

Identification d'un code à bulles par une signature projectivement invariante

Viorica Patraucean

Pierre Gurdjos

Jean Conter

Université de Toulouse (Institut de Recherche en Informatique de Toulouse)

{vpatrauc, conter}@enseeiht.fr gurdjos@irit.fr

Résumé

Nous disposons d'une base de données contenant des images de différentes configurations de cercles coplanaires disposés aléatoirement, appelées « codes à bulles ». Les images des codes à bulles sont prises selon différents points de vue. Étant donnée une nouvelle image (image requête), le but est de retrouver dans la base de données l'image qui contient le même code à bulles que l'image requête. Nous proposons de représenter les images par des signatures projectivement invariantes qui permettent l'identification du code à bulles sans passer au préalable par une reconstruction euclidienne. Les expérimentations réalisées valident la performance de nos algorithmes, en termes de précision et de complexité.

Mots Clef

perspective, valeurs propres, enveloppe convexe, signature.

Abstract

We have at our disposal a database containing images of various configurations of coplanar circles, randomly laid-out, called "Bubble Tags". The images are taken from different viewpoints. Given a new image (query image), the goal is to find in the database the image containing the same bubble tag as the query image. We propose representing the images through projective invariant signatures which allow identifying the bubble tag without passing through an Euclidean reconstruction step. The experiments carried out confirm the efficiency of our approach, in terms of precision and complexity.

Keywords

perspective, eigenvalues, convex hull, signature.

1 Introduction

La calibration de caméra, que l'on retrouve étroitement liée avec les travaux de reconstruction euclidienne, est un point central dans des domaines comme le suivi et l'identification d'objets dans des images, la réalité virtuelle etc. La nécessité de passer par l'étape de calibration est justifiée par le besoin d'avoir des connaissances précises sur les caractéristiques des objets de la scène : position, forme, taille par rapport à un repère choisi. De nombreuses méthodes de calibration de caméra ou de reconstruction euclidienne ont

été proposées dans la littérature. Certaines approches demandent l'usage de mires planes [5] [4] [14] ou ont besoin de plusieurs images d'une scène rigide quelconque pour accomplir cette tâche (on parle alors d'autocalibration [12] [8]). L'usage de projections de motifs circulaires est un sujet ouvert dans le domaine parce qu'il rend possible la reconstruction euclidienne en une seule vue [3] [7] [2].

Dans cet article nous nous intéressons à l'identification, grâce à des images, de configurations de cercles coplanaires, disposés aléatoirement, appelées « codes à bulles ». L'identification pourrait bien sûr se faire en se basant sur les caractéristiques euclidiennes des cercles : coordonnées des centres et rayons. Nous montrons qu'il est possible d'extraire des informations pertinentes pour l'identification du code à bulles sans passer par la phase de reconstruction euclidienne.

Cadre général du problème

Nous disposons d'une base de données contenant un grand nombre d'images de codes à bulles. Les images sont prises avec des caméras non calibrées, selon différents points de vue. Étant donnée une nouvelle image (image requête), le but est de retrouver dans la base de données l'image qui contient le même code à bulles que l'image requête. L'idée est de représenter les images par des signatures convenables qui permettront une identification efficace d'un code à bulles.

Dans la conception d'une telle signature il faut tenir compte du fait qu'il s'agit d'une base de données relativement grande (des milliers, voire des millions d'images). En plus, les signatures extraites à partir d'un même code à bulles dans les phases d'enregistrement dans la base de données et d'identification ne seront pas strictement identiques à cause du changement des conditions de prise de vue, bruit etc. Dans ces conditions, le problème d'identification devient un problème de recherche du plus proche voisin au sens d'une mesure de similarité cohérente avec la signature. La solution évidente dans ce cas est la recherche linéaire, i.e. la signature requête est comparée avec toutes les signatures enregistrées et le candidat qui ressemble le plus (selon la mesure de similarité à définir) est choisi comme correspondant. À cause de la taille de la base de données cette méthode n'est pas applicable en pratique pour des raisons de temps de calcul et de mémoire utilisée. L'usage d'une approche de

type LSH (locality sensitive hashing) [1] [11] s'avère utile pour contrecarrer ces aspects. Dans ce contexte, on choisit de considérer la base de données comme un ensemble P qui contient les signatures des images exprimées par des vecteurs $p_i \in \mathbb{R}^d$. La fonction de calcul de la signature peut s'écrire comme $sig : f \rightarrow P \subset \mathbb{R}^d$, où f est l'image ou des caractéristiques extraites de l'image. Le problème de recherche du plus proche voisin devient alors un problème d'optimisation, ayant comme fonction objectif $\min \left(\text{dist} \left(\text{sig}(\text{requête}), p_i \right) \right)$, où dist est une distance à bien choisir.

Organisation de l'article. Par la suite, nous donnons la formulation du problème d'extraction de la signature et la solution proposée (Section 2). La Section 3 présente les propriétés du plan projectif qui justifient le choix de la signature. Nous détaillons l'algorithme d'identification d'un code à bulles en Section 4. Les sections 5 et 6 sont réservées aux résultats expérimentaux et aux conclusions.

2 Formulation du problème et solution proposée

Étant donnée une image d'un code à bulles prise avec une caméra non calibrée, le but est d'extraire une signature qui permettrait l'identification du code parmi une base d'images de codes à bulles. Une solution évidente serait de calculer les paramètres de la caméra en utilisant une des méthodes proposées en [7] [2] et d'accomplir la reconstruction euclidienne du plan qui contient le code à bulles. La signature pourrait être calculée ensuite comme une concaténation de centres de cercles et leurs rayons. La complexité globale de cette approche est élevée à cause du fait que la reconstruction euclidienne doit être exécutée deux fois : d'abord lors de l'enregistrement et ensuite lors de l'identification.

La solution que nous proposons ne nécessite pas le passage par la phase de reconstruction euclidienne : fondée exclusivement sur des caractéristiques invariantes du plan projectif, elle fournit une signature projectivement invariante.

3 Propriétés remarquables du plan projectif

Dans cette section nous supposons que la caméra utilisée pour prendre les images de code à bulles suit le modèle sténopé. Sous cette hypothèse, tout couple d'images d'un même plan est relié par une homographie H [9, p.7]. Puisque les cercles sont des coniques particulières, ils peuvent être représentés par des matrices symétriques 3×3 :

$$C = \begin{pmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{pmatrix} \quad (1)$$

où (a, b, c, d, e, f) sont les coefficients de la conique C .

Quand elle fait l'objet d'une transformation projective, la conique C se transforme en la conique \tilde{C} selon l'équation suivante [9, p.36] :

$$\tilde{C} = sH^{-T}CH^{-1} \quad (2)$$

avec s un facteur d'échelle non nul. La conique \tilde{C} en (2) pourrait représenter une ellipse, une hyperbole ou une parabole. Nous ne nous intéressons qu'aux homographies qui transforment les cercles en ellipses.

Valeurs propres généralisées et signature absolue.

Considérons deux coniques générales (C_1, C_2) et leurs projections $(\tilde{C}_1, \tilde{C}_2)$ obtenues selon (2). On appelle valeurs propres généralisées du couple $(\tilde{C}_1, \tilde{C}_2)$ les trois solutions réelles $\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\lambda}_3$ pour l'inconnue λ dans l'équation quadrique [6, p.378]

$$\det(\tilde{C}_1 - \tilde{\lambda}\tilde{C}_2) = 0. \quad (3)$$

Les vecteurs $z_k \in \mathbb{C}^3$ qui vérifient l'équation

$$(\tilde{C}_1 - \tilde{\lambda}_k\tilde{C}_2)z_k = 0, k \in 1..3 \quad (4)$$

sont les vecteurs propres généralisés associés [6].

Les valeurs propres généralisées permettent de définir la signature absolue de la famille linéaire $\tilde{D} = \tilde{C}_1 - \tilde{\lambda}\tilde{C}_2$. Cette signature est notée $\sum(\tilde{D}) = |\eta - \nu|$, où η et ν comptent les valeurs propres positives et négatives du \tilde{D} . Par analogie, on utilise des notations équivalentes pour la représentation euclidienne.

Notre méthode d'extraction est fondée sur les propriétés P_1, P_2, P_3 invariantes du plan projectif suivantes :

P_1 L'enveloppe convexe d'un nuage de points coplanaires est invariante par perspective [9, p.515].

P_2 L'ensemble de valeurs propres généralisées de deux coniques générales (C_1, C_2) est invariant par perspective à un facteur d'échelle près [13].

P_3 La signature absolue \sum de la famille linéaire $\tilde{D} = \tilde{C}_1 - \tilde{\lambda}\tilde{C}_2$ est invariante par perspective [6, p.403].

Valeurs propres généralisées et positions relatives de deux cercles. L'étude des valeurs propres généralisées s'avère utile dans notre cas, car celles-ci contiennent des informations sur la position relative de deux cercles [7]. En effet, considérons un couple de cercles (C_1, C_2) appartenant au plan euclidien et l'ensemble de valeurs et des vecteurs propres généralisés associés (λ, z) , calculés selon (3) et (4). En plus, si $(\tilde{C}_1, \tilde{C}_2)$ sont les projections de (C_1, C_2) obtenues selon (2), en vertu de P_2 , on peut écrire que l'ensemble de valeurs et de vecteurs propres généralisés associés est donné par :

$$(\tilde{\lambda}, \tilde{z}) = \left(\frac{s_2}{s_1}\lambda, Hz \right). \quad (5)$$

Par ailleurs, afin de simplifier le calcul, nous représentons les cercles (C_1, C_2) par de matrices canoniques. Soit :

$$C_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}, C_2 = \begin{pmatrix} 1 & 0 & -d \\ 0 & 1 & 0 \\ -d & 0 & d^2 - r^2 \end{pmatrix}. \quad (6)$$

Ainsi, on considère le cercle C_1 de centre l'origine O et de rayon 1 et le cercle C_2 de centre le point $(0, d)$, $d > 0$, et de rayon $r > 0$.

En utilisant Maple pour résoudre (3), on retrouve les expressions des valeurs propres généralisées λ du couple (C_1, C_2) en fonction de d et r :

$$\lambda = \left(\frac{1 + r^2 - d^2 - \sqrt{\beta}}{2r^2}, \frac{1 + r^2 - d^2 + \sqrt{\beta}}{2r^2}, 1 \right)^T, \quad (7)$$

avec $\beta = \alpha(d+r+1)$ et $\alpha = (d-r+1)(d-r-1)(d+r-1)$.

En tenant compte de (5), on peut également écrire (7) comme :

$$\tilde{\lambda} \sim \left(\frac{1 + r^2 - d^2 - \sqrt{\beta}}{2r^2}, \frac{1 + r^2 - d^2 + \sqrt{\beta}}{2r^2}, 1 \right)^T. \quad (8)$$

Soit $\tilde{\lambda} = (\tilde{\lambda}_1 \tilde{\lambda}_2 \tilde{\lambda}_3)^T$. En résolvant l'équation (8) pour d et r et en choisissant les solutions positives, on obtient :

$$\begin{cases} d = \sqrt{\tilde{\lambda}_1 \tilde{\lambda}_2 (\tilde{\lambda}_1 - \tilde{\lambda}_3) (\tilde{\lambda}_2 - \tilde{\lambda}_3) / |\tilde{\lambda}_1 \tilde{\lambda}_2|} \\ r = |\tilde{\lambda}_3| / \sqrt{\tilde{\lambda}_1 \tilde{\lambda}_2} \end{cases} \quad (9)$$

Le couple (d, r) comme un double invariant de deux cercles. L'équation (9) a une importance majeure parce qu'elle exprime le couple (d, r) (inconnu si on ne possède pas la reconstruction euclidienne du plan auquel appartient le couple de cercles (C_1, C_2)) en fonction des valeurs propres généralisées $\tilde{\lambda}$. Celles-ci peuvent être facilement calculées en utilisant une méthode de détection d'ellipses dans une image [10] et en résolvant l'équation (3).

Comme indiqué en [7], un aspect crucial est le tri des valeurs propres généralisées $\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\lambda}_3$. Comme $\tilde{\lambda}_1$ et $\tilde{\lambda}_2$ ont des rôles symétriques en (9), elles peuvent être échangées sans conséquence ; par contre $\tilde{\lambda}_3$ doit impérativement être distinguée de ces deux premières valeurs propres. En tenant compte de P_3 , on peut montrer que :

$$\sum (\tilde{C}_1 - \tilde{\lambda}_{1,2} \tilde{C}_2) \geq 1 \geq \sum (\tilde{C}_1 - \tilde{\lambda}_3 \tilde{C}_2). \quad (10)$$

L'inégalité (10) est un invariant projectif qui nous permet d'établir la distinction entre les trois valeurs propres généralisées [7].

Si $\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\lambda}_3$ sont triées de telle façon que (10) reste valide, alors d et r , exprimés selon (9), sont projectivement invariants en vertu de P_2 .

Par conséquent, étant donné le couple de projections $(\tilde{C}_1, \tilde{C}_2)$, on peut obtenir un double invariant du couple de cercles (C_1, C_2) en suivant l'algorithme (1) :

Notre méthode utilise ce résultat conjointement avec P_1 . On calcule la paire (d, r) pour chaque couple de cercles

Algorithme 1 Calcul d'un double invariant des cercles (C_1, C_2) , étant données leurs projections $(\tilde{C}_1, \tilde{C}_2)$

calculer $(\tilde{C}_1, \tilde{C}_2)$ en utilisant un détecteur d'ellipses [10]
 $\tilde{\lambda} = \text{ValeursPropresGen}(\tilde{C}_1, \tilde{C}_2)$
 tri des solutions $\tilde{\lambda}$ sous la contrainte (10)
 calculer d et r selon (9)

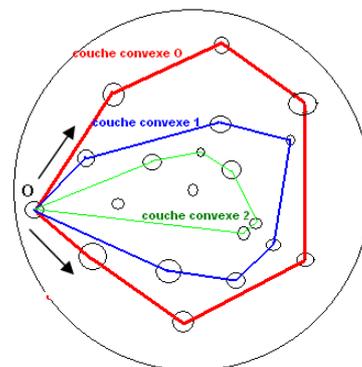


FIG. 1 – Représentation du code à bulles par des couches convexes

voisins dans l'enveloppe convexe et on exprime la signature d'un code à bulles comme la concaténation de ces paires (d, r) . Ainsi, la signature obtenue sera projectivement invariante, donc la reconstruction euclidienne n'est plus nécessaire.

4 Identification d'un code à bulles

L'algorithme mis en place vise à identifier un code à bulles par l'élimination progressive des candidats potentiels. Nous considérons une représentation du code par des couches convexes successives, comme montré dans la figure 1.

Les couches sont calculées de façon à ce qu'elles aient un nœud commun, qui servira comme point de référence (utile pour gérer le problème de rotation du code). Chaque couche sera exprimée comme

$$l_i = \{(d_{ij}, r_{ij})\}, i \in \{1 \dots N_l\}, j \in \{1 \dots N_i\}, \quad (11)$$

où N_l est le nombre de couches convexes et N_i est le nombre de nœuds de la couche i . Ainsi, la signature sera formée par plusieurs *sous_signatures*, une pour chaque couche convexe. L'interrogation de la base de données est faite en commençant avec la sous_signature associée à l'enveloppe convexe externe. À l'issue de cette étape, nous ne gardons comme candidats potentiels que p_i tq $\left(\text{dist}(\text{sig}(\text{requête}), p_i) < d_{\text{seuil}} \right)$. Le processus est répété en utilisant comme requête les couches inférieures suivantes jusqu'à l'obtention d'un voisin qui se trouve à une distance $d < d_{\text{min}}$. À ce moment, une comparaison

complète de « 1 à 1 » est réalisée pour conclure si c'est le bon candidat.

L'algorithme d'enregistrement de codes dans la base de données est présenté dans l'algorithme (2).

Algorithme 2 Enregistrement d'un code à bulles dans la base de données

```

détecter les ellipses dans l'image de référence [10]
calculer les couches convexes du code en choisissant un
point de référence O
calculer les sous_signatures selon (11) et (9), en utilisant
l'algorithme (1)
enregistrer les sous_signatures

```

L'algorithme d'identification débute avec la détection des ellipses dans l'image requête [10]. Ensuite il traite successivement chaque couche convexe du code (algorithme 5) et prend la décision si le code correspondant a été trouvé (algorithme 4). Toutefois, le traitement de la première couche convexe inclut une opération supplémentaire e_3 , nécessaire pour retrouver dans l'image requête le même point de référence et le sens de parcours de l'enveloppe convexe utilisés dans l'image originale. Le but est d'éliminer la rotation éventuelle entre l'image du code originale et l'image requête (algorithme 3).

Algorithme 3 Traitement couche convexe externe

```

(e1) calculer la couche convexe i du code
(e2) calculer la sous_signature de la couche i selon (11)
et (9), en utilisant l'algorithme (1)
(e3) calculer toutes les permutations cycliques de la
sous_signature, dans les deux sens.
(e4) interroger la base de données en utilisant comme
requêtes toutes les sous_signatures calculées auparavant

```

Algorithme 4 Décision

```

si  $dist(requ\hat{e}te, voisins) < d\_seuil$  alors
  garder voisins comme candidats potentiels
fin
si  $cardinal(voisins) == 1$  alors
  vérifier toutes les couches
  si  $dist(voisin, requ\hat{e}te) < d\_min$  alors
    POSITIF
    return voisin
  sinon
    NEGATIF
    return O
  fin
fin

```

5 Résultats expérimentaux

Les résultats que nous présentons ont été obtenus en utilisant des images de synthèse et des images de « codes à

Algorithme 5 Traitement couches convexes inférieures

tantque code pas trouvé **faire**

```

(e1)
(e2)
(e4)
décision
fin tantque

```

pièces » (figure 4). Notre travail a également été évalué dans un contexte de « codes à bulles » industriels soumis à des règles de confidentialité.

Tests sur des images de synthèse. Nous avons utilisé une base de données contenant 300 images de codes générés aléatoirement. Chaque image, de taille 1200x800 pixels, contient un nombre aléatoire de cercles compris entre 80 et 110, avec des rayons prenant des valeurs entre 10 et 20 pixels. Afin d'obtenir la projection équivalente à une prise d'image, nous avons générés aléatoirement des homographies telles que les cercles deviennent des ellipses après avoir appliqué la transformation. Les tests ont visé à tester la robustesse de la méthode vis-à-vis du bruit. Les résultats sont montrés dans les figures 2 et 3. Pour chaque niveau de bruit, nous avons considéré 100 tests (les images requête ont été choisies par des tirages aléatoires) et nous avons calculé les erreurs moyenne et maximale introduites par le bruit. Pour chaque test nous avons retenu les 10 plus proches voisins trouvés dans la base de données, en utilisant comme mesure de similarité la distance euclidienne. Les résultats montrent que jusqu'à un niveau de bruit inférieur à 2.5 pixels, la première couche convexe suffit pour identifier correctement le code. Pour un niveau de bruit supérieur à ce seuil, le correspondant correct a été obtenu en utilisant les deux premières couches convexes. La valeur du seuil est dépendante de la distance minimale entre deux signatures enregistrées.

Tests sur des images de « codes à pièces ». Nous avons considéré 50 configurations différentes de pièces et nous avons pris 2 images pour chacune (exemple figure 4). Une image sert comme référence à enregistrer dans la base de données et l'autre est utilisée comme image requête. La figure 5 montre la distance moyenne (sur les 50 tests) par rapport aux 10 plus proches voisins trouvés en n'utilisant que l'enveloppe convexe externe. Le plus proche voisin est le correspondant correct dans tous les cas.

6 Conclusions

Nous avons décrit une méthode d'identification de configurations des cercles – « code à bulles » – dans des images prises selon différents points de vue. Afin d'éviter une éventuelle étape de reconstruction euclidienne, la signature proposée est projectivement invariante. Elle utilise le fait que l'enveloppe convexe d'un nuage de points coplanaires est invariante par perspective et qu'il est possible d'obtenir un couple des valeurs invariantes de deux cercles en utilisant les valeurs propres généralisées de leurs projections.

En prenant les couples de cercles qui sont voisins dans l'enveloppe convexe et en calculant ces valeurs invariantes pour chacun des couples, on peut obtenir une signature projectivement invariante comme concaténation de ces valeurs. Notre approche permet de réduire la complexité du processus d'identification et, en plus, on est capable de gérer efficacement le problème de rotation du code entre deux images. Les expérimentations menées sur des images de synthèse et sur des images réelles montrent la robustesse au bruit et la performance en termes de temps de calcul.

Références

- [1] A. Andoni and P. Indyk, Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions, *In Proceedings of Symp. Foundations of Computer Science*, 2006.
- [2] Q. Chen, H. Wu and T. Wada, Camera Calibration with two Arbitrary Coplanar Circles, *ECCV*, Vol. 3, pp. 521-532, 2004.
- [3] Y. Chen, I. Horace, Z. Huang and G. Wang, Full Camera Calibration from a Single View of Planar Scene, *Advances in Visual Computing*, Vol. 5358, pp. 815-824, 2008.
- [4] Z. Chen, C. M. Wang and S. Y. Ho, An Effective Search Approach to Camera Parameter Estimation Using an Arbitrary Planar Calibration Object, *PR*, Vol. 26, pp. 655 - 666, 1993.
- [5] O. Faugeras, *Three-Dimensional Computer Vision : A Geometric Viewpoint*, MIT Press, 1993.
- [6] G. H. Golub and C. F. Van Loan, *Matrix computations (3rd ed.)*, Johns Hopkins University Press, 1996.
- [7] P. Gurdjos, P. Sturm, Y. Wu, Euclidean Structure from $N \geq 2$ Parallel Circles : Theory and Algorithms, *Computer Vision ECCV 2006*, pp. 238-252, 2006.
- [8] R. Hartley, An algorithm for self-calibration from several views, *In proceedings of CVPR '94*, pp. 908, 1994.
- [9] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2nd edition, 2003.
- [10] K. Kanatani and N. Ohta, Automatic detection of circular objects by ellipse growing, *International Journal of Image and Graphics*, Vol. 4, pp. 35-50, 2004.
- [11] P. Indik, R. Motwani, Approximate nearest neighbors : towards removing the curse of dimensionality, *Proceedings of the thirtieth annual ACM Symposium on Theory of Computing*, 1998.
- [12] Q.-T. Luong and O. Faugeras, Self-calibration of a moving camera from point correspondences and fundamental matrices, *International Journal of Computer Vision*, Vol. 22, pp. 261-289, 1997.
- [13] J. L. Mundy and A. Zisserman, *Geometric Invariance in Computer Vision*, MIT Press, 1992.
- [14] Z. Y. Zhang, A Flexible New Technique for Camera Calibration, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998.

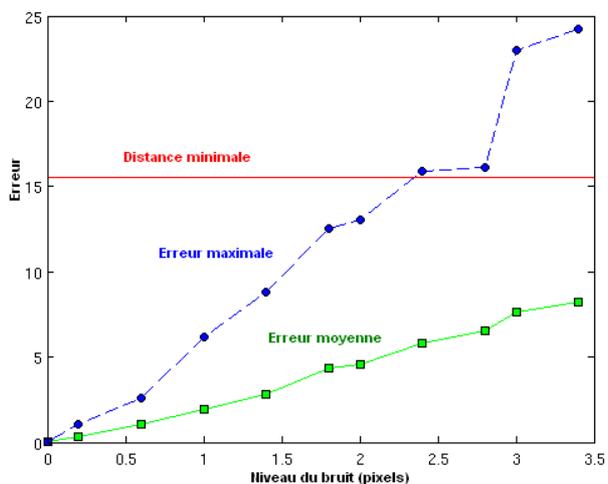
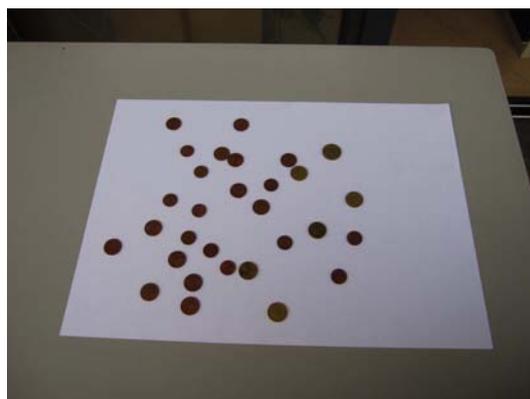
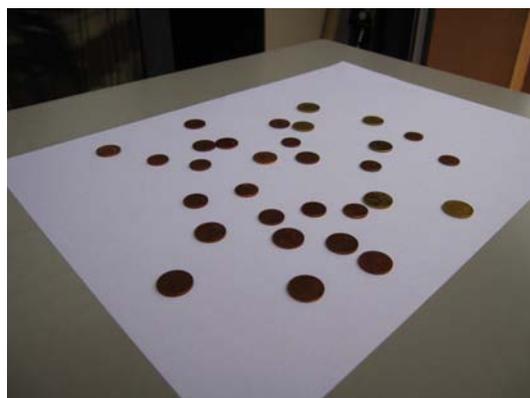


FIG. 2 – Robustesse au bruit de la méthode de calcul de signature sur des images de synthèse. La ligne rouge indique la distance minimale entre deux sous-signatures d’enveloppes convexes externes enregistrées. Les erreurs moyenne (en vert=carré) et maximale (en bleu=rond) ont été calculées sur 100 tests. Pour un niveau de bruit supérieur à 2.4 pixels, l’erreur maximale dépasse la distance minimale, donc le processus d’identification doit prendre en compte une couche convexe supplémentaire.



(a) Image de référence



(b) Image requête

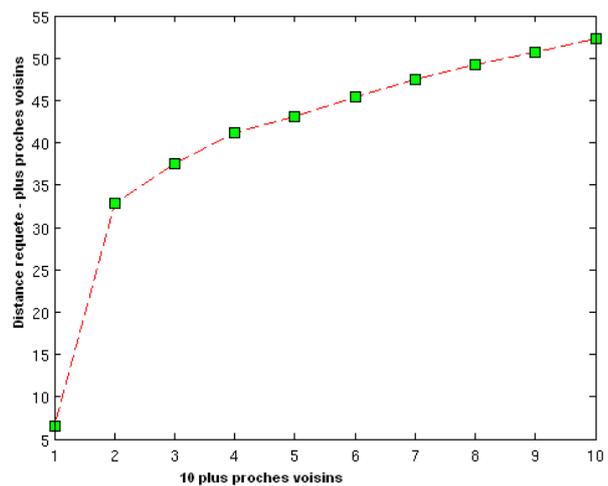


FIG. 3 – Exemple de distances moyennes (calculées sur 100 tests avec des images de synthèse) entre un code requête et les 10 plus proches voisins pour un niveau de bruit de 2.4 pixels. On observe qu’il y a un écart net entre le code correspondant et les autres plus proches voisins.

FIG. 4 – Exemple d’images de code à pièces (a) Image de référence (b) Image requête

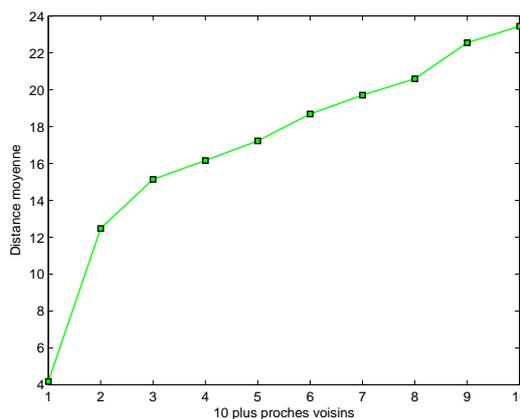


FIG. 5 – Distances moyennes entre les couches convexes externes (calculées sur 50 tests avec des images réelles) d’un code requête et des 10 plus proches voisins. Dans tous les cas, le plus proche voisin est le correspondant correct.