Détection statistique de segments significatifs sur rétine programmable

Nicolas Burrus

15 septembre 2005

Résumé Ce travail se place à l'intersection de deux approches à priori indépendantes du traitement des images : une approche matérielle et algorithmique basée sur l'architecture cellulaire massivement parallèle des rétines artificielles ; et une approche *a contrario* statistiquement fondée de la perception cherchant à minimiser le nombre de paramètres nécessaires pour analyser les images. Nous nous proposons d'étudier la combinaison de ces deux univers à travers un opérateur simple et générique : l'extraction de segments significatifs.

1 Introduction

Récemment, des travaux ont amené à maturité le concept de rétine artificielle, rapprochant les imageurs des rétines biologiques, en intégrant une puissance de calcul directement au niveau des capteurs. Parallèlement à ce mouvement, divers travaux tendent à assainir les algorithmes de traitement des images en développant des modèles statistiques *a contrario* permettant de s'affranchir de seuils ou de quantités numériques définies *a priori*. L'ambition de ce travail est d'utiliser le meilleur des deux approches pour obtenir des algorithmes à la fois efficaces, robustes et sans paramètres.

La partie matérielle et algorithmique s'appuie sur les travaux qui ont été accomplis ces 20 dernières années visant à développer des matrices imageuses composées de pixels intelligents, i.e. ne se contentant pas d'acquérir l'information lumineuse, mais effectuant également des traitements directement dans le plan focal, au même niveau que les capteurs. Les intérêts sont multiples. Les transferts de données sont réduits de façon drastique, seules des données déjà traitées voire structurées étant à transmettre aux étapes de traitement de plus haut niveau, se traduisant par des économies d'énergie très importantes. De plus, comme chaque pixel dispose d'une certaine capacité de calcul, ces structures cellulaires massivement parallèles offrent des performances optimales pour des traitements de bas voire moyen niveau. Finalement, l'encombrement réduit de ces rétines artificielles en fait des éléments très adaptés pour des systèmes embarqués ou de petite taille. Le terme "rétine" vient de l'analogie avec la rétine du monde animal, qui ne se contente pas de recevoir la lumière, mais effectue également une certaine gamme de traitements.

De nombreux modèles ont été développés selon ce principe, analogiques ou numériques, on se référera par exemple à [Paillet, 2001] pour une étude comparative détaillée. Le modèle qui nous a servi de cadre de pensée est une rétine numérique programmable développée à l'ENSTA par Thierry Bernard [Paillet et al., 1999, Bernard, 1992]. Chaque pixel embarque un processeur booléen et un capteur photosensible, aboutissant à une grille de processeurs fonctionnant de façon complètement parallèle. La rétine est généra-lement couplée à un processeur scalaire classique, couramment appelé cortex, permettant de la contrôler et d'analyser les données prétraitées. L'objectif des algorithmes développés dans cet article sera donc de tirer parti au maximum des possibilités de traitement parallèle de l'image, pour ne restituer au cortex que des informations les plus structurées possibles. La rencontre entre rétines et outils statistiques est d'autant plus intéressante que la rétine intègre un sommateur analogique, permettant de calculer des distributions de probabilités de façon extrêmement rapide.

En complément de cette approche algorithmique vient donc une volonté d'analyser les images de façon statistique, avec un minimum de paramètres ou avec des paramètres qui soient fondés sur des grandeurs ayant un sens physique. Récemment, les travaux du CMLA [Desolneux et al., 2003b] ont ouvert la voie dans ce sens, en cherchant à détecter les éléments de la théorie des Gestalts [Wertheimer, 1923, Desolneux et al., 2004] en se basant sur le principe de Helmholtz, un principe qui permet de se passer de modèle

statistique *a priori* en cherchant *a contrario* les objets qui s'éloignent très fortement d'un modèle aléatoire uniforme. Différentes gestalts ont été détectées avec succès en suivant ce principe [Desolneux et al., 2003a, Almansa et al., 2003].

L'approche ascendante de la vision artificielle [Marr, 1982], cherchant à analyser l'image progressivement de façon générique, en partant de traitements simples pour arriver à une analyse de haut niveau, a été largement abandonnée par la communauté faute de résultats pratiques, au profit de traitements dirigés par des applications précises. Cependant, certaines primitives, comme la détection de contours, semblent être suffisamment universelles et utiles pour continuer à mener en parallèle une recherche de traitements génériques et une recherche de traitements adaptés à des applications particulières.

Ce travail s'attache à étudier le potentiel d'un rapprochement entre les systèmes rétiniens et les travaux du CMLA à travers un exemple simple d'opérateur générique utile en milieu artificiel : la détection de segments statistiquement significatifs. De nombreuses approches ont été proposées par le passé. La transformée de Hough [Maître, 1985] sert de base pour beaucoup d'entre elles, et [Desolneux et al., 2000] propose de choisir les pics significatifs d'une version simplifiée de la transformée selon le principe de Helmholtz. Dans un cadre rétinien, les approches plus locales cherchant à connecter des portions de contours sont plus adaptées [Nevatia and Babu, 1980], mais nécessitent de fixer des seuils que nous nous proposons de régler automatiquement par un raisonnement statistique.

Étant donné la forte composante statistique de ce travail, nous ne nous sommes pas attachés à implémenter concrètement les algorithmes dans l'assembleur de la rétine, mais elle a servi de contexte de pensée et d'inspiration algorithmique. La volonté statistique se traduit par l'absence de quelconque paramètre à modifier d'une image à l'autre.

Ce rapport est organisé comme suit : la section 2 donne une description plus détaillée du contexte rétinien, puis la section 3 rappelle les éléments essentiels de la théorie des évènements ϵ -significatifs. Une première approche de l'extraction de segments basée uniquement sur un critère de taille est expliquée dans la section 4, puis une version fondée sur le contraste est donnée dans la section 5. Regorgeant de détails techniques, ces algorithmes sont décrits de façon superficielle, les principes mis en jeu étant plus importants que les détails de mise en oeuvre. La section 6 illustre les algorithmes sur différentes images usuelles et discute des aspects implémentatoires sur une architecture classique et sur un système cellulaire. Finalement, la section 7 fait ressortir l'intérêt de ce travail et discute des résultats, avant de conclure avec la section 8 sur les perspectives à court et moyen termes.

2 La rétine programmable

Nous nous attachons ici à décrire plus précisément le modèle de rétine programmable développé dernièrement à l'ENSTA. Il s'agit d'une grille de 200x200 pixels ou cellules, chacune d'entre elles étant constituée d'un processeur et d'un capteur. Un schéma récapitulatif est donné dans la figure 1. Chaque processeur dispose des capacités suivantes :

- une mémoire de 50 bits;
- des opérations booléennes simples ("ET", "NON", "OU", etc.);
- la possibilité d'accéder à la mémoire de ses 4 voisins.

La rétine fonctionne sur un modèle SIMD : les processeurs reçoivent tous la même instruction au même instant. Il est à noter également que les opérateurs implantables sont nécessairement locaux, chaque processeur ne pouvant lire que la mémoire de ses voisins.

Deux types de données sont extractibles de la rétine et peuvent être analysées par le cortex :

- le contenu de la mémoire, séquentiellement ;
- des descripteurs scalaires, comme le nombre de pixels à 1 dans un certain plan de bit, extrait par un sommateur analogique, donc en temps constant très faible.

Bien entendu, pour obtenir un système performant et pour utiliser au mieux les capacités de la rétine, il est primordial de limiter au maximum les accès séquentiels à la mémoire. Par contre, la possibilité de déterminer le nombre de bits d'un certain plan en un temps extrêmement réduit est idéal pour calculer efficacement des distributions de probabilité. On tâchera donc d'éviter ou au moins de retarder au maximum les échanges massifs d'information entre la rétine et le cortex.

L'acquisition de l'image se fait quant à elle par ensembles de niveau : pour chaque niveau de gris une image binaire est générée, les pixels à 1 étant ceux qui ont un niveau de gris supérieur ou égal au niveau courant. Cette information peut ensuite être encodée pour stocker une image en niveaux de gris sur un minimum de bits.



FIG. 1: Un système à base de rétine programmable : la rétine peut-être vue comme une grille de couples microprocesseur et capteur photosensible. Les processeurs exécutant tous la même instruction, les opérations mémoires reviennent à manipuler des plans de bits (50 dans la dernière génération). Les opérations disponibles se réduisent à des décalages de plans de bits et à des opérations booléennes entre plans. Le cortex peut décider des instructions à envoyer et traite les données transformées par la rétine. Cet environnement matériel, qui peut sembler très restrictif, a pourtant déjà prouvé sa capacité à effectuer nombre d'opérations classiques de bas niveau, par exemple de la détection de mouvement [Manzanera and Richefeu, 2004] ou de la morphologie mathématique [Manzanera, 2000]. Assez paradoxalement, en plus de l'apport en termes de performance, le cadre rétinien, en restreignant le cadre de réflexion, a permis le développement d'algorithmes nouveaux et intéressants également sur une architecture classique.

Pour l'instant, uniquement des opérations locales ont pu être implémentées, mais une nouvelle génération de rétines [Gies and Bernard, 2005] apportera des primitives régionales. La principale amélioration consiste à rendre les connexions entre pixels programmables, de manière à définir des régions connectées. Ensuite, des calculs régionaux peuvent être effectués, le plus simple étant la somme des valeurs des pixels sur une région. Ces calculs sont menés de façon asynchrone, permettant à chaque région de travailler indépendamment. Aussi les algorithmes présentés dans ce rapport, s'ils essayent au maximum de rester dans le cadre de la génération actuelle de rétine, envisagent parfois certaines extensions s'appuyant sur des fonctionnalités des rétines futures. La complexité des algorithmes sur des rétines régionales est analysée dans la section 6.2.

3 Approche statistique et sans paramètres du traitement des images

3.1 Introduction

Cette section a pour but de définir le cadre statistique utilisé dans ce travail, selon les principes énoncés par [Desolneux et al., 2003b]. Cette théorie s'appuie sur le principe de Helmholtz, qui peut s'énoncer de manière intuitive de la façon suivante : "certaines primitives (lignes, courbes, groupements, etc.) sont perceptibles si l'espérance de leur nombre d'apparition dans une image uniformément aléatoire est très faible". En suivant ce principe, pour déterminer si une primitive observée est significative ou non, il est uniquement nécessaire de calculer sa probabilité d'apparition dans un environnement aléatoire uniforme.

Cette approche permet notamment d'élaguer un problème récurrent du cadre bayésien, reposant sur un critère de maximum *a posteriori*, qui nécessite soit de calculer une probabilité à priori, soit de supposer qu'elle est uniformément distribuée, et de se contenter d'un critère de maximum de vraisemblance.

lci, l'a priori porte sur la nature des primitives et sur les critères de décision. Le seul paramètre subsistant ϵ est le nombre maximal d'apparitions espéré dans une image aléatoire pour que l'évènement soit considéré significatif. Cependant dans la plupart des cas $\epsilon = 1$ est une valeur satisfaisante, et son influence est souvent logarithmique, on peut donc fixer sa valeur et ne plus le considérer comme un paramètre.

Le cadre formel a été défini par [Desolneux et al., 2003b], nous reprenons ici les principales définitions.

Definition 1 Soit une configuration d'objets C. Soit $p_{uniform}(C)$ sa probabilité d'apparition dans un environnement aléatoire uniforme. Soit $N_{candidates}$ le nombre d'évènements susceptibles de donner lieu à la configuration C. Le nombre de fausses alarmes de C est :

$$NFA(C) = p_{uniform}(C) \times N_{candidates}$$

Le nombre de fausses alarmes représente le nombre de fois auquel on peut s'attendre à rencontrer une certaine configuration d'objets dans un environnement aléatoire. Par exemple, considérons un ensemble d'objets. Chaque objet peut avoir la couleur bleue ou rouge avec une probabilité $\frac{1}{2}$. On effectue 100 tirages de 5 objets à chaque fois. Soit la configuration C : "les 5 objets sont rouges". Le nombre de fausses alarmes NFA(C) correspond au nombre de fois où on s'attend à avoir un groupe de 5 objets rouges, si les objets étaient distribués aléatoirement. Ici $p_{uniform}(C) = (\frac{1}{2})^5$ et $N_{candidates} = 100$. On obtient donc $NFA(C) = 0.0041 \times 100 = 3.12$. On s'attendrait donc à obtenir un peu plus de 3 groupes de 5 objets rouges sur les 100.

Definition 2 Une configuration C est ϵ -significative si :

 $NFA(C) \leq \epsilon$

En pratique, on choisit presque toujours $\epsilon = 1$. Par la suite, on dira d'un évènement qu'il est significatif s'il est 1-significatif. Ici, le fait d'observer 5 objets rouges n'est pas significatif, puisqu'on s'attend à obtenir cette configuration 3.12 fois. Si par contre on s'intéresse à des groupes de 10 objets rouges, on obtient un nombre de fausses alarmes : $NFA(C) = (\frac{1}{2})^{10} \times 100 = 0.0977$ inférieur à 1, l'évènement est donc significatif.

3.2 Extraction de contours significatifs

Le travail qui suit s'est fortement inspiré des travaux sur les contours significatifs menés dans [Desolneux et al., 2001, Cao et al., 2005] et [Cao, 2004]. Nous résumons ici succinctement leurs méthodes.

Les deux approches s'appuient sur les lignes de niveau, qui sont *a priori* de bonnes candidates puisqu'elles sont orthogonales au gradient et ont une intensité constante. [Monasse and Guichard, 2000] fournit une implémentation rapide d'extraction de lignes de niveau sur une architecture classique. [Desolneux et al., 2003b] définit le nombre de fausses alarmes d'une portion de ligne de niveau *L* en fonction de l'intensité de gradient minimale $\mu(L)$ sur la portion :

$$NFA(L) = P[k \le \mu(L)]^l \times N$$

où $P[k \le \mu(L)]$ est déduite de la distribution des valeurs de gradient sur l'image, N est le nombre de portions de lignes de niveau dans l'image, et l le nombre de points de la portion. Ce nombre de fausse alarmes correspond simplement à la probabilité d'avoir tous les points de L avec un gradient supérieur ou égal à $\mu(L)$, multiplié par le nombre de portions candidates. Ici la supposition *a contrario* réside dans l'hypothèse d'indépendance des intensités de gradient entre les points de la ligne de niveau, ce qui est d'autant plus faux que l'on se situe sur un contour.

D'un point de vue calculatoire, cette méthode requiert le calcul de toutes les lignes de niveau de l'image, puis le calcul de NFA(L) pour toutes les portions de ces lignes. Il s'agit donc d'une méthode computationnellement lourde, et nécessitant des structures de donnée (listes chainées, etc.) complètement inadaptées à un système à base de rétine.

Parallèlement, [Cao, 2004] s'appuie sur un critère de signifiance purement géométrique, ne prenant en compte que la régularité de la direction du gradient le long des portions de ligne de niveau. Le nombre de fausses alarmes devient :

$$NFA(L) = P[k \le \theta_{max}]^l \times N \times N_{\theta}$$

où θ_{max} correspond à l'angle maximal entre les directions de gradient pour deux points consécutifs le long de L, et N_{θ} au nombre d'angles possibles (on discrétise les directions). Ici il y a une double hypothèse de distribution aléatoire uniforme : la distribution de direction du gradient en chaque point est considérée uniforme entre 0 et N_{θ} , et les directions sont considérées indépendantes entre 2 points consécutifs d'une ligne de niveau.

La complexité de l'algorithme est comparable à l'approche de [Desolneux et al., 2001], et donc difficilement plus adaptable. Mais cette approche a donné naissance aux travaux initiaux détaillés dans la section suivante, où l'idée était de sélectionner uniquement des suites de points avec $d\theta_{max}$ inférieur à un certain seuil fixé, rendant le nombre de fausses alarmes uniquement dépendant de la longueur de la suite de points vérifiant le prédicat. Cette simplification a finalement donné lieu à la notion de segments significatifs, caractérisés par le fait que les directions de gradient locales ne s'éloignent pas de la direction principale du segment d'un angle de plus de $\frac{\pi}{4}$. Cette réduction de complexité rend possible une implémentation sur rétine, comme le montrent les sections suivantes.

4 Extraction de segments sur rétine par un critère de taille

Suivant l'approche statistique présentée dans la section précédente, l'algorithme peut se diviser en deux parties :

- la génération de segments candidats;

- la sélection des segments significatifs parmi les candidats.

La génération des segments candidats se fait selon un *a priori* géométrique, puis la sélection s'appuie sur le principe de Helmholtz pour décider de la perceptibilité de chaque candidat. La sélection nous garantit que nous ne laisserons pas passer de candidats non perceptibles, aussi l'étape de génération des candidats doit être suffisamment souple pour couvrir tous les segments potentiels.

4.1 Extraction des segments candidats

Definition 3 Un segment dans la direction d est un ensemble de points connectés tels que la direction locale de chaque point soit comprise entre d - 1 et d + 1.

Cette définition d'un segment donne une certaine robustesse et permet de générer un ensemble de candidats qui soit un sur-ensemble des segments perceptibles dans une image. Un exemple de segment est donné dans la figure 6.

L'algorithme se déroule en plusieurs étapes, récapitulées dans la figure 2. La première étape consiste à extraire les directions locales en chaque point. Puis l'algorithme va détecter les chaînes maximales de points connectés partageant la même direction, ces ensembles de points constitueront les candidats de la section suivante.

4.1.1 Calcul des directions locales

Pour déterminer les directions locales présentes en chaque point, le vecteur gradient n'est pas forcément l'outil le plus adapté, étant donné qu'il peut y avoir plusieurs directions en un point, résultant en un vecteur moyen peu pertinent. De plus, la rétine ne permet pas de stocker beaucoup de directions différentes, se qui conduit à se limiter aux 8 directions possibles sur un voisinage 3x3, comme le montre la figure 3.

Pour savoir si une direction d est présente en 1 point, l'algorithme suivant est appliqué :

- Pour chaque ensemble de niveau :
 - extraction des lignes de niveau (image érodé(image));
 - application de transformées en tout ou rien correspondant aux configurations locales possibles pour d;
 - les points restants contiennent la direction d.

Les transformées en tout ou rien s'implémentent facilement et efficacement sur rétine [Manzanera, 2000]. L'approche par ensembles de niveau se justifie pour trois raisons :

- les lignes de niveau sont orthogonales au gradient et reflètent donc bien les directions locales ;
- en agissant niveau par niveau, plusieurs directions peuvent être extraites en un même point ;
- dans la génération actuelle de rétine, l'acquisition des images se fait par ensembles de niveau, il donc possible de calculer les directions dès l'acquisition.

Finalement, pour chaque direction, une image binaire va être calculée, où la valeur en un pixel est 1 si le pixel contient la direction locale courante, 0 sinon. Chaque image est ensuite combinée avec l'image direction précédente et l'image direction suivante, de manière à obtenir pour chaque direction d: directions(d, x, y) = 1 si d, d - 1 ou d + 1 est présente au point (x, y). Le processus est détaillé sur la figure 4.

Le résultat de l'algorithme est donc 8 images binaires, qui vont être utilisées dans l'étape suivante. Un exemple est donné dans la figure 5.

4.1.2 Extraction des chaînes connectées maximales

L'algorithme d'extraction des segments est présenté dans l'esprit de la rétine locale classique actuellement disponible à l'ENSTA. La section 6.2 discute des possibilités d'implémentation sur une rétine régionale. Pour la rétine classique, la méthode repose sur un rognage itératif des extrémités de segments. A chaque étape, la longueur de chaque chaîne connectée est donc réduite de 2. A l'étape *n*, les points subsistants appartiennent donc à des segments de longueur au moins 2*n*. Quand un point n'est plus connecté avec aucun voisin, on sait donc qu'il correspond au centre géodésique d'une chaîne de longueur exactement 2*n*. Ces points ultimes (leur détection se base sur le principe des érodés ultimes) sont utilisés par la suite comme représentants des segments. Par propagation à chaque itération de l'algorithme, le nombre de points total est attaché aux représentants pour l'étape de sélection. En effet, les segments ont généralement une certaine épaisseur, et si leur longueur diminue de 2 pixels à chaque itération, on ne peut prédire leur largeur.

Le rognage des extrémités La détection des points extrémités est illustrée sur la figure 6. Avant d'être supprimés de l'image de direction courante, ils vont transmettre deux informations à leurs voisins qui deviennent les nouveaux points extrémités :

- le nombre de points cumulés jusqu'à maintenant, + 1 ;
- leur emplacement par rapport à chacun des voisins, pour permettre la reconstruction ultérieure des segments à partir de leur point central représentant.



FIG. 2: La première étape calculer 8 images binaires correspondant à la présence ou non d'une direction en chaque point. Des points représentants des segments sont ensuite extraits par un processus de rognage itératif des extrémités. Ces points sont ensuite filtrés par un critère statistique. Finalement, les segments jugés significatifs peuvent être reconstruits à partir de leurs représentants.



FIG. 3: Les 8 directions possibles sur un voisinage 3x3, avec leurs masques respectifs.

Détection des points ultimes Le processus est illustré dans la figure 7. Il s'agit simplement de la différence entre l'image à l'itération n + 1 et sa reconstruction dans l'image n, en utilisant la connexité liée à la direction générale analysée. Les points ne pouvant pas être reconstruits sont ceux qui n'avaient plus de voisins à l'étape n, et qui correspondent donc à un segment de longueur 2n.

Filtrage des points ultimes Le filtrage des points ultimes se fait ensuite en fonction du nombre de points total du segment en utilisant les critères statistiques de la section suivante.

Reconstruction des segments Une fois que seuls les points représentants significatifs ont été conservés, les segments peuvent être reconstruits dans l'image de direction originale de façon itérative en utilisant les informations de voisinage stockées auparavant.

4.2 Sélection des candidats significatifs

Pourquoi est-ce difficile ? En utilisant l'approche statistique de la section 3, on s'intéresse à la probabilité d'avoir une chaîne de *n* points connectés dans une image de direction donnée. La supposition aléatoire uniforme naturelle ici est de considérer que les points à 1 dans l'image de direction sont répartis uniformément. On peut donc déterminer la probabilité p_{set} qu'un point tiré au hasard soit à 1 dans l'image de direction :

$$p_{set} = \frac{\text{nombre de points à 1}}{\text{nombre de points total}}$$

Cette estimation peut se faire en temps constant sur la rétine en utilisant le sommateur analogique. Un segment est une chaîne de *n* points à 1, sa probabilité d'apparition sous ces hypothèses est donc :

$$p_{segment}(n) = (p_{set})^n$$

Pour calculer le NFA d'un segment, il nous manque encore le nombre de candidats. Malheureusement ce nombre est très difficile à estimer. En effet, l'algorithme considère implicitement toutes les chaînes de l'image dans un certain cône de direction. De plus, il peut y avoir des branchements. Le nombre de



Transformées en tout ou rien pour obtenir chaque image de direction

FIG. 4: Sur chaque ensemble de niveau une extraction de contour donne les lignes de niveau. Un ensemble de transformées en tout ou rien permet de détecter la présence de chacune des 8 directions. Finalement, les images voisines sont regroupées.

FIG. 5: (a) Image originale (b) Image binaire pour la direction horizontale. Un pixel est allumé si la direction y est détectée. (c) Image binaire pour la direction verticale.



FIG. 6: Les points extrémités sont détectées par une batterie de transformées en tout ou rien, correspondant à chacun des cas possibles. A chaque itération, la longueur des segments est réduite de 2 pixels. Quand un pixel est tout seul, on peut donc calculer la longueur du segment correspondant. Le nombre de points total est propagé au cours du processus, permettant de prendre en compte l'épaisseur du segment.



FIG. 7: Le principe est similaire au calcul des érodés ultimes en morphologie mathématique. Si un point n'a plus de voisin, il ne pourra pas être reconstruit dans l'image précédente et aura disparu.

candidats se résume finalement à un calcul de dénombrement délicat. C'est pourquoi le paragraphe suivant tente de simplifier le problème.

Simplification par rectangle englobant Plutôt que de considérer une chaîne connectée comme candidate, nous proposons de nous intéresser à la région définie par son rectangle englobant. La signifiance de ces rectangles viendra du fait qu'ils contiennent beaucoup de points à 1. Le nombre de rectangles orientés selon une certaine direction est cette fois ci simple à calculer (un rectangle peut être défini par un couple de points) :

$$N_{candidates} = \frac{1}{2} \times (N-1) \times (N-2)$$

pour image de taille $N \times N$.

Nous mesurons ensuite le nombre de points à 1 dans le rectangle candidat, et la probabilité p_{rect} d'avoir ce rectangle si les points étaient distribués uniformément devient :

$$p_{rect} = \binom{n}{k} \times (p_{set})^k \times (1 - p_{set})^{(n-k)}$$

où *k* est le nombre de points à 1, et *n* le nombre de points dans le rectangle. Il s'agit donc simplement d'une loi binomiale. En réalité, pour calculer le nombre de fausses alarmes, on ne s'intéresse pas tant à la probabilité d'avoir exactement cette proportion mais plutôt à la probabilité d'en avoir au moins autant, p_{rect} devient donc :

$$p_{rect} = \sum_{i=k}^{n} {n \choose i} \times (p_{set})^{i} \times (1 - p_{set})^{(n-i)}$$

En pratique, cette probabilité peut-être calculée en utilisant la loi log-gamma, qui peut-être estimée facilement par l'approximation de Lanczos. Ensuite, comme $\Gamma(n+1) = n!$ il est immédiat de calculer p_{rect} même pour *n* grand.

Ainsi le NFA peut être calculé :

$$NFA = p_{rect} \times N_{candidates}$$

La contrepartie de cette approche est une sur-estimation du nombre de candidats, puisqu'elle ne prend pas en compte le fait que les pixels soient connectés. La saillance des éléments est donc légèrement sous-estimée, pouvant conduire à rejeter trop sévèrement certains candidats.

Estimation par simulation Si la taille des images est connue à l'avance, il est possible de recourir à une méthode de simulation stochastique pour établir un seuil sur le nombre de points minimal d'un segment en fonction de p_{set} pour qu'il soit significatif. L'algorithme est le suivant :

- générer une image binaire aléatoire selon p_{set} ;

- regarder le nombre de points du segment le plus grand.

On applique ce processus *n* fois, puis on choisit comme seuil de signifiance λ , la taille immédiatement supérieure à la plus grande observée. On peut alors s'assurer que les segments ayant une taille supérieure à λ sont significatifs avec une confiance dépendant de *n*. Numériquement, pour *n* = 100, un encadrement largement majorant par intervalle de confiance donne avec une fiabilité de 95% :

$$P[NFA_{size \ge \lambda} < 1] > 0.92$$

La probabilité que ces segments soient significatifs est donc supérieure à 0.92.

En augmentant *n*, les segments de taille supérieure ou égale à λ peuvent être déclarés significatifs avec une plus grande confiance.

En répétant ce calcul pour différentes valeurs de p_{set} , on obtient une table des seuils. Il suffit ensuite d'appliquer un seuil sur les points ultimes pour ne garder que ceux qui sont significatifs.

La sélection peut donc s'effectuer directement sur la rétine, s'agissant de simples comparaisons d'entiers.



FIG. 8: Nombre de segments saillants en fonction du seuil global sur le gradient sur l'image du bureau. La courbe a été lissée pour rendre compte des tendances globales.

Optimisation du nombre de segments significatifs Cet algorithme n'utilise à aucun moment l'information de gradient. Les directions locales sur les points avec une intensité faible ne sont en général pas pertinentes. Appliquer un seuil sur le gradient avant de calculer les images de directions permet d'élaguer les images binaires en conservant les segments. Le seuil optimal peut être détecté automatiquement, il s'agit du seuil qui permet de faire ressortir au maximum les structures saillantes et donc qui maximise le nombre de segments saillants. Un seuil trop bas surcharge les images de direction, et oblige les segments saillants à être très grands (p_{set} est élevée). En contrepartie, un seuil trop haut finit par briser les segments et diminuer le nombre de structures saillantes. Finalement, la fonction nombre d'éléments saillants en fonction du seuil global se rapproche d'une cloche inversée, un exemple est donné sur la figure 8.

5 Extraction de segments sur rétine par un critère de contraste

Si l'approche précédente a l'avantage d'être implémentable facilement sur un système rétine-cortex, elle n'utilise pas l'information de gradient propre à chaque segment. Par conséquent, les petits segments très contrastés ne sont pas détectés. On se propose donc dans cette section de détecter des segments significatifs en appliquant le principe de Helmholtz sur l'intensité de gradient le long des segments. Le schéma de principe est très similaire à la section précédente, après une extraction légèrement modifiée des segments candidats, un critère statistique est appliqué sur chacun d'entre eux pour ne garder que les candidats significatifs.

5.1 Extraction des segments candidats

La première étape est identique au début de la section 4.1. Les 8 images binaires de directions locales sont extraites de la même façon. Ensuite, un premier filtrage est appliqué sur chacune des images de direction, motivé par les constats suivants :

- un segment a généralement une certaine épaisseur;
- un segment épais avec un gradient peu élevé sur toute l'épaisseur est aussi significatif qu'un gradient très fort sur une faible épaisseur;
- la crête d'un segment correspond à l'ensemble des points de gradient maximal dans la direction orthogonale au segment, un raisonnement similaire à [Canny, 1986] pour la localisation des contours.

Ceci conduit à la volonté d'intégrer les valeurs de gradient sur l'épaisseur des segments, de façon à prendre en compte une éventuelle dilution du gradient sur la largeur. Cette étape permet également d'obtenir une localisation précise des segments.

L'algorithme utilisé pour cette étape se divise en 3 phases :

- calcul des valeurs de gradient en chaque point;
- intégration et regroupement des points sur leur ligne de crête ;
- régularisation de la localisation pour lisser les segments.



FIG. 9: (a) Image originale (b) Zoom sur une portion de l'image de direction horizontale. (c) Image de direction après localisation.



FIG. 10: Pour le calcul de la ligne de crête, à chaque itération les points se soumettent à leur voisin le plus fort et leur donnent leurs intensités de gradients. La localisation est ensuite optimisée de façon à maximiser le nombre de segments locaux.

Calcul des valeurs de gradient Pour l'image de direction *d*, l'intensité du gradient est calculée en chaque point comme la différence entre les valeurs de niveau de gris des pixels dans la direction orthogonale.

Regroupement des points Le regroupement des points s'effectue selon un algorithme local. A chaque itération, chaque point :

- regarde si un de ses 2 voisins dans la direction orthogonale a un gradient plus élevé ;

- si un voisin est plus fort, le point lui donne son intensité de gradient et se supprime.

Ce processus est illustré sur la figure 10. A l'issue de cette étape, les points dans les images de direction n'ont plus qu'une épaisseur de 1 pixel dans la direction orthogonale. A cause des imprécisions de localisation et des légères fluctuations du gradient, les lignes de crêtes résultantes ne sont pas toujours lisses, l'étape suivante cherche donc à déplacer les points décalés pour obtenir un segment régulier.

Régularisation L'étape de régularisation cherche simplement à optimiser le nombre de segment locaux, i.e. de suites de points connectés. Pour chaque point dans une image de direction, on regarde si en le déplaçant d'un pixel dans la direction orthogonale au segment on peut établir une connexion supplémentaire. Le processus est répété jusqu'à convergence. La figure 10 donne un exemple d'exécution de l'algorithme. Cette étape permet de simplifier la tâche d'extraction de segments qui suit. Un exemple de résultat est donné dans la figure 9.

Finalisation Il ne reste plus qu'à appliquer le même algorithme qu'à la section 4.1.2 pour extraire les segments candidats.

5.2 Sélection des candidats

Nous allons appliquer un principe similaire à [Desolneux et al., 2001], mais pour obtenir des résultats plus robustes, on s'intéresse à la valeur moyenne du gradient le long du segment au lieu de considérer le minimum. Ceci nous permet de ne pas avoir à explorer tous les sous-segments possibles pour obtenir des segments significatifs, puisque quelques valeurs faibles ne sont plus éliminatoires.

La supposition ici est que les valeurs de gradient sont sensées être distribuées uniformément parmi tous les segments candidats. On commence donc par calculer cette distribution, par une analyse fréquentielle des valeurs de gradient. Grâce au sommateur analogique, ceci peut être fait très rapidement sur une rétine.

Nombre de fausses alarmes pour les grands segments On peut calculer la moyenne vraie μ et l'écarttype vrai σ de cette distribution :

- N

et

$$\mu = \frac{\sum_{i=0}^{N} |grad(i)|}{N}$$
$$\sigma = \sqrt{\frac{\sum_{i=0}^{N} (|grad(i)| - \mu)^2}{N}}$$

où N est le nombre de points total.

Pour un segment donné, on peut également calculer sa moyenne empirique μ_s . Soit N_s le nombre de points du segment. Si N_s est suffisamment grand (> 30 en pratique), on sait d'après le théorème central limite que μ_s suit une loi normale $L(\mu, \frac{\sigma}{\sqrt{N_s}})$. Ce qui nous intéresse pour calculer le NFA, c'est la probabilité que la moyenne du segment soit supérieur ou égal à μ_s si les points étaient distribués aléatoirement. Si cette probabilité est suffisamment faible, le segment sera considéré significatif.

Cette probabilité correspond à la fonction de répartition de la loi normale :

$$P[k \ge \mu_s] = \frac{1}{2} \times erfc(\frac{\mu_s - \mu}{\sqrt{2} \times \frac{\sigma}{\sqrt{N_s}}})$$

où erfc(x) est le complémentaire de la fonction d'erreur :

$$erfc(x) = \frac{2}{\sqrt{\pi}} \int_{x}^{\infty} e^{-t^2} dt$$

Le nombre de fausses alarmes pour les segments suffisamment grands est donc :

$$NFA_{N>30} = P[k \ge \mu_s] \times N_{segments}$$

où $N_{segments}$ est le nombre total de segments candidats.

Ce critère peut s'interpréter intuitivement : une moyenne très élevée tend à faire diminuer le NFA, et un nombre de points élevé également, en diminuant l'écart-type de la gaussienne $(\frac{\sigma}{\sqrt{N}})$. Ainsi, il faudra une moyenne d'autant plus élevée que le nombre de points est faible pour obtenir un segment significatif.

Nombre de fausses alarmes pour les petits segments L'approximation de la distribution de la moyenne empirique par une loi normale n'est plus valable. Deux options s'offrent à nous :

 utiliser la fonction génératrice de la loi pour déterminer la loi de la moyenne empirique dans le cas général;

utiliser le critère du gradient minimal de [Desolneux et al., 2001].

La première solution est assez lourde en calculs, elle nécessite de calculer la fonction génératrice de la distribution du gradient, de l'élever à la puissance du nombre de points, puis d'effectuer la transformation inverse. En effet, si $X_1 \dots X_n$ sont des variables aléatoires indépendantes et identiquement distribuées, avec comme fonction génératrice ϕ_X , la loi de $X_1 + \dots + X_n$ peut être déduite à partir de $\phi_{X_1+\dots+X_n}(x) = \phi_X(x)^n$.

Cependant, pour les petits segments, l'approche par le minimum semble raisonnable, et donne de bons résultats en pratique. C'est donc celle qui a été choisie.

Le NFA devient donc la probabilité d'avoir tous les points avec une intensité supérieure au minimum multipliée par le nombre de segments :

$$NFA_{N<30} = P[X \ge \lambda]^N \times N_{segments}$$

où λ est le gradient minimal du segment.

Optimisation des segments finaux Il arrive souvent que les segments candidats ne s'arrêtent pas au bord des objets, mais continuent légèrement dans le bruit environnant. La moyenne du gradient sur la longueur totale leur permet de rester significatifs, mais le NFA pourrait être diminué en enlevant les extrémités des segments qui ont un faible gradient et font diminuer la moyenne.

- Il est donc possible d'appliquer l'étape d'optimisation suivante pour chaque segment significatif :
- déterminer l'extrémité avec le gradient le plus faible ;
- si sans ce point le segment est plus significatif, l'enlever et recommencer;
- sinon arrêter.

Les points avec des valeurs en dessous de la moyenne vont avoir tendance à faire augmenter le NFA. De façon opposée, diminuer le nombre de points fait également augmenter le NFA. L'optimum se trouve donc à un compromis entre une moyenne élevée et un nombre de points important.

A propos de l'indépendance des points Pour simplifier l'écriture, nous avons considéré que tous les points d'un segment étaient indépendants. Mais dans une image correctement échantillonnée, 2 pixels ne sont indépendants que si leur distance est supérieure à 2 (la fréquence de Nyquist). Il ne faut donc considérer que $\frac{N}{2}$ points pour le calcul du nombre de fausses alarmes si on a un segment de *N* points.

6 Résultats

6.1 Implémentation sur une architecture classique

Les algorithmes présentés ont été simulés sur un ordinateur classique. Cette section illustre les différentes approches proposées, et discute de leurs qualités respectives.

Résultats basés sur le nombre de points des segments La figure 11 montre les résultats obtenus par un filtrage basé sur le nombre de points des segments uniquement. Cette méthode souffre de plusieurs inconvénients :

- les segments obtenus sont épais et grossièrement localisés;
- les segments peuvent passer d'un objet à l'autre;
- les segments ne sont pas très réguliers ;
- les petits segments très contrastés ne sont pas détectés;
- quand la densité dans les images de direction est trop importante, par exemple dans des textures, des segments précis sont difficilement extractibles. Un seuil de gradient global est ici particulièrement utile.

Ces limitations sont contrebalancées par des avantages significatifs :

- l'extraction des segments candidats est robuste, l'épaisseur des segments permet de s'assurer qu'un chemin connecté sera toujours trouvé;
- en environnement artificiel, les segments principaux sont tous détectés correctement ;
- aucun seuil ni paramètre manuel n'est nécessaire ;
- la détection s'adapte aux différentes conditions de luminosité ou de contraste ;
- la détection s'adapte à l'environnement : très sélective si beaucoup de segments candidats sont présents, beaucoup moins dans un environnement lisse;
- les segments sont détectés individuellement, et donc manipulables séparément ultérieurement, il ne s'agit pas d'un filtre réhausseur de contraste;
- la méthode est rapide et entièrement implantable sur rétine.

Enfin, la complexité de l'algorithme sur une architecture scalaire est modérée si des structures algorithmiques adaptées sont utilisées (files d'attente pour l'extraction des segments par exemple).



FIG. 11: A gauche : image originale. Au milieu : image et segments significatifs par leurs nombre de points. Les points rouges représentent les centres des segments détectés. Le segment en blanc montre un des segments détectés. A droite : segments seuls.

Résultats basés sur le contraste La figure 12 montre les résultats obtenus par un filtrage basé sur le contraste. Ici également, aucun paramètre n'est fixé empiriquement ni modifié entre les différentes images. Le bilan est comparable à la méthode précédente, à quelques différences près :

- les petits segments très contrastés sont bien détectés;
- les segments font un pixel d'épaisseur et sont localisés sur la crête du gradient ;
- l'algorithme est peu sensible à l'application d'un seuil global sur le gradient.

Les textures donnent toujours une forte densité de points dans les images de direction, avec des valeurs de gradient élevée, donnant au final de nombreux segments peu pertinents. Pour compenser ce problème, on pourrait appliquer un filtre de régularité sur les niveaux de gris le long des segments, toujours basé sur le principe de Helmholtz. Les segments devraient alors satisfaire les deux critères de saillance pour être conservés.

6.2 Implémentabilité sur rétine?

Les résultats présentés ont été réalisés sur un ordinateur traditionnel, mais en prenant soin de n'utiliser que des opérateurs implémentables sur une rétine cellulaire. Cette section analyse plus précisément la complexité des différentes techniques sur une rétine synchrone et sur une rétine asynchrone (régionale).

Calcul des images directions Se situant au plus bas niveau, l'extraction des images de directions se fait de façon optimale sur une rétine synchrone (et donc sur une rétine asynchrone également). Le processus pourrait même être intégré à l'étape d'acquisition si le nombre et la mise en oeuvre des transformées en tout ou rien arrivait à être optimisé pour être suffisamment rapide. Il n'est pas nécessaire non plus d'effectuer les calculs pour chaque ensemble de niveau, 50% des niveaux sont suffisants en pratique. Le processus général, massivement parallèle, ne dépend donc pas de la taille de l'image, et ne dépend que du nombre de niveaux de gris et du nombre de transformées à effectuer, des quantités constantes.

Extraction des segments Sur une rétine synchrone, l'extraction des segments est sub-optimale. A chaque étape seuls les pixels actifs dans la direction courante sont utilisés. La reconstruction des points ultimes fait intervenir encore moins de pixels. En outre, le nombre d'itérations de l'algorithme dépend de la taille du segment le plus long, autrement dit de la taille de l'image. Il est clair que cette étape ne tire pas pleinement parti des possibilités de la rétine synchrone, et c'était prévisible, nous manipulons des entités distinctes qui sont intrinsèquement régionales.

Sur une rétine régionale, la complexité serait diminuée de façon drastique. En effet, les chaînes connectés peuvent être détectées en temps constant, en liant chaque pixel à son voisin s'ils font partie d'un même segment local. Ensuite, des quantités régionales pourront être calculées simultanément sur tous les segments.

Calcul des distributions La quantité de pixels sur une image de direction donnée est en temps constant et négligeable grâce au sommateur analogique. p_{set} est donc optimale à calculer sur les deux modèles de rétine. Pour la distribution des valeurs des crêtes de gradient (approche par contraste), les deux rétines peuvent une fois de plus faire appel à leur sommateur analogique.

Aussi bien pour le nombre de points que pour l'estimation de la moyenne de chaque segment, une implémentation en temps constant est possible sur les deux rétines, de façon différente :

- pour la rétine synchrone, les valeurs doivent être agrégées au cours de l'extraction des segments, et les points ultimes contiennent finalement les quantités globales aux segments;
- pour la rétine asynchrone, les caractéristiques peuvent être regroupées de façon asynchrone par propagation.

Analyse de la saillance Pour le critère basé sur le nombre de points, il suffit de seuiller les points ultimes avec un seuil dépendant du nombre de points dans l'image de direction. Tout ceci est donc réalisable directement sur la rétine.

Pour le calcul du NFA en fonction du contraste, il est nécessaire de recourir au cortex. Cependant, un certain nombre d'optimisations sont envisageables pour élaguer au maximum les segments qui n'ont aucune chance d'être significatifs. Il est par exemple possible de calculer le nombre de points minimal pour être significatif dans le cas où la moyenne est inférieure à un certain seuil. Pour les candidats restants, il faut transmettre leurs valeurs caractéristiques au cortex, pour que celui-ci décide de leur signifiance et



FIG. 12: A gauche : image originale. Au milieu : image et segments significatifs par leur contraste. A droite : segments seuls.

renvoie les instructions à la rétine. Le coût de transfert de l'information resterait donc très réduit par rapport à une architecture classique, conformément à l'esprit des systèmes rétiniens.

A propos de la mémoire La dernière génération de rétine dispose de 50 bits de mémoire. Quoique préoccupé par un souci de limitation des besoins de stockage, notre travail n'a pas été jusqu'à s'assurer que 50 bits seraient effectivement suffisants. En effet, les images de directions prennent 8 bits, les calculs sur des entiers (gradients, nombre de points) également, il sera donc nécessaire d'optimiser l'utilisation des bits disponibles pour les calculs temporaires. Il est à noter que les calculs de segments peuvent être menés indépendamment direction par direction, limitant la consommation mémoire.

7 Discussion

Ce travail a plusieurs intérêts. Premièrement, il a montré que le couple système rétinien et critères statistiques sans paramètres peuvent cohabiter pour fournir un système efficace et robuste. L'application envisagée se voulait suffisamment simple pour être réalisable rapidement, mais suffisamment complexe pour faire ressortir un certain nombre de besoins pratiques :

- l'asynchronisme des rétines régionales est essentiel pour arriver à faire de la vision moyen niveau sur puce cellulaire;
- la rétine et le cortex doivent pouvoir échanger des informations plus riches qu'une quantité scalaire globale, notamment des quantités scalaires régionales.

En complément, le cadre de pensée rétinien a apporté un algorithme original de détection de segments, peu coûteux, et complètement parallélisable, propriété utile y compris sur des architectures classiques avec la tendance à la vectorisation des processeurs. Il apporte aussi sa volonté de réduction de l'information, d'une matrice de pixels nous avons abouti à un ensemble réduit (moins de 200 en moyenne sur une image 256x256) de structures perceptuellement saillantes.

Le modèle statistique *a contrario* formalisé par [Desolneux et al., 2003b] a quant à lui montré sa souplesse, en pouvant être appliqué à différentes hypothèses. Agissant en tant que filtre, il permet à la fois d'assigner une valeur de saillance à chaque objet et de déterminer un seuil ϵ qui soit lié à une grandeur physique, finalement fixé une fois pour toute à 1 en pratique. Le modèle ne se passe pas de tout *a priori* :

- la nature des candidats doit être définie a priori ;
- les critères à analyser sur les candidats également sont choisis par l'utilisateur.

Le principal intérêt réside dans la nature de l'a priori, il ne s'agit pas de quantités numériques, mais de critères justifiables par le raisonnement. Quant aux candidats, l'important est de générer un sur-ensemble des structures potentiellement intéressantes, puisque le filtrage permettra d'enlever celles qui ne sont pas significatives.

Une objection couramment faite à l'encontre de cette méthode statistique est la globalité du processus, en effet, les distributions de directions et de gradient sont déterminées globalement à toute l'image. Ceci a l'avantage de prendre en compte l'environnement, mais une image est souvent divisée en régions très différentes, qui mériteraient des traitements indépendants. Ainsi, une image composée de moitié de ciel bleu puis de champs donne un sens discutable à une distribution globale du gradient ("blue sky effect"). Idéalement, il faudrait donc appliquer la détection séparément sur les régions suffisamment différentes, en se basant sur une segmentation. La segmentation elle-même pourrait se baser sur le principe de Helmholtz pour décider de la saillance de chaque région.

8 Conclusion

Nous avons montré la pertinence de notre approche sur un opérateur générique simple. La nature des éléments à détecter a été choisie *a priori*. La justification formelle des informations à utiliser pour des opérateurs génériques reste un problème ouvert, incluant généralement le contraste, la régularité, voire des éléments issus de théories de la perception comme les Gestalts.

Cependant, au moins deux extensions paraissent naturelles : la généralisation à des courbes significatives, définies par une régularité dans les changements de directions et une analyse multi-échelle, permettant d'extraire des structures plus globales.

Une spécificité des systèmes rétiniens non exploitée dans ce travail est le rebouclage : les informations extraites de la rétine peuvent être utilisées par le cortex pour modifier en retour les traitements effectués

par la rétine, de la simple adaptation de contraste au changement radical des algorithmes. Diriger les opérateurs de base par les briques de plus haut niveau en temps réel est une possibilité qui devrait permettre aux applications futures de se passer de paramètres tout en optimisant chaque étape.

Références

- A. Almansa, A. Desolneux, and S. Vamech. Vanishing points detection without any a priori information. 25 (4) :502–507, april 2003.
- T.M. Bernard. *Des Rétines artificielles intelligentes*. PhD thesis, Université Paris-Sud, UFR Scientifique d'Orsay, Octobre 1992.
- J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.
- F. Cao. Application of the Gestalt principles to the detection of good continuations and corners in image level lines. *Computing and Visualisation in Science*, 7 :3–13, 2004.
- F. Cao, P. Musé, and F. Sur. Extracting meaningful curves from images. J. Math. Image and Vison, 2005.
- A. Desolneux, L. Moisan, and J.-M. Morel. Meaningful alignments. 40(1):7–23, 2000.
- A. Desolneux, L. Moisan, and J.-M. Morel. Edge detection by helmholtz principle. Journal of Mathematical Imaging and Vision, 14(3) :271-284, 2001. URL http://citeseer.ist.psu.edu/ desolneux00edge.html.
- A. Desolneux, L. Moisan, and J.-M. Morel. A grouping principle and four applications. 25(4) :508–513, 2003a.
- A. Desolneux, L. Moisan, and J.-M. Morel. Maximal meaningful events and applications to image analysis. *Annals of Statistics*, 31(6) :1822–1851, 2003b. submitted july 2000.
- A. Desolneux, L. Moisan, and J.-M. Morel. *Seeing, Thinking and Knowing*, chapter Gestalt Theory and Computer Vision, pages 71–101. A. Carsetti ed., Kluwer Academic Publishers, 2004.
- V. Gies and T.M. Bernard. Increasing interconnection network connectivity for reducing operator complexity in asynchronous vision systems. In *DGCI*, pages 1–10, 2005.
- H. Maître. Un panorama de la transformation de hough, 1985.
- A. Manzanera. Vision Artificielle Retinienne. PhD thesis, Ecole Nationale Superieure des Telecommunications, Novembre 2000.
- A. Manzanera and J. Richefeu. A robust and low cost change detection algorithm based on *σ*-δ background estimation. Technical report, ENSTA/LEI, 2004. URL http://www.ensta.fr/~manzaner/.
- D. Marr. Vision. W.H.Freeman and Company, 1982.
- P. Monasse and F. Guichard. Fast computation of a contrast invariant image representation. In *IEEE Transactions on Image Processing*, volume 9, pages 860–872, 2000.
- R. Nevatia and K. R. Babu. Linear feature extraction and description. Computer Vision Graphics and Image Processing, 13:257–269, 1980.
- F. Paillet. Intégration et évaluation de Rétines Artificielles Numériques Programmables de hautes performances. PhD thesis, Université Paris 6, Septembre 2001.
- F. Paillet, D. Mercier, and T.M. Bernard. Second generation programmable artificial retina. In *Proc. IEEE* ASIC/SOC Conf., pages 304–309, september 1999.
- M. Wertheimer. Untersuchungen zur lehre der gestalt, 1923.