiérarchique est décrite en détail dans la section 4. Enfin, nous démontrons l'efficacité de la méthode dans la section 5 et concluons en section 6.

2 Relaxation Probabiliste

Dans cette section, nous résumons pour le lecteur le cadre probabiliste développé par Christmas et al. dans [6]. Soit deux graphes complets G^m et G^s (respectivement, le graphe modèle le graphe scène), l'objectif de l'appariement est de trouver la meilleure correspondance entre chaque sommet du modèle et de la scène. Dans notre formalisme, $G = (V, E, X)$ où E représente l'ensemble des arêtes, V l'ensemble des sommets et X l'ensemble de leurs mesures unaires associées (dans notre cas, un descripteur SIFT). Le cas de l'isomorphisme de sous-graphes est traité en ajoutant le nœud nul $v_0 \in V^m$ au graphe modèle. En d'autres termes, tous les nœuds étrangers au modèle dans la scène sont tout simplement étiquetés nuls.
Comme dans des travaux similaires, la méthode a besoin de deux types de mesures probabilistes pour estimer la probabilité de correspondances entre les noeuds de la scène et du modèle : (a) la probabilité $p(u_\alpha \leftarrow v_i|x_\alpha)$ d'un appariement noeud-à-noeud en utilisant les attributs unaires uniquement ($u_\alpha \in V^s$, $x_\alpha \in X^s$ et $v_i \in V^m$), et (b) une fonction de compatibilité entre arêtes qui décrit l'affinité entre deux paires locales présumées :

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

avec $e_{\alpha\beta} \in E^s$. Après avoir initialisé les probabilités avec la mesure (a), la relaxation itère jusqu'à convergence selon la règle de mise à jour suivante :

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

où

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

Pour plus de détails, nous renvoyons le lecteur à l'article original [6].

3 Graphe de proximité

Bien que Christmas et al. [6] aient formulé le problème d'appariement avec des graphes complets (i.e. $\forall i \neq j$, $v_i, v_j \in V \times V \Rightarrow e_{ij} \in E$), cela n'est habituellement pas faisable en terme de complexité. Un point très important pour notre système est donc d'être en mesure d'assouplir les contraintes spatiales entre des éléments éloignés. Curieusement, cela reste compatible avec le mécanisme de relaxation de [6] à condition que nous forcions la fonction de densité à valoir zéro lorsque l'arête n'existe pas :

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

Ainsi, on définit simplement le graphe de proximité comme un graphe dans lequel les caractéristiques lointaines ne sont pas connectées. Formellement, nous limitons l'ensemble des arêtes à :

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

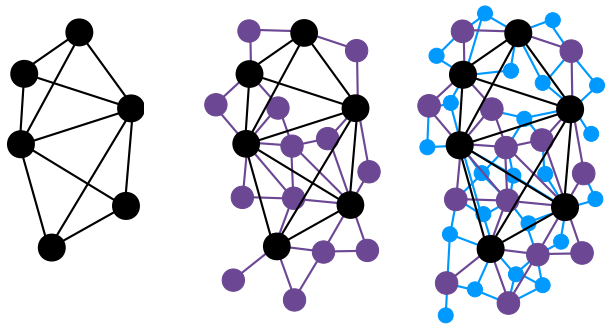
où $\mathbf{p} = (p_x, p_y)$ dénote la position d'un point d'intérêt, σ son échelle et χ est une constante. Cette définition induit plusieurs propriétés intéressantes pour notre application :

- La topologie du graphe est indépendante de l'échelle, c'est à dire que les structures du graphe modèle et du graphe de scène sont invariantes à la taille de l'objet dans l'image.
- Chaque arête du graphe représente une connexion stable. En effet, du point de vue d'un point d'intérêt, le bruit sur la position relative des autres points d'intérêt augmente avec leur distance dans l'espace-échelle pyramidale (i.e. les points plus gros paraissent plus proches).
- Le graphe de proximité permet de réduire sensiblement la charge de calcul tout en améliorant dans le même temps les performances de détection (section 5).
- Globalement, le graphe présente une structure hiérarchique centralisée (voir figure 1.(c)). Cela est du au fait que les patches plus grand possèdent plus de connexions.
- Aucune contrainte de planarité n'est imposée. Contrairement à une triangulation de Delaunay classique [8], notre graphe n'est pas affecté par la disparition de noeuds due au bruit.

4 Appariement pseudo-hiérarchique

Globalement, l'appariement de graphe est traité par une approche descendante qui commence par l'échelle la plus grossière et termine avec la plus petite (contrairement aux vraies approches hiérarchiques). Pour chaque niveau d'échelle, la relaxation probabiliste est exécutée afin de déterminer la meilleure correspondance possible entre un sous-ensemble du graphe modèle et un sous-ensemble du graphe de scène. Grâce à cette restriction, notre méthode est très rapide. L'algorithme complet est détaillée dans l'algorithme 1, mais nous détaillons maintenant les différentes étapes.

4.1 Décomposition du graphe



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

FIGURE 1 – *Décomposition du graphe modèle (ici, 3 niveaux). Les caractéristiques plus petites sont incorporées au fur et à mesure.*

Tout d'abord, nous décomposons le graphe modèle en un ensemble de sous-graphes $\{G_l^m\}_{l=1}^L$ en se basant sur l'échelle des points d'intérêt. Pour chaque niveau l, seuls les éléments dont l'échelle est supérieure à un seuil s_l sont conservés (pour le noeud nul, $\sigma_0 = \infty$ par convention). Plus précisément, les seuils sont définis de telle sorte que le plus gros soit égal à une fraction $\rho \in [0, 1]$ du rayon w_{obj} de l'objet modèle, et le plus petit à l'échelle minimum σ_{min} :

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

Par conséquent, $G_l^m = (V^{m,l}, E^{m,l}, X^{m,l})$ avec$V^{m,l} = \{\forall i \, v_i^m | \sigma_i^m > s_l\}$ (et ainsi de suite pour $E^{m,l}$ et $X^{m,l}$). Un exemple d'une telle décomposition est présenté dans la figure 1. Notez que la topologie graphique ne change pas à travers les niveaux, i.e. $\forall l < l'$, $E^{m,l} \subseteq E^{m,l'} \subseteq E^m$.

4.2 Graphe d'association

Comme dans d'autres articles traitant de sujets similaires [11, 12], nous introduisons la notion de graphe d'association pour décrire l'espace discret des hypothèses de correspondance entre les nœuds du modèle et de la scène.
Formellement, le graphe d'association $A = (V^A, E^A, X^A, Y^A)$ représente les hypothèses candidates examinées durant l'appariement ainsi que leurs relations d'influence réciproque. Ici, V^A est l'ensemble des hypothèses, $X^A = \{p^{(n)}\}$ les probabilités correspondantes estimées à l'itération n, E^A l'ensemble des arêtes et Y^A leur poids associé issu de l'équation (1). Une illustration d'un tel graphe est donné dans la figure 2.(a). Dans la suite de cet article, nous désignons par une hypothèse $h_{i\alpha} \in V^A$ un couple entre un nœud du modèle et un noeud de la scène $h_{i\alpha} = (v_i, u_\alpha)$ et une hypothèse nulle par $h_{0\alpha} = (v_0, u_\alpha)$.
Avant d'expliquer comment construire V^A et E^A à partir du modèle et du graphe scène, nous allons maintenant décrire un ensemble d'opérations communes à tous les niveaux hiérarchiques, exécutées sur le graphe d'association, avant, pendant et après le processus de relaxation :

Élagage dynamique du graphe. Pour augmenter encore les performances, le graphe d'association est élagué à chaque itération de relaxation en éjectant les hypothèses pour lesquelles le noeud scène associé correspond au nœud nul avec une certaine confiance (généralement, plus de 99,9%).

Extraction des détections. Enfin, après l'achèvement du processus de relaxation, le graphe d'association est traité pour en extraire les détections. Premièrement, nous appliquons la règle du MAP pour chaque nœud de la scène, i.e. nous éliminons toute hypothèse non-maximale en terme de probabilité a posteriori. De plus, chaque hypothèse nulle est également supprimé. Il reste un ensemble de composantes connexes $\{C_k = \{h_{i\alpha}\}\}$, chacune d'elles représentant une détection unique dans l'image scène. Notez que le sous-graphe modèle $C_k^m = \{v_i\}$ et le sous-graphe scène $C_k^s = \{u_\alpha\}$ dérivés de C_k sont également connexes dans leur graphe respectif étant donné la construction du graphe d'association (éq. (4), voir section suivante).

4.3 Initialisation de l'appariement

Le sous-graphe grossier G_1^m est utilisé pour l'appariement initial. Comme ce graphe contient un petit nombre de caractéristiques, le calcul est presque instantané. Nous détaillons ici les opérations nécessaires :

Génération des hypothèses. Avant le processus de relaxation, les attributs unaire des nœuds sont utilisés pour fixer les probabilités de départ :

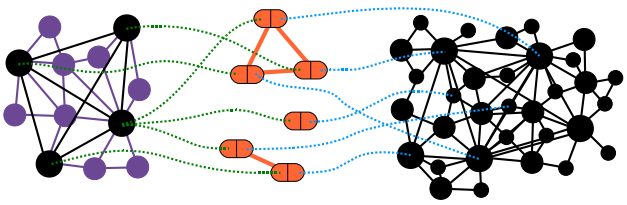
$$\begin{aligned} p^{(0)}(u_\alpha \leftarrow v_i) &= p(u_\alpha \leftarrow v_i | \mathbf{x}_\alpha) \\ &= \frac{p(\mathbf{x}_\alpha | u_\alpha \leftarrow v_i) p(u_\alpha \leftarrow v_i)}{\sum_{v_j \in V^m} p(\mathbf{x}_\alpha | u_\alpha \leftarrow v_j) p(u_\alpha \leftarrow v_j)} \end{aligned}$$

avec $p(u_\alpha \leftarrow v_i) = cste$ puisqu'on ne peut pas l'estimer, et :

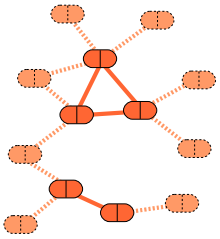
$$p(\mathbf{x}_\alpha | u_\alpha \leftarrow v_i) = \begin{cases} \phi_i(\mathbf{x}_\alpha) & \text{si } \phi_i(\mathbf{x}_\alpha) > \varepsilon_1, \\ 0 & \text{sinon.} \end{cases} \quad (6)$$

Dans le cas où $p^{(0)}(u_\alpha \leftarrow v_i)$ est nulle, alors l'hypothèse n'est pas considérée. Nous avons supposé que le bruit de mesure sur les descripteurs SIFT suit une distribution gaussienne, c'est-à-dire $\phi_i(\mathbf{x}_\alpha) = \mathcal{N}(x_\alpha; x_i, \Sigma)$ avec une variance uniforme. En outre, si v_i est le noeud nul, alors on impose $p(\mathbf{x}_\alpha | u_\alpha \leftarrow v_0) = \eta_1$ (voir section 5.1 pour savoir comment régler ε_1 et η_1).

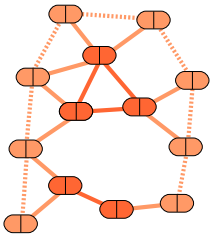
Génération des arêtes. En regardant l'éq. (2), on voit que deux hypothèses ne doivent être connectées que si leur compatibilité d'arêtes n'est pas nulle. Puisque nous avons déjà forcé la compatibilité à être nulle pour chaque paire d'hypothèses dont les noeuds correspondants ne sont pas



(a) Exemple d'un appariement initial (grossier)



(b) Ajouter des nouvelles hypothèses



(c) et les connecter.

FIGURE 2 – *(a) Illustration du graphe d'association (noeuds oranges) entre le graphe modèle (à gauche) et le graphe scène (à droite). (b), (c) : Algorithme de mise à jour (voir le texte).*

liés dans le graphe modèle ou dans le graphe de scène, par définition de (4), il suffit de simplement itérer sur chaque arête du modèle e_{ij} et chaque arête de la scène $e_{\alpha\beta}$, en reliant à chaque fois les hypothèses $h_{i\alpha}$ et $h_{j\beta}$ (à noter que le nœud nul est connecté à tous les autres noeuds dans le graphe modèle, y compris lui-même), afin d'initialiser complètement E^A.

En pratique, la compatibilité d'arêtes $p(e_{\alpha\beta}|u_\alpha \leftarrow v_i, u_\beta \leftarrow v_j) = y_{i\alpha,j\beta} \in Y^A$ est estimée en extrayant 4 invariants de e_{ij} et $e_{\alpha\beta}$:

– La longueur de l'arête $e^{(1)}_{\alpha\beta} = \|\mathbf{p}_\alpha - \mathbf{p}_\beta\| / (\sigma_\alpha + \sigma_\beta)$,

– l'angle de l'arête $e^{(2)}_{\alpha\beta} = \theta_{\alpha\beta} - \theta_\alpha$,

– la différence d'échelle $e^{(3)}_{\alpha\beta} = |\sigma_\alpha - \sigma_\beta| / max(\sigma_\alpha, \sigma_\beta)$ et

– La différence d'angle $e^{(4)}_{\alpha\beta} = \theta_\alpha - \theta_\beta$

où θ désigne l'orientation d'un point d'intérêt ou d'une arête. Nous avons supposé quatre distributions gaussiennes indépendantes par rapport au descripteur d'arête $\left\{e^{(n)}_{ij}\right\}_{n=1}^{4}$ pour calculer la compatibilité finale. De même qu'avant, si le résultat est inférieur à un seuil constant ε_2, l'arête est ignorée, et lorsque l'arête modèle contient le noeud nul, le résultat vaut η_2 (voir la section 5.1).

4.4 Mise à jour du graphe d'association

Après la première relaxation en utilisant G^m_1, on obtient un ensemble de composantes connexes, chacune correspondant à une détection localisée dans l'image scène. La plupart de ces composantes ne contiennent qu'une seule paire, i.e. un descripteur de la scène était semblable à un descripteur du modèle, mais aucun autre paire en accord n'a été trouvée dans le voisinage. Nous estimons que ces détections sont insuffisantes et les éliminons.

Puis, le reste de l'algorithme de mise à jour consiste à raffiner itérativement le modèle (à savoir ajouter les caractéristiques plus petites du modèle) en élargissant les composantes connexes dans le graphe de scène (à savoir essayer d'ajouter les voisins). L'étape d'expansion est elle-même divisée en deux étapes : d'abord, ajouter de nouvelles hypothèses impliquant les voisins de noeuds détectées (fig. 2.(b)) et, ensuite, pour relier les nouvelle hypothèses entre elles (fig. 2.(c)). La procédure complète est résumée dans l'algorithme 1.

Algorithm 1 Algorithme complet de la procédure d'appariement pseudo-hiérarchique.

Initialisation (niveau $l = 1$) :

1. Pour chaque $v_i \in V^m_1$ et pour chaque $u_\alpha \in V^s$:
   Essayer de générer une hypothèse $h_{i\alpha}$ (section 4.3).
2. Pour chaque $e_{ij} \in E^m_1$ et Pour chaque $e_{\alpha\beta} \in E^s$:
   Si $h_{i\alpha} \in V^A$ et $h_{j\beta} \in V^A$: essayer de générer une arête entre elles (section 4.3).

Mise à jour : Pour chaque $l \in [2..L]$:

1. Répéter R fois (nombre d'itération de relaxation) :
   - Exécuter une itération de relaxation (éq. (2)).
   - Élaguer le graphe d'association (section 4.2).
2. Appliquer le MAP et extraire l'ensemble de composantes connexes $\{C_k\}_{k=1}^{C}$ (section 4.2).
3. Si $l = L$: sortir et retourner l'ensemble des $\{C_k\}$.
4. Créer une liste vide T.
5. Pour chaque composante connexe C_k, $k \in C$ (section 4.4) :
   Calculer l'ensemble des noeuds voisins dans la scène $N^s_k = \{u_\beta \in V^s_l | u_\alpha \in C^s_k, u_\beta \notin C^s_k, e_{\alpha\beta} \in E^s\}$.
   Pour chaque $u_\beta \in N^s_k$ et chaque $v_j \in V^m_l$:
   - Essayer de générer une nouvelle hypothèse $h_{j\beta}$.
   - Si succès : connecter $h_{j\beta}$ avec C_k et ajouter $h_{j\beta}$ à T.
6. Pour chaque hypothèse $h_{j\beta} \in T$ (section 4.4) :
   Pour chaque v_k voisin de v_j et chaque u_γ voisin de u_β :
   Si $h_{k\gamma} \in T$: ajouter une arête entre $h_{j\beta}$ et $h_{k\gamma}$

5 Expérimentations

5.1 Apprentissage des paramètres

Paramètres indépendants Le cadre probabiliste développé par Christmas et al. [6] ne nécessite pas d'hyperparamètres (contrairement à RANSAC, par exemple). Toutefois, nous avons à apprendre à la place les constantes ε_1, ε_2, η_1 et η_2 au cours d'une phase de pseudo-apprentissage indépendante du modèle.

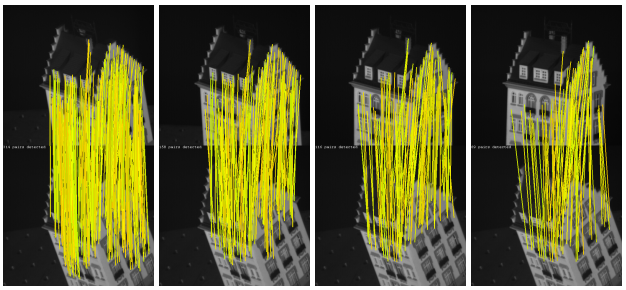
Concrètement, nous avons calibré le seuil ε_1 (éq. (6)) de manière à éliminer 99% des hypothèses candidates. C'est plutôt généreux, puisque cela équivaut virtuellement à utiliser un dictionnaire visuel de seulement $1/1\% = 100$ mots. Pour cela, nous avons extrait un grand nombre de descripteurs SIFT dans des images naturelles et avons effectué des comparaisons aléatoires. Ensuite, η_1 a été fixé à l'espérance

de la formule (6) lorsque deux descripteurs aléatoires sont utilisés, car cela correspond à une comparaison entre un descripteur connu et un inconnu (le nœud nul).
Pour fixer la valeur de η_2, nous avons supposé une répartition uniforme sur les intervalles des quatre invariants (section 4.3), respectivement 2, 2π, 1 et 2π, de sorte que $\eta_2 = 1/(8\pi^2)$. Nous avons alors fixé arbitrairement le seuil ε_2 à $\eta_2/10$.
Enfin, le nombre d'itérations de relaxation R a été fixé à 2 sans observer de perte notable de performances, preuve que le processus de relaxation converge très rapidement.

Paramètres dépendants du modèle Le paramètre χ contrôle le compromis entre un graphe de proximité densément connecté et une rapidité de détection élevée. En conséquence, nous fixons ce paramètre à sa valeur minimale à condition que les caractéristiques du modèle sont suffisamment connectées (i.e. $|E^m| \,/\, |V^m| \approx 8$). Dans la plupart des cas, une valeur de $\chi = 1$ produit de bons résultats lorsque σ correspond au rayon d'un patch SIFT. L'influence de ρ et L est étudiée dans les expériences suivantes.

5.2 Robustesse aux déformations 3D

La robustesse au changement de point de vue 3D est présentée à travers la base CMU-hotel [13]. Nous avons comparé des paires d'images séparées par un nombre d'images allant de $\Delta = 20$ à $\Delta = 80$ en utilisant les points d'intérêt SIFT. Les résultats de la figure 3 montrent que la méthode proposée réussit à reconnaître les points présents sur les différentes façades malgré un changement de point de vue important.



$\Delta = 20$	$\Delta = 40$	$\Delta = 60$	$\Delta = 80$
314 paires	158 paires	116 paires	69 paires

FIGURE 3 – *Résultats d'appariements entres des paires d'images séparées par Δ frames de la base CMU-hotel [8] (des points SIFT sont utilisés au lieu de balises posées manuellement). L'approche proposée reste robuste à un changement de point de vue 3D important.*

5.3 Comparaison avec des méthodes existantes

Étant donné que notre méthode est à cheval sur deux domaines (à savoir, l'appariement des graphes et la détection d'objets), il est difficile de se comparer avec des méthodes d'appariement de graphes existantes. En effet, notre algorithme nécessite l'existence, pour chaque nœud d'une échelle et d'une orientation - en plus de leur position dans l'espace et de leur descripteur. En outre, nos graphes modèle et scène doivent avoir une structure spécifique (i.e. un graphe de proximité). Malheureusement, ces conditions ne sont pas remplies dans la plupart des base de test, comme par exemple la base CMU-hotel [13, 8] (30 points d'intérêt manuellement définis, échelle non disponible). Au lieu de cela, nous avons comparé contre certaines méthodes de détection d'objets plus traditionnelles de l'état de l'art :
– un RANSAC basique [2] (avec une homographie)
– Locally Optimized RANSAC (LO-RANSAC) [14, 4] (similitude 2D suivie d'une homographie)
– la méthode de Lowe [1] (vote/Hough suivi d'une transformation affine)

Base d'évaluation. Nous avons filmé deux courtes vidéos avec une caméra Sony Handycam (720x480 px). Comme les vidéos ont été prises dans des conditions réalistes pour un robot d'intérieur, les vidéos contiennent naturellement une variété de bruits divers, dont du flou de bougé, de l'entrelacement vidéo, un éclairage criard. Les vidéos ont été échantillonnées pour obtenir un ensemble de 400 images (1160 noeuds par graphe scène en moyenne). Deux objets ont été utilisés pour évaluer notre méthode (fig. 5), chacun d'eux apparaissant environ 200 fois dans l'ensemble de données de test. Une image en gros plan de chaque objet a été utilisée pour construire le modèle (respectivement 225 et 1093 sommets dans les graphes modèles) et pour initialiser les autres méthodes.

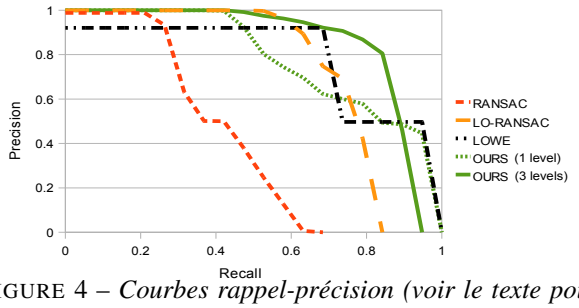


FIGURE 4 – *Courbes rappel-précision (voir le texte pour plus de détails).*

Résultats expérimentaux. Les résultats sont indiqués dans la fig. 4 en termes de courbes rappel-précision. Précision et rappel sont définis comme N_c/N_d et N_c/N_g, respectivement, avec N_c le nombre de détections correctes, N_d le nombre total de détections et N_g le nombre de boîtes de la vérité terrain (le plus haut est la courbe, meilleur est le résultat). Nous avons utilisé un seuil sur la cardinalité des composantes connexes (i.e. nombre de paires) pour générer les courbes avec notre méthode. Quelques exemples de détection sont présentés dans la fig. 5.
Globalement, la méthode proposée surpasse les autres. Nous expliquons ce fait principalement par notre procédure hiérarchique et par la distance utilisées entre les points d'intérêt. En effet, l'utilisation d'une hiérarchie (courbe "OURS 3 levels" dans la fig. 4) augmente notoirement les performances par rapport à la même méthode sans hié-

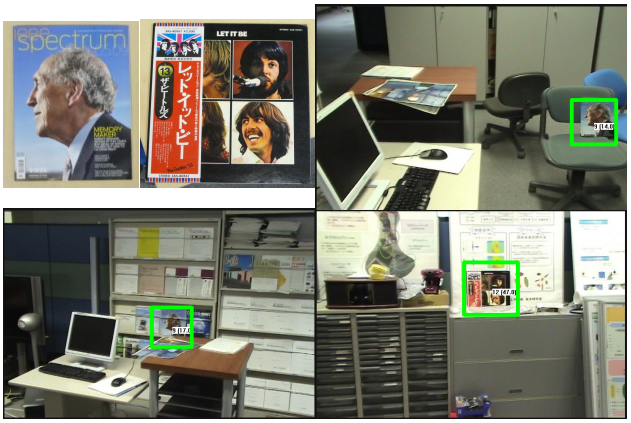


FIGURE 5 – *Objets modèles (en haut à gauche) et exemple de détections (seuil fixé à 95% de précision).*

rarchie (courbe "OURS 1 level"). Par ailleurs, une distance absolue entre keypoints est plus robuste au bruit, bien qu'elle génère plus d'hypothèses de paires.

Influence des paramètres L et ρ. Nous avons tour à tour fait varier ρ et L, chaque fois en fixant l'autre paramètre à sa valeur optimale. Fait intéressant, les performances de détection maximale sont atteintes pour des valeurs intermédiaires de ρ et L, à savoir entre 0.2 et 0.3 pour ρ et entre 3 et 6 pour L. Pour des valeurs élevées de ρ, il ne reste plus assez de caractéristiques dans G_1^m et la détection devient logiquement impossible. Le nombre de niveaux n'a pas une grande importance tant que $L \geq 3$, fixer L à 3 semble donc être le meilleur compromis car le temps de détection augmente linéairement avec L.

Temps d'exécution. Nous avons mesuré les temps moyens de traitement pour détecter les deux objets du modèle (deux graphes modèle de 225 et 1093 noeuds contre un graphe scène de 1160 nœuds en moyenne) avec différents niveaux d'optimisation :
– avec des graphes complets [1] comme dans [6] : $\approx 10^5$ s,
– avec des graphes de proximité ($L = 1$) : 2,58 s,
– avec des graphes de proximité et une pseudo-hiérarchie ($\rho = 0.2$, $L = 3$) : 0,027 s.
Comme on peut le remarquer, il y a une différence de 5 ordres de grandeur entre la première et la deuxième option, et encore un écart de 2 ordres de grandeur entre la deuxième et la troisième option. Au final, la vitesse de détection de l'article original [6] a été améliorée d'un facteur 10^6. En outre, notre méthode semble très compétitive par rapport aux détecteurs de l'état-de-l'art qui affiche chacun des temps de détection moyens de 100 ms environs.

6 Conclusion

Nous avons montré qu'une relaxation pseudo-hiérarchique peut être efficace en termes de temps de calcul et de performances de détection. Elle surpasse plusieurs méthodes de l'état de l'art en termes de courbes rappel-précision, et le temps de détection a été réduit de plusieurs ordres de grandeurs par rapport à l'approche originale grâce au graphe de proximité et à une procédure d'appariement multi-niveau novatrice. Nous pensons que ces résultats sont très encourageants et nous essayerons d'améliorer cet aspect ainsi que d'étendre notre méthode à la reconnaissance de classes d'objets dans des travaux futurs.

1. Ce résultat a été extrapolé à partir du nombre de noeuds et d'arêtes dans le graphe. Pour référence, un appariement entre deux graphes complets de 163 et 120 sommets prend environ 13 s.

Références

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

[2] Martin A. Fischler et Robert 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.

[3] Fred Rothganger, Svetlana Lazebnik, Cordelia Schmid, et Jean Ponce. 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *International Journal of Computer Vision*, 66(3) :231–259, 2006.

[4] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, et Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. Dans *Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[5] V. Ferrari, T. Tuytelaars, et L.J. Van Gool. Simultaneous object recognition and segmentation by image exploration. Dans *ECCV*, 2004.

[6] William J. Christmas, Josef Kittler, et Maria Petrou. Structural matching in computer vision using probabilistic relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17 :749–764, 1995.

[7] S. Gold et A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18 :377 – 388, 1996.

[8] Tiberio S. Caetano, Julian J. McAuley, Li Cheng, Quoc V. Le, et Alex J. Smola. Learning graph matching. Dans *International Conference on Computer Vision*, 2007.

[9] Boris Epshtein et Shimon Ullman. Feature hierarchies for object classification. Dans *International Conference on Computer Vision*, pages 220–227, 2005.

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

[11] Sergey Melnik, Hector Garcia-Molina, et Erhard Rahm. Similarity flooding : A versatile graph matching algorithm and its application to schema matching. Dans *ICDE*, 2002.

[12] Lorenzo Torresani, Vladimir Kolmogorov, et Carsten Rother. Feature correspondence via graph matching : Models and global optimization. Dans *European Conference on Computer Vision*, pages 596–609, 2008.

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

[14] Ondřej Chum, Jií Matas, et Josef Kittler. Locally optimized ransac. *Pattern Recognition*, pages 236–243, 2003.