



**HAL**  
open science

## Reconnaissance d'objets par indexation géométrique étendue.

Bart Lamiroy, Patrick Gros

► **To cite this version:**

Bart Lamiroy, Patrick Gros. Reconnaissance d'objets par indexation géométrique étendue.. Actes des Journées ORASIS GDR-PRC Communication Homme-Machine, May 1996, Clermont-Ferrand, France. pp.19–24. <inria-00590066>

**HAL Id: inria-00590066**

**<https://inria.hal.science/inria-00590066v1>**

Submitted on 5 May 2011

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# Reconnaissance d'objets par indexation géométrique étendue.

Bart Lamiroy & Patrick Gros  
Équipe MOVI  
GRAVIR – IMAG & INRIA Rhône-Alpes  
655, avenue de l'Europe  
38330 Montbonnot Saint Martin, FRANCE

e-mail : `Bart.Lamiroy@imag.fr`

## 1 Introduction.

Plusieurs travaux traitent déjà du problème de la reconnaissance. Certains utilisent le paradigme de prédiction – vérification. Leur efficacité dépend en général de l'implémentation du parcours d'arbre qui est inhérent à ce problème. Lamdan et Wolfson [7] introduisaient les premiers la notion de *Geometric Hashing* ou l'indexation géométrique. Cette méthode repose, comme celles de la classe précédente, sur l'hypothèse d'un mouvement rigide entre l'image inconnue et son modèle. Par contre, la recherche arborescente est réduite à une indexation par tables de *hachage*, ce qui réduit considérablement la complexité du problème.

Pour s'affranchir de la contrainte de rigidité, d'autres auteurs ont essayé d'implémenter des systèmes basés sur des critères plus topologiques [6]. Ils sont malheureusement très sensibles au bruit. D'autres méthodes, comme celles basées sur des approches stochastiques, ont l'inconvénient d'être très fortement liées à la modélisation initiale du problème, et par conséquent, manquent d'une flexibilité indispensable.

Notre méthode se rapproche plus du *Geometric Hashing*. Par contre cette dernière résout uniquement le problème de l'indexation, et non pas celui de la mise en correspondance. Nous nous proposons de faire les deux simultanément, permettant ainsi de retrouver rapidement un (ou des) modèle(s) dans une image inconnue et de donner des informations quantitatives quant à la position des primitives qui ont participé à la reconnaissance et leur mise en correspondance avec celles du modèle trouvé.

## 2 Principes généraux

Notre approche s'appuie principalement sur les travaux décrits dans [5] et est reprise en plus détaillé dans [8]. Nous nous servons d'un algorithme de mise en correspondance de primitives pour déterminer le modèle qui correspond le mieux à une image inconnue. Nous esquisserons rapidement cet algorithme, et nous reviendrons sur la partie indexation et la complexité dans la section suivante.

Certaines similarités avec [9] pourraient être remarquées. Nous montrerons dans la conclusion qu'il y a une différence d'approche entre ces deux méthodes.

## 2.1 Mise en correspondance de 2 images.

L'hypothèse générale qui régit cet algorithme consiste à dire que le mouvement apparent entre deux images qui représentent un même objet, peut être approché par une similitude quand la différence de point de vue n'est pas très grande [5].

Pour l'exploiter dans une méthode de mise en correspondance, il suffit de calculer localement, pour chacune des primitives le mouvement apparent. Un critère global permettra ensuite de sélectionner la meilleure solution. En pratique nous énumérons toutes les configurations de deux segments concourants dans chacune des images. Pour chacune d'entre elles nous calculons le quasi-invariant [2] associé, formé par l'angle et le rapport de longueurs des deux segments intervenant. On peut donc considérer que les couples de configurations qui ont des quasi-invariants associés semblables forment des appariements plausibles.

Chaque couple de configurations ainsi sélectionné définit un mouvement apparent (exact) qui est une similitude. En votant dans l'espace des paramètres de cette famille de transformations nous trouverons un amas plus dense à l'endroit qui définit le mouvement apparent global entre les deux images. Il suffit de prendre les paires de configurations qui participent à cet amas pour obtenir les appariements définitifs entre les primitives des deux images.

## 2.2 Reconnaissance.

L'algorithme décrit ci-dessus est maintenant facilement adaptable à la recherche d'un modèle (parmi plusieurs) qui correspond au mieux à une image inconnue. En ayant, au préalable, stocké les quasi-invariants de chaque modèle dans une table de hash bidimensionnelle dont les clefs de hashage sont les composantes de ces derniers, on peut confronter notre image à cette base de modèles.

L'algorithme se décompose en deux parties principales :

1. On extrait les quasi-invariants de l'image. Ensuite, pour chaque invariant, nous recherchons dans la table de hash les invariants qui lui correspondent. Ceci nous donne une liste d'appariements plausibles entre invariants, et par conséquent entre primitives. Ces primitives n'appartiennent plus à un seul modèle comme dans le cas précédent, mais à un ensemble de modèles différents.
2. Pour chaque couple faisant intervenir un modèle particulier, on exprime un vote dans l'espace de transformations qui lui est propre. À la fin, il suffit de prendre le modèle qui contient le plus grand amas de votes mutuellement consistants.

## 3 Algorithme

Dans cette partie nous reviendrons plus en détail sur la partie indexation. Nous donnerons ensuite quelques éléments de la complexité de l'algorithme.

### 3.1 Indexation

Le contexte de reconnaissance et de mise en correspondance étant défini dans la partie précédente, nous allons détailler notre façon d’indexer les quasi-invariants.

Il est clair [1, 3] que des méthodes d’indexation ne sont optimales que quand les clés ont une distribution uniforme dans leur espace d’indexation. Si on considère un segment sur lequel on définit comme origine  $\mathcal{O}$  l’une de ses extrémités, la distribution des quasi-invariants  $(\theta, \rho)$  angle et rapport des longueurs formés par des segments aléatoires partageant l’extrémité  $\mathcal{O}$ , ne sera pas uniforme. L’uniformité de cette distribution est pourtant indispensable. Pour une utilisation optimale de la table d’indexation et afin de minimiser la perte de précision lors du stockage dans certaines régions, nous appliquerons une transformation sur nos invariants initiaux pour obtenir une répartition uniforme.

L’angle  $\theta$ , a déjà les bonnes propriétés. Par contre, il est difficile de comparer deux valeurs dans l’intervalle  $[0, 2\pi[$  puisqu’elles sont définies modulo  $\pi$ . Il est évident, par contre, que l’on peut toujours considérer le plus petit angle formé par les deux segments. On obtient ainsi des valeurs dans l’intervalle  $[0, \pi[$ . Celles-ci conservent la propriété de la distribution uniforme et éliminent la contrainte sur la comparaison.

Nous pouvons résoudre le problème de la distribution non-uniforme du rapport de longueurs  $\rho$  de façon similaire. Au lieu de considérer la valeur  $\rho = \frac{l_1}{l_2}$ , qui a une répartition non-uniforme, nous pouvons utiliser  $\rho' = \ln(\rho) = \ln(l_1) - \ln(l_2)$  qui se comporte exactement comme nous le souhaitons.

Le problème de l’indexation étant résolu, le stockage des invariants des modèles ne pose plus de problème. Dans chaque entrée de notre table d’indexation correspondant à la discrétisation d’un quasi-invariant particulier, nous stockons les références des modèles (et leur configurations qui ont engendré l’invariant).

Puisque nous avons affaire à des informations visuelles réelles, les invariants calculés sont assujettis à du bruit. De plus, leur valeur n’est que quasi-invariante par rapport aux transformations projectives qui régissent le passage 3D-2D. Il est donc nécessaire de comparer les invariants d’une image avec ceux des modèles à un  $\varepsilon$  près. Cet  $\varepsilon$  est indépendant de la valeur des invariants. Il est donc possible de définir la table d’indexation comme une grille  $\varepsilon \times \varepsilon$ .

Supposons maintenant que nous voulons comparer un invariant d’une image à ceux de la base de modèles à  $\varepsilon$  près. Nous pouvons procéder selon l’une des deux façons suivantes (l’une optimise le temps d’exécution, l’autre, l’espace mémoire utilisé).

- Puisque  $\varepsilon$  est connu au moment de la construction de la base de modèles, il est possible d’enregistrer plusieurs instances du même invariant dans cette base. Pour que l’invariant recouvre la totalité la zone d’incertitude de rayon  $\varepsilon^1$  (et sachant que la granularité de la base des modèles est de  $\varepsilon \times \varepsilon$ ) il nous suffit de copier une instance dans chacune des cases voisines. Au moment de la reconnaissance il suffit de vérifier toutes les instances qui se trouvent à l’endroit défini par un invariant de l’image inconnue pour connaître tous ceux qui se trouvent dans un rayon de  $\varepsilon$ .

---

1. En réalité,  $\varepsilon$  est un vecteur multidimensionnel, définissant une “ellipsoïde” d’incertitude autour de chaque invariant. Nous supposons, sans perte de généralité, qu’il s’agit d’un même scalaire dans toutes les directions. Nous pourrions ainsi parler d’un “rayon” d’incertitude comme s’il s’agissait d’un cercle.

- Il est possible d’obtenir des résultats similaires de n’enregistrer qu’une seule instance de chaque invariant par modèle. Au moment de la reconnaissance il suffit de regarder dans toutes les cases dans un voisinage de  $\varepsilon$  autour de l’invariant de l’image à reconnaître.

### 3.2 Complexité

Nous revenons ici sur la complexité des deux parties spécifiées dans la section 2.2. Dans ce qui suit nous supposons que  $i$  représente la quantité d’invariants dans l’image à reconnaître, tandis que  $m$  est le nombre de modèles présents dans la base des modèles.

1. La partie liée directement à l’indexation qui correspond à la confrontation des invariants de l’image à la base des modèles est indépendante de  $m$  et est linéaire en  $i$ . On peut donc dire que cette partie est de complexité  $\mathcal{O}(i)$ . Par contre, le résultat de l’opération est une liste qui dépend bien de  $m$ .
2. La partie de vote est purement linéaire en nombre d’appariements trouvés lors de l’étape précédente. Or étant donné que celui-ci est de l’ordre de  $\mathcal{O}(i \times m)$ , la totalité de cette partie a la même complexité.

## 4 Résultats

Nous avons effectué deux types de test. L’un utilisant des images et des modèles synthétiques, l’autre des réels. La mise en œuvre du premier type était la suivante : nous avons pris environ 200 images sur une sphère de vue d’un modèle CAO. Ces images ont été traitées automatiquement afin d’obtenir 34 “aspects” représentatifs de l’objet. La notion d’*aspect* faisant ici référence à des aspects visuels, et non pas à la notion topologique que l’on rencontre plus souvent dans la littérature. Ces aspects formaient la base de modèles.

Nous avons obtenu un taux de reconnaissance de 100% lorsque les aspects étaient présentés comme images à reconnaître. Lorsque nous avons présenté les 200 images initiales au système, nous avons obtenu un taux de reconnaissance de 92,6%, la plupart des erreurs provenant d’aspects dégénérés. En guise de comparaison nous avons effectué le même test en supprimant la partie de calcul de transformation dans le vote, le réduisant ainsi à un simple vote majoritaire pour chaque modèle. Le taux de reconnaissance a chuté à 18,8%.

En utilisant des images et des modèles réels, nous avons confronté le système à deux configurations de test différentes : la première montre des images contenant des objets sous un aspect similaire que l’un de ceux dans la base de modèles. La seconde présente des objets inconnus ou des objets connus sous un aspect inconnu. Dans le premier cas, le taux de reconnaissance est de 94%. Dans le second cas les poids associés aux modèles ressortant du processus de vote sont nettement inférieurs aux poids des modèles trouvés dans le premier cas. Ceci confirme donc bien l’efficacité et les limites de notre approche.

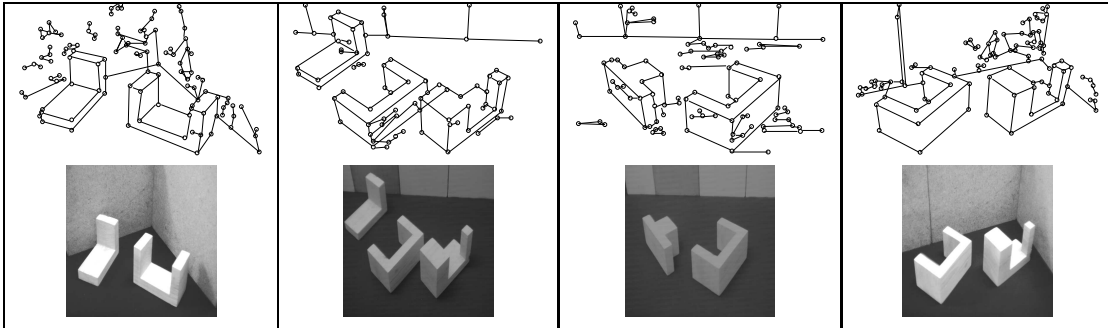


FIG. 1 – Exemples d'images utilisées pour la reconnaissance.

## 5 Discussion.

Cette méthode présente des avantages par rapport à la méthode présentée par Lamdan et Wolfson [7]. Premièrement, le fait d'utiliser des configurations plus contraignantes que des configurations aléatoires de 4 points introduit un facteur topologique. Étant donné que ces configurations correspondent à des "réalités" dans l'image, elles sont par construction plus discriminantes. Il y a toutefois un facteur encore plus important : l'utilisation des quasi-invariants et le vote de Hough qui leur est associé. Dans [4] une sévère critique est émise quant à la stabilité de la méthode de la transformée de Hough. La conclusion des auteurs prouve qu'à partir d'une complexité moyenne des images ou de la base de modèles la méthode acquiert un comportement complètement aléatoire. Notre approche nous préserve de ce défaut par le biais des invariants, et dans une moindre mesure, de l'utilisation de votes plus discriminants. Les invariants empêchent l'algorithme de mettre en correspondance n'importe quelle combinaison de configurations et le dirigent vers les configurations qui ont une grande probabilité d'être porteuses d'information pertinente. Leur utilisation réduit en même temps considérablement la complexité du problème, ce qui contribue à sa rapidité et sa robustesse.

D'autres travaux utilisant l'indexation d'invariants existent [9]. Ces méthodes s'appuient principalement sur l'utilisation d'invariants projectifs pour la modélisation et la reconnaissance des objets. Nous nous distinguons nettement de cette approche par la différence même des invariants utilisés. En effet, des invariants projectifs requièrent des configurations plus complexes que celles que nous utilisons. Par conséquent, l'extraction d'invariants s'en trouve affectée. Ils deviennent plus sensibles au bruit dans les images et à cause de la complexité des configurations qui interviennent, ils sont aussi plus sensibles aux occlusions dans celles-ci. Finalement, puisqu'il s'agit d'invariants projectifs, seules les classes d'objets projectivement équivalents peuvent être distinguées entre elles.

Notre approche évite ces inconvénients par l'utilisation de configurations plus simples et encore plus locales. Elle devient donc plus robuste aux occlusions et au bruit. La contrainte de vote assure la cohérence globale de notre méthode.

### 5.1 Poursuite du travail

L'inconvénient des méthodes basées sur l'extraction d'invariants géométriques est le fait qu'elles soient basées sur une segmentation préalable. D'autres méthodes d'indexation qui

s'affranchissent de cette segmentation existent [10], mais elles sont moins robustes au changement d'éclairage et de luminosité. Il serait intéressant d'intégrer les deux méthodes afin d'obtenir un schéma de reconnaissance global, utilisable dans un plus grand éventail de situations.

## Références

- [1] G. Bebis, M. Georgiopoulos, and N. da Vitoria Lobo. Learning geometric hashing functions for model-based object recognition. In *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, pages 543–548. IEEE, June 1995.
- [2] T.O. Binford and T.S. Levitt. Quasi-invariants: Theory and exploitation. In *Proceedings of DARPA Image Understanding Workshop*, pages 819–829, 1993.
- [3] L. Grewe and A.C. Kak. Interactive learning of a multiple-attribute hash table classifier for fast object recognition. *Computer Vision and Image Understanding*, 61(3):387–416, May 1995.
- [4] W.E.L. Grimson and D.P. Huttenlocher. On the sensitivity of the Hough transform for object recognition. In *Proceedings of the 2nd International Conference on Computer Vision, Tampa, Florida, USA*, pages 700–706, 1988.
- [5] P. Gros. Matching and clustering: Two steps towards object modelling in computer vision. *The International Journal of Robotics Research*, 14(6):633–642, December 1995.
- [6] R. Horaud and H. Sossa. Polyhedral object recognition by indexing. *Pattern Recognition*, 28(12):1855–1870, 1995.
- [7] Y. Lamdan and H.J. Wolfson. Geometric hashing: a general and efficient model-based recognition scheme. In *Proceedings of the 2nd International Conference on Computer Vision, Tampa, Florida, USA*, pages 238–249, 1988.
- [8] B. Lamiroy and P. Gros. Rapid object indexing and recognition using enhanced geometric hashing. In *Proceedings of the 4th European Conference on Computer Vision, Cambridge, England*, volume 1, pages 59–70, April 1996. Postscript version available at [ftp://ftp.imag.fr/pub/MOVI/publications/Lamiroy\\_eccv96.ps.gz](ftp://ftp.imag.fr/pub/MOVI/publications/Lamiroy_eccv96.ps.gz).
- [9] C.A. Rothwell. *Object Recognition Through Invariant Indexing*. Oxford Science Publications, 1995.
- [10] C. Schmid and R. Mohr. Combining greyvalue invariants with local constraints for object recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, San Francisco, California, USA*, m4 1996. Accepted for CVPR 96. [ftp://ftp.imag.fr/pub/MOVI/publications/Schmid\\_cvpr96.ps.gz](ftp://ftp.imag.fr/pub/MOVI/publications/Schmid_cvpr96.ps.gz).