Chapitre 5

Formes paramétrées

5.1 Descripteurs d'objet : du local au global

Ce chapitre traite de la modélisation d'objet dans les images, dans le but de permettre leur détection (Y a-t-il un objet dans l'image ?) et leur localisation (Où se trouve-t-il ?). Le modèle peut être produit *hors-ligne* à partir d'exemples, *via* une phase d'apprentissage, ou bien acquis *en ligne* au cours du traitement. L'apprentissage hors-ligne de modèle est en général employé pour la détection ou la reconnaissance d'instances d'objet générique (ex : piétons, voitures, visages,...). La difficulté principale est alors d'être suffisamment robuste à la variabilité plus ou moins grande qui peut affecter les objets, tout en restant assez discriminant pour ne pas créer trop de faux positifs. La modélisation en ligne est plutôt employée dans le cas de suivi ("tracking") d'objet dans une séquence vidéo, ou de ré-identication d'objets détectés précédemment. La difficulté principale est ici d'être assez flexible pour permettre au modèle de supporter les déformations inhérentes à l'objet (rotations 3d, changements d'échelles,...) ou dues à son environnement (changements d'éclairage, occultations,...), tout en distinguant l'objet de son arrière-plan.

L'approche adoptée dans ce chapitre, qui fonde la description globale sur des mesures statistiques (histogrammes, cooccurrence,...) de valeurs relatives à la géométrie locale, est dominante en modélisation d'objets, car elle possède une robustesse intrinsèque à l'occultation, un phénomène courant en analyse d'images. Dans les sections suivantes, on évoquera les descripteurs locaux avec les différents types d'invariance associés, puis on résumera les approches fondées sur l'analyse dite par «fenêtre glissante», qui consistent à appliquer une mesure statistique sur une portion de l'image correspondant à une hypothèse de localisation. Puis, dans la suite du chapitre, on développera les approches dite «par parties», fondées sur la transformée de Hough, qui modélise l'objet par la cooccurrence d'éléments locaux, et permettent de détecter le ou les objets dans l'image directement, sans faire d'hypothèses de localisation. Les transformées de Hough étant en outre adaptées aux formes analytiques (i.e. définies par une équation), on développera aussi la détection de droites et de cercles dans ce cadre.

5.1.1 Descripteurs locaux invariants

Nous avons déjà vu dans le chapitre 3 l'importance des dérivées partielles spatiales en tant que descripteurs de la géométrie locale. Certaines de ces dérivées, ou combinaisons de ces dérivées présentent de plus certaines propriétés d'invariance. On a déjà mentionné l'invariance par rotation du laplacien, mais cette propriété s'étend à toute une famille de mesures appelées les invariants différentielles. Si l'on suppose que ces invariants sont générées par des expressions polynomiales à partir des dérivées partielles jusqu'à un certain ordre, il est démontré [14] qu'on peut générer une famille finie $\mathcal{R}$ d'invariants générateurs, c'est-à-dire telle que tout polynôme de dérivées partielles invariants par rotation doit s'écrire aussi comme un polynôme d'éléments de $\mathcal{R}$. Ainsi à l'ordre 2, on

obtient l'ensemble de 5 descripteurs suivants :

$$\mathcal{R} = \begin{pmatrix} I \\ I_x^2 + I_y^2 \\ I_{xx}I_x^2 + 2I_xI_yI_{xy} + I_{yy}I_y^2 \\ I_{xx} + I_{yy} \\ I_{xx}^2 + 2I_{xy}^2 + I_{yy}^2 \end{pmatrix}. \tag{5.1}$$

Le lecteur attentif aura reconnu, dans l'ordre, le niveau de gris ($\mathcal{R}_0$), le carré du module du gradient ($\mathcal{R}_1 = I_g^2$), la dérivée seconde dans la direction du gradient multipliée par le carré du module du gradient ($\mathcal{R}_2 = I_{gg}I_g^2$), le laplacien($\mathcal{R}_3$), et le carré de la norme de Frobénius de la matrice hessienne ($\mathcal{R}_4$). La famille est génératrice dans le sens que tout autre polynôme de dérivées partielles à l'ordre 2 invariant par rotation s'exprime aussi comme un polynôme des éléments de $\mathcal{R}$. Par exemple :

$$\mathcal{K}_I = \mathcal{R}_1 \times \mathcal{R}_3 - \mathcal{R}_2 = I_{xx}I_y^2 - 2I_xI_yI_{xy} + I_{yy}I_x^2 \tag{5.2}$$

$$\mathcal{B}_I = \frac{1}{2}(\mathcal{R}_3^2 - \mathcal{R}_4) = I_{xx}I_{yy} - I_{xy}^2 \tag{5.3}$$

Avec $\mathcal{K}_I = I_{tt}I_g^2$, la dérivée seconde dans la direction de l'isophote multipliée par le carré du module du gradient, et $\mathcal{B}_I$ le déterminant de la matrice hessienne. Ces deux grandeurs, appelées respectivement *cornerness* et *blobness* par Lindeberg [27] sont utilisées comme détecteurs de points d'intérêt.

Pour obtenir l'invariance au zoom, ces descripteurs sont calculés à plusieurs échelles, de manière à obtenir un descripteur de taille $n_i \times n_s$, où n_i est le nombre d'invariants ($n_i = 5$ à l'ordre 2, $n_i = 9$ à l'ordre 3), et n_s est le nombre d'échelles utilisées. Schmid et Mohr [40] ont introduit l'utilisation de ces descripteurs locaux pour l'indexation d'images dans une application de recherche par le contenu.

Un autre type d'invariance couramment recherché en analyse d'image est l'invariance par changement de contraste, qui correspond aux combinaisons de dérivées partielles qui sont inchangées par l'application d'une fonction croissante f du niveau de gris sur l'image. Comme $\frac{\partial f(I)}{\partial u} = f' \frac{\partial I}{\partial u}$, cette propriété s'applique aux fonctions de la forme $g(\frac{P(I)}{Q(I)})$, où P et Q sont des polynômes de dérivées partielles de même degré. C'est ainsi le cas de toutes les mesures angulaires :

— argument du gradient
— direction de l'isophote
— arguments des vecteurs propres de la matrice hessienne

C'est aussi le cas pour la courbure de l'isophote $\kappa_I = -\frac{I_{tt}}{||\nabla I||}$ (voir section 5.2.3), qui présente ainsi la double propriété d'invariance par changement de contraste, et d'invariance par rotation (étant défini comme rapport de quantités invariantes par rotation).

5.1.2 Description par fenêtre glissante

L'approche la plus directe, et la plus couramment employée, pour construire un descripteur global d'objet à partir de descripteurs locaux consiste à réaliser des mesures intégrales sur des portions de l'image correspondant à des hypothèses de localisation. Le descripteur global ainsi produit est ensuite envoyé à

un algorithme de classification qui renvoie une étiquette définissant la classe de l'objet, qui peut être binaire (objet / fond) dans le cas d'une détection mono-classe.

La mesure produite est souvent une statistique du premier ordre, représentée par un histogramme de valeurs prises par les descripteurs locaux. L'utilisation des histogrammes d'orientation du gradient a été fortement popularisée par Dalal et Triggs dans leur descripteur HOG (Histogram of Oriented Gradients [10]), ainsi que par Lowe dans son descripteur SIFT (Scale Invariant Feature Transform [28]). Ces descripteurs sont essentiellement fondés sur :

1. une quantification assez forte des orientations du gradient (souvent 8 directions/sens, auxquelles s'ajoute une valeur spéciale correspondant à l'absence de structure lorsque le module du gradient est trop faible).
2. un découpage de l'espace correspondant à la localisation considérée, permettant de réaliser un compromis entre représentations globale et locale, et de modéliser des relations spatiales dans les objets (i.e. des statistiques d'ordre supérieur), que ne pourraient pas représenter un unique histogramme.
3. un histogramme des orientations du gradient calculé dans chaque portion de la fenêtre de localisation, l'effectif compté en chaque pixel étant souvent pondéré, d'une part par le module du gradient, et d'autre part en fonction de sa position plus ou moins centrale dans la fenêtre de calcul.
4. une concaténation de tous les histogrammes partiels pour former le vecteur descripteur global.

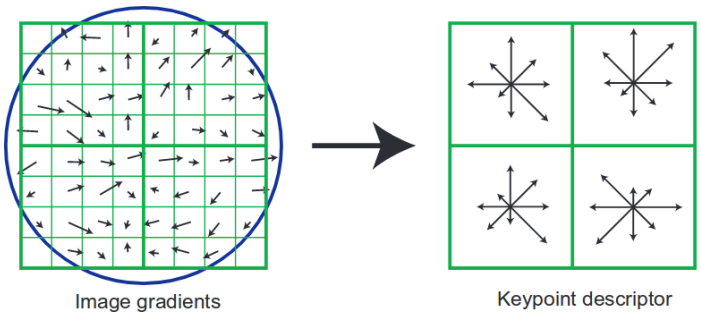


FIGURE 5.1 – Exemple de construction du descripteur SIFT avec un découpage en 4 régions, et 8 orientations du gradient. Tiré de [28].

La figure 5.1 montre un exemple de descripteur SIFT. Les histogrammes d'orientation sont réputés pour bien discriminer deux objets différents, mais il ne sont évidemment pas invariants par rotation. Cependant il est intéressant de noter que le descripteur SIFT est rendu invariant par rotation en considérant l'orientation de façon relative, c'est-à-dire qu'au lieu de choisir l'un des axes cartésien comme angle nul, il considère l'orientation locale au centre de la fenêtre (donnée par l'argument du gradient) comme origine.

Une autre technique qui entre dans le cadre des représentations à fenêtre glissante est la représentation par sac de mots visuels [8] ou BoVW pour Bag of Visual Words, qui consiste à :

— utiliser un algorithme de quantification vectorielle, ou de clustering, tel que les K-moyennes, pour construire un dictionnaire (codebook) d'éléments locaux [48] (Apprentissage non supervisé, voir Fig. 5.2(a)).
— construire un modèle des objets d'intérêts (classes) en recueillant des statistiques (histogrammes) de ces éléments locaux (les «mots» du dictionnaire visuel) à partir d'exemples (Apprentissage supervisé, voir Fig. 5.2(b)).
— représenter la portion d'image inconnue par un histogramme de mots visuels, qu'on soumet à classification (Détection, voir Fig. 5.2(c)).

Les représentations par BoVW ont eu beaucoup de succès pour la catégorisation d'images, la reconnaissance d'objets ou de lieux, grâce à leur simplicité, à leur flexibilité (elles peuvent être calculées sur des points d'intérêt, des contours, ou même de façon dense dans toute l'image), et à leur invariance géométrique (étant fondées sur des histogrammes d'éléments locaux, elles sont robustes à tout type de déformation). Mais c'est justement leur grand robustesse aux déformations qui constituent leur limite en termes de discrimination : elles ne tiennent aucun compte des relations spatiales entre les élements locaux. Les approches hiérarchiques telles que celle proposées dans [25] permettent d'obtenir un compromis entre discrimination et invariance en calculant plusieurs histogrammes dans des sous-régions de la fenêtre de localisation.

Les réseaux de neurones convolutionnels (CNN) pour la classification d'images peuvent également être utilisés en détection dans une approche par fenêtre glissante. Cette variété des réseaux de neurones multicouches où les mêmes neurones sont appliqués pour toutes les positions de la donnée d'entrée (image) réalisent donc des opérations de convolution dont les valeurs de noyaux sont apprises par entraînement sur des exemples. Ces réseaux ont été utilisés il y a plusieurs décennies pour l'analyse d'images, mais ce n'est que depuis les années 2010 que leur emploi s'est généralisé dans toutes les applications de la vision artificielle, à commencer par la classification d'images [37]. Cette explosion s'explique par une amélioration spectaculaire des résultats observés lorsque l'augmentation significative du nombre de couches de neurones (réseaux profonds, ou DNN) a été rendue possible, principalement grâce aux éléments suivants :

1. augmentation de la puissance de calcul et apparition du parallélisme massif dans les processeurs many-core (GP-GPU)
2. proposition de nouvelles techniques pour améliorer la rétropropagation du gradient et la convergence des réseaux (fonctions d'activation, taux d'apprentissage, normalisations, étapes de rétropropagation par «paquets», etc. voir [2])
3. existence de bases de données gigantesques pour l'apprentissage (internet, crowd sourcing,...)

Les réseaux de neurones convolutionnels qui ont permis une chute drastique du taux d'erreur en classification d'images [24] sont fondés sur une architecture similaire à celle de la figure 5.3. L'avantage déterminant de ces techniques est de supprimer le problème de la sélection de caractéristiques, qu'elles soient locales ou globales, puisqu'elles deviennent un sous-produit de l'apprentissage, qui peut d'ailleurs être récupéré : on peut par exemple brancher les couches basses d'un réseau appris pour une certaine application, sur de nouvelles couches hautes qu'on va entraîner pour une autre application. La contrepartie est une difficulté certaine pour comprendre et maîtriser l'apprentissage, difficulté liée à la variété des architectures et au nombre important d'hyper-paramètres [2]. La taille des

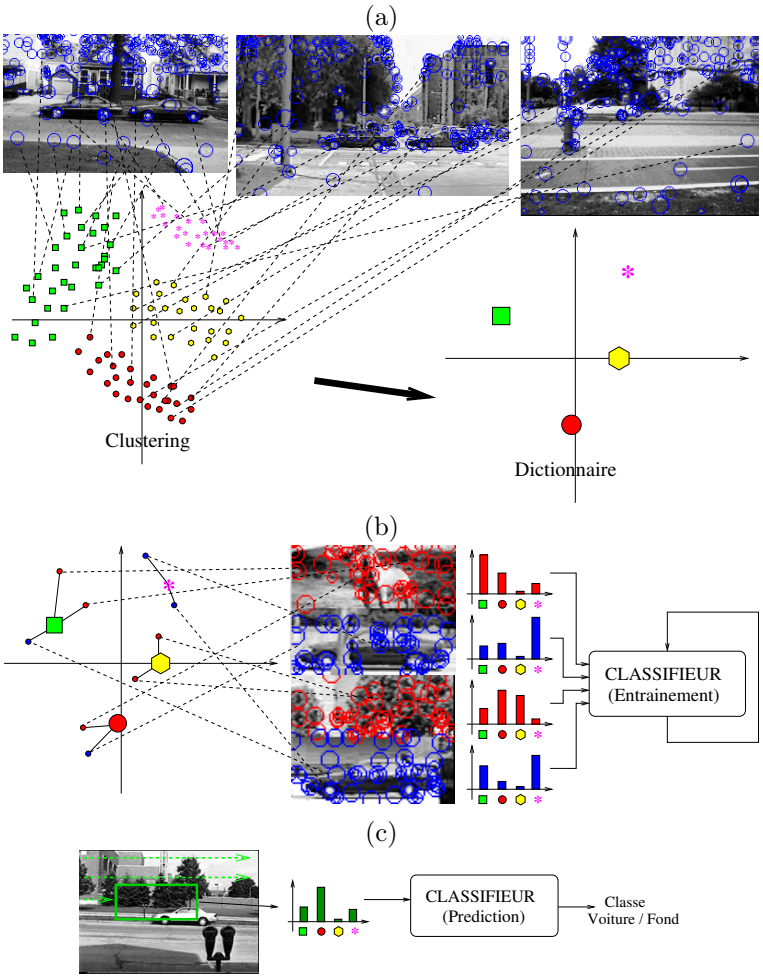


FIGURE 5.2 – Représentation d'objets par sacs de mots visuels (BoVW). (a) *Apprentissage non supervisé* : on extrait les descripteurs locaux autour d'un grand nombre de points (ici les points de type «blob» à plusieurs échelles), et on applique un algorithme de clustering pour construire un dictionnaire visuel. (b) *Apprentissage supervisé* : des histogrammes de mots visuels sont extraits d'exemples positifs (bleus) et négatifs (rouges), par extraction des descripteurs locaux et recherche de plus proche voisin dans le dictionnaire. Puis les descripteurs globaux ainsi formés sont utilisés pour entraîner un classifieur. (c) *Détection* : pour chaque hypothèse de localisation on calcule l'histogramme de mots visuels, qui est soumis au classifieur pour prédiction de la classe.

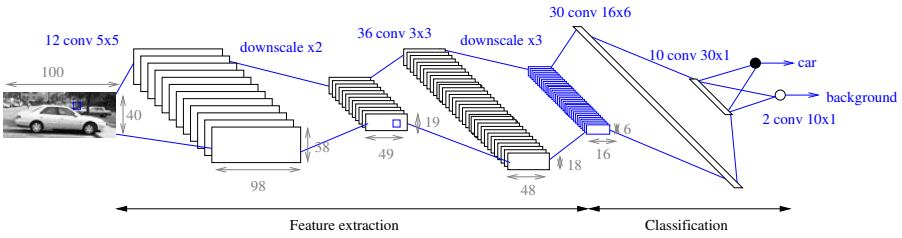


FIGURE 5.3 – Un exemple d'architecture CNN pour la classification d'images. La première couche apprend 12 caractéristiques *locales* sous la forme de 12 noyaux de convolution 5×5. Après sous-résolution ($\times 2$), la deuxième couche convolutionnelle apprend 36 caractéristiques *régionales* sous la forme de 36 noyaux de convolution $12 \times 3 \times 3$. Après une nouvelle sous-résolution($\times 3$), la troisième couche convolutionnelle apprend 30 caractéristiques *globales* sous la forme de 30 noyaux de convolution $36 \times 16 \times 3$. Les deux dernières couches sont totalement connectées, et apprennent respectivement 30×10 et 10×2 poids. Les valeurs de sortie dans les deux neurones doivent correspondre à une probabilité de classe «voiture» et de classe «fond» respectivement.

réseaux en nombre de connexions apprises génère aussi en général une grande complexité de l'algorithme de prédiction / détection résultant, mais qui est relativisée par le très fort niveau de parallélisme potentiel existant dans ces réseaux.

Une autre contrepartie des DNN est la nécessité de disposer d'un très grand nombre d'exemples d'apprentissage, ce qui est problématique pour de nombreuses applications. On notera néanmoins que, contrairement aux approches analytiques mentionnées précédemment pour produire des descripteurs invariants, les DNN permettent d'apprendre explicitement l'invariance désirée en appliquant aux données d'apprentissage toutes les transformations voulues : rotations, homothéties, changements de contraste, occultations, bruits... ce qui permet de multiplier les données exemples potentiellement à l'infini.

Les modèles neuronaux de classification sont limitées en termes de détection aux approches «fenêtre glissante» dans la mesure où elles nécessitent une ou plusieurs hypothèses de taille et d'aspect pour les données d'entrée, et fournissent un vecteur de probabilités de classes en sortie. Des modèles spécialisés dans la détection [38] permettent d'aller au-delà de cette limitation en combinant classification et régression de la position. C'est le cas par exemple des réseaux dits de «proposition de régions» (Region Proposal Networks) qui prennent en entrée une image de taille quelconque et fournissent en sortie des positions de fenêtres correspondant à des objets potentiels, sur lesquels une branche de classification est appliquée. Voir Figure 5.4.

5.2 Transformées de Hough analytiques

La suite de ce chapitre traite des représentations de formes par parties, c'est-à-dire des modèles associées aux transformées de Hough. Bien qu'associées à ce qui est peut-être l'algorithme de vision par ordinateur le plus ancien [18],

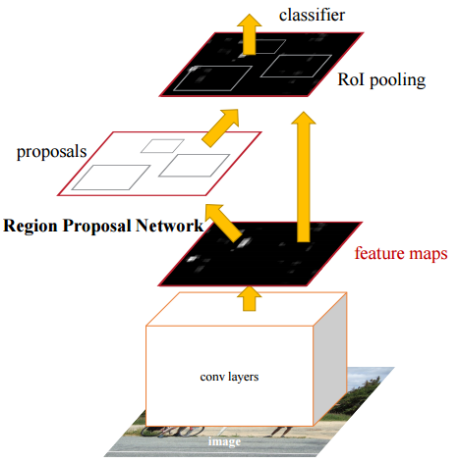


FIGURE 5.4 – Architecture du réseau *Faster R-CNN*, figure tirée de [38]. Noter que les réseaux de proposition de régions et de classification partagent les couches basses d'extraction de caractéristiques.

ces représentations ont suscité un intérêt quasiment constant depuis plus de 50 ans, comme l'atteste un article comparatif de 2015 [34], qui mentionne plus de 2 500 articles, dont environ une centaine postérieurs à l'année 2000. Cet intérêt s'explique par l'élégance et la polyvalence de l'approche, qui permet de détecter des formes analytiques comme des formes quelconques, et qui se prête à une très large variété d'optimisations et de variations.

Le principe général de la modélisation par Transformée de Hough (TH) consiste à représenter une forme par la cooccurrence l'éléments locaux (parties), qu'on va chercher à indexer efficacement en fonction d'une caractéristique liée à l'apparence locale. La TH elle-même consiste alors à projeter les données d'une image inconnue (pixels, contours, points particuliers,...) dans un espace de paramètres (ou espace de Hough), dont chaque point correspond à une position particulière de l'objet modélisée dans l'image, puis à rechercher les points d'accumulation dans l'espace de Hough, qui correspondent aux positions les plus probables des objets recherchés.

Par rapport aux approches précédemment évoquées dites par fenêtre glissante, l'avantage des représentations de Hough est de fusionner en quelque sorte localisation et classification, dans la mesure où elles ne nécessitent pas d'hypothèse de localisation, mais fournissent au contraire les localisations les plus probables. En revanche ce ne sont pas des méthodes de classification dans la mesure où elles ne fournissent pas directement de mesure d'appartenance (étiquette ou probabilité) à une classe donnée.

5.2.1 Principes généraux

Si l'on considère une image I multi-dimensionnelle comme un sous-ensemble de $\mathbb{R}^n$, on définit une forme analytique à partir de son équation paramétrique :

$$\mathcal{C}^{\mathbf{a_0}} = \{\mathbf{x} \in \mathbb{R}^n ; \phi(\mathbf{x}, \mathbf{a_0}) = 0\}, \tag{5.4}$$

où $\mathbf{x}$ est la variable spatiale, et $\mathbf{a_0} \in \mathbb{R}^m$ est une constante paramétrique. Si l'on considère maintenant un point particulier $\mathbf{x_0} \in \mathbb{R}^n$ de l'espace image, l'ensemble défini par :

$$\mathcal{D}^{\mathbf{x_0}} = \{\mathbf{a} \in \mathbb{R}^m; \phi(\mathbf{x_0}, \mathbf{a}) = 0\}, \tag{5.5}$$

où $\mathbf{a}$ est la variable paramétrique, est une surface dans l'espace paramétrique m-dimensionnelle, qui est la projection, ou forme duale du point $\mathbf{x_0}$. La somme de toutes les projections de I est appelée *la transformée de Hough* de I relativement à ϕ :

$$\Gamma_I^\phi = \sum_{\mathbf{x} \in I} \mathbb{1}_{\mathcal{D}^{\mathbf{x}}}, \tag{5.6}$$

avec $\mathbb{1}_A$ la fonction indicatrice de l'ensemble A. Les formes les plus représentatives dans l'image I sont finalement détectées en cherchant les maxima de Γ_I^ϕ. La figure 5.5 illustre les principes de dualité et de projection entre les espaces image et de Hough (de paramètres), sur l'exemple de la détection de droite, en paramétrisation polaire.

En pratique, dans les approches classiques, les espaces image $\mathcal{E}$ et paramètre $\mathcal{P}$ sont tous deux discrétisés, et la transformée de Hough (i.e. le résultat de la projection de tous les points de l'espace image dans l'espace des paramètres) est calculée à partir d'un nombre limité de points, les contours en général. De plus, la projection est habituellement réalisée selon l'une des deux techniques duales :

— La projection *one-to-many*, ou divergente (Fig. 5.6(a)).

— La projection *many-to-one*, ou convergente (Fig. 5.6(b)).

La projection one-to-many est définie comme plus haut par $\Gamma_I^\phi = \sum_{\mathbf{x} \in I} \mathbb{1}_{\mathcal{D}^{\mathbf{x}}}$, soit l'union (somme des fonctions indicatrices) de toutes les courbes duales associées aux points du contour. La projection many-to-one est définie par $\hat{\Gamma}_I^\phi = \sum_{S \subset I, |S| = \dim(\mathcal{P})} \mathbb{1}_{\{\mathbf{a}_S\}}$, où $\{\mathbf{a}_S\} = \bigcap_{\mathbf{x} \in S} \mathcal{D}^{\mathbf{x}}$, est un point unique de $\mathcal{P}$, qui représente la seule courbe de $\mathcal{E}$ contenant tous les points de S. Dans les sections suivantes, nous développons une approche différente pour les transformées de Hough analytiques, qui est la projection one-to-one, qui permet, grâce à l'estimation des dérivées partielles, (1) de projeter chaque point de l'espace image en un unique point de l'espace des paramètres, et (2) de calculer la TH de façon *dense* pour chaque pixel, sans avoir à extraire les contours préalablement.

5.2.2 Ordre 1 : droites

Pour la détection de droites, on utilise l'équation polaire plutôt que cartésienne [13], pour des raisons d'uniformité de représentation des droites dans l'espace des paramètres discrétisés (dans une équation cartésienne du type $y = ax + b$, la moitié des droites de l'espace aura son paramètre a dans l'intervalle $[-1, +1]$, et l'autre moitié dans l'union d'intervalles $]-\infty, -1] \cup [+1, +\infty[$). Donc si la variable d'espace est $\mathbf{x} = (x, y)$ et la variable paramétrique est $\mathbf{a} = (\theta, \rho)$, l'équation paramétrique de la droite $\mathcal{C}^{(\theta,\rho)}$ est :

$$x \cos\theta + y \sin\theta = \rho. \tag{5.7}$$

La courbe duale de la droite est donc la courbe sinusoïdale $\mathcal{D}^{(x,y)}$, qui a la même équation, mais avec les rôles de (x, y) et de (θ, ρ) inversés. De façon classique,

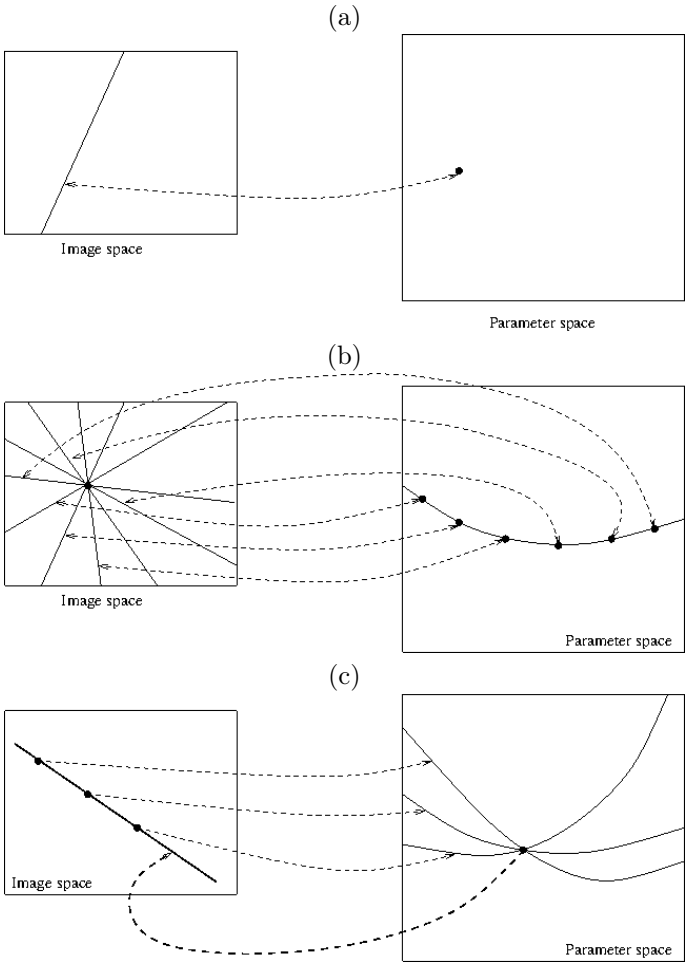


FIGURE 5.5 – Principe de la transformée de Hough : (a) Chaque point de l'espace des paramètres (ici un couple de coordonnées polaires (θ, ρ)) correspond à une unique forme dans l'espace image (ici une droite). (b) Chaque courbe de l'espace des paramètres (ici une sinusoïde) correspond à un unique point ou, de manière équivalente, à un faisceau de formes (ici des droites) dans l'espace image. (c) Réciproquement, différents points appartenant à la même forme dans l'espace image forment un faisceau de courbes dans l'espace des paramètres, qui converge vers l'unique point qui définit la forme correspondante.

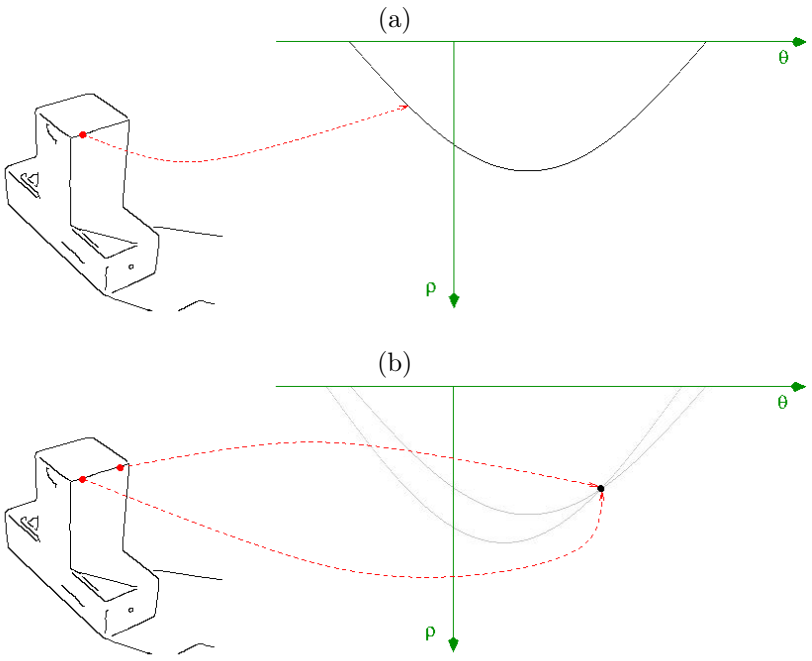


FIGURE 5.6 – Approches classiques des transformées de Hough analytiques calculées sur les contours : (a) Projection one-to-many, (b) Projection many-to-one.

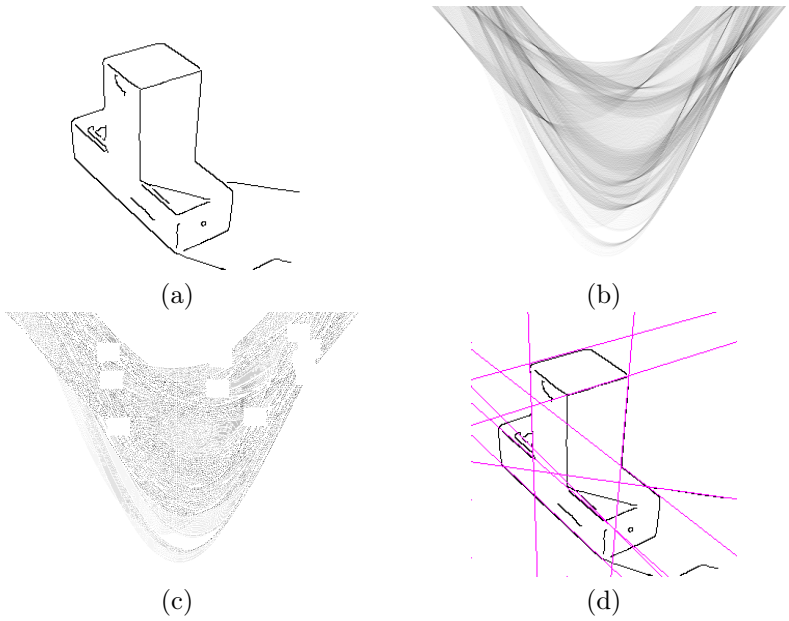


FIGURE 5.7 – Transformée de Hough classique one-to-many. (a) Image de contours. (b) Transformée de Hough one-to-many. (c) Extraction des 10 plus grands maxima locaux de la TH. (d) Rétroprojection des 10 meilleurs maxima dans l'image originale.

on réduit donc l'ensemble des points «votants» aux points jugés significatifs (les contours), et on initialise la transformée de Hough à un tableau identiquement nul. Puis, pour chaque point de contours (méthode one-to-many), ou pour chaque couple de points du contours (méthode many-to-one), on incrémente dans la TH tous les points de l'espace des paramètres formant la sinusoïde correspondant au point votant (méthode one-to-many) ou bien l'unique point correspondant à la droite passant par les deux points de contours (méthode many-to-one). Une fois l'ensemble des points (ou des couples) de l'image de contours projetés, les meilleures formes candidates sont détectées en calculant les maxima de la transformée de Hough. Voir Figure 5.7 pour une illustration de l'approche one-to-many.

Mais comme nous l'avons vu au chapitre 3, les contours se calculent à partir des dérivées partielles, en particulier le vecteur gradient. Or, puisque la direction orthogonale au gradient est celle de l'isophote, en tout point $\mathbf{x}$ de l'image tel que $||\nabla I(\mathbf{x})|| \neq 0$, on peut déduire directement l'équation de la droite qui passe par $\mathbf{x}$ des coordonnées du vecteur gradient (voir Figure 5.8). Le principe de la Transformée de Hough Dense (THD) one-to-one est de généraliser ce principe à tous les pixels par un processus de vote. Plus précisément pour les droites, en utilisant la paramétrisation polaire (θ, ρ), où ρ est la distance entre la droite et l'origine, et θ est l'angle entre la normale et l'axe des abscisses, s'il y a une

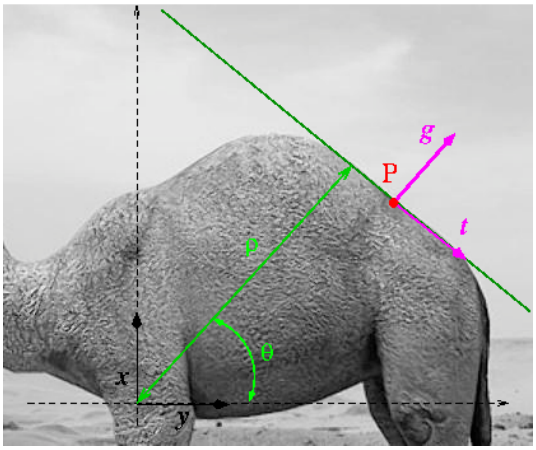


FIGURE 5.8 – Gradient, isophote et droite passant par un point P dans une image en niveaux de gris.

droite passant par le point $\mathbf{x}$, on doit avoir :

$$\theta_{\mathbf{x}} = \arg \nabla I, \tag{5.8}$$

$$\rho_{\mathbf{x}} = \frac{|\mathbf{x}.\nabla I|}{||\nabla I||}. \tag{5.9}$$

c'est-à-dire que $\theta_{\mathbf{x}}$ correspond à la direction du vecteur gradient, et $\rho_{\mathbf{x}}$ à la distance entre l'origine et la droite passant par $\mathbf{x}$ et perpendiculaire au vecteur gradient. Pour évaluer l'importance du point $\mathbf{x}$ vis-à-vis de la présence d'une droite, il est naturel d'utiliser l'intensité de la dérivée première, c'est-à-dire le module du gradient $||\nabla I|| = \sqrt{I_x^2 + I_y^2}$ (Voir Figure 5.9(b) et (c)).

La table 5.1 montre l'algorithme complet de calcul de la THD 1-to-1 pour la détection de droites, directement déduit des équations précédentes. La figure 5.10 montre un exemple de résultat de la THD comparé avec une version classique de la TH 1-to-many. Noter que bien que calculée de façon dense sur tous les pixels, la transformée de Hough Dense (en bas) est en fait plus éparse que la TH 1-to-many, car le nombre de votes au final est plus faible. Mais les points d'accumulation sont clairement visible, et d'autre part la THD peut être rendue plus lisse par (1) l'utilisation de plusieurs échelles d'estimation (voir Table 5.1), et (2) l'interpolation du vote sur des cellules voisines, voir [31].

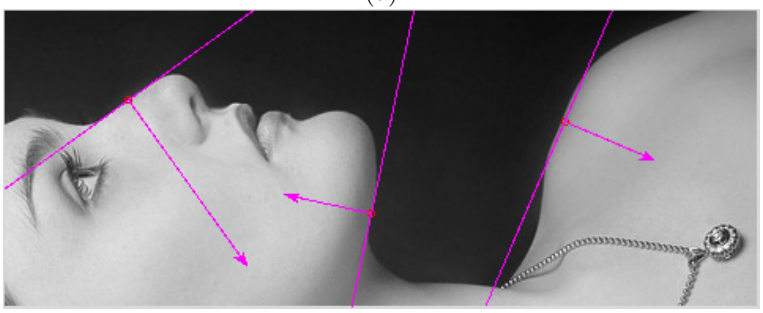
5.2.3 Ordre 2 : cercles

Pour la détection des cercles, la variable paramétrique est [13] $\mathbf{a} = (\mathbf{c}_x, \mathbf{c}_y, r)$ et l'équation du cercle $\mathcal{C}^{\mathbf{a}}$ est :

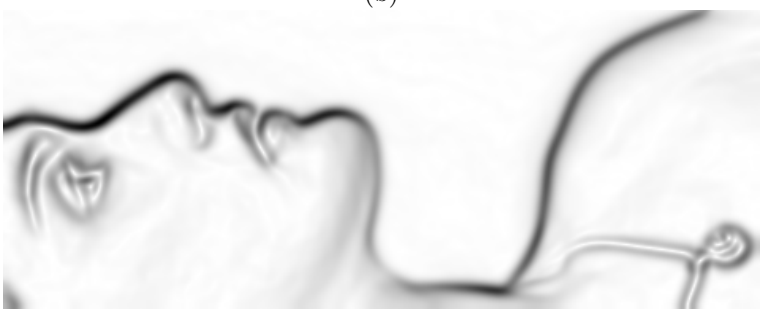
$$(x - \mathbf{c}_x)^2 + (y - \mathbf{c}_y)^2 = r^2 \tag{5.10}$$

La forme duale du cercle est donc $\mathcal{D}^{(x,y)}$, la surface d'un cône. L'espace de Hough étant de dimension 3, la TH 1-to-many est encore plus coûteuse à calculer que pour les droites. Les algorithmes les plus utilisés dans la littérature sont

(a)



(b)



(c)

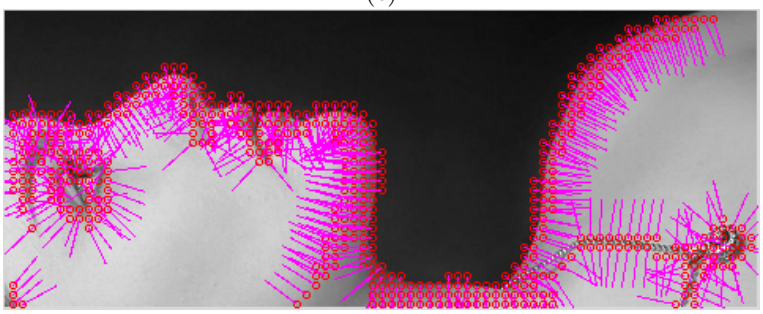


FIGURE 5.9 – Transformées de Hough Denses pour la détection des droites : l'échelle d'estimation utilisée ici est $\sigma = 3,0$. (a) La droite est estimée en trois points particuliers (représentés par les cercles rouges) à partir de leur vecteur gradient (représenté ici perpendiculaire à la droite). (b) La norme du gradient est utilisée pour pondérer les votes. (c) Vote dense pour la direction du gradient, représentée ici seulement pour des points dont la norme du gradient est supérieure à 2,5.

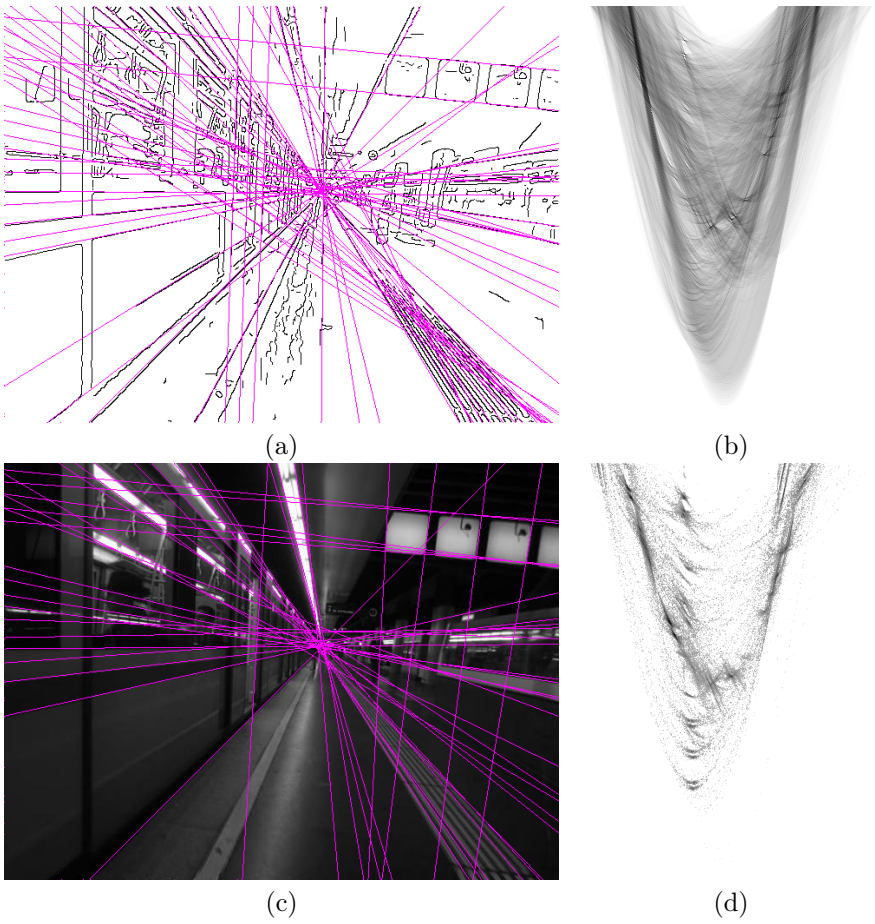


FIGURE 5.10 – Comparaison pour la détection de droites entre Transformée de Hough classique 1-to-many (b) calculée sur les contours (a) et la Transformée de Hough Dense 1-to-1 (d) calculée sur le niveau de gris (c).

TABLE 5.1 – DHT 1-to-1 multi-échelles pour la détection de droites

```
Γ = function Hough_Lines (Image I)
   forall scale σ ∈ {σ_1, ..., σ_n}
      forall pixel p = (p_x, p_y)
         ∇I ← (I_x^σ(p), I_y^σ(p))
         if ||∇I|| > 0:
            d ← p_x I_x + p_y I_y
            ρ ← |d| / ||∇I||
            θ ← arctan(I_y / I_x)
            Γ(ρ, θ) ← Γ(ρ, θ) + σ||∇I||
         endif
      endfor
   endfor
end
```

plutôt des approches many-to-1, ou plus exactement des versions approchées dites randomisées [51], qui tirent au hasard des triplets de points à partir des pixels de contours, et font voter chaque triplet dans l'espace de Hough pour l'unique point correspondant au cercle passant par les trois points. Il faut aussi mentionner les approches en 2 passes telles que [52], qui calculent d'abord une première TH 1-to-many 2d en réduisant les paramètres au centre du cercle, puis font voter les pixels autour des meilleurs centres pour trouver les meilleurs rayons.

Il faut cependant noter que, comme pour les droites, la connaissance des dérivées partielles doit permettre de connaître directement l'équation d'un cercle passant par un point donnée. En effet, ce cercle est directement lié à la courbure de l'isophote, mentionnée dans le chapitre 3. Nous allons d'abord établir la relation précise qu'il existe entre les deux.

On sait qu'en chaque point $\mathbf{x}$ tel que $||\nabla I(\mathbf{x})|| > 0$, les vecteurs gradient et isophote forment un repère local $(\mathbf{x}, \mathbf{g}, \mathbf{t})$ qui fournit un système de coordonnées locales au premier ordre (Voir Fig. 5.11). La courbe isophote qui passe par $\mathbf{x}$ peut être paramétrisée par la coordonnée curviligne s, définie par :

$$I(\mathbf{g}(s), \mathbf{t}(s)) = \text{Cte} = I(\mathbf{x}) \tag{5.11}$$

Si l'on considère le déplacement d'un point le long de cette courbe isophote, où la coordonnée curviligne s est assimilée au temps, le repère local $(\mathbf{x}, \mathbf{g}, \mathbf{t})$ correspond au repère de Frenet de la courbe isophote. Dans ce cadre, la courbure de l'isophote peut être définie par l'accélération radiale $\ddot{\mathbf{g}}(s)$, lorsque le déplacement se fait à vitesse unitaire, c'est-à-dire $\dot{\mathbf{t}}(s) = 1$ (On utilise ici les notations cinématiques $\dot{\mathbf{u}} = \frac{\partial \mathbf{u}}{\partial s}$ et $\ddot{\mathbf{u}} = \frac{\partial^2 \mathbf{u}}{\partial s^2}$).

Si l'on dérive l'équation 5.11 par rapport à s, on obtient :

$$\dot{\mathbf{g}} I_g + \dot{\mathbf{t}} I_t = 0. \tag{5.12}$$

Comme $I_t = 0$, si $I_g \neq 0$, on aura $\dot{\mathbf{g}} = 0$. En dérivant encore l'équation 5.12

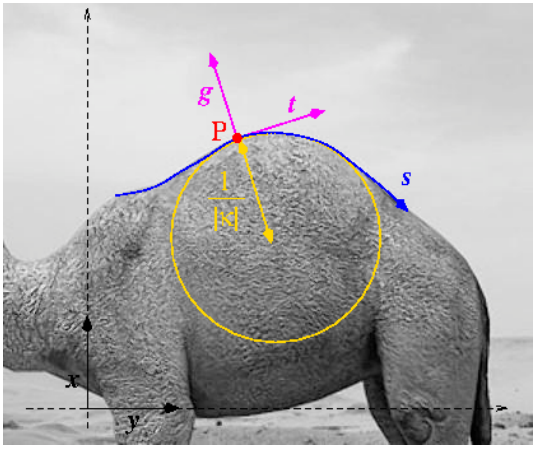


FIGURE 5.11 – Gradient, isophote et courbure estimés au point P dans une image en niveaux de gris.

par rapport à s, on a :

$$\dot{\mathbf{g}}^2 I_{gg} + \ddot{\mathbf{g}} I_g + 2\dot{\mathbf{g}}\dot{\mathbf{t}} I_{gt} + \dot{\mathbf{t}}^2 I_{tt} + \ddot{\mathbf{t}} I_t = 0. \tag{5.13}$$

Finalement, puisque $I_t = 0$, $\dot{\mathbf{t}} = 1$, et $\dot{\mathbf{g}} = 0$ (si $I_g \neq 0$), on obtient :

$$\ddot{\mathbf{g}} = -\frac{I_{tt}}{I_g}. \tag{5.14}$$

Et en reprenant l'expression cartésienne de ces dérivées dans le repère local (voir Chap. 3), la courbure de l'isophote κ se calcule donc comme suit (voir aussi [45]) :

$$\kappa = \ddot{\mathbf{g}} = -\frac{I_{xx} I_y^2 - 2 I_{xy} I_x I_y + I_{yy} I_x^2}{||\nabla I||^3} \tag{5.15}$$

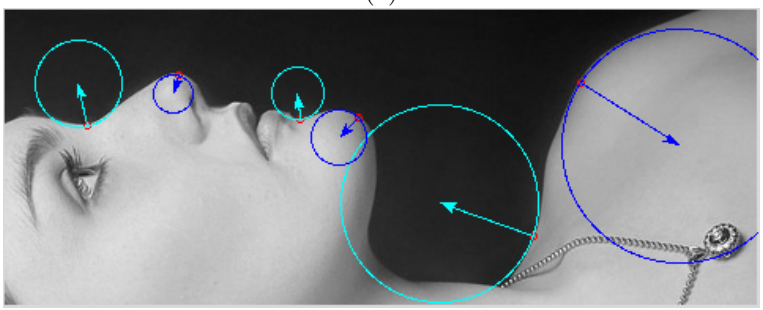
La valeur absolue de la courbure de l'isophote vaut l'inverse du rayon du cercle osculateur à la courbe isophote (Voir Fig. 5.11), et son signe fournit la polarité de la courbure (positive : plus clair à l'intérieur). Ainsi pour les cercles, en utilisant la paramétrisation (C, r), où $C \in \mathbb{R}^2$ est le centre et r le rayon du cercle, s'il existe un cercle passant par le point $\mathbf{x}$ on doit avoir :

$$r_{\mathbf{x}} = \frac{1}{|\kappa_{\mathbf{x}}|}, \tag{5.16}$$

$$\overrightarrow{\mathbf{x}C_{\mathbf{x}}} = \frac{\nabla I}{\kappa_{\mathbf{x}} ||\nabla I||}. \tag{5.17}$$

c'est-à-dire que le rayon $r_{\mathbf{x}}$ est l'inverse de la courbure absolue $\kappa_{\mathbf{x}}$ calculée au point $\mathbf{x}$ en utilisant l'équation 5.15, et le centre $C_{\mathbf{x}}$ est obtenu en traçant depuis $\mathbf{x}$ le vecteur ayant pour norme le rayon, pour direction celle du gradient et pour sens le signe de la courbure. Ici encore on peut évaluer l'importance du point $\mathbf{x}$ vis-à-vis de la présence d'un cercle par l'intensité de la dérivée seconde, c'est-à-dire la norme de Frobenius de la matrice hessienne $||H_I||_F = \sqrt{I_{xx}^2 + 2I_{xy}^2 + I_{yy}^2}$.

(a)



(b)

(c)

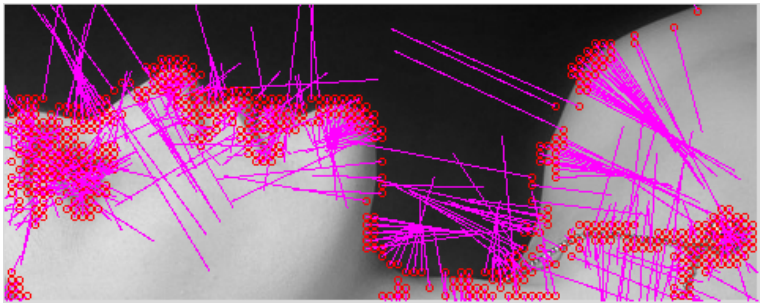


FIGURE 5.12 – Transformées de Hough Denses pour la détection des cercles : l'échelle d'estimation utilisée ici est $\sigma = 3,0$. (a) Le cercle est estimée en quelques points particuliers (représentés par les cercles rouges) à partir du vecteur gradient et de la courbure de l'isophote. Ici les segments avec une flèche ont la direction du vecteur gradient, leur norme vaut l'inverse de la courbure absolue de l'isophote, et leur sens est le même que celui du vecteur gradient lorsque la courbure est positive (cercles bleus) et le sens opposé lorsque la courbure est négative (cercles cyan). (b) La norme de Frobenius de la matrice hessienne est utilisée pour pondérer les votes. (c) Vote dense pour la position du centre de cercle osculateur, représenté ici seulement pour des points dont la norme de Frobenius de la matrice hessienne est supérieure à 0,5.

Le principe de la THD au deuxième ordre est illustré sur la Figure 5.12. La table 5.2 montre l'algorithme complet de calcul de la THD 1-to-1 pour la détection de cercles. La figure 5.13 compare les sorties de la TH 1-to-many et de la THD 1-to-1, ainsi que leur version en 2 passes, qui sont de fait plus souvent utilisées en raison de leur meilleure précision.

TABLE 5.2 – DHT 1-to-1 multi-échelles pour la détection de cercles.

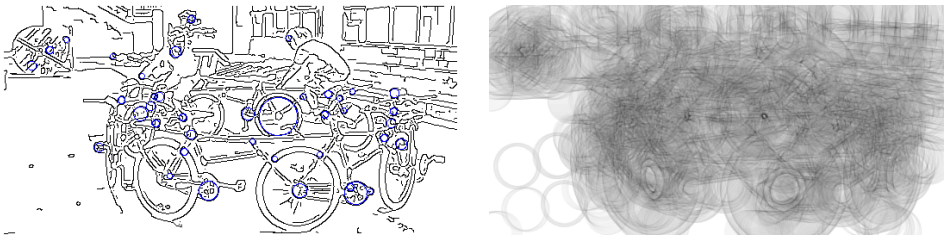
Γ = function Hough_Circles_V1 (Image I)
  forall scale $\sigma \in \{\sigma_1, \ldots, \sigma_n\}$
    forall pixel $\mathbf{p} = (\mathbf{p}_x, \mathbf{p}_y)$
      $\nabla I \leftarrow (I_x^\sigma(\mathbf{p}), I_y^\sigma(\mathbf{p}))$
      $H_I = \begin{pmatrix} I_{xx}^\sigma(\mathbf{p}) & I_{xy}^\sigma(\mathbf{p}) \\ I_{xy}^\sigma(\mathbf{p}) & I_{yy}^\sigma(\mathbf{p}) \end{pmatrix}$
      if $||H_I||_F > 0$:
        $\kappa \leftarrow I_{xx} I_y^2 - 2 I_{xy} I_x I_y + I_{yy} I_x^2$
        $\rho \leftarrow \frac{||\nabla I||^3}{\kappa}$
        $(\mathbf{c}_x, \mathbf{c}_y) = (\mathbf{p}_x, \mathbf{p}_y) - \frac{\nabla I ||\nabla I||^2}{\kappa}$
        $\Gamma(\mathbf{c}_x, \mathbf{c}_y, \rho) \leftarrow \Gamma(\mathbf{c}_x, \mathbf{c}_y, \rho) + \sigma^2 ||H_I||_F$
      endif
    endfor
  endfor
end

5.3 Transformées de Hough généralisées

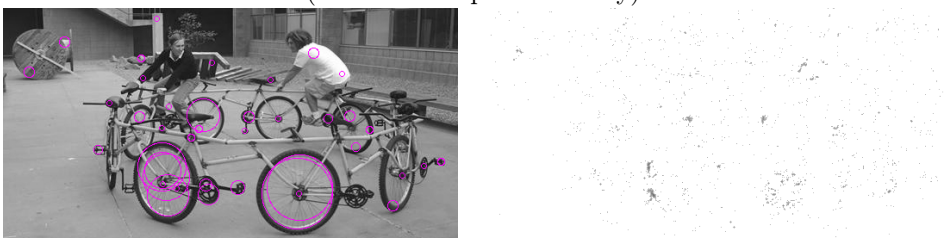
L'avantage principal de la modélisation de Hough est de se prêter à tout type de paramétrisation, y compris celle qu'on peut construire pour une forme quelconque, ce qui permet de représenter un objet réel, dans le but de détecter des instances de cet objet dans des images. C'est l'objet de la transformée de Hough généralisée. On présente dans cette partie d'abord les premières représentations issues de la R-table, qui permettent de représenter de façon plus ou moins flexible un objet particulier, puis on aborde des techniques fondées sur les forêts aléatoires, qui vise à représenter une catégorie beaucoup plus large d'objets.

5.3.1 R-Tables et modèles implicites de forme

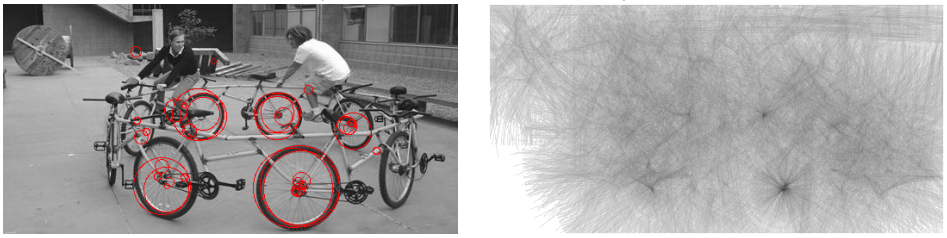
Les premières généralisations des modèles de Hough sont dues à Ballard [1] et sont fondées sur une paramétrisation implicite d'une courbe quelconque, où chaque point de la courbe est représenté par un index i relatif à son apparence (typiquement, l'orientation locale de la courbe). Plusieurs points de la courbe de référence (prototype) partagent le même index i. Le principe de représentation appelé R-table, consiste à choisir un point de référence (disons le centre) dans l'objet, et à enregistrer, sous la forme d'un tableau T, pour chaque $T(i)$



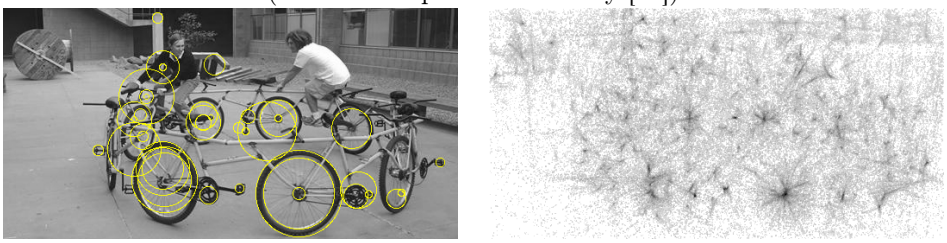
(1 : TH classique 1-to-many)



(2 : THD 1-to-1, Table 5.2)



(3 : TH en 2 passes 1-to-many [52])



(1 : THD en 2 passes 1-to-1 [31])

FIGURE 5.13 – Comparaison de différentes Transformées de Hough pour la détection de cercles. Les 40 meilleurs cercles sont affichés. Les algorithmes (1) et (3) opèrent sur les contours, et les algorithmes (2) et (4) sur le niveau de gris. Pour (1) et pour (2), l'espace de Hough est en 3d, on montre ici seulement le plan d'équation $r = 25$. Pour (3) et pour (4) on montre la transformée 2d correspondant à la sortie de la première passe (localisation du centre).

l'ensemble des vecteurs correspondant aux positions relatives de tous les points de la courbe ayant l'index d'apparence i, par rapport au centre de l'objet.

La figure 5.14 illustre ce principe sur un prototype de cheval. Les trois index représentés ici correspondent à des points ayant des apparences similaires, mais des positions différentes. La R-table représentée à droite code donc la cooccurrence des 3 éléments locaux identifiés dans le prototype, sous la forme d'un tableau de listes $T(i) = \left\{Tv_j^i\right\}_{0 \leq j < N_i}$, où N_i est le nombre de déplacements (vecteurs) associés à l'index d'apparence i.

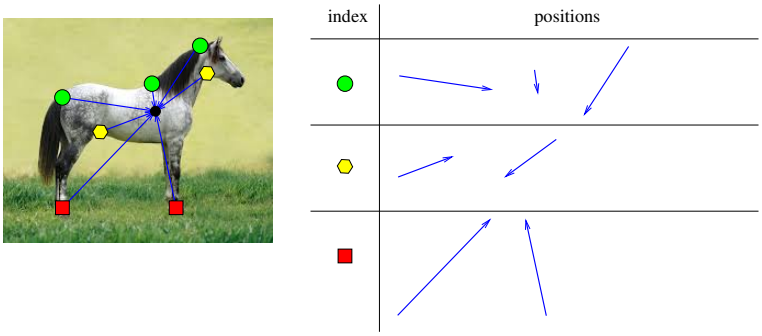


FIGURE 5.14 – Construction de la R-table à partir d'un prototype.

La Tranformée de Hough Généralisée (THG), qui correspond à la phase de détection sur une image inconnue I, consiste ensuite à calculer, pour tous les pixels $\mathbf{x}$ (ou pour un sous-ensemble de pixels), l'index d'apparence associé à chaque pixel : $i(I, \mathbf{x})$, et à faire voter ce pixel pour tous les centres possibles d'objet qui le contiendrait, c'est-à-dire à toutes les positions $\left\{\mathbf{x} + Tv_j^{i(I,\mathbf{x})}\right\}_{0 \leq j < N_{i(I,\mathbf{x})}}$. Formellement, la THG H est donc définie comme suit :

$$H(\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{D}} \sum_{j=0}^{N_{i(I,\mathbf{y})}} \delta(\mathbf{x}, \mathbf{y} + Tv_j^{i(I,\mathbf{y})}). \tag{5.18}$$

où $\mathcal{D}$ est l'ensemble des pixels votants, et $\delta(\mathbf{x}, \mathbf{y})$ est la fonction d'identité de Kronecker.

L'ensemble des pixels votants $\mathcal{D}$, ainsi que la fonction d'apparence $i(I, \mathbf{x})$ peuvent s'adapter au contexte et à la puissance de calcul : chez Ballard [1], $\mathcal{D}$ était l'ensemble des contours, et i l'orientation local du contour, tandis que dans des travaux plus récents, chez Leibe et al [26], $\mathcal{D}$ est un ensemble de points d'intérêts, tandis que i correspond au mot du dictionnaire visuel associé à une approche sac-de-mots (voir Section 5.1.2).

La THG peut aussi s'appliquer à une approche dense multi-échelles telle que celle présentée dans les sections précédentes pour les formes analytiques. Dans ce cas, le calcul se fait sur tous les pixels, en utilisant un index correspondant à une dérivée partielle ou à une combinaison invariante de ces dérivées (ex : l'argument du gradient, invariant au changement de contraste). Dans ce cas il peut être utile, comme pour les formes analytiques, de pondérer les votes par une mesure évaluant la pertinence de l'index utilisé pour ce pixel (ex : le module du gradient). La figure 5.15 montre un exemple d'application de la THG dense.

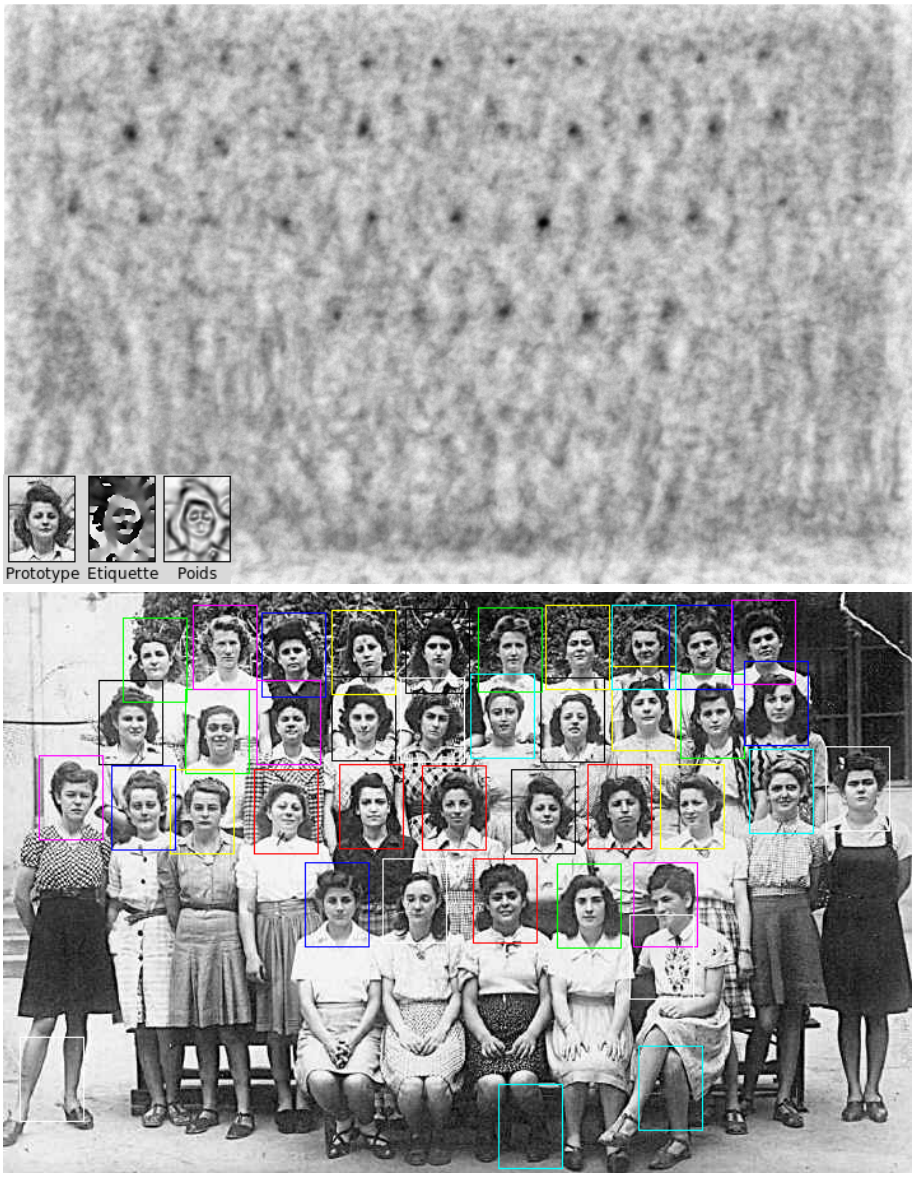


FIGURE 5.15 – Transformée de Hough généralisée dense. En haut, la THG dense, avec le prototype utilisé en incrustation : l'index (étiquette) correspond à l'argument du gradient, le poids au module du gradient. En bas, les 40 meilleures détections correspondant aux 40 principaux maxima locaux de la THG. La couleur du rectangle correspond aux rangs des détections : noire pour les 5 meilleures, rouge pour les 5 suivantes, puis verte, jaune, bleue, magenta, cyan et enfin blanche pour les 5 dernières.

5.3.2 Forêts de Hough

Les représentations d'objet fondées sur la R-Table présentées dans la section précédente sont plutôt adaptées pour les modélisations *hors-ligne* d'objets précis, dans la mesure où elle n'utilise pas de phase d'apprentissage supervisé (elle peuvent néanmoins se fonder sur un apprentissage non supervisé pour la construction du dictionnaire d'index). Elles sont donc pertinentes pour le suivi ou la ré-identification d'un objet connu, mais beaucoup moins pour la détection d'une catégorie d'objets, car la R-Table, qui contient toutes les occurrences d'éléments locaux rencontrées dans le (ou les) prototype(s) ne permet pas de représenter une variabilité géométrique raisonnable sans occuper une place mémoire déraisonnable.

Pour remédier à cela, Gall et al [15] ont proposé une représentation dite par *Forêt de Hough*, qui combine les R-Tables et les forêts aléatoires. La classification par une forêt aléatoire [4] consiste à soumettre un descripteur $\mathbf{u}$ de classe inconnue à un ensemble de fonctions correspondant aux sommets non terminaux (les nœuds) des arbres, en commençant par le nœud racine du premier arbre. La fonction associée au nœud prend autant de valeurs différentes que le nœud possède de sommets fils. La valeur de la fonction détermine ainsi la prochaine fonction qui sera appliquée au descripteur $\mathbf{u}$, et ainsi de suite jusqu'à ce $\mathbf{u}$ parvienne à un sommet terminal (une feuille), qui elle, est associée à une étiquette de classe c_1. Le même parcours est réalisé sur le deuxième arbre pour obtenir une classe c_2, etc. La décision finale est ensuite établie par consensus sur l'ensemble des N classes $\{c_k\}$ obtenues, par exemple un vote majoritaire.

Dans une forêt de Hough [15], les arbres sont binaires : chaque nœud est associé à un prédicat booléen lié à l'apparence locale, par exemple $f_I(\mathbf{x}) < f_I(\mathbf{y}) + \delta$, où f_I est une caractéristique scalaire extraite de l'image I. De plus les feuilles ne sont pas associées à une classe unique, mais sont formées d'un ensemble d'éléments locaux (ici des imagettes) qui vont être utilisés pour la réalisation du vote dans la THG.

Une forêt de Hough est construite lors d'une phase d'apprentissage, à partir d'un très grand nombre d'imagettes, sur lesquelles on calcule un certain nombres de caractéristiques scalaires. Les imagettes «positives», c'est-à-dire celles qui viennent d'un exemple d'objet de la classe recherchée, sont associé à une position relative par rapport au centre de l'objet, comme on avait pour la R-Table. L'objectif de l'apprentissage est d'indexer les imagettes de la base d'apprentissage en regroupant les éléments similaires, en termes d'apparence, de classe, et de position relative pour les imagettes positives. Idéalement ces groupes doivent correspondre aux feuilles des arbres aléatoires, sachant que chaque nœud d'un arbre correspond à un sous-ensemble d'imagettes de la base. Au début de la création de chaque arbre, on tire au hasard un (grand) ensemble d'imagettes de la base pour former la racine de l'arbre. Puis on crée un premier prédicat qui va diviser l'ensemble en deux groupes, puis on recommence à diviser récursivement chaque sous-ensemble jusqu'à l'obtention d'un critère d'arrêt (profondeur maximum ou nombre d'imagettes minimum). La figure 5.16 montre un exemple d'arbre de Hough.

Au cours de l'apprentissage, chaque nouveau nœud est formé en choisissant aléatoirement un prédicat d'apparence, de façon à optimiser deux critères :

1. l'entropie de classes, pour séparer au maximum les classes représentées dans chaque feuille.

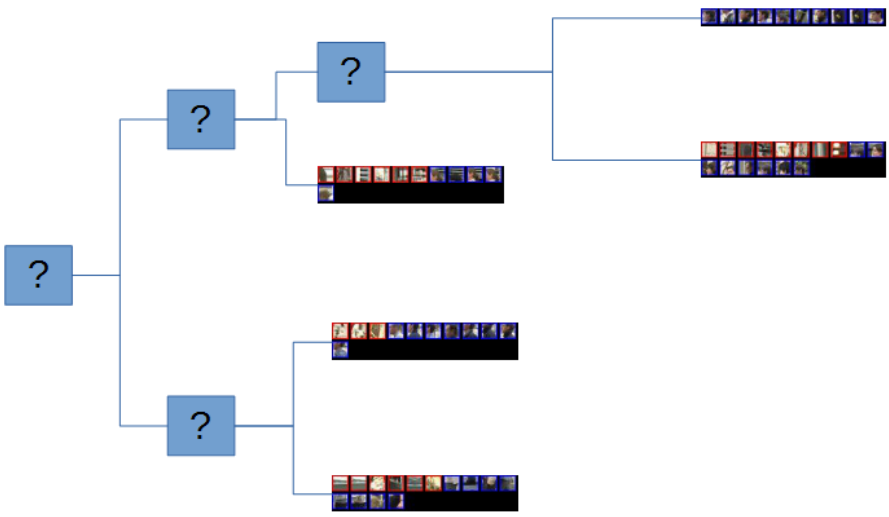


FIGURE 5.16 – Un exemple d'arbre de Hough, construit à partir d'un échantillon de 39 imagettes positives (en bleu) et de 23 imagettes négatives (en rouge), partitionné en 5 feuilles sur des prédicats relatifs à l'apparence.

2. la variance spatiale des vecteurs positions associés aux imagettes positives, pour avoir des votes le moins dispersés possible

La figure 5.17 montre un exemple de bonne et de mauvaise feuille, au sens de ces critères.

Une fois le modèle construit en apprentissage hors ligne, la détection (en ligne) consiste à soumettre chaque pixel (ou un sous-ensemble de pixels ayant un minimum de structure), pour chaque arbre de la forêt de Hough, à l'ensemble des questions sur son apparence locale, puis à réaliser l'ensemble des votes correspondant aux imagettes positives de la feuille d'arrivée.

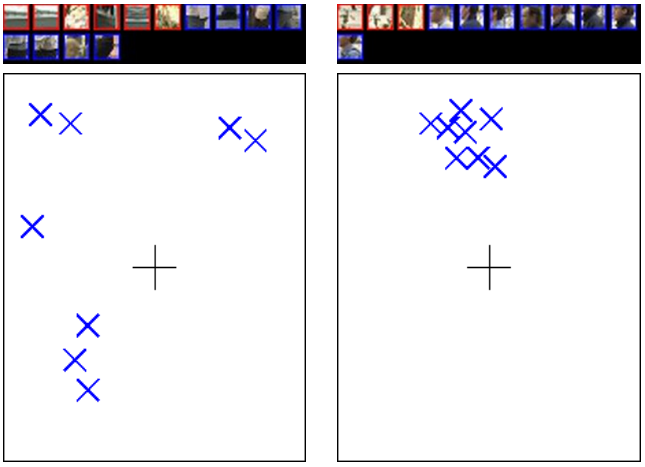


FIGURE 5.17 – Deux exemples de feuilles, représentées par leur sous-ensemble d'imagettes (en haut), et les positions relatives associées aux imagettes positives (en bas). À gauche, une feuille médiocre (grosse entropie de classes et votes positifs dispersés). À droite, une feuille efficace (majorité de labels positifs et votes positifs concentrés).