

# Détection de pupille par combinaison des critères morphologiques et colorimétriques.

Benjamin Raynal, Venceslas Biri

raynalb@esiee.fr, biri@univ-mlv.fr

Université Paris-Est

Laboratoire d'Informatique de l'Institut Gaspard Monge

UMR CNRS 8049

5 bd Descartes, 77454 Marne la Vallée Cedex 2, France

---

## Abstract

*Dans cet article nous étudions le problème de la détection de pupilles dans une image de résolution moyenne. On considérera que la connaissance de la position globale de l'oeil est déjà acquise, pour se concentrer sur le problème de la détection de la pupille dans une zone restreinte. Nous proposerons pour cela une méthode combinant des critères morphologiques et colorimétriques. Cette combinaison permettra, comme nous le démontrerons, une grande robustesse de détection. Cet algorithme sera également optimisé, afin d'être utilisable en temps réel.*

*We study in this article the problem of pupil detection in a medium resolution image. The global location of the eye is supposed to be known and we focus on the pupil detection in a restricted area of the initial image. Our new method combine morphologic and colorimetric criteria to obtain a very robust detection of the eye center.*

---

## 1. Introduction

Le suivi du regard est une discipline de vision par ordinateur, consistant à étudier la position des yeux de l'utilisateur, et à partir de là, à déterminer la direction fixée par celui-ci. Les applications du suivi du regard [Duc02] sont diverses, par exemple :

**Interface homme-machine** La principale application du suivi du regard est son utilisation en tant que système de pointage, dans le cas où l'utilisateur est dans l'incapacité d'utiliser ses bras, du fait de sa situation, ou de sa condition.

**Etude de conception** On peut aussi utiliser le suivi du regard sur des utilisateurs témoins, devant une affiche où un site internet, pour analyser quelles parties sont étudiées ou ignorées, afin d'optimiser leur impact.

**Compression vidéo** Il est également envisageable d'utiliser ce type de méthodes pour déterminer les régions d'intérêt dans une vidéo, afin de régler la qualité de compression en conséquence.

**Etudes cognitives** Une tout autre utilisation du suivi du regard consiste à l'utiliser pour étudier la façon dont on perçoit une scène, un texte, et comment on les analyse.

Les algorithmes de suivi du regard comportent généralement une phase de détection de l'œil, de façon globale, suivie d'une phase de détection du centre de l'iris et de la pupille. Cette dernière se doit d'être robuste et précise, car d'elle dépend, bien entendu, la précision du suivi du regard.

Les principaux problèmes liés aux algorithmes de détection de pupille existants sont :

- le coût financier, certains nécessitant d'utiliser un type de caméra particulier (caméras infrarouges, par exemple), d'une valeur souvent très élevée (entre 4000 et 50000 euros).
- la sensibilité au changement d'utilisateur ou de luminosité (cas notamment des méthodes utilisant un réseau de neurones [ZY02]).
- la durée et la difficulté de calibration de l'algorithme .

Ces problèmes rendent difficile d'accès le suivi du regard aux personnes en ayant le plus besoin : les personnes lourdement handicapées.

De plus, un algorithme de détection de pupille doit valider les contraintes suivantes :

**Rapidité** Permettre l'utilisation dans une application temps réel.

**Précision** Détecer le plus finement possible la pupille.

**Robustesse** Insensible au bruit et à l'environnement.

**Efficacité** Ne donner de résultat que si celui-ci est pertinent.

Nous présentons dans cet article une méthode de détection de pupille utilisant en entrée une image de résolution moyenne de l'oeil, et répondant aux critères ci-dessus. Cette méthode, de moindre coût car ne nécessitant aucun matériel hautement spécifique, sera conçue pour être temps réel. De plus, elle sera robuste pour tout utilisateur et dans un grand nombre d'environnements, d'illumination différente.

## 2. État de l'art

Si les techniques pour détecter la position globale de l'oeil sont nombreuses (détection des clignements de l'oeil [KT02], utilisation de caméra IR [JZ02, JZ05b, JZ05a], réseaux de neurones [SW95, VRR98, BMM00], données anthropomorphiques [SYW96], détection de la couleur de la peau [SW95, VRR98], utilisation de marqueurs [KR99]), celles pour détecter le centre de la pupille le sont moins.

On peut classifier ces méthodes en deux familles :

1. celles utilisant un apprentissage, basées sur l'apparence.
2. celles basées sur un modèle.

La première famille contient toutes les méthodes utilisant un réseau de neurones [SW95, VRR98], un modèle de Markov caché, ou toute méthode utilisant une base de données préalablement traitée. Du fait justement qu'elles nécessitent un apprentissage, elles sont sensibles aux changements d'utilisateur, voire même au changement de luminosité. De plus, cela implique une durée d'apprentissage, souvent non négligeable, voire une certaine difficulté à effectuer celui-ci pour un utilisateur quelconque. Ceci rend donc les algorithmes de cette famille particulièrement inadaptés à l'utilisation dans un suivi du regard utilisable par des personnes lourdement handicapées.

La seconde famille contient toutes les méthodes utilisant les propriétés de la pupille pour détecter celle-ci :

**forme** en partant de la carte des contours de l'image, on essaye alors de détecter la forme ressemblant le plus à un cercle, en utilisant par exemple la transformée circulaire de Hough [KBS75].

**teinte** En utilisant la couleur de la pupille, on cherche à détecter les composantes connexes de cette couleur répondant à divers critères géométriques [MMM06, KR99].

Un sous ensemble de cette seconde famille contient les méthodes utilisant un premier résultat approximatif, comme les méthodes infrarouges, où la pupille est détectée grossièrement grâce au reflet de la lampe IR, où les méthodes se basant sur les résultats précédents, en utilisant par

exemple le filtre de Kalman [ZJFL02]. A partir de ce premier résultat, il est plus aisément de détecter quels points appartiennent au contour de la pupille, et alors de calculer l'ellipse passant au mieux par ces points, au moyen de la méthode du moindre carré [ZY02].

Notre méthode sera quant à elle basée sur un modèle, et n'utilisant aucun résultat préalable, autre que la détection globale de la zone de l'oeil.

## 3. Architecture

Notre méthode peut se décomposer en deux passes successives :

- Une première passe permettra, dans toute l'image, de trouver la position approximative de la pupille.
- Une seconde détection permettra de localiser plus précisément le centre de la pupille, en utilisant le résultat de la première passe.

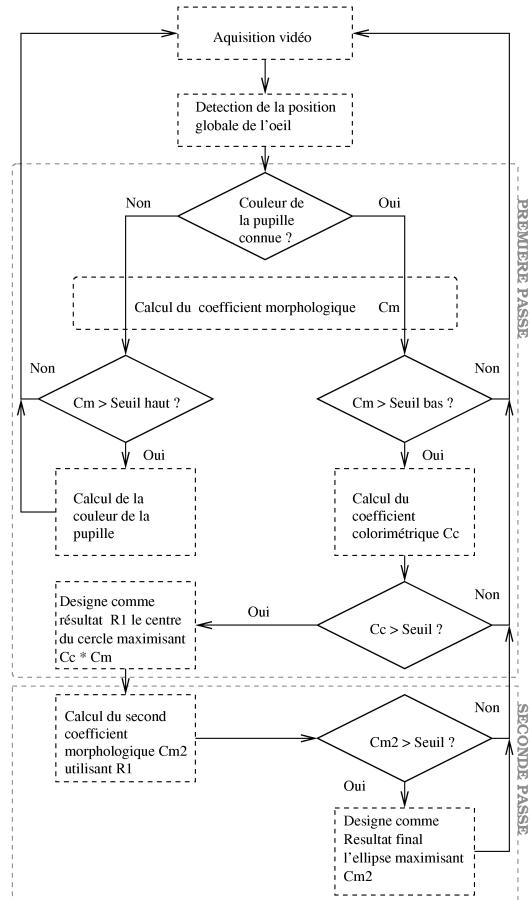


Figure 1: Architecture globale de notre algorithme.

Lors de la première passe, on calculera pour les candidats deux coefficients : un coefficient morphologique, basé

sur la forme circulaire de la pupille, et un coefficient colorimétrique, basé sur la couleur de la pupille. Pour qu'un candidat soit sélectionné, il doit avoir chacun de ses coefficients au dessus de leur seuil respectif. Si aucun candidat n'est sélectionné, l'algorithme ne retourne pas de résultat. Dans le cas contraire, il utilise comme premier résultat le candidat dont le produit des deux coefficients est maximal.

Durant la seconde passe, on calculera un second coefficient morphologique plus précis, en utilisant le résultat de la première passe.

Une première phase consistera à trouver la couleur de la pupille, qui servira par la suite à calculer le coefficient colorimétrique. Pour cela, on commence par une phase de détection des cercles ayant un coefficient morphologique au dessus d'un seuil haut, afin d'être sur de ne pas avoir une détection erronée. Le premier cercle au dessus de ce coefficient sera utilisé pour calculer la couleur de la pupille.

#### 4. Coefficient morphologique

Le coefficient morphologique doit permettre de mesurer la ressemblance du contour de la pupille à un cercle où à une ellipse. Il est calculé sur une image binaire des contours  $C$ , produite par l'application du filtre de Canny [Can86]. Ce filtre à plusieurs propriétés appréciables :

**bonne détection** faible taux d'erreur dans la signalisation des contours.

**bonne localisation** minimisation des distances entre les contours détectés et les contours réels.

**clarté de la réponse** une seule réponse par contour, insensible au bruit.

Cet opérateur est donc bien adapté à nos besoins, la propriété de bonne localisation permettant de ne pas perdre de précision, et la clarté de la réponse permettant de traiter le problème du bruit.

#### 4.1. Choix entre ellipse et cercle

La pupille n'est circulaire dans l'image que lorsque l'œil est orienté vers la caméra. Lorsqu'il est orienté dans d'autres directions, la forme de la pupille dans l'image s'approche plus d'une ellipse. Cependant, alors qu'un cercle n'est caractérisé que par trois paramètres (ses coordonnées dans le plan, et son rayon), une ellipse en utilise cinq (ses coordonnées dans le plan, ses deux rayons axiaux, son angle de rotation).

Sachant que la complexité de l'algorithme dépendra du nombre de ces paramètres, nous avons porté notre choix sur la détection de cercle, afin d'avoir un temps de calcul le plus restreint possible. Une détection d'ellipse sera envisageable lorsque les variations des paramètres auront été fortement restreintes.

#### 4.2. Méthode de calcul

Pour calculer le coefficient morphologique d'un cercle de centre  $(x_c, y_c)$  et de rayon  $r_0$ , on va appliquer le masque adapté  $M$  sur l'image binaire des contours  $C$  (dont les valeurs sont égales à 1 en cas de présence d'un contour et 0 dans le cas contraire) grâce à la formule suivante :

$$k_m(x_c, y_c, r_0) = \frac{\sum_{m \in M} C(x_c + x_m, y_c + y_m) * M(x_m, y_m)}{\sum_{m \in M} M(x_m, y_m)}$$

Les masques sont bien entendus centrés en  $(0, 0)$ .

#### 4.3. Conception du masque

Une première idée consiste à utiliser un simple masque représentant le cercle de rayon  $R_0$  désiré, de façon discrète. Cependant, du fait de l'imperfection de la circularité de la pupille dans l'image, peu de points du contour correspondent à un masque donné.

En nous inspirant des masques proposés par Tian et al. [TKC99], nous avons utilisé des masques d'anneaux, d'épaisseur proportionnelle fixe  $E$ . Pour tout cercle de rayon  $r_0$ , correspond un anneau de rayon interne  $r_1 = r_0 - \frac{E}{2}$  et de rayon externe  $r_2 = r_0 + \frac{E}{2}$ . Cela permet certes de capturer l'intégralité du contour visible de la pupille, mais entraîne évidemment une erreur de précision sur le centre de la pupille.

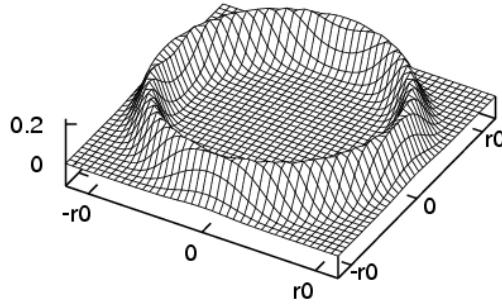
Pour limiter la marge d'erreur, nous avons décidé de pondérer les valeurs du masque, en utilisant une fonction gaussienne  $f$  sur la distance au cercle :

$$f(x, y) = \frac{1}{\pi E} e^{-\frac{(r_0 - \sqrt{x^2 + y^2})^2}{E}}$$

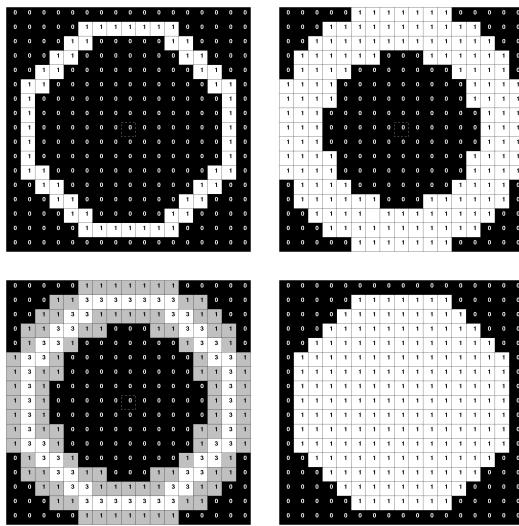
Ceci nous permet d'avoir tous les points du contours de la pupille masqués, sans trop toutefois sacrifier la précision.

#### 4.4. Optimisation du calcul

Pour calculer l'ensemble des coefficients morphologiques, en chaque point de l'image  $I$ , et pour un ensemble de rayons entiers  $R$ , l'utilisation de convolutions se traduit par une complexité en  $o(I_w * I_h * \text{card}(M_R))$ , où  $\text{card}(M_R)$  représente le nombre de points total pour l'ensemble des masques de rayon appartenant à  $R$  (voir l'algorithme 1).



**Figure 2:** Valeurs de la fonction gaussienne  $f$



**Figure 3:** Les différents types de masques correspondant à un même cercle : en haut, à gauche, masque de cercle, à droite, masque d'anneau. En bas, à gauche, masque d'anneau pondéré, à droite, masque de disque.

---

#### Algorithm 1: Algorithme par convolution

---

**Data:** image  $I$ , double  $seuil$ , image  $M$   
**Result:** set<point>  $res$   
**begin**  
 set<point>  $res \leftarrow \emptyset$ ;  
**foreach** point  $p \in I$  **do**  
 double  $km \leftarrow 0$ ;  
 double  $d \leftarrow 0$ ;  
**foreach** point  $m \in M$  **do**  
 $km \leftarrow km + I(p+m) * M(m)$ ;  
 $d \leftarrow d + M(m)$ ;  
**end**  
 $km \leftarrow km/d$ ;  
**if**  $km > seuil$  **then**  
 $res \leftarrow res \cap \{p\}$ ;  
**end**  
**end**  
**end**

---

En constatant que les pixels de valeur non nulle dans la carte des contours ne représentent qu'une petite part de la totalité de l'image, on peut utiliser la symétrie des masques pour réduire la complexité à  $o(card(C) * card(M_R))$ , où  $card(C)$  est le nombre de points de valeurs non nulles dans l'image. On utilise pour cela un accumulateur  $A$ , c'est à dire un tableau de mêmes dimensions que l'image, où seront stockées les valeurs de coefficient morphologique de chaque point (voir l'algorithme 2).

---

#### Algorithm 2: Algorithme avec accumulateur

---

**Data:** image  $I$ , double  $seuil$ , set<point>  $C$ , image  $M$   
**Result:** set<point>  $res$   
**begin**  
 set<point>  $res \leftarrow \emptyset$ ;  
 image  $A$ ;  
 double  $size \leftarrow 0$ ;  
**foreach** point  $m \in M$  **do**  
 $| size \leftarrow size + M(m)$ ;  
**end**  
 $seuil \leftarrow seuil * size$ ;  
**foreach** point  $c \in C$  **do**  
**foreach** point  $m \in M$  **do**  
 $| A(c+m) \leftarrow A(c+m) + M(m)$ ;  
**if**  $A(c+m) > seuil$  **then**  
 $| res \leftarrow res \cap \{(c+m)\}$ ;  
**end**  
**end**  
**end**  
**end**

---

L'utilisation de l'algorithme 2 plutôt que l'algorithme 1 permet de diviser la complexité du calcul du coefficient morphologique par  $\frac{I_w * I_h}{card(C)}$ , avec  $card(C) \ll I_w * I_h$ .

## 5. Coefficient colorimétrique

Le coefficient colorimétrique est utile en complément du coefficient morphologique, et permet de détecter des pupilles dans des yeux moins ouverts. En effet, l'ajout d'une contrainte sur un critère autre que la forme permet d'être moins restrictif sur celle-ci, sans pour autant sacrifier à la robustesse.

Ce coefficient sera calculé à partir de la couleur de référence de la pupille, et de la couleur moyenne à l'intérieur du cercle dont on veut le coefficient.

Dans toute cette section, les couleurs seront des vecteurs de trois dimensions (R,V,B), de valeur comprises dans l'intervalle  $[0, 1]$ .

### 5.1. Calcul de la couleur moyenne dans un cercle

Pour calculer la couleur moyenne dans un cercle de centre  $(x_c, y_c)$  et de rayon  $r_0$ , on va utiliser le masque  $M_D$  du disque de rayon  $r_0$  discréteisé, et l'appliquer à l'image couleur  $I$  de

la façon suivante :

$$\bar{V}(x_c, y_c, r_0) = \frac{\sum_{m \in M_D} I(x_c + x_m, y_c + y_m) * M_D(x_m, y_m)}{\sum_{m \in M_D} M_D(x_m, y_m)}$$

## 5.2. Acquisition de la couleur de la pupille

Pour pouvoir calculer ce coefficient, il nous faut d'abord acquérir la couleur de la pupille. Pour cela, on va détecter un cercle ayant une forte probabilité de correspondre à la pupille, en ne sélectionnant les candidats qu'ayant un coefficient morphologique très élevé. Lorsqu'un tel cercle est trouvé, on peut calculer la couleur de l'iris  $V_I$  en utilisant la formule de calcul de la couleur moyenne défini plus haut.

## 5.3. Méthode de calcul

Pour obtenir le coefficient colorimétrique d'un cercle  $C$ , on calcule la couleur moyenne  $\bar{V}_C$  de son intérieur, puis on utilise la formule suivante :

$$k_c(x_c, y_c, r_0) = 1 - \frac{\|\bar{V}_C - V_I\|_2}{\sqrt{3}}$$

## 6. Raffinement du résultat

Pour gagner en précision, on utilise ce premier résultat comme base de départ pour une détection plus fine : nous allons maintenant chercher à détecter des ellipses, de centre proche de celui du premier résultat, et de rayons avoisinant celui du premier résultat.

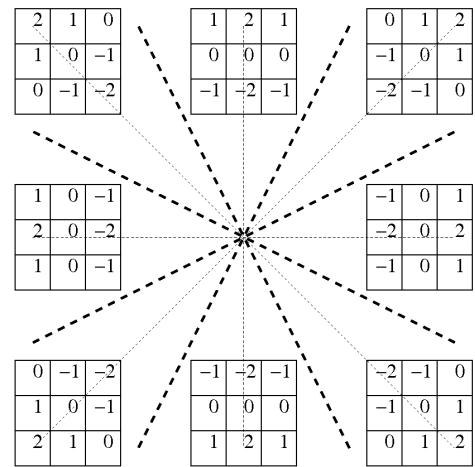
Afin de supprimer le plus de contours inutiles (n'appartenant pas au contours de la pupille), nous avons développer un gradient radial, dont la direction est orientée à partir d'un centre donné, ici, le centre du premier résultat. C'est sur la carte des contours donnée par ce gradient que nous appliquons la détection d'ellipses.

### 6.1. Gradient radial

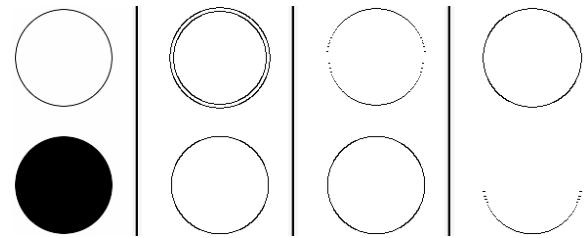
Nous avons cherché à concevoir le gradient le plus adapté à nos besoins, ayant les mêmes propriétés que le filtre de Canny, mais optimisant les contours de toute forme convexe entourant un point donné. En nous inspirant des travaux de Li et al. [LP06], nous avons décidé d'utiliser un gradient défini en fonction d'un centre  $C$  donné.

Soit  $I$  l'image de base,  $G_C$  est l'image qui en chaque point  $P$  contient l'approximation de la dérivée du point correspondant, selon l'angle  $\theta$  que fait  $\vec{CP}$  avec le vecteur  $(1, 0)$ .  $G_C$  est obtenue par convolution de  $I$  par  $N_\theta$ ,  $N_\theta$  étant un noyau de convolution défini en fonction de  $\theta$  (voir la figure 4).

Une fois  $G_C$  obtenu, on en extrait les maxima locaux



**Figure 4:** Les différents noyaux de convolution selon la position par rapport au centre.



**Figure 5:** De gauche à droite : image de base, image traitée avec Canny, image traitée avec notre opérateur centré sur le disque du bas, image traitée avec notre opérateur centré sur le cercle du haut.

selon la direction du vecteur  $\vec{CP}$ . On seuille ensuite le résultat par hysteresis.

Les propriétés de notre filtre sont les suivantes :

- marque bien, et **uniquement** les contours d'une zone foncée vers une zone plus claire, en partant du centre, perpendiculaires à la direction au centre.
- une bonne localisation (meilleure que Canny, dans certains cas).
- une réponse minimale.

La première propriété peut paraître étrange, mais est utile pour pouvoir détecter plus efficacement certains yeux clairs qui ont un bord de pupille mince plus sombre (voir la figure 5). Alors que le filtre de Canny donnera dans ce cas deux contours très proches, notre opérateur n'en donnera qu'un. Pour détecter également les contours d'une zone claire vers une zone foncée, il suffirait d'utiliser les valeurs absolues de  $G_C$ .

Ce filtre a pour autre propriété intéressante de très bien

déetecter les contours perpendiculaires aux rayons partant de son centre, et d'ignorer ceux qui y sont parallèles. Il en découle une très bonne détection des contours d'un cercle contenant le centre du filtre, et une bien moins bonne pour ceux ne le contenant pas (voir les dernières colonnes de la figure 5). C'est donc une sécurité supplémentaire qui permet de renforcer la contrainte de robustesse.

## 6.2. Détection d'ellipse dans la zone restreinte

Une fois notre nouvelle carte des contours calculée, on l'utilise comme base pour détecter des ellipses avec de fortes contraintes, décrites ci-dessous. Soit  $r_0$  le rayon du premier résultat,  $C_0$  son centre, et  $\epsilon$  une constante fixée. Définissons la fenêtre  $W$  de coté  $r_0$ , et de centre  $C_0$ . Les ellipses détectées sont celles dont le centre se situe dans  $W$ , et dont les rayons axiaux sont compris dans  $[r_0 * (1 - \epsilon), r_0 * (1 + \epsilon)]$ .

Les masques utilisés pour détecter ces ellipses sont des masques d'anneaux ellipsoïdaux pondérés, selon une adaptation de la formule utilisée pour pondérer nos masques d'anneaux : pour une ellipse d'axe semi mineur  $r_x$  parallèle à l'axe des abscisses et d'axe semi majeur  $r_y$ , on définit

$$\alpha_{x,y} = \sqrt{\frac{x^2}{r_x^2} + \frac{y^2}{r_y^2}}$$

le coefficient de mise à l'échelle de l'ellipse pour qu'elle passe par le point  $(x,y)$ . On peut alors définir la fonction de pondération en fonction de la distance à l'ellipse comme ci-dessous :

$$g(x,y) = \frac{1}{\pi E} e^{-\left(\left|\frac{\alpha_{x,y}-1}{\alpha_{x,y}}\right| \sqrt{x^2+y^2}\right)^2}$$

## 7. Résultats expérimentaux

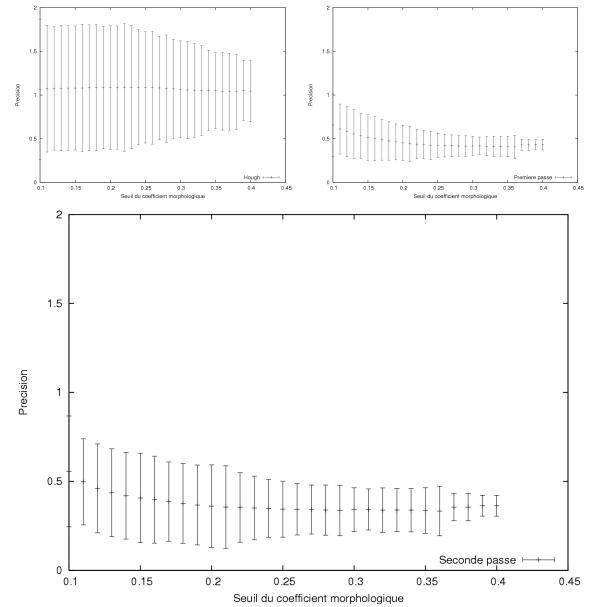
Dans le but de mesurer la précision et l'efficacité de notre algorithme, nous l'avons testé sur 6 séries d'images d'œil, provenant de différents utilisateurs aux couleurs de prunelles/d'iris différentes. Chaque série contient 51 images de résolution moyenne de 150\*150 pixels et représentant la zone globale de l'œil.

A partir de ces données, nous avons compté le nombre



**Figure 6:** Echantillon d'images de test.

de résultats obtenu pour notre algorithme, avant et après



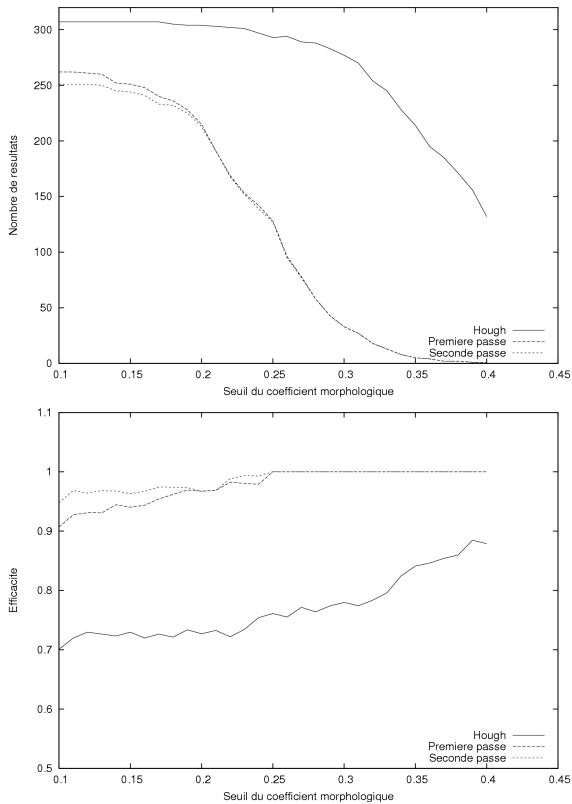
**Figure 7:** Selon le seuil de coefficient morphologique. En haut à gauche, la transformée de Hough, à droite, la première passe de notre algorithme. En bas, seconde passe de notre algorithme.

la seconde passe, et le meilleur résultat de la transformée circulaire de Hough, pour avoir une référence. Nous avons également mesuré la précision des algorithmes, en utilisant la distance entre le centre trouvé par l'algorithme et celui du cercle désigné manuellement, servant de référence. Cette distance est ensuite divisée par le rayon du cercle de référence, afin d'avoir une mesure relative. Enfin, pour évaluer l'efficacité, nous utilisons la proportion de résultats valides (à une distance raisonnable du centre de référence), par rapport au nombre de résultats total.

On constate que le nombre de résultats obtenus diminue en même temps que la précision augmente, ce qui est normal, étant donné que l'on supprime les résultats les moins probants pour gagner en précision.

On observe que la seconde passe apporte un gain de précision non négligeable, mais entraîne également une baisse du nombre de résultats. Cependant, pour un seuil du coefficient morphologique aux alentours de 0.2, on obtient un bon gain de précision, sans trop perdre de résultats. (voir la figure 7). Il faut aussi prendre en compte que les résultats sont ici des moyennes, une adaptation plus fine des seuils en fonction de l'utilisateur donnera de meilleurs résultats.

Pour ce qui est de la vitesse d'exécution, notre algorithme tourne sur la machine de test (Pentium 4 2.8Ghz) à 50 Hz environ.



**Figure 8:** Nombre de résultats et efficacité selon le seuil de coefficient morphologique (Seuil du coefficient colorimétrique à 0.85).

## 8. Conclusion

Nous avons développé un algorithme de détection de pupilles répondant aux critères fixés :

**précision** étant donnée la qualité de l'image, et les besoins matériels, notre algorithme à une précision tout à fait acceptable, grâce à la seconde passe.

**efficacité** L'utilisation de seuils nous assure que le résultat obtenu est de même forme et de même couleur que la pupille. Du fait de la localité de l'image, la probabilité que ce résultat corresponde à la pupille est très élevé.

**robustesse** du fait de l'utilisation du filtre de Canny pour produire les données de départ, notre algorithme est insensible au bruit, ainsi qu'à la couleur des yeux, grâce à notre méthode d'acquisition de la couleur de la pupille.

**rapidité** avec une fréquence de 50 Hz en moyenne sur une machine de milieu de gamme, notre algorithme peut être utilisé pour une application temps réel.

Cet algorithme permet donc de faire du suivi du regard en temps réel, tout en ne nécessitant qu'une webcam de bonne qualité, pour un prix avoisinant les 100 euros.

Enfin, la calibration de l'algorithme ne nécessite que peu de

réglages, dont beaucoup peuvent être automatisés en utilisant des données fournies par d'autres parties du logiciel de suivi du regard. Tout ceci permet l'utilisation de cet algorithme dans un logiciel accessible aux personnes lourdement handicapées, notre objectif est donc atteint.

Pour améliorer les résultats, diverses pistes sont envisageables : utiliser le filtre de Kalman [Kal60], afin de s'aider des résultats précédents pour déterminer le centre de la pupille et ainsi gagner en efficacité, où encore utiliser des techniques sous pixels [ZY02], afin de gagner en précision.

## References

- [BMM00] BETKE M., MULLALLY W., MAGEE J.: Active detection of eye scleras in real time. In *Proceedings of the IEEE Workshop on Human Modeling, Analysis and Synthesis, June 2000*. (2000).
- [Can86] CANNY J.: A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 8, 6 (1986), 679–698.
- [Duc02] DUCHOWSKI A. T.: A breadth-first survey of eye-tracking applications. *Behav Res Methods Instrum Comput* 34, 4 (November 2002), 455–470.
- [JZ02] JI Q., ZHU Z.: Eye and gaze tracking for interactive graphic display. In *Int. Symp. on Smart Graphics* (2002).
- [Kal60] KALMAN R.: A new approach to linear filtering and prediction problems. *Transactions of the ASME-Journal of Basic Engineering* 82, Series D (1960), 35–45.
- [KBS75] KIMME C., BALLARD D., SKLANSKY J.: Finding circles by an array of accumulators. *Commun. ACM* 18, 2 (1975), 120–122.
- [KR99] KIM K., RAMAKRISHNA R.: Vision-based eye-gaze tracking for human computer interface. In *IEEE International Conf. on Systems, Man, and Cybernetics, Tokyo, Japan* (1999).
- [KT02] KAWATO S., TETSUTANI N.: Detection and tracking of eyes for gaze-camera control. In *VI '02 : Vision Interface 2002* (2002).
- [LP06] LI D., PARKHURST D.: Open-source software for real-time visible-spectrum eye tracking. In *COGAIN Conference* (2006).
- [MMM06] MERAD D., METZ S., MIGUET S.: Eye and gaze tracking algorithm for collaborative learning system. In *International Conference on Informatics in Control, Automation and Robotics (IFAC/ICINCO 2006)* (2006).
- [SW95] SCHIELE B., WAIBEL A.: Gaze tracking based on face-color. In *International Workshop on Automatic Face and Gesture Recognition, Zurich* (1995), pp. 344–349.
- [SYW96] STIEFELHAGEN R., YANG J., WAIBEL A.: A

model-based gaze tracking system. In *IEEE International Joint Symposia on Intelligence and Systems* (1996), pp. 304–310.

[TKC99] TIAN Y., KANADE T., COHN J.: Dual-state parametric eye tracking. In *International Conference on Face and Gesture Recognition*, 1999. (1999).

[VRR98] VARCHMIN A., RAE R., RITTER H.: Image based recognition of gaze direction using adaptive methods. *Lecture Notes in Computer Science 1371* (1998), 245+.

[ZJ05a] ZHU Z., JI Q.: Eye gaze tracking under natural head movements. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (2005), vol. 1, pp. 918–923.

[ZJ05b] ZHU Z., JI Q.: Robust real-time eye detection and tracking under variable lighting conditions and various face orientations. *Comput. Vis. Image Underst.* 98, 1 (2005), 124–154.

[ZJFL02] ZHU Z., JI Q., FUJIMURA K., LEE K.: Combining kalman filtering and mean shift for real time eye tracking under active ir illumination. In *ICPR '02: Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 4* (2002).

[ZY02] ZHU J., YANG J.: Subpixel eye gaze tracking. In *FGR '02: Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition* (2002).