



DÉPARTEMENT TSI

Vision par ordinateur et raisonnement dans les images

Brique VOIR - Module VORAS

Isabelle BLOCH, Henri MAÎTRE, Florence TUPIN

ENST, département TSI, CNRS UMR 5141 LTCI, 46 rue Barrault, 75634 Paris Cedex 13
Tél : 01 45 81 75 85, Fax : 01 45 81 37 94, E-mail : Isabelle.Bloch@enst.fr

Mai 2005

Table des matières

1	Les principales méthodes de recalage	5
1.1	Introduction	5
1.2	Les types de problèmes de recalage	6
1.3	Les transformations admissibles	6
1.4	Les primitives utilisées	7
1.5	Fonctions de similarité / dissimilarité	7
1.5.1	Similarité entre points en correspondance	8
1.5.2	Correspondance inconnue	9
1.5.3	Fonctions de similarité entre images (pour des recalages mono-modalité)	10
1.5.4	Fonctions de similarité entre images (pour des recalages multi-modalités)	10
1.6	Optimisation	12
1.7	Conclusion	12
2	Graphes en traitement d'images et en reconnaissance des formes	15
2.1	Introduction	15
2.1.1	Exemples de graphes en traitement d'images	15
2.2	Approches mono-graphes	16
2.2.1	Méthodes de segmentation	16
2.2.2	Graphes et champs de Markov	19
2.2.3	Graphes et reconnaissance des formes	21
2.3	Appariement de graphes	22
2.3.1	Isomorphismes de sous-graphes	23
2.3.2	Isomorphismes de sous-graphes avec tolérance d'erreurs	24
2.3.3	Appariement par algorithmes approximatifs	25
2.3.4	Cas des informations topographiques	28
2.4	Conclusion	29

3	Des images 2D aux objets à 3D	31
3.1	Les graphes d'aspect	32
3.1.1	Référentiel, perspective orthographique ou cavalière	32
3.1.2	Définitions	33
3.1.3	Visibilité	34
3.1.4	Les cas compliqués	37
3.1.5	Reconnaissance d'un objet	38
3.1.6	Stockage des graphes	38
3.1.7	Extension aux objets non polyédriques	39
3.2	Les lignes caractéristiques	39
3.2.1	Les silhouettes	39
3.2.2	Lignes paraboliques, et ligne singulière	42
3.3	La forme à partir de l'ombrage	43
3.3.1	Mise en équation	44
3.3.2	Résolution du problème	46
3.3.3	Conclusions	49
4	Les grammaires en traitement des images	51
4.1	Les grammaires formelles	51
4.1.1	Un exemple	51
4.1.2	Définition	52
4.1.3	les deux usages d'une grammaire	52
4.1.4	Les divers types de grammaires	55
4.2	Extensions indispensables en traitement d'images	56
4.2.1	Les grammaires stochastiques	56
4.2.2	Grammaires tolérantes	57
4.3	Quelles grammaires pour l'analyse de scène et le traitement des images?	57
4.4	Les grammaires pour le graphique	58
4.4.1	Les contours à 2D	58
4.4.2	Les descriptions des objets tri-dimensionnels	60
4.5	Les grammaires pour les images	60
4.5.1	Grammaires et détection	62
4.5.2	Grammaires et textures	62
4.6	Des usages plus avancés	63

Chapitre 1

Les principales méthodes de recalage

Chapitre encore incomplètement rédigé...

1.1 Introduction

Recalage = mise en correspondance spatiale (même point géométrique \rightarrow mêmes coordonnées)

Autres termes : (co)registration, normalisation spatiale (en imagerie médicale, surtout inter-sujets)

Intérêt et nécessité du recalage :

- usage intensif d’images multimodalités, fusion
- stéréo, reconstruction 3D, GIS
- recalage avec un modèle (carte, atlas, formes connues a priori), RDF, interprétation de scènes

Domaines d’application variés.

Exemples en imagerie médicale : usage intensif d’images médicales multiples. Applications : diagnostic, planification, guidage (traitement, chirurgie), monitoring, recherche (anatomo-fonctionnelle)

Différents types d’images, de principes physiques différents, dédiées à des applications différentes Résolution, qualité, type d’informations, distorsions éventuelles : autant de paramètres variables d’une modalité ou d’une image à l’autre. Donc à prendre en compte

Formulation générale :

$$\min_{t \in \mathcal{T}} f(I_1, t(I_2)) \quad (1.1)$$

- I_1 et I_2 images à recaler (ou informations extraites de ces images)
- t : transformation
- \mathcal{T} : ensemble des transformations possibles / admissibles

Composantes d’un système de recalage :

- nature de la transformation (t et son domaine de définition \mathcal{T})

- primitives utilisées (sur quoi portent t et f)
 - critère de similarité f
 - méthode d'optimisation (minima locaux, multi-échelles)
- non indépendants entre eux, et non indépendants du type d'images, des modalités, et du problème de recalage posé.

Difficultés liées

- à la complexité des problèmes
- à la discrétisation des images
- à l'évaluation de la qualité du recalage

1.2 Les types de problèmes de recalage

- 2D/2D/, 2D/3D, 3D/3D
- recalage d'images mono-modales
- recalage d'images multi-modales
- recalage d'images multi-temporelles
- recalage modalité / modèle (ex : carte, atlas)
- recalage inter-patients
- ...

1.3 Les transformations admissibles

Les principales classes de transformations linéaires sont les suivantes :

- transformations rigides, composées uniquement de translations et rotations ;
- transformations affines : toujours linéaires, ces transformations ont plus de degrés de liberté que les précédentes, elles peuvent par exemple inclure des changements d'échelle, et elles transforment des lignes parallèles en lignes parallèles ;
- transformations projectives (les transformations linéaires les plus générales) qui mettent en correspondance des lignes non parallèles : elles sont souvent utilisées pour mettre en correspondance une image 3D avec une image de projection 2D puisqu'elles incluent des transformations de perspective.

Dans certaines applications, comme en imagerie médicale pour des organes fortement déformables, les transformations linéaires ne suffisent pas à prendre en compte la complexité des mouvements et donc des différences possibles entre deux images. Des transformations non-linéaires (une droite peut alors être transformée en courbe) doivent alors être utilisées, parmi lesquelles on trouve essentiellement :

- les transformations polynomiales ;
- les fonctions de base radiale, définies comme des combinaisons linéaires de fonctions de base telles que des gaussiennes, des fonctions multiquadratiques, des « thin-plate splines », etc. ;
- des déformations de forme libre, exprimées comme des combinaisons linéaires de fonctions splines définies sur une grille de points de contrôle ;
- des déformations élastiques, un exemple de déformation physique dans lequel le champ de déformation $u(x, y, z)$ suit l'équation de Navier :

$$\mu \nabla^2 u(x, y, z) + (\lambda + \mu) \nabla(\nabla \cdot u(x, y, z)) + f(x, y, z) = 0$$

où f représente des forces externes appliquées au corps élastique et λ et μ sont des constantes d'élasticité (le calcul de ce type de déformation est coûteux) ;

- des transformations fluides suivant l'équation de Navier-Stokes, similaire à l'équation précédente mais où u est remplacé par le champ de vitesse.

L'application de ces transformations nécessite des procédures d'interpolation et de rééchantillonnage (voir la brique ANIM).

1.4 Les primitives utilisées

Du point de vue des primitives utilisées (les éléments auxquels les transformations sont appliquées pour évaluer la fonction de similarité), les méthodes de classification se répartissent en deux catégories : celles utilisant des primitives extraites des images, ne représentant qu'une partie de son contenu, et celles utilisant tout le contenu des images.

Pour le premier type d'approche, les primitives choisies doivent être faciles à obtenir (par des méthodes de détection ou de segmentation), robustes au bruit et aux artefacts potentiels, visibles dans les deux images à recaler, stables d'une acquisition à l'autre, bien distribuées dans l'ensemble de la zone à recaler. Elles peuvent être extrinsèques, c'est-à-dire ajoutées à l'objet imagé (c'est le cas par exemple de cadres stéréotaxiques utilisées pour certaines applications en imagerie cérébrale, de marqueurs, etc.). Ce type de primitive n'est pas toujours disponible et souvent assez contraignant. Dans la plupart des cas, des primitives intrinsèques sont donc préférées, c'est-à-dire des primitives directement reliées au contenu des images. Il peut s'agir de :

- points (points de repères remarquables, carrefours, bifurcations, coins...)
- structures linéiques géométriques ou différentielles (polygones, segments, courbes, contours, lignes de crête...)
- primitives géométriques et différentielles (lignes de crête, etc.)
- surfaces et régions...

Plus les primitives sont de dimension élevée, plus elles fournissent de redondance et donc potentiellement une plus grande robustesse. Une question importante qui jouera sur les méthodes décrites par la suite est de savoir si la correspondance entre les primitives des deux images est connue ou non, ou si l'on peut l'établir avec facilité.

Dans le deuxième type d'approche, l'étape de détection ou de segmentation disparaît. Toute l'information contenue dans les images est utilisée pour trouver la transformation. Dans ce cas la question importante et sur laquelle nous allons revenir est de savoir si les intensités des deux images sont en relation fonctionnelle.

1.5 Fonctions de similarité / dissimilarité

Quelques exemples :

- recherche du point le plus proche
- minimisation de la distance entre points de repère
- maximisation de l'aire d'intersection de surfaces
- alignement des axes d'inertie
- corrélation entre les deux images (spatial / Fourier)
- fonction de similarité quadratique

- minimisation de la variance des rapports d'intensité
- minimisation de la variance dans des partitions de l'image de référence projetées sur l'image à recalcr
- maximisation de l'information mutuelle de l'histogramme conjoint
- minimisation de la dispersion de l'histogramme conjoint
- comptage des changements de signe
- minimisation de la médiane du carré des erreurs
- minimisation de mesures fondées sur la matrice de co-occurrence
- utilisation de la transformée de distances
- critères bayésiens
- ...

1.5.1 Similarité entre points en correspondance

Hypothèses :

- même nombre de points n
- correspondance entre x_i et y_i connue
- dimension quelconque
- pas de points aberrants (robuste au bruit gaussien)

Définition du critère :

$$E = \sum_{i=1}^n \|x_i - (r(y_i) + t)\|^2$$

Translation optimale : mise en correspondance des centres d'inertie des deux ensembles de points

Rotation optimale : formule directe en 2D, méthode des quaternions en 3D

Représentation d'une rotation d'axe \vec{u} et d'angle θ : (s, v) et $(-s, -v)$ avec :

$$s = \cos \frac{\theta}{2}$$

$$v = \sin \frac{\theta}{2} \vec{u}$$

Relation d'équivalence : $\mathcal{R}(q, q') \Leftrightarrow q = -q'$

\mathcal{R}^3 est isomorphe à $\mathcal{Q}_1/\mathcal{R}$ \mathcal{R}^3 est isomorphe à \mathcal{P}^3
--

$$Rx = q \times x \times \bar{q}$$

Application au recalage rigide : minimisation de $E = \sum_{i=1}^n \|x_i - (r(y_i) + t)\|^2$

$$E = \sum_{i=1}^n |x_i - q \times y_i \times \bar{q}|^2$$

$$\begin{aligned}
&= \sum_{i=1}^n |x_i - q \times y_i \times \bar{q}|^2 |q|^2 \\
&= \sum_{i=1}^n |x_i \times q - q \times y_i \times \bar{q} \times q|^2 \\
&= \sum_{i=1}^n |x_i \times q - q \times y_i|^2 = \sum_{i=1}^n q^t A_i^t A_i q
\end{aligned}$$

La rotation optimale s'obtient en calculant les valeurs propres de la matrice A :

$$A = \sum_{i=1}^n A_i^t A_i.$$

Solution du problème de minimisation aux moindres carrés = quaternion qui est le vecteur propre de norme 1 correspondant à la plus petite valeur propre de A .

1.5.2 Correspondance inconnue

– Méthode de Fourier en 2D :

$$I_2 = I_1 + t \Rightarrow TF^{-1}[TF(I_2)TF^*(I_1)/|TF(I_2)TF^*(I_1)|] = \delta(x_0, y_0)$$

– Réduire la complexité :

- associations progressives entre primitives
- recalage entre les primitives les plus pertinentes, puis ajout progressif des autres
- contraintes géométriques, topologiques, etc.
- mise en correspondance de graphes
- distance entre surfaces

Carte de distances (voir brique ANIM). Voir figure 1.1.

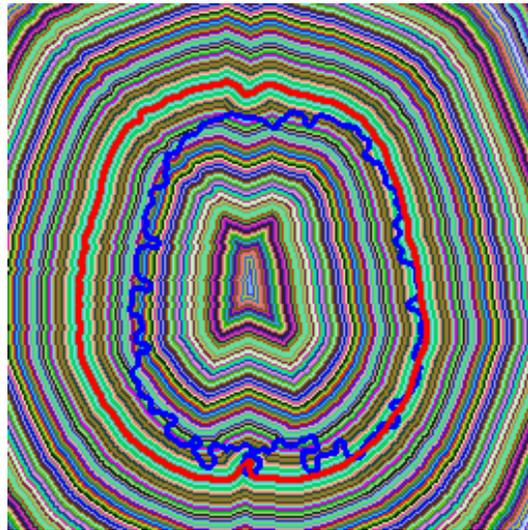


FIG. 1.1 – Utilisation d'une carte de distance pour le recalage.

Méthode ICP (Besl & McKay) : calcul itératif à la fois de la correspondance et de la transformation.

1.5.3 Fonctions de similarité entre images (pour des recalages mono-modalité)

Quadratique :

$$E(\Theta) = \sum_x [I_{ref}(x) - I_{rec}(T_\Theta(x))]^2$$

Quadratique avec normalisation d'intensité :

$$E(\Theta) = \sum_x \left[\frac{\bar{I}_{rec}}{\bar{I}_{ref}} I_{ref}(x) - I_{rec}(T_\Theta(x)) \right]^2$$

Corrélation :

$$R(\Theta) = \frac{\sum_x [I_{ref}(x) - \bar{I}_{ref}] [I_{rec}(T_\Theta(x)) - \bar{I}_{rec}]}{\sqrt{\sum_x [I_{ref}(x) - \bar{I}_{ref}]^2 \sum_x [I_{rec}(T_\Theta(x)) - \bar{I}_{rec}]^2}}$$

(maximum de corrélation pour la « bonne » transformation)

Similarité robuste :

$$E(\Theta) = \sum_x \rho [I_{ref}(x) - I_{rec}(T_\Theta(x))]$$

$\rho =$ M-estimateur

La figure 1.2 présente quelques exemples de fonctions ρ et de leur dérivée. La quadratique classique a une dérivée linéaire, qui donne donc un gradient du critère d'optimisation d'autant plus fort que les erreurs sont grandes. Pour éviter l'influence de ces grandes erreurs, la quadratique tronquée permet d'annuler la dérivée si la valeur absolue de l'erreur dépasse un certain seuil, au prix de discontinuités dans la dérivée. L'estimateur de Geman-McLure atténue la quadratique, avec une fonction de la forme :

$$\rho(x) = \frac{x^2}{x^2 + C}$$

où x désigne l'erreur et C est une constante, dont la dérivée a la forme :

$$\rho'(x) = \frac{2xC}{(x^2 + C)^2}$$

et est donc continue. Des critères convexes (contrairement aux deux précédents) ont également été proposés. Par exemple la fonction de Huber a une forme quadratique pour les faibles erreurs, puis linéaire. Sa dérivée est linéaire pour les faibles erreurs, constantes pour les erreurs dépassant un certain seuil, et continue.

1.5.4 Fonctions de similarité entre images (pour des recalages multi-modalités)

Uniformité inter-images : partition de l'image de référence en régions homogènes et projection (spatiale) sur l'image à recaler, puis minimisation de :

$$E(\Theta) = \sum_{regions\ g} \frac{N_g \sigma_g(T_\Theta(x))}{N \mu_g(T_\Theta(x))}$$

$$\sigma_g(T_\Theta(x)) = \sqrt{\sum_{x, I_{ref}(x)=g} [I_{rec}(T_\Theta(x)) - \mu_g(T_\Theta(x))]^2}$$

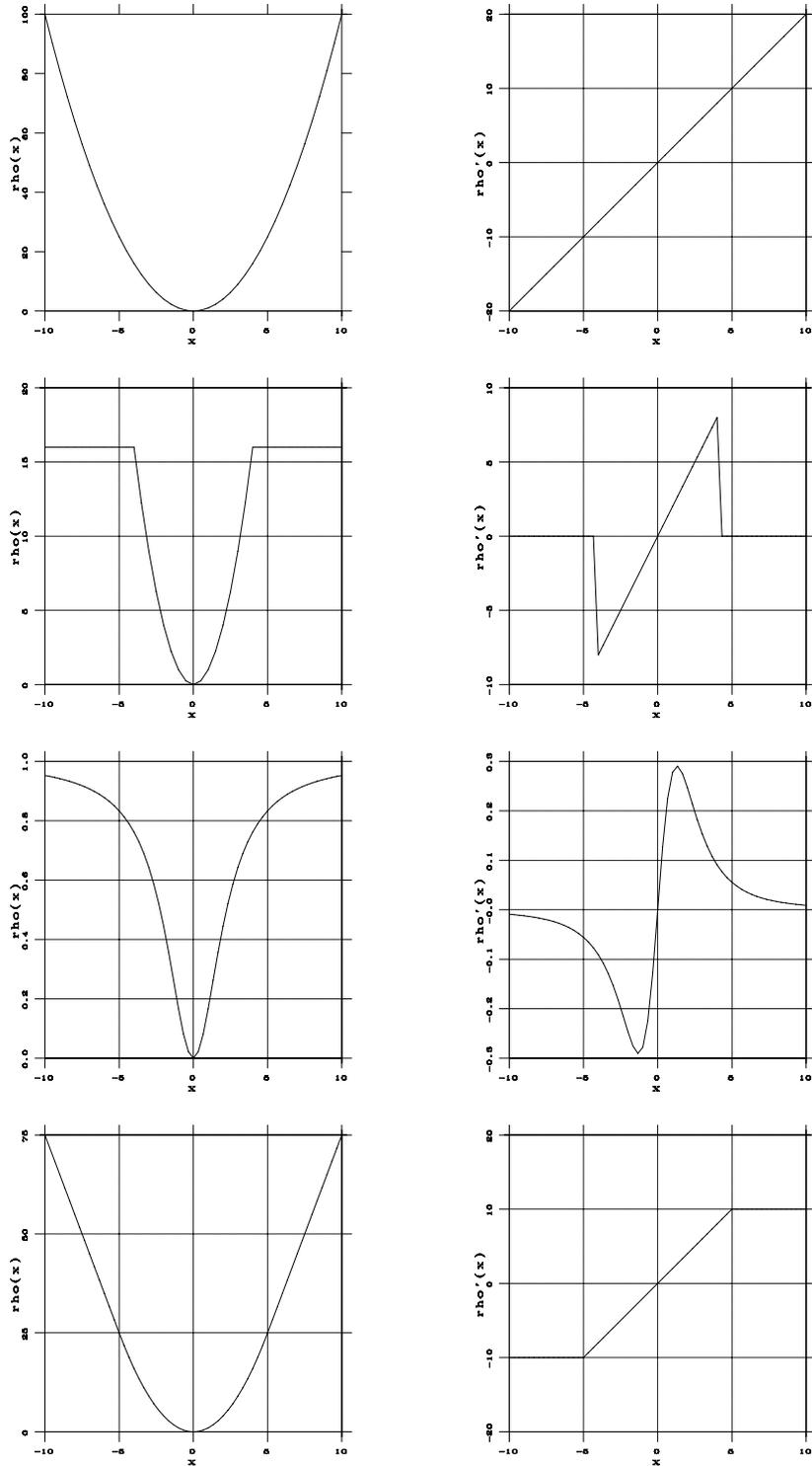


FIG. 1.2 – Quelques exemples de M-estimateurs. A gauche, fonction ρ , à droite, sa dérivée. De haut en bas : quadratique, quadratique tronquée, estimateur de Geman-McLure, estimateur de Huber.

$$\mu_g(T_\Theta(x)) = \frac{1}{N_g} \sum_{x, I_{ref}(x)=g} I_{rec}(T_\Theta(x))$$

Exploitation de l'histogramme conjoint : maximisation de l'information mutuelle

$$E(\Theta) = - \sum_g \sum_k p(g, k) \log \frac{p(g, k)}{p(g)p(k)}$$

Similarité robuste :

$$E(\Theta) = \sum_{regions\ g} \frac{N_g}{N} \tilde{\sigma}_g(T_\Theta(x))$$

$$\tilde{\sigma}_g(T_\Theta(x)) = \sqrt{\sum_{x, I_{ref}(x)=g} \rho[I_{rec}(T_\Theta(x)) - \tilde{\mu}_g(T_\Theta(x))]}$$

$$\tilde{\mu}_g(T_\Theta(x)) = \underset{\mu_g}{\operatorname{argmin}} \sum_{x, I_{ref}(x)=g} \rho[I_{rec}(T_\Theta(x)) - \mu_g]$$

1.6 Optimisation

Problème des minima locaux, d'où importance de l'initialisation :

- connaissance de la position du patient en imagerie médicale, des paramètres d'acquisition, ou autres informations a priori
- axes d'inertie
- essais à partir de quelques positions et choix du meilleur résultat

Algorithmes classiques d'optimisation : gradient, gradient conjugué, Powell, simplexe, Levenberg-Marquardt, Newton-Raphson, hashage géométrique

Autres solutions pour sortir des minima locaux : diminution progressive du pas, optimisation stochastique, algorithmes génétiques, recuit simulé

Méthodes adaptées à la fonction de coût (exemple : ICP)

Recherche par graphes, ou programmation dynamique

Multi-échelles, multi-résolutions : convergence plus rapide, moins de problèmes de minima locaux

1.7 Conclusion

Difficultés : validation, robustesse au bruit et aux dissimilarités, complexité

Validation : précision intrinsèque de l'algorithme, précision de la mesure, robustesse, fiabilité, ressources requises, complexité algorithmique, vérification des hypothèses, utilisation en pratique (exemple : contraintes particulières de la routine clinique)

Problèmes des objets de test : idéaux, pas de cas rares ou pathologiques.

Exemple d'études en recalage d'images médicales : Vanderbilt

Automatique : souvent visé mais pas toujours souhaitable Interactif (assistance par logiciel de visu) : lourd, surtout en 3D, peu reproductible Semi-automatique : interaction soit

au niveau de l'initialisation (d'où réduction de l'espace de recherche, position initiale proche de la solution donc moins de problèmes de minima locaux), soit au niveau du contrôle

Chapitre 2

Graphes en traitement d'images et en reconnaissance des formes

2.1 Introduction

La description du contenu d'une image se fait le plus souvent par une représentation topologique dans laquelle les relations contextuelles tiennent une grande importance. Un outil de représentation naturel d'une scène est donc un graphe dont les nœuds représentent les objets présents et les arcs les relations que l'on souhaite mettre en évidence. Un des intérêts principaux d'une telle représentation est sa compacité et par voie de conséquence les gains de place et de temps de calcul qui lui seront associés. Les travaux sur l'utilisation des graphes en traitement d'images et en reconnaissance des formes remontent au début des années 70 pour l'interprétation automatique de dessins polygonaux [53] et se poursuivent aujourd'hui [1].

Ce chapitre se scinde en deux grandes parties. La première est dédiée aux approches mono-graphe dans lesquelles la scène est représentée sous forme d'un graphe qui est ensuite traité pour extraire l'information cherchée (segmentation, contours, objets spécifiques,...). la seconde partie traitera de la mise en correspondance entre deux graphes, l'un extrait généralement des données et l'autre étant un modèle de l'objet ou de la scène étudiée.

2.1.1 Exemples de graphes en traitement d'images

Dans toute la suite, nous noterons un graphe G par le couple (X, E) , où X est l'ensemble des sommets et E l'ensemble des arcs. L'ordre n du graphe est le nombre de nœuds et sa taille m le nombre d'arcs. On trouvera dans [13] les définitions et les algorithmes classiques en théorie des graphes. On pourra également se reporter au livre de Gondran et Minoux[27].

Les graphes seront généralement attribués aussi bien au niveau des nœuds que des arcs. On définit un graphe relationnel attribué par $G = (X, E, \mu, \nu)$, avec :

- $\mu : X \rightarrow L_X$ fonction d'attribution des attributs aux sommets (interpréteur de sommets)
- $\nu : E \rightarrow L_E$ fonction d'attribution des attributs aux arcs (interpréteur d'arcs)

Le graphe le plus couramment utilisé est le graphe des pixels, où chaque pixel représente

un nœud du graphe et chaque arc est défini par la 4- ou la 8-connexité. Une représentation plus compacte de l'image est donnée par exemple par le graphe d'adjacence (GRA) construit à partir d'une sur-segmentation. Dans ce cas, on associe souvent aux nœuds représentant les régions des attributs comme le niveau de gris moyen, la surface, des indices texturaux, etc. Quant aux arcs entre deux régions adjacentes, on leur associe la longueur du contour ou une mesure de contraste. Des graphes également couramment utilisés sont les graphes de Voronoï construits à partir d'un nuage de points ou, de façon duale, la triangulation de Delaunay [40].

Il est important de noter que de nombreux graphes en traitement d'images sont des graphes planaires (i.e pour lesquels il existe une représentation graphique dans laquelle les arcs ne se coupent pas [27]). Ceci présente l'intérêt de rendre certains algorithmes de complexité polynomiale plutôt qu'exponentielle.

Il existe d'autres graphes comme les graphes aléatoires (typiquement utilisés dans les champs markoviens), les graphes flous et les graphes d'attributs flous [55].

2.2 Approches mono-graphes

Nous avons regroupé dans cette partie les approches utilisant un seul graphe. Il s'agit essentiellement de méthodes de segmentation visant à partitionner l'image en régions homogènes et de méthodes de reconnaissance des formes visant à extraire un sous-graphe du graphe initial.

2.2.1 Méthodes de segmentation

De nombreuses méthodes de segmentation s'appuyant sur la théorie des graphes ont été proposées. L'idée de base est la suivante : à partir d'un graphe construit sur les pixels de l'image (les arcs étant définis par la 4-connexité), on cherche à partitionner le graphe de telle sorte que les parties obtenues correspondent à des zones "homogènes" de l'image (au sens de l'attribut des nœuds).

Arbre couvrant de poids minimal

Une première approche proposée par Constantinidès [52] se fonde sur la recherche de l'arbre couvrant de poids minimal pour réaliser la segmentation. Le graphe est construit en pondérant les arcs par la valeur absolue de la différence de niveaux de gris, ou tout autre attribut significatif pour la segmentation considérée, des sommets qu'ils relient (figure 2.1). Sur ce graphe, on calcule alors l'arbre couvrant de poids minimal. On supprime les arêtes les plus coûteuses du graphe pour obtenir la segmentation. En effet, on obtient alors une forêt, dans laquelle chaque arbre représente une zone homogène (puisque toutes les arêtes ont des coûts faibles, et donc lient des sommets dont les attributs sont proches). Le critère de suppression des arêtes peut porter sur la valeur de l'arête ou sur le nombre total d'arêtes à supprimer.

La recherche de l'arbre couvrant de poids minimal se fait par l'algorithme de Kruskal (complexité en $O(n^2 + m \log_2(m))$) ou par l'algorithme de Prim (en $O(n^2)$) [13]. Dans les

deux cas l'idée est de trier les arêtes du graphe par ordre de poids croissants et de les ajouter pour constituer l'arbre.

C'est une méthode de segmentation bien adaptée aux images sans bruit, qui n'introduit pas d'information sur la forme des régions cherchées, et qui suppose que les régions à segmenter sont bien délimitées les unes par rapport aux autres. Une mise en échec sera par exemple due à l'existence d'une connexion entre deux régions similaires mais appartenant à des objets différents.

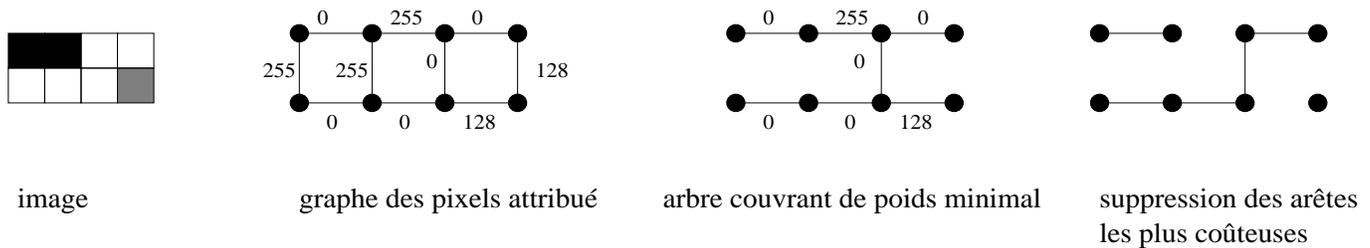


FIG. 2.1 – Segmentation par arbre couvrant de poids minimal

Coupe de capacité minimale

Dans le même ordre d'idée, des méthodes de segmentation recherchant la coupe de capacité minimale ont été proposées [80]. A nouveau, le principe est de rechercher un partitionnement du graphe en sous-graphes. Une coupe partitionne un graphe en deux parties A et B telles que $A \cup B = X$ et $A \cap B = \emptyset$, et la capacité de la coupe est définie par :

$$cut(A, B) = \sum_{x \in A, y \in B} w(x, y)$$

avec $w(x, y)$ capacité de l'arc (x, y) . Le graphe considéré est à nouveau le graphe des pixels en 4-connexité, mais cette fois-ci le poids des arcs est une fonction de similarité (et non l'inverse comme précédemment). La coupe de poids minimale va donc déconnecter les pixels très dissemblables (figure 2.2).

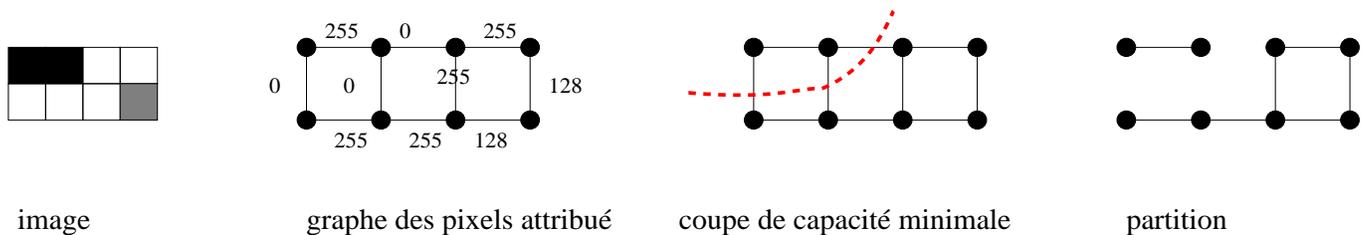


FIG. 2.2 – Segmentation par coupe de capacité minimale

La recherche de la coupe de capacité minimale se fait par l'algorithme de Ford et Fulkerson [13]. Sa complexité est en $O(mnc_{max})$ où c_{max} est la capacité maximale des arcs. Même si des variantes plus rapides ont été proposées, les temps de calcul restent relativement longs. Une extension naturelle qui permet d'accélérer le traitement de l'image consiste à effectuer une sur-segmentation puis à construire le graphe d'adjacence des régions.

Le problème de la coupe de capacité minimale est que la capacité est toujours fonction du nombre d'arcs de la coupe, et donc que pour des poids positifs, on aura toujours intérêt à partitionner le graphe de telle sorte que le moins d'arcs possible intervienne dans la coupe. Ce phénomène est illustré sur la figure 2.3, avec un graphe de 6 sommets et des arcs de poids a ou b inversement proportionnels à la distance ($b < a$).

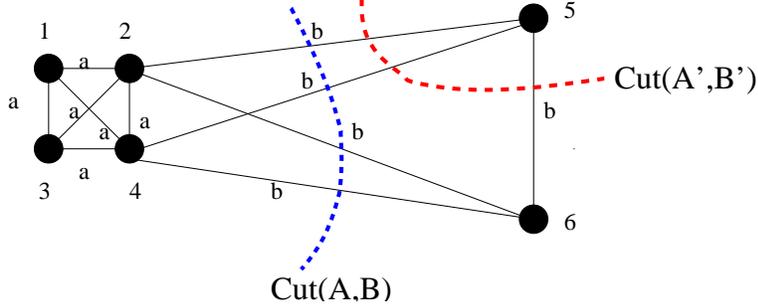


FIG. 2.3 – Influence du nombre de sommets dans la coupe : $Cut(A, B) = 4b$, $Cut(A', B') = 3b$

Considérons les deux coupes suivantes : $(A = \{1, 2, 3, 4\}, B = \{5, 6\})$ et $(A' = \{1, 2, 3, 4, 6\}, B' = \{5\})$. On a alors :

$$cut(A, B) = 4b \quad \text{et} \quad cut(A', B') = 3b$$

et il est donc toujours moins coûteux d'isoler un nœud, même si structurellement la partition (A, B) serait préférable. Pour remédier à ce problème Shi et Malik ont proposé une nouvelle définition du coût d'une coupe [62], qui supprime l'influence du nombre d'arcs dans la coupe :

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, G)} + \frac{cut(A, B)}{assoc(B, G)}$$

avec $assoc(A, G) = \sum_{a \in A, x \in G} w(a, x)$ Pour cette nouvelle définition, la contribution de B' à la valeur de la coupe sera de 100 % puisque $assoc(B', G) = cut(A', B')$. On obtient $Ncut(A, B) = \frac{4b}{12a+4b} + \frac{2}{3}$ et $Ncut(A', B') = \frac{3b}{12a+7b} + 1$, ce qui résout bien le problème mentionné précédemment. On peut également définir une mesure de l'association normalisée à l'intérieur de chacune des parties de la partition par :

$$Nassoc(A, B) = \frac{assoc(A, A)}{assoc(A, G)} + \frac{assoc(B, B)}{assoc(B, G)}$$

On peut alors montrer la propriété suivante :

$$Ncut(A, B) = 2 - Nassoc(A, B)$$

et donc minimiser la coupe normalisée revient à maximiser l'association correspondante. On retrouve une propriété similaire à celle de la classification pour laquelle maximiser la variance inter-classes revient à minimiser la variance intra-classes.

La recherche de la coupe normalisée minimale est un problème NP-complet. Néanmoins en se plaçant dans le domaine réel, une solution discrète approximée peut être trouvée de façon efficace. En effet, posons $x = (x_i)$, $x_i = 1$ si $i \in A$, $x_i = -1$ sinon, une représentation vectorielle de la partition. Soit $D = diag(d_i)$, $d_i = \sum_j w_{ij}$ et $W = (w_{ij})$ représentant les matrices de pondération (w_{ij} étant la capacité d'un arc), alors on peut montrer que :

$$\min_x Ncut(x) = \min_y \frac{y^T (D - W) y}{y^T D y}$$

avec $y_i \in \{1, -b\}$ ($b = \frac{\sum_{x_i > 0} d_i}{\sum_{x_i < 0} d_i}$, $y_i = \frac{1}{2}[(1 + x_i) - b(1 - x_i)]$) et $y^T D 1 = 0$.

La solution de ce système est donnée par le vecteur propre associé à la seconde plus petite valeur propre de la matrice :

$$D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}$$

La recherche des valeurs et vecteurs propres est en $O(n^3)$ mais on peut la réduire à $O(n)$ pour des matrices très creuses, ce qui est le cas ici. Bien sûr, en pratique le vecteur propre y obtenu ne prend pas seulement les valeurs 1 et $-b$, et il est nécessaire de choisir une valeur seuil pour partitionner le graphe. On peut prendre la valeur 0 ou tester un ensemble de valeurs et calculer la coupe normalisée correspondant à chaque fois, puis prendre la partition correspondant au minimum.

En fonction de la qualité de la partition obtenue, l'algorithme est relancé sur l'une ou l'autre des parties seulement, jusqu'à ce que la valeur de la coupe obtenue soit trop forte. Des exemples de segmentation sont montrés sur la figure 2.4.

Systèmes à base de règles

De très anciens travaux ont proposés une interprétation automatique des images à partir du graphe des régions d'une sur-segmentation [68]. L'interprétation se fait le plus souvent à partir d'un système à base de règles sur les objets (taille, couleur, texture,...) et sur les relations les liant (au dessus de, à l'intérieur de, à côté de, etc.).

Ce formalisme très général a ensuite été repris pour de nombreuses applications, notamment avec des généralisations utilisant des graphes d'attributs flous [55] et sur des graphes polymorphiques [11].

2.2.2 Graphes et champs de Markov

La théorie des graphes peut être reliée aux modèles markoviens selon deux points de vue.

D'une part, toutes les applications markoviennes se font sur des graphes aléatoires, que ce soit celui des pixels pour des applications de bas-niveau (segmentation, classification, restauration,...), ou un graphe construit sur des primitives de plus haut niveau (points caractéristiques, régions, etc. pour la reconnaissance d'objets ou l'interprétation de la scène). On pourra se référer à [64], chap.6, pour des exemples d'utilisation.

D'autre part, la théorie des graphes donne un moyen de calculer la configuration maximisant la probabilité a posteriori du champ, ou de façon équivalente minimisant son énergie.

Les premiers travaux établissant ce résultat sur des images binaires datent de 1989 [29]. Le graphe utilisé est le graphe des pixels de l'image auquel on ajoute une source S (correspondant au label 0) et un puits (correspondant au label 1). Les capacités des arcs sont alors définies comme suit (fig.2.5) :

- les arcs terminaux sont pondérés par l'attache aux données pour le label associé à S ou P (noté $V_c(y_i|x_i)$ pour une observation y_i au site i et un label x_i);
- les arcs entre sites voisins sont pondérés par $V_c(0,1)$, potentiel de la clique pour une configuration des pixels avec des labels différents (de valeur β ici).

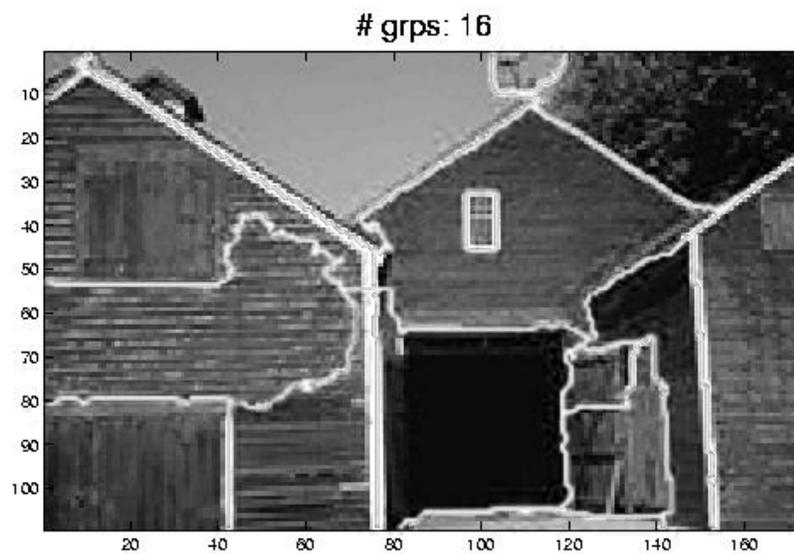
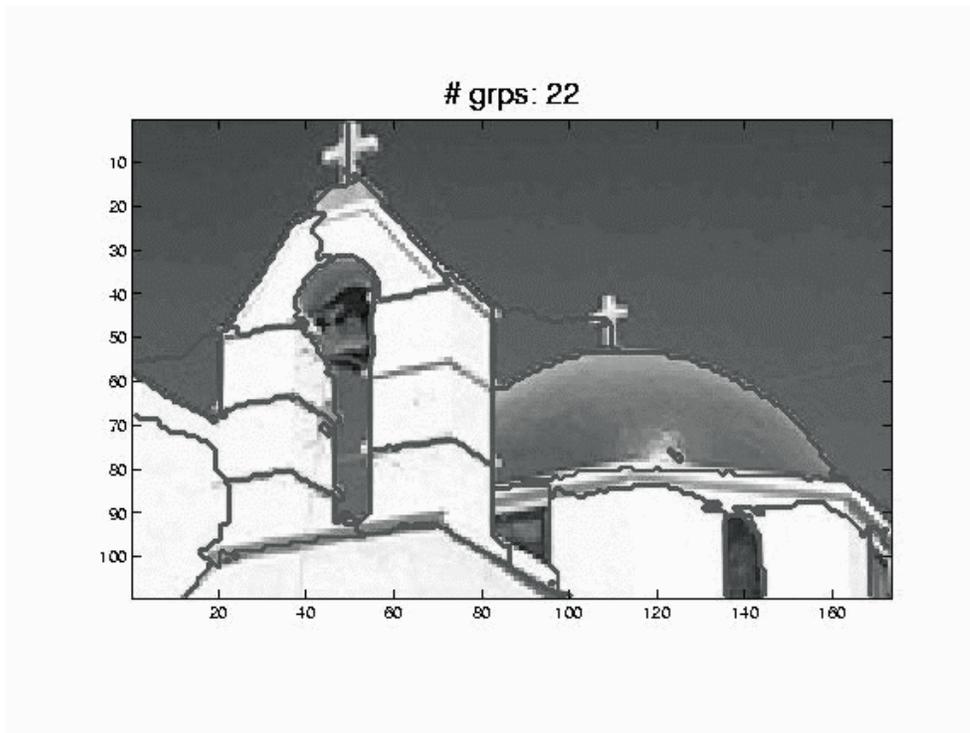


FIG. 2.4 – Résultats de segmentation par coupe normalisée (<http://www.cs.berkeley.edu/projects/vision/Grouping/manmade.html>). Les poids sont définis par $w_{ij} = \exp\left(-\frac{|F(i)-F(j)|_2^2}{\sigma_F^2}\right)$. Dans le premier exemple $F(i)$ est le niveau de gris du point et dans le second un vecteur d'indices texturaux.

La solution du maximum a posteriori est alors donnée par la coupe séparant la source du puits de capacité minimale, en choisissant comme étiquette d'affectation pour un site l'arc terminal (à la source ou au puits) qui est coupé. En effet, l'énergie s'écrit de la façon suivante :

$$\mathcal{U}(x|y) = \sum_i V_c(y_i|x_i) + \sum_{(i,j)} \beta(x_i - x_j)^2$$

pour un modèle d'Ising. La capacité d'une coupe séparant les sites en deux sous-ensembles E_S (pour ceux qui restent reliés à la source) et E_P (pour ceux qui restent reliés au puits), s'écrit :

$$cut(E_S, E_P) = \sum_{i \in E_S} V_c(y_i|1) + \sum_{i \in E_P} V_c(y_i|0) + \sum_{(i \in E_S, j \in E_P)} \beta$$

ce qui s'identifie directement à $\mathcal{U}(x|y)$ avec la convention précédente ($x_i = 1$ pour $i \in E_S$, $x_i = 0$ pour $i \in E_P$). La solution trouvée correspond *exactement* au MAP.

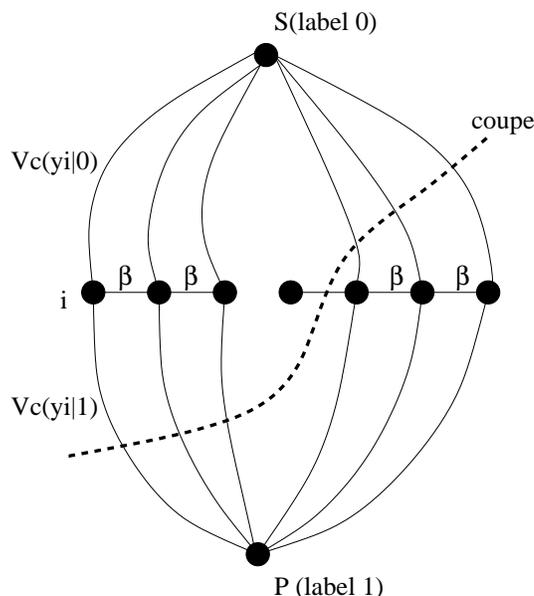


FIG. 2.5 – Définition du graphe pour la recherche du MAP dans un modèle markovien binaire

Les travaux de Boykov et Veksler [73] proposent une généralisation à des images en niveaux de gris, pour de la restauration ou de la segmentation (un formalisme est développé dans chaque cas). Comme on ne peut ajouter à chaque fois qu'une source et qu'un puits au graphe des pixels, l'idée est de définir des "mouvements" de la solution : l'échange entre deux labels α/β (tous les pixels possédant l'un des deux labels peuvent prendre l'autre, et l'expansion d'un label α (tous les pixels peuvent prendre ce label). Les capacités du graphe sont calculées à partir du résultat courant. Un faible nombre d'itérations permet de converger vers une solution. On peut montrer que ces algorithmes sont plus rapides qu'un recuit simulé, mais contrairement au cas binaire précédent la solution trouvée n'est pas exacte.

2.2.3 Graphes et reconnaissance des formes

Les graphes sont très largement utilisés pour la reconnaissance des formes [50]. Le principe est généralement le suivant : l'objet est défini par un ensemble de primitives qui constituent

les nœuds du graphe, des relations binaires de compatibilité entre primitives constituent les arcs du graphe. Une clique du graphe représente alors un sous-ensemble de primitives compatibles 2 à 2 qui est une configuration possible de l'objet. La reconnaissance se fait alors par la détection des cliques maximales du graphe (i.e des cliques auxquelles on ne peut plus rajouter aucun sommet). Si plusieurs cliques maximales existent, on définit une fonction de coût pour effectuer le choix.

La recherche des cliques maximales d'un graphe est un problème NP-complet. Elle s'effectue en construisant un arbre de décision (un nœud de l'arbre correspond à une clique du graphe) qui est élagué afin de ne pas réengendrer les mêmes cliques¹.

Un exemple d'application est la reconstruction automatique de bâtiments 3D à partir de données cadastrales vectorisées et d'un couple stéréo [35]. A l'intérieur de chaque bâtiment (dont les limites sont données par le cadastre), un ensemble d'hypothèses de plans 3D est généré à partir de la carte des hauteurs de la stéréovision directe. Les plans sont ensuite filtrés avec des contraintes sur la forme cherchée et ils définissent un ensemble de facettes. Un graphe de compatibilité est construit dans lequel un nœud correspond à une facette et une arête indique que les 2 nœuds liés appartiennent à une surface admissible 3D. Les cliques maximales donnent alors un toit possible pour le bâtiment (la recherche se fait par une méthode hybride, non exhaustive). Une fonction de mérite prenant en compte l'adéquation de la surface aux hauteurs mesurées et la complexité de la solution permet ensuite de choisir la solution. La figure 2.6 montre des exemples de bâtiments reconstruits.

Dans le même ordre d'idée, on peut utiliser des grammaires de graphes afin de générer les occurrences possibles d'un objet ou analyser des données [7] [21].

2.3 Appariement de graphes

De nombreux problèmes de traitement d'images peuvent se poser en termes d'appariement de deux graphes : un graphe modèle de référence représentant l'objet cherché et un graphe de données déduit de l'image à analyser. C'est par exemple le cas en imagerie médicale où on dispose d'un atlas anatomique avec lequel on cherche à mettre l'image en correspondance ; ou en imagerie aérienne ou satellitaire avec une carte ou un système d'information géographique ; ou encore en reconnaissance d'objets (par exemple des caractères) où un modèle de l'objet à détecter va être construit. Plusieurs degrés de difficultés peuvent être pris en compte. Dans le cas le plus simple (le plus contraint), on cherche un isomorphisme entre les deux graphes, ou plus généralement un isomorphisme de sous-graphes [26] [79]. En effet, l'acquisition de l'image entraîne souvent des parties cachées de l'objet (voir le cours sur les graphes d'aspects) et donc des parties manquantes dans le graphe des données par rapport au graphe modèle. Dans la réalité, le graphe des données est souvent bruité et le graphe modèle éventuellement incomplet, on cherche alors un isomorphisme de sous-graphes avec tolérance d'erreurs [61] [20] [70] [59] [78] [19]. Dans ce dernier cas, on peut utiliser des algorithmes exacts (lorsque le nombre de nœuds est petit) ou des algorithmes approximatifs.

¹Soit S un nœud de l'arbre de recherche T , et soit x le premier fils de S à être exploré. Si tous les sous-arbres de $S \cup \{x\}$ ont été générés, il suffit d'explorer les fils de S qui ne sont pas adjacents à x .

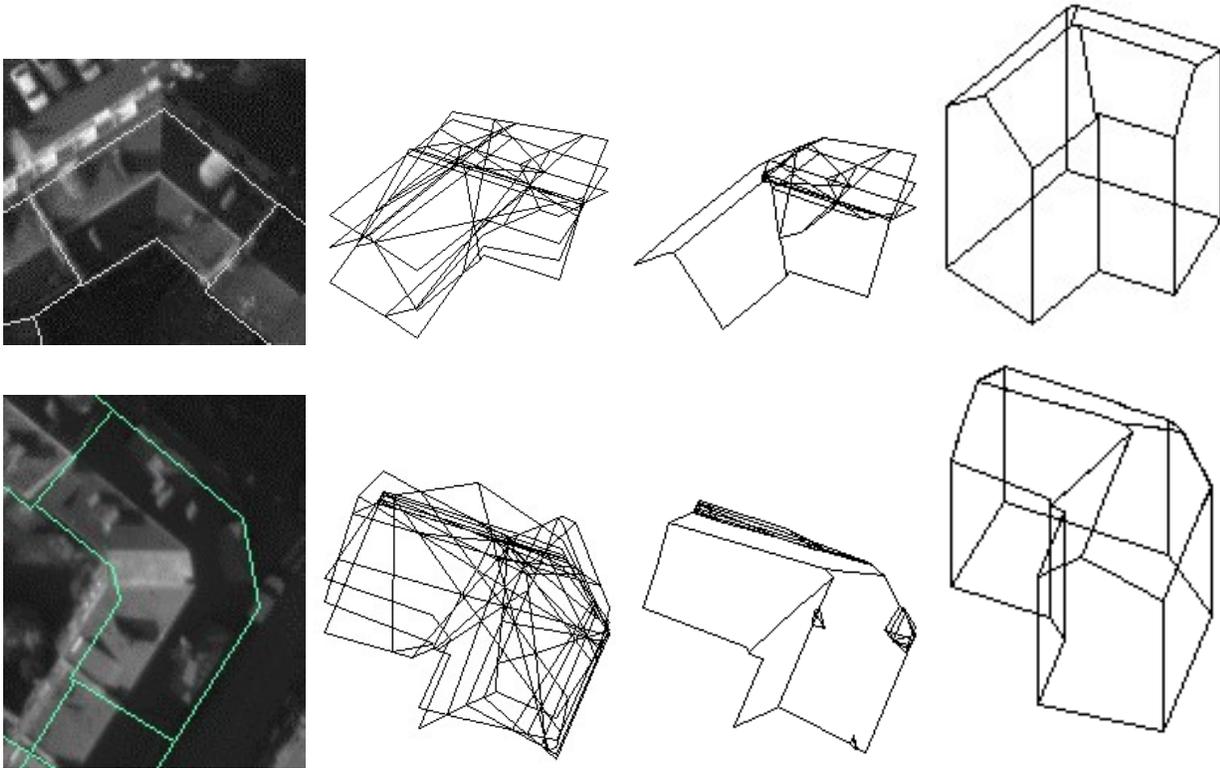


FIG. 2.6 – Détection du bâti par recherche de la clique maximale [35]. De gauche à droite : image originale et relevé cadastral associé ; hypothèses de plans générées ; solution obtenue après heuristique ; solution finale de reconstruction automatique.

2.3.1 Isomorphismes de sous-graphes

Rappelons qu'un isomorphisme de graphe est une fonction bijective $f, f : G \rightarrow G'$ telle que :

- $\forall x \in X, \exists !x' = f(x) \in X'$
- $\forall e = (x_1, x_2), \exists e' = (f(x_1), f(x_2))$

et plus généralement dans le cas où les graphes sont attribués :

- $\mu(x) = \mu'(f(x))$
- $\forall e = (x_1, x_2), \exists e' = (f(x_1), f(x_2)) / \nu(e) = \nu'(e')$ et réciproquement,

où μ et ν (μ' et ν') sont les fonctions d'attribution des arcs et des nœuds de G et G' .

On trouve deux définitions pour l'isomorphisme de sous-graphes entre deux graphes G_1 et G_2 ($|G_1| > |G_2|$). Dans la plus stricte, on cherche un isomorphisme entre un sous-graphe de G_1 et G_2 alors que dans l'autre, on considère également un sous-graphe de G_2 .

Le problème de la recherche d'un isomorphisme de sous-graphes est un problème NP-complet², excepté dans le cas où les graphes sont planaires où la complexité devient polynomiale (cette situation est assez fréquente en traitement d'images). Il existe deux grands algorithmes de recherche. Le premier consiste à construire un graphe d'association G_a , où

²Pour l'isomorphisme de graphes, la question reste ouverte de savoir s'il s'agit d'un problème P ou NP-complet.

chaque sommet correspond à l'association entre deux sommets de même attribut et chaque arc correspond à des associations de sommets compatibles (i.e reliés dans les graphes d'origine G_1 et G_2 par des arcs de même attribut le cas échéant). Chaque clique du graphe d'appariement correspond alors à un isomorphisme de sous-graphe (au sens large). On recherche dans G_a la plus grande ou la "meilleure" clique maximale au sens d'un critère donné (un exemple de graphe d'appariement est illustré figure 2.7). Un exemple pour de la mise en correspondance en stéréo-vision est donné dans [31].

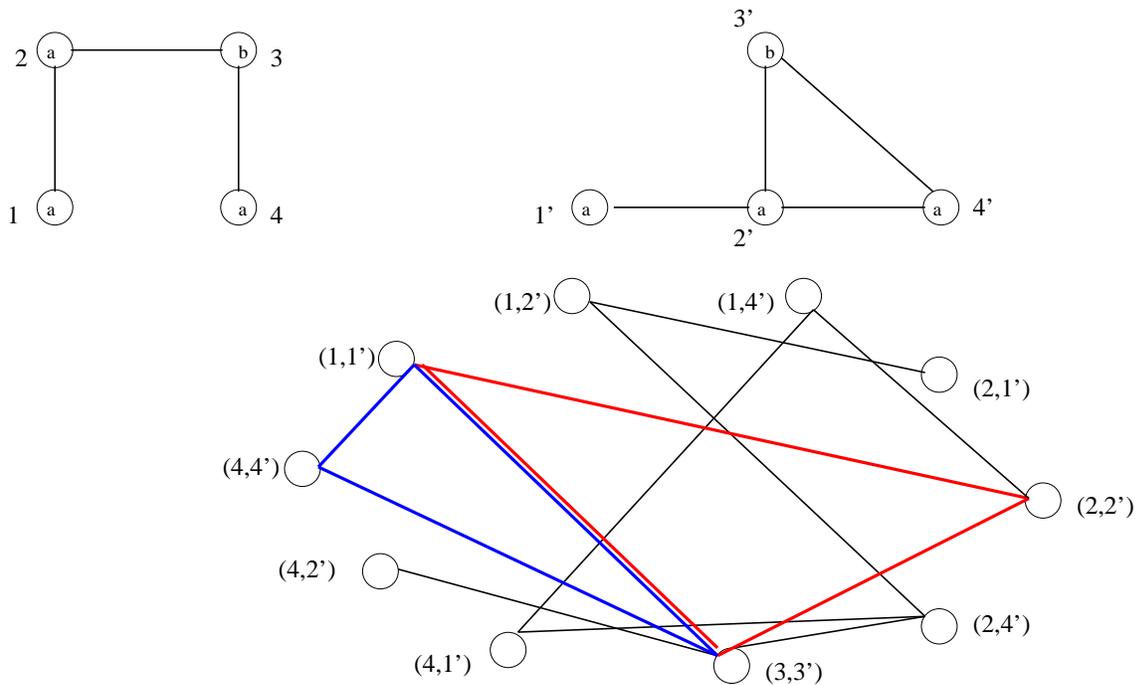


FIG. 2.7 – Deux graphes et leur graphe d'association. Les deux plus grandes cliques maximales sont indiquées en gras (trait plein et en pointillés).

Un autre algorithme est l'algorithme de Ullman [72] qui consiste à construire un arbre de décision, chaque feuille correspondant à un ensemble d'associations compatibles. En cas d'échec (cas d'isomorphisme de sous-graphes strict ici), on remonte dans l'arbre pour essayer une nouvelle association. Pour accélérer la recherche, une matrice d'association permet de vérifier s'il existe des appariements possibles pour tous les sommets restants. La complexité est dans le pire cas en $O(m^n n^2)$ (n ordre de G et m de G' avec $n < m$).

2.3.2 Isomorphismes de sous-graphes avec tolérance d'erreurs

Dans le monde réel, les images et les graphes qui en sont déduits sont bien sûr sujets à de multiples distortions et donc bruités (au niveau des nœuds comme à celui des arcs) et incomplets. L'idée est donc de définir une distance entre graphes et de rechercher le sous-graphe du graphe des données à distance minimale du graphe de référence G . On parle dans ce cas d'isomorphisme de sous-graphes avec tolérance (ou correction) d'erreurs ou d'isomorphisme inexact.

Il existe des algorithmes optimaux (l'algorithme A^* et ses variantes [50]) qui assurent une solution exacte au prix d'une complexité exponentielle, et de nombreux algorithmes

approximatifs (algorithmes génétiques, recuit simulé, réseaux de neurones, relaxation probabiliste,...). Ils minimisent itérativement une fonction de coût et sont mieux adaptés à de grands graphes car plus rapides, mais leur convergence n'est pas assurée.

Une distance entre graphes souvent utilisée est la distance d'édition. Elle consiste à définir des opérations d'édition et leur coût (substitution de l'attribut d'un sommet, de l'attribut d'un arc, suppression d'un sommet, d'un arc, insertion d'un arc,...). Un graphe édité est un graphe qui a subi une séquence d'opérations d'édition dont le coût est la somme des coûts élémentaires. La distance d'édition est définie comme le coût minimal du graphe édité pour lequel on a un isomorphisme de sous-graphe avec le graphe de référence (cet isomorphisme existe toujours pour la séquence d'édition triviale qui consiste à supprimer tous les sommets de G' et à les remplacer par des sommets de G).

Le principe de l'algorithme A^* est la construction d'un arbre de recherche par appariement successif des sommets avec évaluation de la fonction de coût à chaque état (seuls les états de coûts inférieurs sont ensuite propagés). Une amélioration possible est d'estimer une borne inférieure des coûts futurs pour ne pas propager inutilement des branches (ce qui est fait en associant chaque nœud au nœud le plus proche indépendamment des arcs).

Dans le cas d'une grande base de données de graphes modèles qu'on recherche dans une image, il peut être très intéressant de préconditionner les graphes modèles en les décomposant en sous-graphes communs de sorte à ne pas faire plusieurs fois un même appariement [10].

Cette méthode a en particulier été adaptée dans les travaux de F. Fuchs [25] pour la reconstruction 3D de bâtiments. Il dispose d'un ensemble de graphes 3D de référence (représentant différentes formes de toits possibles, à 2-pans, à 4-pans, etc... et généralisés à l'aide d'une grammaire), qui sont stockés sous forme décomposée en adaptant l'approche de Messmer. Un graphe de données est construit mélangeant des informations polymorphes 2D et 3D (linéaires, planes, ponctuelles). Ce graphe est ensuite associé à tous les graphes de la base avec correction d'erreurs et la meilleure solution est gardée. La figure 2.9 montre un exemple de reconstruction obtenue.

L'une des difficultés majeure de la distance d'édition est la définition des coûts d'édition élémentaires. Ceux-ci sont généralement choisis empiriquement en fonction de l'application visée.

2.3.3 Appariement par algorithmes approximatifs

Dans ce cadre, la fonction de coût utilisée s'écarte d'une distance d'édition. Le problème se rapproche d'ailleurs plus d'un problème d'étiquetage puisque le graphe de référence représente des étiquettes possibles pour les données, entre lesquelles on veut respecter certaines relations. Dans ce cas, plusieurs nœuds du graphe de données peuvent être associés à un nœud du graphe modèle, et on introduit également le nœud vide pour prendre en compte le bruit des données ou l'incomplétude du modèle. Beaucoup de travaux se sont appuyés sur une approche probabiliste, notamment ceux de l'équipe de E. Hancock [76] [16], et de J. Kittler [14]. La solution cherchée maximise alors la probabilité a posteriori de f conditionnellement aux graphes modèle et de données. Les difficultés résident alors dans l'expression des probabilités (plusieurs formes ont été proposées faisant des hypothèses simplificatrices d'indépendance différentes), et dans le schéma d'optimisation (recuit simulé, recuit simulé par champ moyen, méthodes de relaxation avec règles de mise à jour, etc.). Un exemple de ce type d'approche

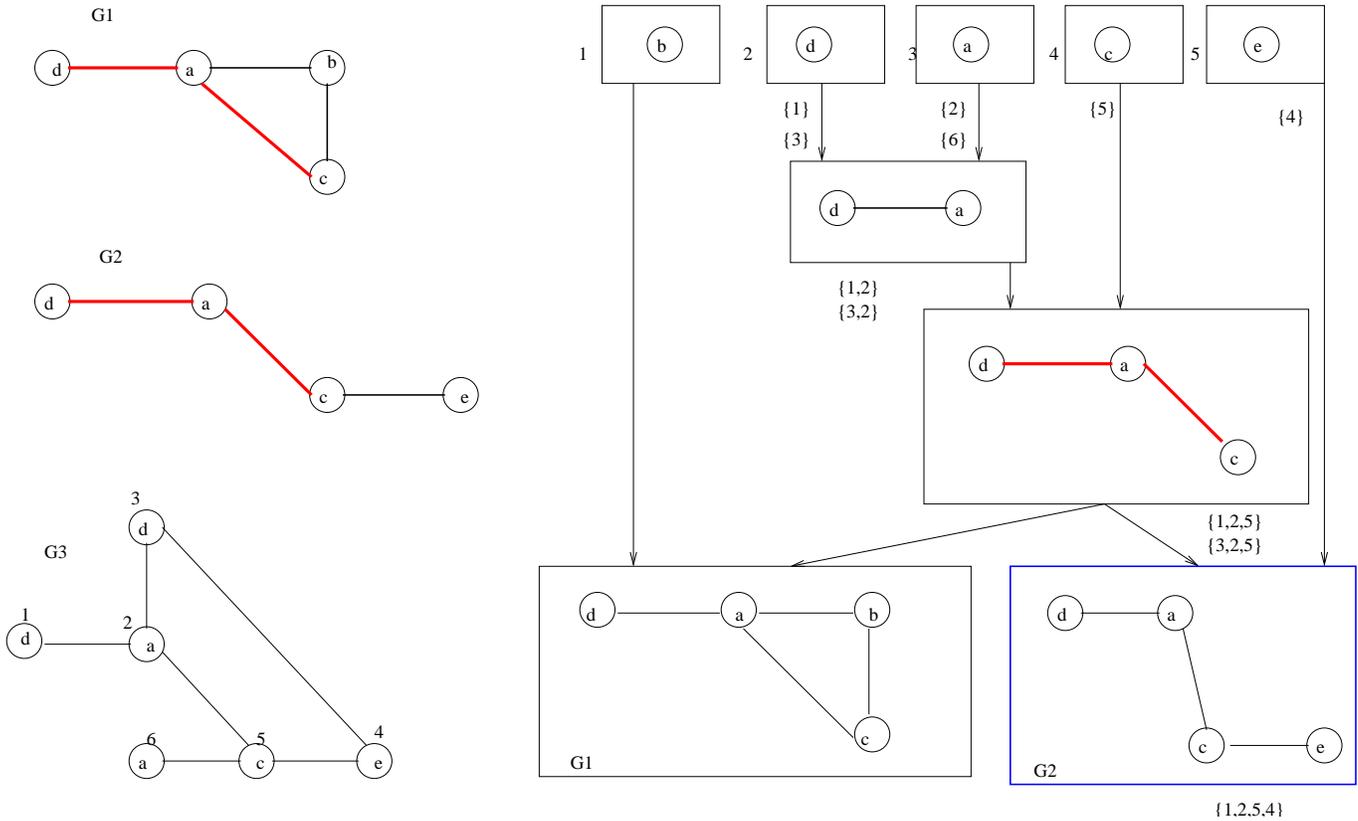


FIG. 2.8 – Exemple de décomposition en éléments communs de deux graphes G_1 et G_2 et des étapes de mise en correspondance avec un graphe G_3 . Une grande partie des opérations d'appariement n'est faite qu'une seule fois puisque le sous-graphe $\{d, a, c\}$ est commun à G_1 et G_2 (thèse de B. Messmer [49].)

est décrit plus en détails ci-dessous. Néanmoins, il n'est pas nécessaire de passer par un formalisme probabiliste. La fonction de coût peut être définie de façon plus intuitive par une combinaison de fonctions de similarité entre les noeuds et les arcs associés. Les algorithmes d'optimisation peuvent être très divers, par exemple des algorithmes génétiques, des algorithmes gloutons, une méthode tabou, ou des algorithmes d'estimation de distributions (EDAs) [4].

Un exemple d'approche probabiliste [14]

Dans ces travaux le problème est formalisé comme un problème d'étiquetage et on cherche à attribuer à chaque nœud du graphe la meilleure étiquette au sens du MAP (local) :

$$\max_{\omega_\lambda \in \Omega} P(\theta_i = \omega_\lambda | x_{j,j \in N_0}, A_{ij,j \in N_i})$$

avec N_0 l'ensemble des nœuds du graphe de données, x_j le vecteur d'attribut du nœud j , N_i l'ensemble des voisins de i et A_{ij} l'attribut de l'arc reliant les nœuds i et j , θ_i le label du nœud i du graphe des données, et Ω l'ensemble des labels du graphe modèle.

On peut alors montrer sous certaines hypothèses d'indépendance que la règle de mise à

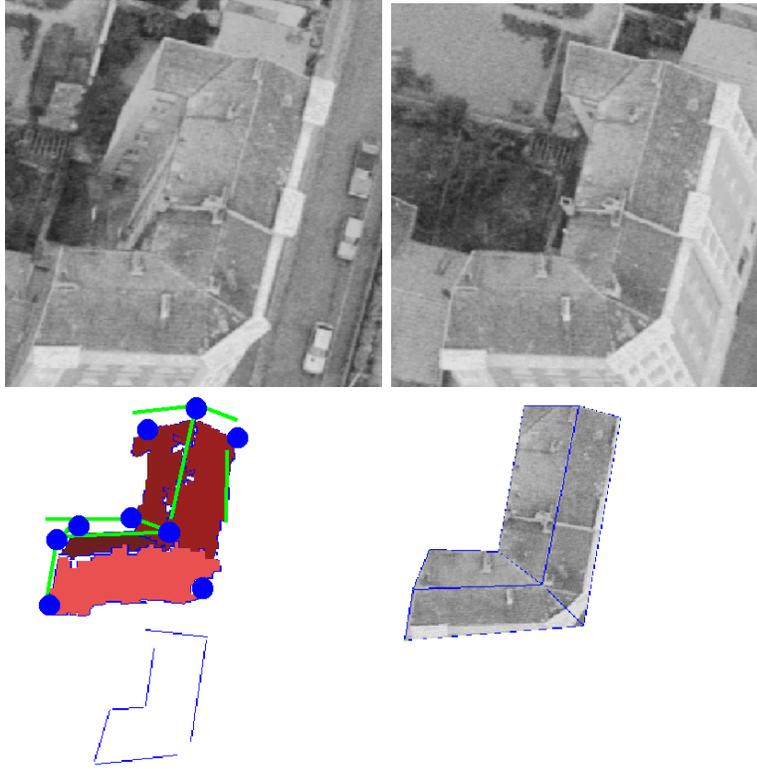


FIG. 2.9 – Exemple de reconstruction 3D par appariement entre un ensemble de graphes modèles et un graphe de données [25].

jour d'un nœud s'écrit :

$$P^{(n+1)}(\theta_i = \omega_{\theta_i}) = \frac{P^{(n)}(\theta_i = \omega_{\theta_i})Q^{(n)}(\theta_i = \omega_{\theta_i})}{\sum_{\omega_\lambda} P^{(n)}(\theta_i = \omega_\lambda)Q^{(n)}(\theta_i = \omega_\lambda)}$$

avec

$$Q^{(n)}(\theta_i = \omega_\alpha) = \prod_{j \in N_i} \sum_{\omega_\beta \in \Omega} P^{(n)}(\theta_j = \omega_\beta) \times p(A_{ij} | \theta_i = \omega_\alpha, \theta_j = \omega_\beta)$$

$Q^{(n)}(\theta_i = \omega_\alpha)$ exprime le soutien des autres objets de la scène et leurs relations binaires pour l'affectation de l'étiquette ω_α au nœud i . La probabilité $p(A_{ij} | \theta_i = \omega_\alpha, \theta_j = \omega_\beta)$ représente un coefficient de compatibilité pour les deux labels $\omega_\alpha, \omega_\beta$ pour le vecteur d'attribut A_{ij} de l'arc (i, j) . Elle est généralement choisie gaussienne. Quant à l'initialisation, elle n'utilise que les attributs des nœuds :

$$P^{(0)}(\theta_i = \omega_{\theta_i}) = P(\theta_i = \omega_{\theta_i} | x_i)$$

La relaxation stochastique s'arrête lorsque les nœuds atteignent des probabilités proches de 1.

Un exemple d’approche par algorithme génétique

Dans ce cas, la fonction de coût peut être définie à partir de fonctions de similarité entre les arcs et les nœuds (indépendamment d’une modélisation probabiliste). On peut par exemple combiner des attributs de niveaux de gris ou texturaux sur les nœuds et des relations géométriques sur les arcs [12]. Soit, D le graphe des données et M le modèle, c_N et c_E sont des fonctions de similarité entre les nœuds et les arcs :

$$\begin{aligned} c_N(a_D, a_M) &= \beta |g_D(a_D) - g_M(a_M)| + (1 - \beta) |w_D(a_D) - w_M(a_M)| \\ c_E((a_D^1, a_D^2), (a_M^1, a_M^2)) &= ||v(a_D^1, a_D^2) - v(a_M^1, a_M^2)|| \end{aligned}$$

avec $g(a)$ le niveau de gris associé au nœud a , w un indice textural, et $v(a_i, a_j)$ le vecteur entre a_i et a_j . La fonction de coût de l’appariement h est alors :

$$f(h) = \frac{\alpha}{|N_D|} \sum_{a_D \in N_D} c_N(a_D, h(a_D)) + \frac{1 - \alpha}{|E_D|} \sum_{(a_D^1, a_D^2) \in E_D} c_E((a_D^1, a_D^2), (h(a_D^1), h(a_D^2)))$$

Le coût $c_N(a_D, \emptyset)$ qui permet d’associer un nœud du graphe de données au nœud vide \emptyset doit être choisi. L’optimisation peut se faire par algorithme génétique. Dans ce cas, une solution correspond à un chromosome, chaque gène correspondant à l’association entre deux nœuds. Dans ce cas comme la présence d’un arc n’est pas explicitement prise en compte dans la solution, la vérification des contraintes sur les arcs -solutions admissibles- se fait a posteriori.

Un exemple de mise en correspondance de structures du visage est présenté sur la figure 2.10. Comme aucun nœud vide n’est introduit dans le modèle, une partie des cheveux à gauche dans l’image est associée au nœud “pupille” du graphe modèle.

2.3.4 Cas des informations topographiques

Un cas très particulier est celui dans lequel les graphes représentent des informations topographiques, c’est à dire dans lesquels les localisations et les positions relatives des nœuds sont porteuses d’informations. Dans ce contexte, la mise en correspondance revient à rechercher une transformation géométrique permettant de passer de l’un à l’autre.

Il peut s’agir d’une transformation rigide plus ou moins complexe (translation, translation + rotation, similitudes -translation + rotation + facteur d’homothétie) ou d’une transformation élastique avec une variabilité sur la position des nœuds du graphe.

Pour les transformation rigides, il existe de nombreuses méthodes de mise en correspondance :

- recherche du maximum de corrélation (pour une translation seulement)
- transformée de Hough (si la taille de l’espace des paramètres n’est pas trop grande)
- méthodes par génération et propagation d’hypothèse [57].

Pour les méthodes élastiques, on commence en général par rechercher une transformation rigide, puis on effectue de petits déplacements aléatoires des sommets en gardant les positions qui diminuent une fonction de coût (méthode de perturbation). Le graphe initial peut être constitué par les nœuds d’une grille fixe (pas en distance constant) ou au contraire par la détection de points d’intérêts particuliers. Les attributs associés aux nœuds sont souvent les coefficients obtenus localement par convolution avec des ondelettes de Gabor, et celui associé aux arcs correspond directement au vecteur géométrique reliant les nœuds. Ces techniques sont très utilisées en reconnaissance de visages [77] [42] [39].

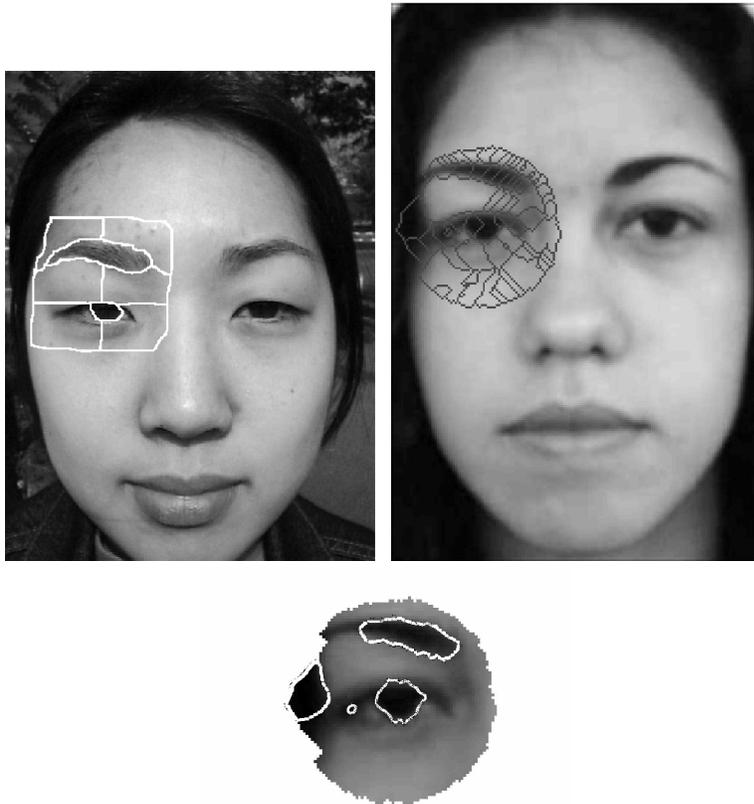


FIG. 2.10 – Graphe modèle (en haut à gauche) construit manuellement -chaque région correspond à un nœud, graphe des données construit par une sur-segmentation de l'image (à droite), résultat de la mise en correspondance (en bas) [12].

2.4 Conclusion

La représentation par graphe est un outil qui est utilisé de façon très diverse en traitement des images. Même si des applications de bas niveau comme la segmentation peuvent tirer avantageusement parti des techniques fournies par la théorie des graphes, leur utilisation est le plus souvent dédiée à l'interprétation d'images et à la reconnaissance des formes par appariement d'un graphe de données et d'un graphe modèle.

Chapitre 3

Des images 2D aux objets à 3D

La détermination d'éléments de géométrie tridimensionnelle d'une scène à partir d'une seule image (donc d'une représentation purement bidimensionnelle) est possible de multiples façons [37]. De nombreux indices conduisent à des informations de position relative, de distance à l'observateur, d'orientation ou de disposition dans l'espace. Chacun de ces indices est souvent insuffisant à une reconstruction exacte et complète de l'espace (contrairement à ce que l'on peut faire par exemple par stéréovision ou triangulation active), mais combinés, ces éléments suffisent amplement à une interprétation élaborée pour satisfaire aux tâches comme la préhension des objets ou l'orientation et le guidage d'un mobile par exemple.

Les effets les plus souvent mis en jeu pour estimer l'information tridimensionnelle sont les suivants :

- la parallaxe, c'est à dire le déplacement relatif des objets en fonction de leur distance à l'observateur et leur masquage mutuel [44] ;
- la perspective, c'est à dire la distorsion de la géométrie apparente en fonction de la distance des divers points, qui entraîne la déformation des objets (par exemple un carré ressemble à un losange), ainsi que la variation apparente des structures homogènes (gradient des textures [65, 66]) ;
- la luminosité qui évolue de deux façons différentes : d'une part par des " effets de brume " (éclaircissement des arrières plans, perte de contraste, éventuellement modification du contenu chromatique), d'autre part par des effets d'orientation (dégradés pour les matériaux diffus, reflets pour les matériaux brillants) ;
- enfin de très nombreux effets culturels associés à une interprétation de la scène comme par exemple la taille apparente des objets, le rôle des ombres projetées, la forme et la position des divers éléments, etc.

Il est très difficile d'isoler le rôle de chacun ces critères dans une situation réelle et il n'est pas du tout prouvé que chacun d'eux soit nécessaire. Néanmoins, un gros effort a été porté depuis les années 70 pour mettre au point des techniques capables de traiter de façon isoler chacun d'eux, au moins dans des expériences de laboratoire où ils sont bien contrôlés.

C'est dans cet esprit que nous présentons ici quelques méthodes et leurs résultats.

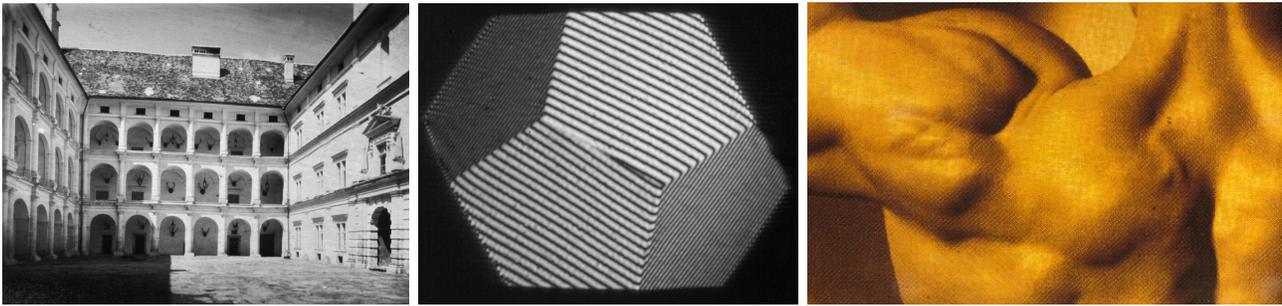


FIG. 3.1 – A gauche, les éléments principaux de la perception du relief sont la perspective et la parallaxe. Au centre, ce sont les gradients de la texture qui donnent l’information de relief. A droite, c’est le modelé des luminances.

3.1 Les graphes d’aspect

Une première famille de méthodes se penche particulièrement sur les objets manufacturés et leur ” apparence ”. L’objectif de ces méthodes est de reconnaître un objet issu d’une famille d’objets déjà connus, et son attitude, c’est à dire son orientation par rapport à l’observateur.

En pratique on se limite aux seuls objets polyédriques et on suppose que l’on a été capable d’extraire de ces objets toutes les arêtes de façon parfaite. A ce titre on pourrait considérer que les méthodes de graphes d’aspects sont une sous-classe des méthodes de lignes caractéristiques que nous verrons au prochain chapitre.

3.1.1 Référentiel, perspective orthographique ou cavalière

On se place toujours dans le cas d’une caméra parfaite (vision sténopique), on attache l’objet à un référentiel 3D (cf. 3.2) et on repère l’observateur par ses deux angles sur la sphère de Gauss.

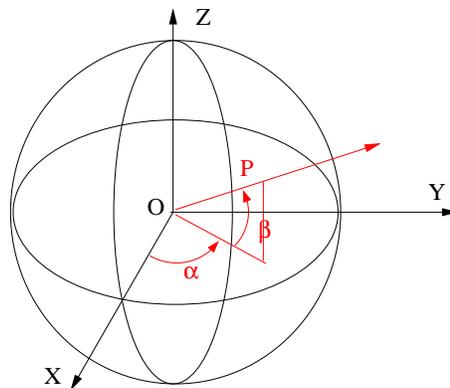


FIG. 3.2 – Le référentiel utilisé est attaché à l’objet. L’observateur est à une distance D dans la direction repérée par α et β

La distance d’observation est une variable importante des problèmes de vision, mais on choisit souvent de l’ignorer, en particulier chaque fois que l’on observe l’objet à grande distance. On se place alors dans la situation de ”perspective orthographique”, c’est à dire

d'observation à l'infini. Dans cette configuration les lignes parallèles de l'objet apparaissent parallèles sur l'image, ce qui n'est pas le cas général dans une projection perspective à distance finie, dite "projection cavalière". Le choix d'un graphe d'aspect en perspective orthographique est souvent plus simple qu'un graphe d'aspect en perspective cavalière¹ (cf. Figure 3.3).

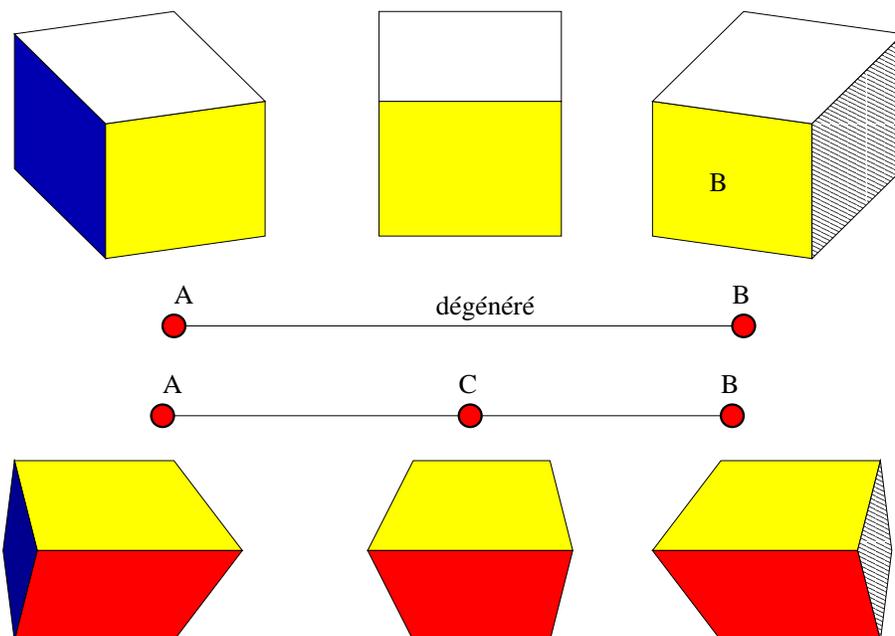


FIG. 3.3 – Dans une projection orthographique de ce cube (en haut), la position où les faces latérales du cube ne sont pas vues, correspond à une position unique de la sphère à l'infini. Ce n'est pas le cas de la projection perspective (ou cavalière) (en bas) pour laquelle il existe une famille plus large de telles positions.

3.1.2 Définitions

L'idée des graphes d'aspects est issue des travaux pionniers de Koenderink et van Doorn qui, dès 1977 ont proposé cette représentation pour expliquer nos capacités de reconnaissance des formes. Elle concerne les objets polyédriques à volume connexe (1 seule composante par objet) et à faces opaques (pas de transparence). L'objet est traditionnellement décomposé en une structures de faces, chaque face étant constituée d'un plan et des arêtes qui limitent ce plan. Les arêtes sont listées de façon à définir une normale extérieure unique à la facette.

$$\text{objet polyédrique} = \{\text{faces}\} = \{\text{plans} + \{\text{arêtes}\}\}$$

On appelle **graphe objet** le graphe ayant pour nœuds les sommets du polyèdre et pour arcs ses arêtes.

Une image, après le processus de détection que nous avons supposé parfait, est alors constituée de la projection (orthographique ou cavalière) des arêtes ou des portions d'arêtes visibles. La projection d'une arête ou d'une portion d'arête est appelée ligne de l'image. Une jonction est l'intersection d'au moins 2 lignes de l'image.

¹L'adverbe « souvent » exprime ici le fait que de nombreux objets manufacturés ont des faces parallèles.

On définit un **graphe image** comme le graphe dont les nœuds sont les jonctions et les arcs les lignes. Tout objet vérifiant les propriétés énoncées plus haut produit, de quelque point d'observation que l'on le regarde, un graphe de polyèdre planaire².

On appelle **graphe topologie image** ou GTI, le graphe obtenu en représentant chaque face vue dans l'image par un nœud et chaque ligne par un arc (fig. 3.4).

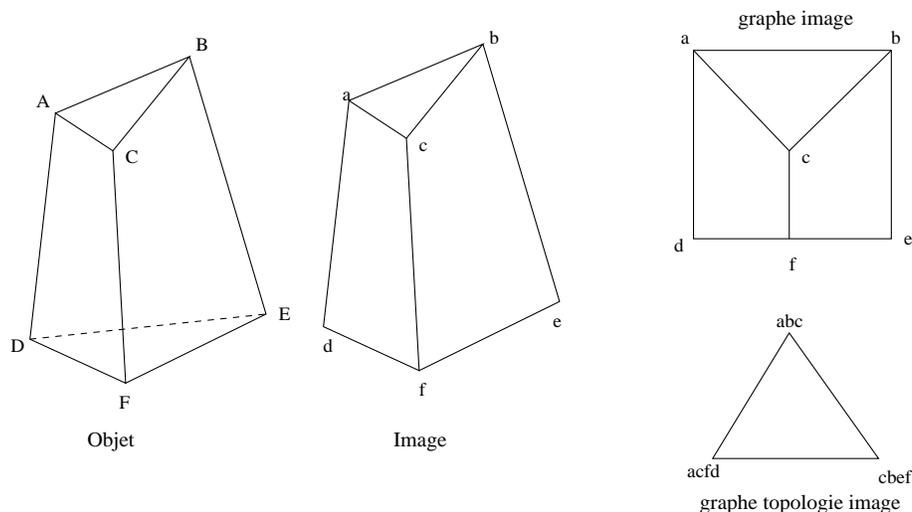


FIG. 3.4 – Partant de l'objet, on construit l'image par projection. Le graphe image en est déduit et par dualité le GTI. Ici les graphes sont limités aux faces vues à la différence de ceux représentés en figure 3.5

Dans l'approche proposée par Koenderinck et van Doorn, deux images sont équivalentes si leurs graphes sont isomorphes (c'est à dire que l'on ne s'intéresse plus qu'à leur topologie et non à la métrique sous-jacente). Deux images équivalentes seront probablement obtenues par un déplacement faible de l'observateur.

On appelle **aspect** une classe d'équivalence des graphes d'images.

Le nombre des aspects d'un objet lorsque l'observateur décrit toutes les positions d'observation (c'est à dire toute la sphère de Gauss) est généralement de l'ordre du carré du nombre de faces. Par exemple, si le polyèdre est convexe et général, le nombre d'aspect est : $\mathcal{O} = n^2 - n$.

On appelle **graphe d'aspect** le graphe qui regroupe tous les aspects de l'objet. Nous allons voir tout d'abord comment on le calcule connaissant l'objet, puis nous présenterons la façon de reconnaître l'aspect d'un objet connaissant son graphe image (fig. 3.5).

3.1.3 Visibilité

Soit \vec{d} la normale à une face dirigée vers l'extérieur et \vec{p} la direction d'observation. Si $\vec{p} \cdot \vec{d} > 0$, l'observateur peut voir la face, à la condition qu'elle ne soit pas cachée par une autre partie de l'objet. Si $\vec{p} \cdot \vec{d} < 0$, l'observateur ne verra jamais la face. La condition $\vec{p} \cdot \vec{d} = 0$ indique que l'observateur est dans le plan de la face, ce sera une situation limite du graphe

²c'est à dire dont les arêtes ne s'intersectent pas.

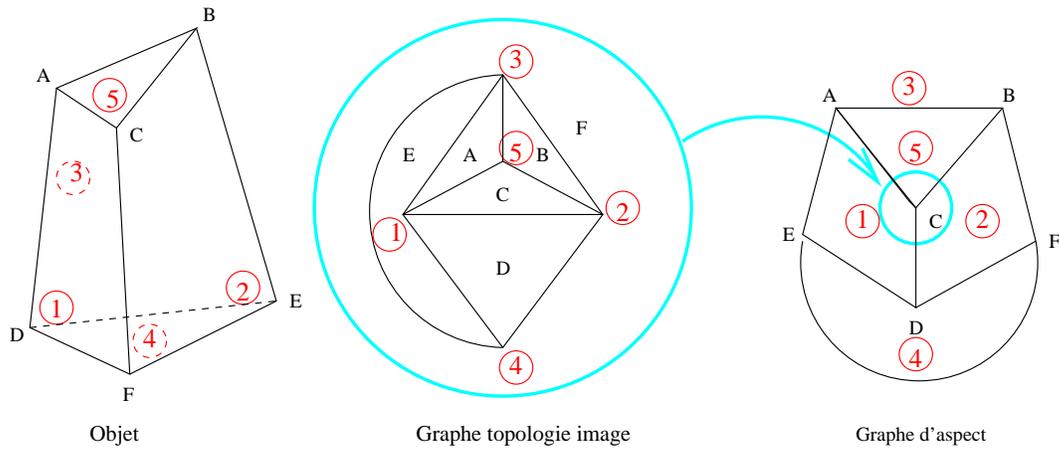


FIG. 3.5 – Le graphe d’aspect peut être vu comme le graphe dual du graphe topologie image du même objet mais sans parties cachées. L’aspect représenté à gauche est porté par le nœud C du graphe d’aspect.

d’aspect. Cette condition est attachée à la face elle-même et détermine un cercle sur la sphère de Gauss (fig. 3.6).

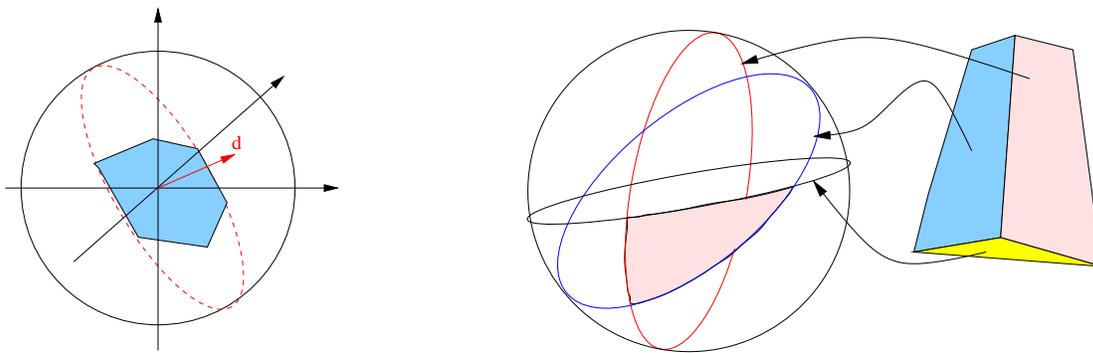


FIG. 3.6 – A gauche : pour une seule facette, Le grand cercle obtenu par intersection du plan contenant la facette et la sphère de Gauss définit les deux sous-espaces de visibilité de la facette. A droite, pour un polyèdre convexe, chaque aspect correspond à une cellule de la sphère.

Dans le cas d’un polyèdre convexe, il est alors très simple de construire le graphe d’aspect. Pour un tel polyèdre, la condition $\vec{p} \cdot \vec{d} > 0$ est nécessaire et suffisante à la visibilité de la face et la sphère de Gauss est alors découpée par autant de grands-cercles³. Les aspects se déduisent de ces cercles. Une zone connexe de la sphère de Gauss est un aspect, lorsqu’on se déplace sur la sphère, si l’on traverse un grand-cercle, on fait disparaître l’une des faces de l’aspect. Si l’on remonte le long d’un grand-cercle, à l’un des sommets, on a fait disparaître simultanément 2 faces (cf. fig. 3.7).

Dans le cas d’un polyèdre \mathcal{P} non-convexe, la situation est plus complexe, en effet, l’aspect dépend non seulement des faces vues, mais également de la géométrie des parties partie-

³Rappelons que nous observons l’objet en perspective orthographique, en perspective cavalière, ces grands-cercles seraient des cercles quelconques.

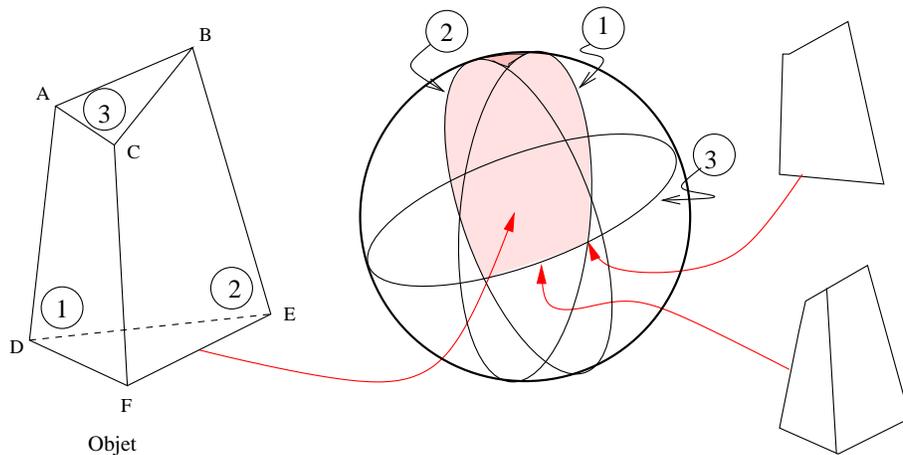


FIG. 3.7 – La sphère de Gauss et les aspects associés à la pyramide tronquée de gauche. L’aspect représenté à gauche appartient au triangle marqué. L’association des 2 quadrilatères (à droite) correspond à un point de vue sur le grand cercle 3. La représentation par un seul quadrilatère (à droite en haut) correspond au point unique à l’intersection des cercles 3 et 2.

ment masquées (cf. fig. 3.8).

On considère alors le polyèdre convexe \mathcal{Q} englobant le polyèdre \mathcal{P} . \mathcal{Q} partage un certain nombre de faces avec \mathcal{P} . Ces faces seront soit entièrement vues soit entièrement cachées. les seules faces posant problèmes sont donc celles qui n’appartiennent pas au polyèdre convexe.

On définit une concavité de l’objet comme une composante connexe de l’enveloppe convexe moins l’objet (cf. fig. 3.8).

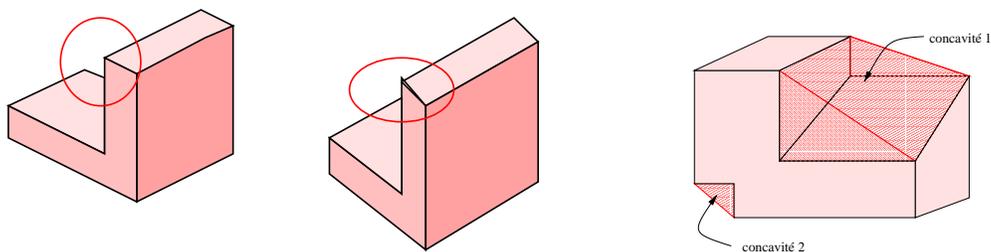


FIG. 3.8 – La visibilité des zones non-convexes d’un objet non-convexe peut varier en fonction de la position d’observation, même sans traverser de face de l’objet (à gauche). Pour un objet non-convexe, on appelle concavité tout volume connexe compris entre l’objet et son enveloppe convexe. Cet objet comporte 2 concavités (à droite).

On montre que la visibilité des faces d’une concavité n’est pas affectée par les faces des autres concavité. On traite donc séparément chaque concavité.

3.1.4 Les cas compliqués

Un coin et une ligne

Une première famille de situations, assez simples, correspond à l'occlusion d'un coin par une ligne ou d'une ligne par un coin (voir figure 3.9). Dans ces cas de figures, un nouveau grand cercle est introduit dans le graphe d'aspects, correspondant à une facette virtuelle portée par le plan contenant le coin et l'arête. Il y a autant de tels plans que de couples "coin-arête" n'appartenant pas à une même face de l'objet dans la cavité.

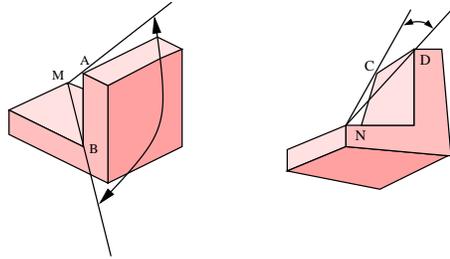


FIG. 3.9 – A gauche, l'arête AB est amenée à masquer le coin M selon le déplacement de l'observateur de part et d'autre du plan MAB. A droite, c'est le coin N qui peut masquer l'arête CD.

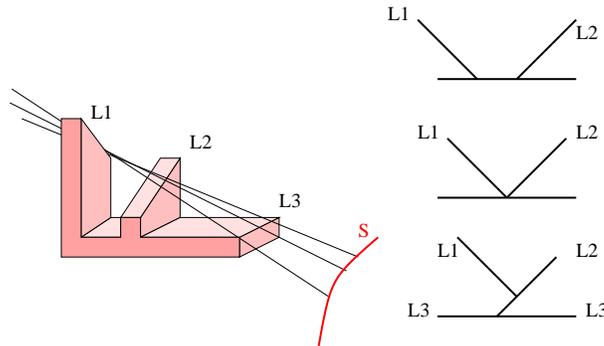


FIG. 3.10 – Dans le cas de 3 arêtes non-coplanaires, les configurations de masquage sont complexes et la surface d'intersection S ne peut être déterminée simplement.

Trois arêtes non coplanaires

Cette configuration est plus complexe. Elle est décrite sur la figure 3.10. lorsque l'observateur se déplace autour de l'objet, il observe des configurations variées. La limite de ces configurations est obtenue par l'intersection de la nappe réglée qui s'appuie simultanément sur deux des surfaces avec la sphère. Cette surface est un parabolôïde hyperbolique et l'intersection est une fonction de degré 4 qui ne peut être connue que point à point. Ce sont de telles courbes qui subdivisent à nouveau les domaines déjà existants.

3.1.5 Reconnaissance d'un objet

La construction du graphe d'aspect d'un objet par la démarche précédente est une opération longue car un polyèdre de n faces a en général $\mathcal{O}(n^2)$ régions. On garde alors pour chaque région l'aspect correspondant à chaque domaine. Observant un objet, on détecte les arêtes, on reconstruit alors l'aspect vu et on identifie l'aspect à l'un des aspects stockés. Cette opération même de comparaison est lourde puisqu'elle consiste en un isomorphisme de graphes (graphes planaires heureusement). De plus elle doit être répétée de nombreuses fois puisqu'à priori tous les aspects doivent être testés. Si l'on a de nombreux objets dans la base, l'étape de reconnaissance devient tout à fait impraticable.

Pour la simplifier, on utilise souvent des heuristiques qui consistent à recenser des configurations particulières qui seront discriminantes. Ce sont par exemple les coin à angle droit qui se traduiront par des apparences de "L", de "T", ou de "Y". Ces structures sont elles mêmes assemblées si besoin dans des structures plus complexes, par exemple : "un T compris entre 2 L", et ainsi de suite.

La reconnaissance d'un objet se fait alors en "lisant" le descriptif littéral de l'aspect et en y recherchant les attributs les plus discriminants. On tire bénéfice à ce niveau d'algorithmes d'exploration de graphes, ou de techniques inspirées des grammaires et des automates finis.

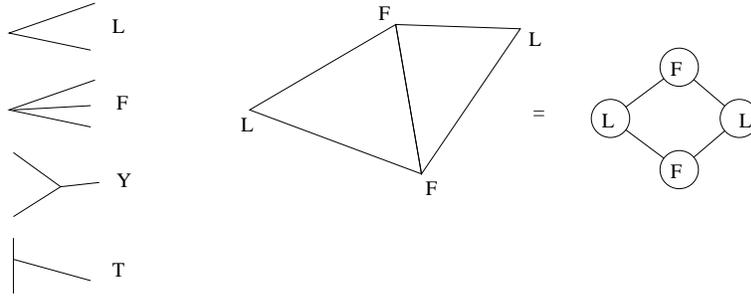


FIG. 3.11 – Pour simplifier la reconnaissance, on utilise les primitives de base de gauche et l'on décompose l'objet selon un graphe simplifié à droite.

3.1.6 Stockage des graphes

Un autre problème délicat est l'archivage des graphes d'aspects et des sphères de visibilité de ces graphes. Il n'est généralement pas possible d'archiver la représentation sous forme de liste de contraintes par les équations analytiques des lignes de séparation des aspects. On préfère souvent adopter un étiquetage de la sphère. Cela nécessite tout d'abord de choisir un paramétrage (angles d'Euler θ et ϕ généralement, mais parfois z et θ), puis un échantillonnage. La discrétisation de la sphère s'accompagne :

1. toujours d'une imprécision sur la position des frontières,
2. parfois d'une perte des aspects dont le domaine de visibilité est très réduit (inférieur au pas d'échantillonnage).
3. et donc dans ce dernier cas d'incohérence dans les transitions entre aspects.

L'échantillonnage de la sphère de visibilité n'est souvent pas très commode. On remplace alors la sphère par un cube de visibilité dont l'échantillonnage régulier est beaucoup plus

simple. Ceci entraîne cependant une inhomogénéité de la résolution en fonction de la direction d'observation qu'il faut pouvoir prendre en compte.

3.1.7 Extension aux objets non polyédriques

L'extension des graphes d'aspects aux objets non polyédriques est assez délicate et n'a pas conduit aujourd'hui à des applications vraiment convaincantes. L'idée de base consiste à remplacer les arêtes des graphes d'aspects par les lignes singulières (voir section 3.2) : lignes de maximum de courbure extrême, lignes paraboliques, silhouettes et limbes. Les arêtes des polyèdres comme lignes de maximum de courbures maximales entrent bien dans cette généralisation. Des travaux ont été conduits dans cette direction par van Effelterre sur des tores et par Ponce et al. sur des surfaces algébriques d'équations relativement simples. Néanmoins, la complexité de telles approches est prohibitive (pour une surface algébrique de degré d , la complexité de stockage est en $\mathcal{O}(d^{12})$!

3.2 Les lignes caractéristiques

3.2.1 Les silhouettes

La ligne caractéristique la plus importante d'un objet est sa **silhouette**. C'est une ligne généralement virtuelle de l'objet en tous les points de laquelle les rayons issus de l'observateur sont tangents à l'objet et ne rencontrent plus l'objet. Par projection, les silhouettes créent les contours de l'objet dans l'image. Il y a généralement une discontinuité de l'image perpendiculairement au contour et souvent continuité à l'intérieur du contour le long du contour (voir figure 3.12), voir [38].

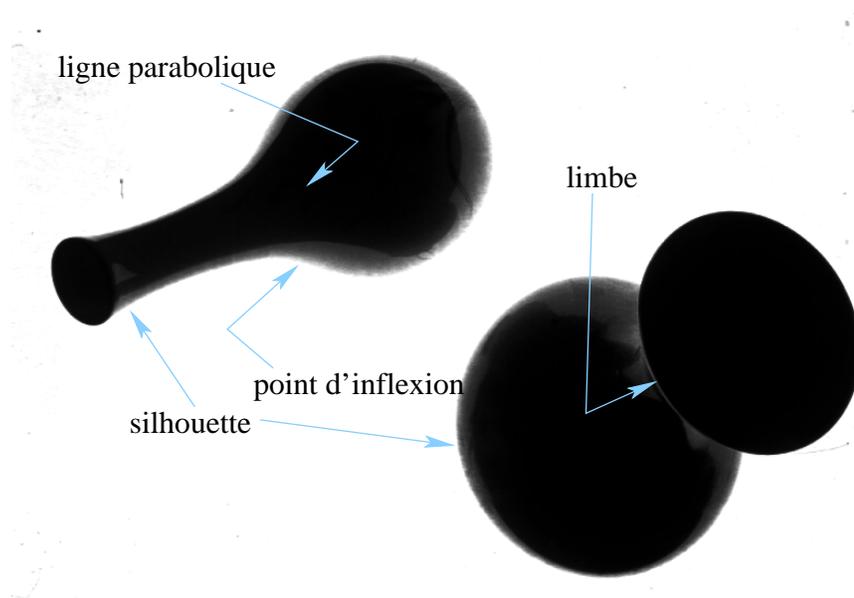


FIG. 3.12 – Les lignes caractéristiques.

Lorsque le rayon issu de l'observateur est tangent à l'objet puis rencontre de nouveau l'objet, il crée un **limbe**. En projection, un limbe donne souvent un "contour interne" dans le plan image. Il y a souvent discontinuité de l'intensité à la traversée des contours internes.

"Silhouettes" et "Limbes" sont donc généralement détectables à partir d'une image. Mais ces lignes ne sont pas nécessairement des lignes particulières de l'objet, ce sont souvent des courbes tridimensionnelles de la surface de l'objet et leur connaissance ne nous informe pas beaucoup sur la forme de l'objet.

Des silhouettes 2D aux courbures 3D

Lorsque la surface de l'objet est deux fois différentiable, elle admet en chaque point une courbure gaussienne, produit des 2 courbures principales ou produit des 2 courbures mesurées selon un repère orthogonal tangent à la surface (cf. cours d'ANIM : [6] chapitre 15.3). La courbure gaussienne K_g est un invariant du repère. A partir de la courbure K_{2D} mesurée dans l'image le long de la silhouette, on peut déduire l'aspect de l'objet et le signe de la courbure radiale K_{rad} mesurée le long du rayon d'observation par la formule :

$$K_{2D} = \frac{dK_g}{K_{rad}}$$

où d est le grandissement optique.

- si $K_{2D} > 0$ alors la surface est localement elliptique convexe car la surface est convexe, or la courbure gaussienne ne peut être que positive car si K_{rad} était négative la surface ne serait pas vue, ce qui indique que K_g est bien positif.
- si $K_{2D} < 0$ alors la surface est localement hyperbolique et la silhouette concave.
- si $K_{2D} = 0$ alors on est situé sur une ligne parabolique (de courbure gaussienne nulle) de l'objet.

L'ensemble de ces remarques conduit à interpréter certaines images par des modèles localement simples (voir figure 3.13).

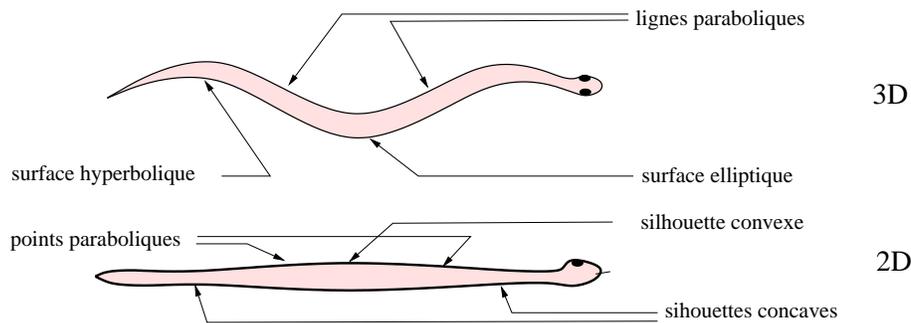


FIG. 3.13 – Interprétation des courbures à partir des silhouettes lors du passage de 3D à 2D.

Les silhouettes des objets de révolution

Les objets de révolution forment une classe particulière d'objets dont on peut apprendre beaucoup par l'analyse de leur silhouette.

Tout d'abord il est important de constater qu'un objet de révolution autour d'un axe droit (donc un objet symétrique dans tout plan axial de section) ne se projette généralement

pas selon une figure symétrique dans le plan de l'image (cf. figure 3.14). Ceci contrarie notre expérience car un observateur, confronté à l'examen d'un tel objet, le place au centre de son champ de visée et se met donc dans une position particulière où la symétrie est conservée.

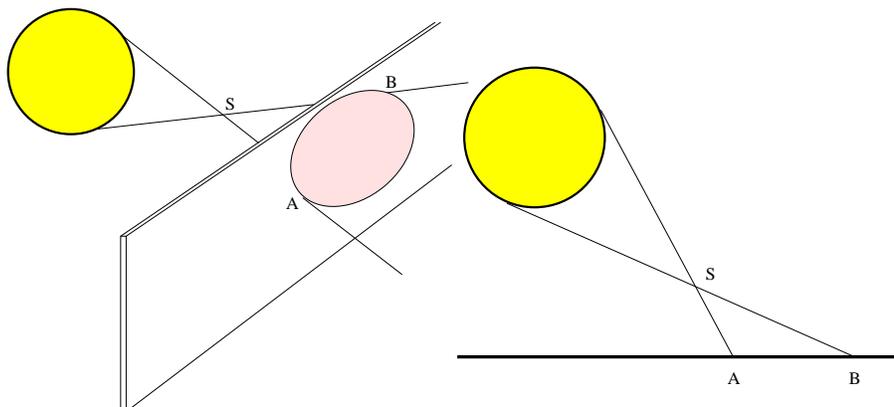


FIG. 3.14 – Lors d'une projection, l'image d'une sphère n'a généralement pas une symétrie de révolution.

La première chose que l'on détermine aisément à partir des silhouettes est la projection de l'axe de révolution de l'objet. On utilise pour cela le fait que les lignes de courbure gaussienne nulle des objets de révolution sont des cercles et que les plans tangents en ces lignes ont pour enveloppes des cônes dont le sommet se trouve sur l'axe de révolution. Par suite, dans l'image, par deux points de silhouette de courbure nulle correspondant à un même grand cercle les tangentes se coupent sur la projection de l'axe de rotation.

La détection des points de courbure nulle est généralement aisée, ainsi que la détermination des tangentes en ces points (cf. figure 3.15). Comme on ne sait généralement pas appairer automatiquement les paires de points des silhouettes droite et gauche, on procède par transformation de Hough [6] (c'est-à-dire par une technique de vote) ou par une démarche combinatoire exhaustive (Ponce et Chelberg, 87).

Un autre problème d'intérêt est la détection de l'attitude de l'objet (en l'occurrence la détermination de l'orientation de l'axe de l'objet dans le repère de la caméra). L'image de l'axe (obtenue comme ci-dessus) permet de déterminer le plan contenant l'axe. Si l'on dispose de l'image d'une section de l'objet, on peut émettre des hypothèses complémentaires sur cet axe. Une section d'objet de révolution est un cercle dont la projection est une ellipse. A partir de la connaissance de l'ellipse on détermine 4 hypothèses correspondant à 2 cercles se projetant selon cette ellipse et pour chacun à deux objets (l'un au dessus, l'autre au dessous du cercle). Les plans contenant les cercles définissent l'information manquante sur l'orientation de l'axe de l'objet (1 juste, 3 fausses que l'on sépare généralement par des tests sur l'objet), on obtient ces plans en déterminant les plans cycliques du cône (diagonalisation de la matrice représentant l'ellipse) (Dhome et Richetin - 90).

Si l'on dispose de 3 vues d'un objet de révolution, on peut déterminer totalement sa géométrie (Dhome et Richetin 88).

Si l'objet est connu, on peut avec sa seule silhouette déterminer son attitude (2 solutions) (Horaud et Brady 87).

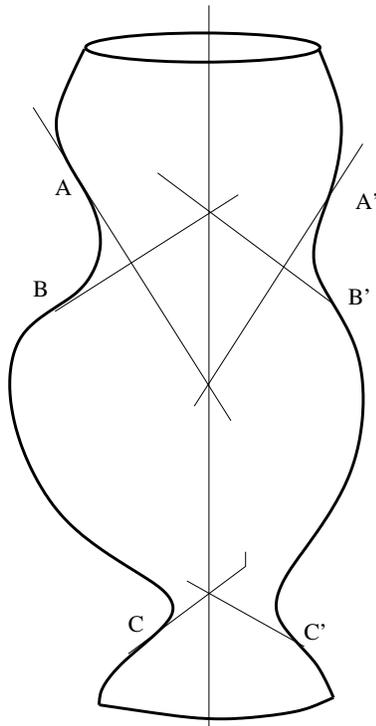


FIG. 3.15 – Un objet de révolution n’a pas une image symétrique par rapport à un axe. mais deux points d’inflexion de même altitude ont, dans l’image, des tangentes qui se coupent sur l’image de l’xe de révolution de l’objet.

Que faire de nombreuses silhouettes ?

Lorsque l’on connaît un grand nombre de silhouettes vues de points de vues différents dont on connaît la position, on peut calculer une **enveloppe visuelle** de l’objet par intersection de tous les cônes s’appuyant sur les silhouettes et issus des centres de projection. L’enveloppe visuelle, introduite par Laurentini [43] a fait l’objet de nombreux travaux [44, 44] aussi bien pour leur calcul rapide [67] que pour leur raffinement [41, 18] à l’aide, en particulier, de multi-stéréovision [30, 60].

3.2.2 Lignes paraboliques, et ligne singulière

Si la surface est localement cylindrique (l’une des deux courbures principales est nulle) ou si la surface est localement plane (les deux courbures principales sont nulles), alors la courbure de la silhouette est nulle (point d’inflexion de la courbure), nous venons de le voir. En dehors des silhouette, une ligne de courbure gaussienne nulle (appelée ligne parabolique de la surface) se projette selon une ligne virtuelle généralement indécélable sur l’image car la luminosité ne subit généralement pas de discontinuité, non plus que sa première dérivée. Néanmoins ces lignes, comme les lignes de maximum de la courbure maximale, sont considérées comme importantes pour décrire la forme d’un objet en projection et constituent l’essentiel des ” lignes de construction ” utilisées par les caricaturistes par exemple.



FIG. 3.16 – Statue africaine : image originale à gauche et enveloppe obtenue à partir de 36 silhouettes par intersection des cônes s'appuyant sur ces silhouettes (merci à F. Schmitt et C. Hernandez).

3.3 La forme à partir de l'ombrage

Dans cette section on étudie l'information apportée par l'évolution de l'intensité apparente de la surface d'un objet homogène en fonction de son orientation par rapport à l'observateur d'une part et par rapport à la source lumineuse d'autre part. Ce domaine est appelé *shape from shading* dans la littérature anglo-saxonne [32]. il a acquis une très grande importance depuis les années 90, mais il a fait l'objet de calculs réguliers depuis plus de 100 ans, en particulier pour déterminer le relief lunaire à partir d'une simple image.

Considérons donc un objet homogène, constitué d'un seul matériau uniforme et considérons-le également éclairé par une source uniforme (par la suite on choisira généralement un point source à l'infini dans une direction fixée). Pour un observateur dans une position donnée, la luminosité d'un point de l'objet sera d'autant plus forte :

- qu'il recevra plus de lumière de la source (donc sera mieux orienté par rapport à elle),
- qu'il en renverra une plus grande fraction vers l'observateur (donc qu'il sera mieux orienté par rapport à lui).

Si l'on connaît la position et l'énergie de la source, si l'on connaît la position de l'observateur et si l'on connaît la courbe de réflectance bidirectionnelle de l'objet (une propriété physique du matériaux qui constitue l'objet), on peut déduire beaucoup de la mesure de l'intensité recueillie en chaque pixel. On constate tout d'abord que l'œil humain est très capable de reconstruire une information tridimensionnelle à partir d'une telle image (cf. fi-

gure 3.17). Cette reconstruction est qualitative mais généralement très pertinente. Elle nous permet sans problème d'évaluer le relief sur une piste de ski par exemple ou sur une plage de sable. Elle peut cependant se faire tromper car des formes " en bosse " ou " en creux " ont généralement la même apparence pour deux sources de lumière symétriques.



FIG. 3.17 – Forme à partir de l'ombrage : A gauche, surface de Mercure vue par la sonde Mariner, au centre un bas-relief de pierre, à droite, un bébé. L'œil interprète aisément le relief à partir des seules ombres.

3.3.1 Mise en équation

Nous allons mettre en équations la forme à partir de l'ombrage dans un cas particulièrement simple auquel on se ramène très souvent : celui d'un éclairage frontal, c'est-à-dire pour lequel la source lumineuse et l'observateur sont confondus (cf. figure 3.18). On note par z la profondeur et par x et y les deux variables d'espace décrivant l'objet. On note par \vec{s} la pente de l'objet au point (x,y) :

$$s = \|\vec{\nabla}z\| = \sqrt{p^2 + q^2}$$

avec :

$$p = \frac{\partial z}{\partial x} \quad \text{et} \quad q = \frac{\partial z}{\partial y}$$

la fonction de réflectance bidirectionnelle $R(\vec{E}, \vec{O}, \vec{N})$, fonction de l'angle d'éclairage \vec{E} , de l'angle d'observation \vec{O} et de la normale locale \vec{N} à l'objet, s'écrit, dans le cas d'un éclairage et d'une observation frontaux, sous la forme très simple :

$$R(\vec{E}, \vec{O}, \vec{N}) = R(\vec{N}) = R(s)$$

A ce point, on choisit généralement une loi de réflectance bidirectionnelle particulière, le plus souvent celle d'un matériau parfaitement diffusant (réflexion lambertienne), régie par la formule :

$$R(\vec{N}) = k|\vec{S} \cdot \vec{N}|$$

où k est un scalaire qui prend en compte les propriétés du matériaux et de l'optique. Nous le ferons à la Section sec :lambert.

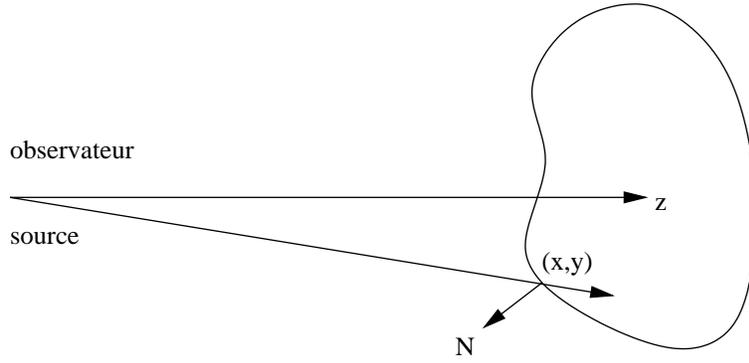


FIG. 3.18 – Forme à partir de l’ombrage. Schéma simplifié de mise en équations.

Comme la surface est éclairée de face :

$$\vec{S} = S \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$$

et \vec{N} vaut :

$$\vec{N} = \frac{1}{\sqrt{1+p^2+q^2}} \begin{pmatrix} -p \\ -q \\ 1 \end{pmatrix}$$

La fonction de réflectance s’écrit donc :

$$R(\vec{N}) = \frac{kS}{\sqrt{1+s^2}} \quad \text{avec } s^2 = p^2 + q^2 \quad (3.1)$$

Cette fonction est maximale lorsque la surface est perpendiculaire à la direction d’éclairage ($p = q = 0$). Elle vaut alors $E_{max} = kS$. On peut donc récrire la fonction d’éclairement comme :

$$R(\vec{N}) = \frac{E_{max}}{\sqrt{1+s^2}}$$

ou :

$$R(p, q) = \frac{E_{max}}{\sqrt{1+p^2+q^2}} \quad (3.2)$$

On reconnaît dans cette écriture un problème aux dérivées partielles de Cauchy.

On remarque que la luminosité est minimale ($E = 0$) pour $s = +\infty$, c’est-à-dire pour $p = \pm\infty$ ou pour $q = \pm\infty$. En ces points, la surface est orthogonale au rayon et nous avons donc affaire à des points de la silhouette de l’objet.

L’équation 3.1 peut s’écrire sous la forme :

$$R(p, q) = f(p^2 + q^2) \quad (3.3)$$

appelée équation eikonale dans la littérature de la physique en particulier (c’est celle qui régit la propagation dans les milieux d’indice variable, comme par exemple la propagation dans

les fibres à indice variable). Si la fonction f est inversible (c'est généralement le cas car c'est souvent une fonction strictement décroissante qui vaut 0 à l'infini), alors on peut écrire :

$$\left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2 = f^{-1}(x, y)$$

La difficulté du SfS (*Shape from Shading*) consiste tout d'abord à s'assurer de l'existence des solutions de ce problème, puis de leur nombre, enfin de proposer des algorithmes pour les calculer. Malheureusement tout ces problèmes sont aujourd'hui encore mal résolus.

3.3.2 Résolution du problème

L'hypothèse lambertienne

En fait nous n'aborderons le SfS qu'à partir de l'hypothèse lambertienne (comme d'ailleurs la majorité des auteurs), c'est-à-dire celle d'un matériau parfaitement diffusant. Sous cette hypothèse, la luminance de l'objet s'exprime à partir du flux incident Φ par la formule :

$$L = \frac{\rho\Phi}{\pi\|\vec{S}\|} |\vec{S} \cdot \vec{N}| \quad (3.4)$$

et l'éclairement est proportionnel à :

$$E \propto k \cdot |\vec{S} \cdot \vec{N}|$$

Avec ces hypothèses, l'équation du SfS devient :

$$p^2 + q^2 = \left(\frac{E_{max}}{E(x, y)}\right)^2 - 1 \quad (3.5)$$

qui est une équation d'Hamilton Jacobi du premier ordre qui se résoud avec des conditions aux limites :

- z est connu sur le contour de l'objet (conditions de Diriclet),
- ou p et q sont connus sur le contour de l'objet (conditions de Neumann).

Il est clair en effet qu'en chaque point de l'objet on ne dispose que d'une seule mesure (la valeur $E(c, y)$) pour déterminer 2 inconnues, p et q . Il est donc important de disposer d'information supplémentaire. Celle-ci est en partie apportée par les conditions aux limites. On y ajoute souvent des contraintes supplémentaires (régularité de la surface par exemple), sans lesquelles le problème n'a pas de solution.

Sous ces hypothèses, le problème continu est parfois bien posé (c'est à dire qu'il peut avoir des solutions en nombre fini). Le problème discret est beaucoup plus difficile à étudier.

Le choix des variables

Plusieurs solutions ont été proposées dans la littérature. Certaines sont des solutions locales qui résolvent le problème à partir d'un point où la solution est connue, par une propagation. D'autres sont des approches globales où l'on recherche une solution en tous les points simultanément. Les solutions diffèrent également par la paramétrisation choisie. On

peut ainsi considérer comme inconnue principale l'altitude z ou les deux composantes p et q . z se déduit alors par intégration. Mais on peut aussi choisir les angles de Gauss ou des lignes trigonométriques de ces angles. l'objectif de ces changements de variables peut varier : on cherche parfois à obtenir des équations linéaires, d'autres-fois des équations découplées. On peut au contraire chercher à éviter des variables pour lesquelles la fonction R pourrait prendre des valeurs indéfinies (c'est le cas de la réflectance sur les silhouettes). Bien sûr ces contraintes sont généralement contradictoires et expliquent la multiplicité des schémas.

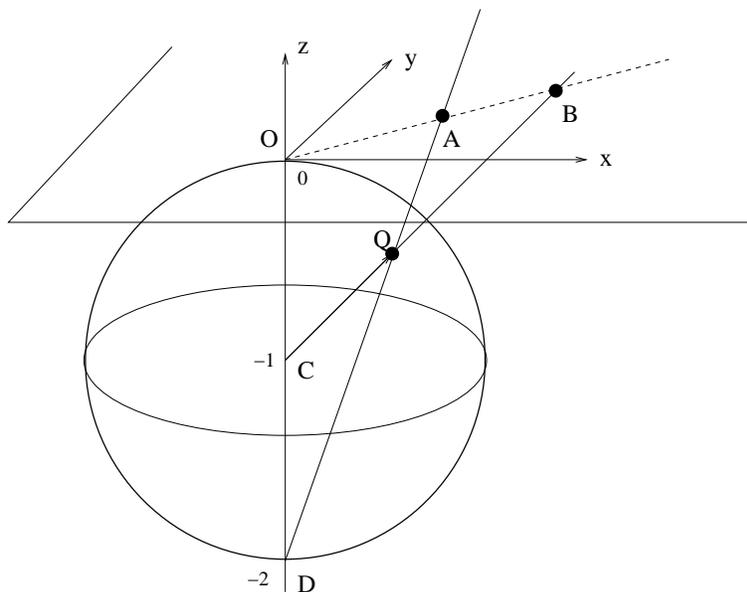


FIG. 3.19 – Paramétrisation des équations du Sfs. La sphère de Gauss est de centre $C(0, 0 - 1)$ et de rayon 1. Le plan est tangent au point $O(0, 0, 0)$. La paramétrisation naturelle de la surface utilise le point Q qui représente le vecteur normal à la surface $\vec{N} = CQ$. D'autres paramétrisations utilisent le point $A(f, g, 0)$ ou le point $B(p, q, 0)$.

La figure 3.19 précise ces variables :

- les coordonnées cartésiennes : x, y, z
- les composantes du vecteur tangent à la surface

$$\begin{cases} p = \frac{\partial z}{\partial x} \\ q = \frac{\partial z}{\partial y} \end{cases}$$

- Le point B :

$$\begin{cases} f = \frac{2p}{1 + \sqrt{1 + p^2 + q^2}} \\ g = \frac{2q}{1 + \sqrt{1 + p^2 + q^2}} \end{cases}$$

- les angles de la normale \vec{N} :

$$\begin{cases} \tau \\ \sigma \end{cases}$$

Solution locale de Pentland

Dans cette solution on choisit pour variables les angles τ et σ de la normale à la surface. On fait l'hypothèse qu'en chaque pixel la surface est localement sphérique (approximation par

la sphère osculatrice). La surface de l'objet est alors classifiée en champs : plans, cylindriques, en selle de cheval, concave ou convexe [54].

On calcule globalement sur un petit voisinage $\tan \tau$, on en déduit $\cos \sigma$. Par intégration on détermine z . Malheureusement, une telle approche très simple ne garantit pas l'intégrabilité de la solution le long d'un circuit fermé en raison du bruit naturellement présent sur le signal et de l'accumulation de petites erreurs le long du parcours d'intégration. La solution trouvée est généralement déchirée aux points où se referme le balayage de l'objet. Par ailleurs, la valeur de $\cos \sigma$ se déduisant par une équation algébrique des données et de la valeur précédemment calculée de $\tan \tau$, n'est pas nécessairement inférieure à 1.

Afin d'améliorer les résultats, on peut procéder à une restauration itérative contextuelle (de type markovien) pour imposer à la solution finale un compromis entre l'adéquation aux mesures locales et la vérification de l'intégrabilité sur des parcours fermés.

Solution locale de Tsai et Shah

Dans cette approche [69], on choisit une paramétrisation par les coordonnées cartésiennes : (x, y, z) et on procède à une linéarisation de R , après sa discrétisation aux éléments finis, par un développement limité. Ainsi, l'équation :

$$f(E_{i,j}, z_{i,j}, z_{i-1,j}, z_{i,j-1}) = 0$$

devient :

$$= R(z_{i,j} - z_{i-1,j}, z_{i,j} - z_{i,j-1}) - E_{i,j}$$

on procède ensuite par résolution itérative de Jacobi (voir [54]) :

$$z_{i,j}^k + 1 = \Phi[f(E_{i,j}, z_{i,j}^k, z_{i-1,j}^{k+1}, z_{i,j-1}^{k+1})]$$

Très simple, cette solution a donné des résultats médiocres dès que les surfaces sont un peu complexes, mais elle est le point de départ de nombreux autres schémas plus complexes et plus robustes.

Solution de Ikeuchi et Horn

Ils choisissent une paramétrisation qui évite les valeurs infinies de p ou q sur les frontières [34] :

$$m = \frac{2p}{1 + \sqrt{1 + p^2 + q^2}} \quad n = \frac{2q}{1 + \sqrt{1 + p^2 + q^2}}$$

sous la contrainte : $m^2 + n^2 \leq 4$. Ils résolvent ensuite une équation intégrale du type :

$$\int \int [(E(x, y) - R(m(x, y), n(x, y)))^2 + \lambda(m_x^2 + m_y^2 + n_x^2 + n_y^2)] dx dy = \min$$

qui conduit, après discrétisation à la résolution itérative d'un grand système d'équations.

Les schémas variationnels sont comparés dans [33].

Solutions continues

Une solution mathématiquement très élégante a été proposée par Lions, Rouy et Tourin [47, 58] utilisant les solutions des équations de viscosité. Ce sont les seules approches qui semblent aujourd’hui avoir reçu une preuve complète de convergence dans leur cas discret, sous certaines hypothèses. Dans cette approche, une solution de viscosité est obtenue comme fonction valeur d’un problème de contrôle optimal attaché aux équations d’Hamilton-Jacobi parmi les fonctions de classe C^1 (il est cependant nécessaire de modifier la notion classique de dérivabilité). Les conditions aux limites (Diriclet ou Neumann) ne sont pas suffisantes pour assurer la convergence. Des preuves d’existence de la solution de l’équation eikonale ont été apportées dans [58], mais aussi l’impossibilité de résoudre le problème en présence d’un seul point singulier (infinité de solution). Par exemple, une sphère sera identique à toutes les formes de la figure 3.20.

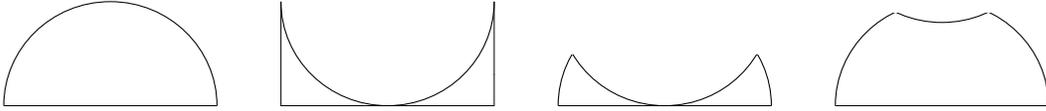


FIG. 3.20 – Toutes ces surfaces auront la même image lorsqu’elles seront observées d’un point sur l’axe.

Une autre famille d’approches consiste à ne s’intéresser qu’aux lignes particulières de l’objet pour lesquelles l’équation du SfS n’a plus qu’une inconnue par point de mesures. C’est la méthode dite des « lignes caractéristiques ». Ces lignes caractéristiques sont généralement issues du point de maximum de luminance (le point en réflexion spéculaire) et sont les lignes de pente maximale.

A partir d’un point A où z_A est connu, on calcule l’altitude d’un point infiniment voisin B par :

$$z_B = z_A + \left(\frac{\partial z}{\partial x}\right)_A dx + \left(\frac{\partial z}{\partial y}\right)_A dy + o(dx, dy)$$

ou par :

$$p_B = p_A + \left(\frac{\partial p}{\partial x}\right)_A dx + \left(\frac{\partial p}{\partial y}\right)_A dy$$

ou par développement en série entière (sous l’hypothèse que la surface est C_∞).

Toutes ces approches rencontrent de très difficiles problèmes lors de la discrétisation.

3.3.3 Conclusions

Le ” shape from shading ” demeure l’un des problèmes très mal résolus de la vision par ordinateur, d’une part parce que sa théorie est mal maîtrisée malgré des recours à des importants outillages mathématiques, d’autre part parce que sa résolution pratique est en butte à des écueils nombreux et imprévisibles. La sensibilité des algorithmes d’inversion au bruit présent dans les données, la discrétisation des signaux, les problèmes aux limites (sur les lignes de silhouette en particulier) se conjuguent à la méconnaissance des fonctions de

réflectance des matériaux, à l'ignorance de la position précise des sources et de l'observateur, à l'inhomogénéité des matériaux, pour mettre en échec beaucoup de reconstructions. Une étude très complète et récente vient faire le point sur l'état des connaissances sur ce sujet [17].

Chapitre 4

Les grammaires en traitement des images

Les méthodes à base de grammaires font partie des méthodes **structurales** de description et d'analyse des images [2]. Elles reposent sur une modélisation souvent assez abstraite de l'image et s'inspirent des techniques utilisées en littérature pour décrire le langage naturel, puis reprises en mathématique pour décrire les chaînes de symboles. Les grammaires sont donc des techniques de description « de haut-niveau » (puisque travaillant sur des primitives abstraites qui ne sont pas immédiatement accessibles par l'analyse). Ce sont aussi des approches **syntaxiques** qui essaient « d'expliquer » comment sont agencées les primitives de l'image. Les méthodes syntaxiques et les grammaires sont présentées dans des ouvrages généraux dont : [51, 8, 3]. L'œuvre de K.S. Fu a été en majorité consacrée aux approches syntaxiques et il a fourni une très volumineuse littérature sur ce sujet [22].

4.1 Les grammaires formelles

Les grammaires formelles ont été définies pour représenter les règles de construction de structures chaînées comme les phrases du langage courant (Chomsky 1959). Nous en décrirons le fonctionnement simplifié à partir d'un exemple, puis nous examinerons les deux extensions majeures qui les rendent adaptées au traitement des images.

4.1.1 Un exemple

Considérons la phrase suivante :

X = Le tracteur rêveur a archivé les lointaines catastrophes devant le Sénat.

Elle se décompose selon la formule suivante :

$$X = \text{GS.GV.GCOD.GCL}$$

où l'on reconnaît :

Une analyse plus détaillée permettrait de décomposer chaque composante selon une représentation plus fine :

$$\text{GS} = \text{article.nom.adjectif}$$

GS	groupe sujet	<i>Le tracteur rêveur</i>
GV	groupe verbe	<i>a archivé</i>
GCOD	groupe complément d'objet direct	<i>les lointaines catastrophes</i>
GCL	groupe complément de lieu	<i>devant le Sénat</i>

pour conduire à une représentation plus détaillée, par exemple :

$$X = \text{ar-}$$

ticle.nom.adjectif.verbe_auxillaire.participe_passé.article.adjectif.nom.preposition.article.nom

Cette décomposition est la dernière que l'on peut faire selon les règles de la syntaxe. C'est une décomposition terminale de la phrase utilisant les règles de la grammaire du langage français.

Mais notre décomposition est « formelle » en ce qu'elle ne s'intéresse pas au sens de la phrase (qui est ici incertain ...) mais à la seule conformité de la construction par rapport aux règles.

4.1.2 Définition

Une grammaire \mathcal{G} est constituée des éléments suivants :

- un élément « symbole » ou un élément « phrase » représenté ici par X ,
- un ensemble N d'éléments non-terminaux : $N = \text{GS, GV, GCOD, ...}$
- un ensemble d'éléments terminaux : $T = \text{article, auxillaire, adjectif, nom, ...}$
- un ensemble de règle d'écriture R (ici les règles de la grammaire française).

$$\mathcal{G} = X, N, T, R$$

4.1.3 les deux usages d'une grammaire

Le mode génératif

Dans ce mode, on part d'une grammaire $\mathcal{G} = \{X, N, T, R\}$ et de l'élément symbole, puis on applique successivement, selon un ordre que l'on choisit, les règles d'écriture possibles. Lorsqu'aucune nouvelle règle ne peut s'appliquer (on est donc arrivés à une chaîne d'éléments terminaux), on a produit une « phrase » de la grammaire.

Si l'on applique toutes les combinaisons possibles des règles à l'axiome symbole, l'ensemble des phrases ainsi obtenues constituent le « langage » \mathcal{L} associé à la grammaire \mathcal{G} .

L'opération de concaténation utilisée dans les grammaires, est associative et, généralement, non commutative, elle agit sur l'ensemble $\mathcal{X} = N \cup T$. On dote cet ensemble d'un élément neutre λ tel que $\lambda a = a \quad \forall a \in \mathcal{X}$. L'ensemble ainsi constitué forme un monoïde.

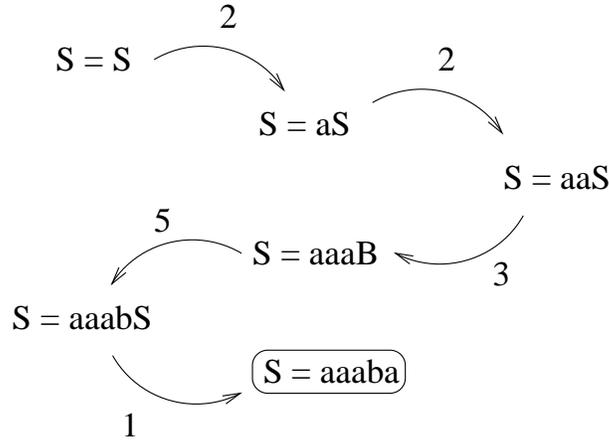
Illustrons le mode génératif à partir d'une grammaire particulière¹.

La grammaire est définie à partir des éléments suivants :

$$N = \{S, B\} \quad T = \{a, b\}$$

¹Cette grammaire a été proposée par Ledley, pour décrire les chromosomes. Nous en reparlerons.

$$R = \begin{pmatrix} X :: a & 1 \\ X :: aX & 2 \\ X :: aB & 3 \\ B :: b & 4 \\ B :: bX & 5 \end{pmatrix}$$



La phrase calculée par la séquence d'opérations 22351 est donc $aaaba$.

Le mode analytique

Il permet de répondre aux questions suivantes : « Etant donné une phrase X , appartient-elle au langage \mathcal{L} engendré par la grammaire \mathcal{G} ? Si oui, quelle est la structure de cette phrase ? ».

La technique adoptée pour répondre à ces questions consiste à faire une recherche exhaustive des solutions par un parcours de graphe (voir figure 4.1). S'il existe un chemin qui conduit de X à une suite des seuls éléments terminaux, la phrase appartient au langage et « s'explique » par la séquence des règles qui ont permis d'y aboutir. Sinon elle n'appartient pas au langage. S'il existe plusieurs chemins différents, alors on dit que la phrase est ambiguë. S'il existe une seule phrase ambiguë, alors on dit que le langage et la grammaire sont eux-mêmes ambigus.

L'inférence de grammaire

Cette troisième catégorie de problèmes est très différente des précédentes, et beaucoup plus difficile. Elle consiste à déterminer quelles règles de grammaire $\{R_i\}$ permettent de construire un langage \mathcal{L} sachant l'ensemble des primitives $\mathcal{X} = T \cup N$ et disposant d'un ensemble d'exemples de phrases du langage : $\mathcal{I} \subset \mathcal{L}(G)$.

Ce problème s'appelle l'**inférence de grammaire**. c'est un problème qui ne peut se résoudre exactement que dans des cas simples (grammaire « peu compliquée » et alphabet réduit) [23, 24, 56, 48] et [15] chapitre 7. La solution de ce problème passe tout d'abord par la reconnaissance de l'équivalence entre les grammaires régulières (voir ci-dessous) et les automates finis. Un automate est un processus qui agit sur un système défini par ses états. Lors d'une transition R , il fait passer le système d'un état Q à un état Q' en produisant un

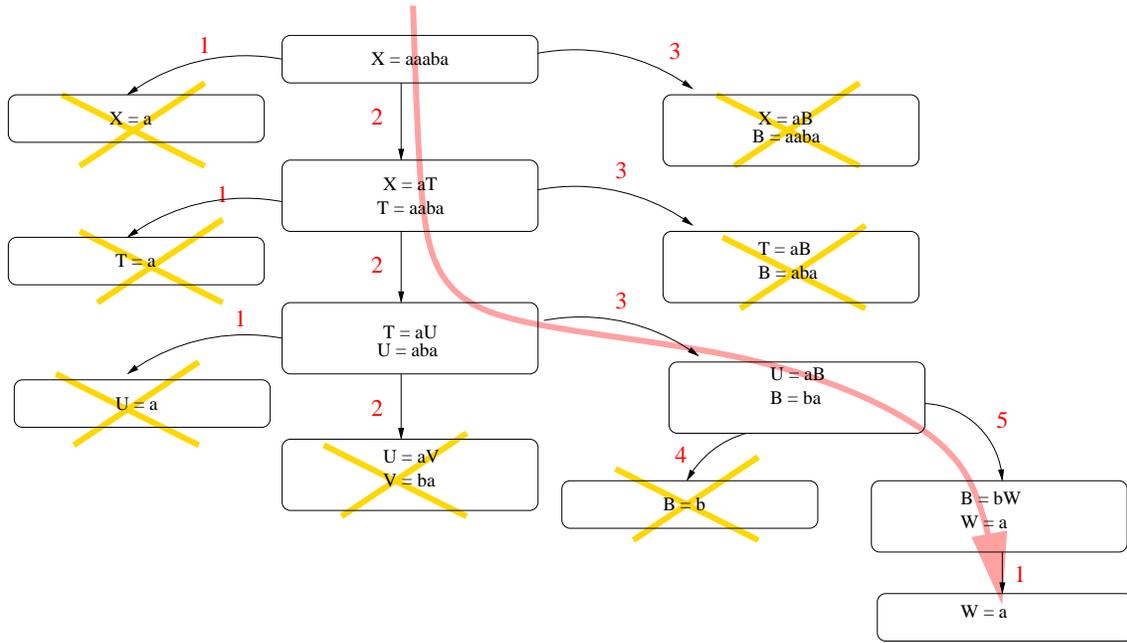


FIG. 4.1 – Parcours du graphe construit à partir de $X = aaaba$ les croix marquent les expressions interdites par la grammaire. L'aboutissement à la formule $W = a$ exprime la reconnaissance de la phrase X par le langage.

élément a membre d'un alphabet T . Un système est défini par son état initial S et évolue vers un état final S' . Pour aller de S à S' , l'automate produit une séquence $\{a_1 a_2 \dots a_n\}$ qui sera identifiée aux mots du langage. On sent qu'il existe de nombreuses similitudes entre automates finis et langages. Cette similitude est détaillée dans [51] par exemple. On utilise l'analogie automates finis / grammaire pour tirer profit de la grande quantité de résultats connus en théorie des automates. On sait par exemple définir des automates dérivés d'un automate donné en regroupant des états. Cela permet d'éviter les situations ambiguës dans les grammaires en créant des classes d'équivalence de symboles et de mots. On sait également démontrer l'accessibilité de certains états et, partant, de certains mots du langage.

S'appuyant sur ces résultats, la théorie des graphes nous montre qu'il existe généralement une infinité de grammaires qui produisent un langage donné ou un sous-ensemble \mathcal{I} d'un langage \mathcal{L} donné.

En particulier, il existe la **grammaire universelle** qui autorise la production de tous les symboles à partir de tout état :

$$\forall i \forall a \in \mathcal{X} \exists x_j : x_i :: ax_j$$

Il existe également la grammaire canonique, elle aussi dégénérée, qui à tout élément $x = a_1.a_2.\dots.a_n$ de \mathcal{L} fait correspondre la règle :

$$\forall x = a_1.a_2.\dots.a_n \exists R : X :: a_1.a_2.\dots.a_n$$

Pour réduire l'espace des solutions, on impose que \mathcal{I} soit un **échantillon complet** par rapport à la grammaire \mathcal{G} , c'est-à-dire :

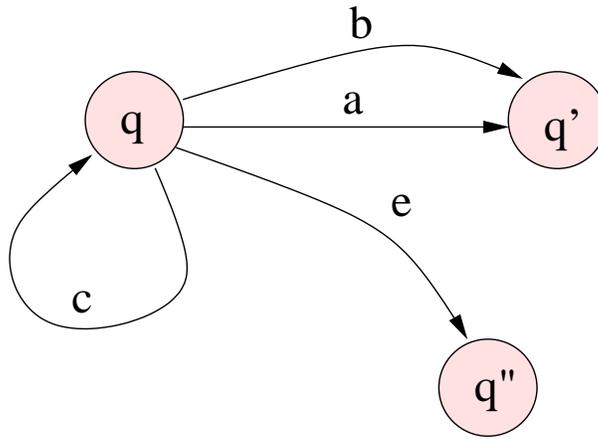


FIG. 4.2 – Un automate fait passer d'un état q à des états q', q'', \dots par des fonctions de production délivrant les symboles a, b, \dots

- $\mathcal{I} \subset \mathcal{L}(G)$
- l'alphabet de \mathcal{I} contient tout les éléments de T ,
- toutes les règles R sont utilisées au moins une fois dans la génération des phrases de \mathcal{I} .

On démontre alors qu'avec un échantillon \mathcal{I} complet, on sait énumérer toutes les solutions régulières à partir de l'**automate canonique maximum** ($\text{ACM}(\mathcal{I})$). Malheureusement cette solution est généralement trop coûteuse car le nombre de partitions de $\text{ACM}(\mathcal{I})$ est en $\mathcal{O}(\exp(N))$ si l'alphabet a N lettres.

On a donc en pratique recours à des heuristiques qui ont reçu des noms classiques dans la littérature des automates :

- l'algorithme $UV^k W$ qui recherche dans l'échantillon la séquence qui peut le plus souvent se factoriser dans les échantillons (ce sera V qui se factorise k fois), on remplace alors V^k par un nouveau symbole (par exemple U), puis on recommence la recherche de la plus grande séquence, et ainsi de suite ;
- l'algorithme des k-finales.

4.1.4 Les divers types de grammaires

Les grammaires formelles ont trouvé de très nombreuses applications en informatique théorique aussi bien dans la conception des langages informatiques que pour concevoir les compilateurs ou assurer des démonstrations. Il y a de grandes différences de complexité entre les grammaires informatiques les plus simples et les langages naturels rencontrés usuellement [51]. On a classé les diverses grammaires par ordre décroissant de complexité en fonction des primitives que manipulent les règles. Si l'on écrit une règle sous la forme :

$$R = \{\alpha :: \beta\}$$

le tableau 4.1 indique les classes ainsi construites :

Dans la pratique, en traitement des images, nous nous efforcerons de rester dans le domaine aisément soluble des grammaires régulières. Mais nous serons amenés à étendre ce cadre avec les deux extensions ci-dessous.

Grammaire	α	β	difficulté
Type 0 : à structure de phrase	$\in \mathcal{X}$	$\in \mathcal{X}$	non résolu
Type 1 : dépendant du contexte	$\alpha = \gamma.A.\delta$ $A \in \mathcal{X}$	$\beta = \gamma.B.\delta$ $\gamma, \beta, B \in \mathcal{X}$	parfois possible
Type 2 : indépendant du contexte	$A \in N$	$\beta \in \mathcal{X}$	souvent possible
Type 3 : régulier	$A \in N$	$a.B \quad a \in T$ $B.a \quad b \in N$	facile

TAB. 4.1 – Les diverses grammaires et leur complexité

4.2 Extensions indispensables en traitement d'images

4.2.1 Les grammaires stochastiques

Cette extension intervient lorsque plusieurs règles de transcription se présentent concurrentement mais ne se rencontrent pas avec des probabilités identiques dans la construction des diverses phrases du langage [71, 51].

On associe à la règle de réécriture : $\alpha_i \in \mathcal{X} :: \beta_{ij} \in \mathcal{X}$ avec $i = 1 \dots j$ et $j = 1, \dots, n_i$, la probabilité P_{ij} vérifiant : $0 \leq P_{ij} \leq 1$ et

$$\forall i \sum_{j=1}^{n_i} P_{ij} = 1$$

Partant du symbole phrase X , on aboutit à une phrase x par la succession des règles r_i , $i = 1, \dots, n_x$, formant un chemin dénoté ρ . On peut alors définir la probabilité d'obtenir x par le chemin ρ par :

$$P(x|\rho) = P(r_{n_x}|r_{n_x-1}, r_{n_x-2}, \dots, r_1)$$

et, dans le cas où les règles sont indépendantes les unes des autres :

$$P(x|\rho) = \prod_{i=1}^{n_x} P(r_i)$$

Lorsqu'une grammaire est ambiguë il existe de multiples façons de se rendre de X à x . Pour chaque façon repérée par l'indice j nous calculons une probabilité que nous dénotons $P_j(x)$ et définissons alors la probabilité de la phrase par :

$$P(x) = \sum_{j=1}^k P_j(x)$$

Une grammaire est auto-consistante si :

$$\sum_{x \in \mathcal{L}} P(x) = 1$$

Quoiqu'intéressante en théorie, cette expression n'est souvent pas d'un grand intérêt pratique car l'utilisateur se trouve plutôt dans une situation inverse. Disposant d'un langage \mathcal{L} , il observe des phrases S de probabilité $P(S)$ fort différentes. Il souhaite retrouver les probabilités élémentaires des transitions P_{ij} . Ce problème est infiniment plus compliqué et n'a pas de solution simple dans le cas général.

4.2.2 Grammaires tolérantes

Une grammaire est tolérante si elle accepte des phrases qui ne s'expriment pas rigoureusement à partir d'une séquence du langage, mais qui n'en diffèrent que par un petit nombre d'erreurs. Une grammaire tolérante accepte une phrase $y \in (T \cup N)^n$ telle que $T \notin \mathcal{L}(\mathcal{G})$ si sa distance à une phrase de $\mathcal{L}(\mathcal{G})$ est inférieure à une distance donnée.

Traditionnellement, on accepte comme erreurs possibles les défauts suivants :

- le remplacement d'un symbole par un autre
- la perte d'un symbole,
- l'ajout d'un symbole.

L'algorithme de Wagner et Fisher est un outil très connu pour trouver la distance minimale entre deux phrases [74].

4.3 Quelles grammaires pour l'analyse de scène et le traitement des images ?

Nous venons de voir à travers des exemples que les grammaires traitent des chaînes, c'est-à-dire des structures monodimensionnelles ordonnées. Ceci ne les prédisposent pas à travailler sur des images qui par nature sont bidimensionnelles et résident par nature dans un espace sans ordre (voir chapitre sur la couleur de [6]).

Les primitives du traitement des images (contours, régions, objets, ...) sont naturellement bidimensionnelles. L'application d'une règle par « concaténation » est donc naturellement ambiguë (voir figure 4.3). Pour pouvoir utiliser le formalisme des grammaires, il est indispensable de pouvoir introduire une structure d'ordre dans les primitives que l'on peut traiter. Cela peut se faire de 3 façons :

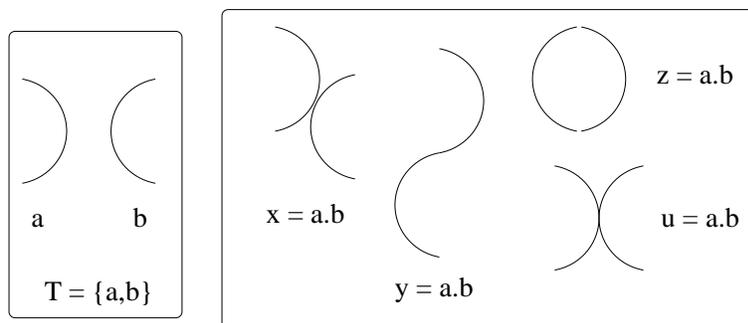


FIG. 4.3 – Les primitives bidimensionnelles ne présentent pas d'ordre naturel et la formule $a.b$ peut s'interpréter de multiples façons.

1. en dotant les primitives a de points caractéristiques et en ajoutant à chaque règle R une extension indiquant en quel point les primitives se combinent (cela revient en pratique à multiplier les règles) ; par exemple : $a(1).b(1)$ peut s'écrire $a.(1,1)b$, identifiant ainsi l'opérateur $.(1,1)$.
2. en imposant à chaque primitive a une « tête » et une « queue » de façon que la règle $a.b$ applique toujours la tête de la primitive b à la queue de la primitive a (cela revient en pratique à multiplier les primitives).
3. en créant des « bi-grammaires » c'est-à-dire des grammaires qui appliquent conjointement une règle en ligne et une règle en colonne (on multiplie aussi les règles).

La complexité de ces situations est illustrée sur la figure 4.4.

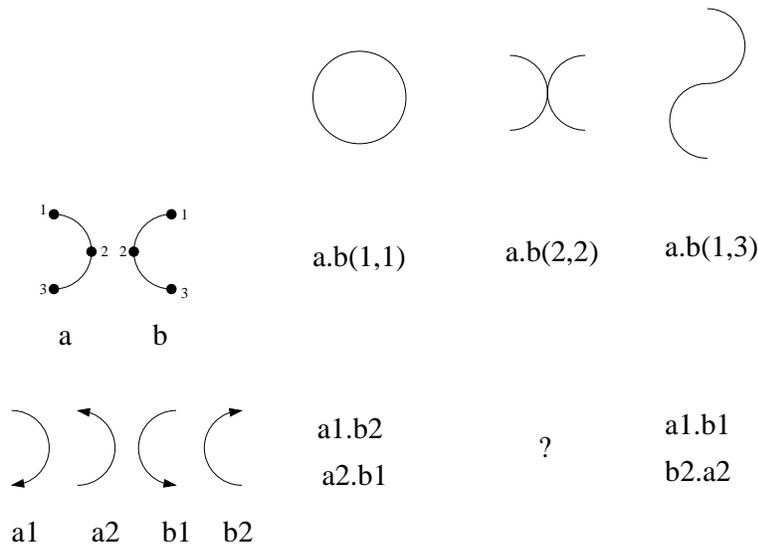


FIG. 4.4 – Comment ordonner les primitives. Ligne du haut, 3 figures simples. Ligne du centre : on multiplie les règles en définissant des points d'application des primitives. Ligne du bas, on multiplie les primitives en les ordonnant. Tous les cas ne peuvent s'exprimer avec tous les choix.

4.4 Les grammaires pour le graphique

Les dessins au trait et les graphique se prêtent bien à une représentation par grammaires car ils offrent des structures généralement 1D qui se suivent de façon ordonnée le long d'un contour.

4.4.1 Les contours à 2D

Une littérature volumineuse, mais pas toujours très intéressante, a été consacrée dès les années 70 à des grammaires ayant pour objectif de décrire :

1. des objets manufacturés ; outils, objets usinés, mobilier, etc., essentiellement pour des applications de robotique : manipulation, usinage, évitement d'obstacles.

2. des caractères et des symboles pour la lecture automatique et la lecture des symboles : par exemple pour décrire la variété des tiretés dans les cartes ou les plans, ou pour les légendes des cartouches, etc.

3. quelques objets naturels simples : chromosomes en biologie, mollusques, etc.

Les primitives (éléments terminaux de la grammaire) sont souvent très simples : des segments de droite dans diverses orientations ou de diverses longueurs, des arcs de cercle, des segments courbes attribués parfois de "concavité" [28, 63, 56]. Lorsque sur des structures si simples on met des règles elles-mêmes très simples, on obtient des outils assez limités mais parfois très utiles pour distinguer des objets dans des contextes limités [9]. Mais il est souvent nécessaire de complexifier les primitives pour rendre compte des variabilités des objets rencontrés. On aboutit alors à des grammaires très efficaces, mais souvent très spécifiques (voir figure 4.5).

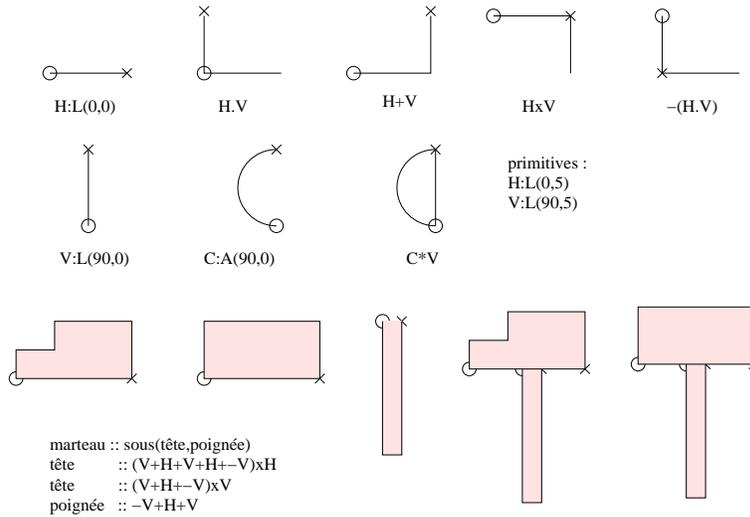


FIG. 4.5 – Exemple d’objets simples décrits par des grammaires de contours : la grammaire de Masini (1984). Les primitives sont dotées de 2 attributs qui spécifient leurs associations.

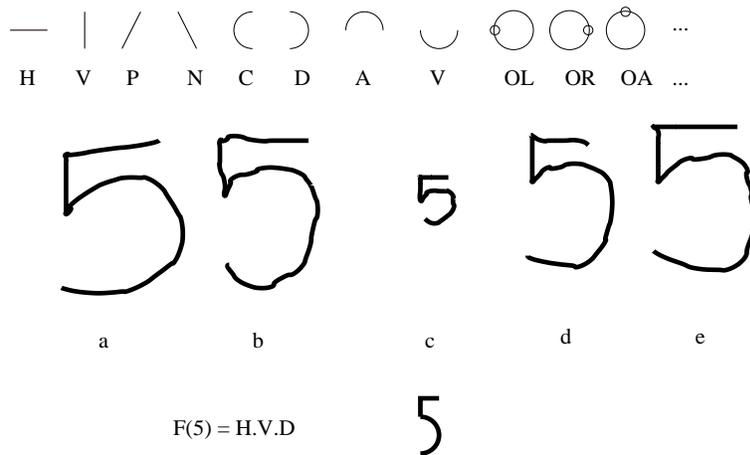


FIG. 4.6 – Exemple d’une grammaire développée pour la description des chiffres sur les chèques.

Parmi ces grammaires, celle de Ledley a été précurseur [45]. Elle permettait de reconnaître divers types de chromosomes (en fonction de leur nombre de « pattes »). Ledley codait le chromosome avec 4 éléments terminaux exprimant la courbure (forte, faible ou nulle) et la concavité des morceaux de contours (voir figure 4.7). Les éléments non-terminaux comportaient les bras (A), les pentes douces (B), les virages à droite (R), les virages à gauche (L) et les côtés (S). Les règles de réécriture sont typiquement celles écrites ci-dessous :

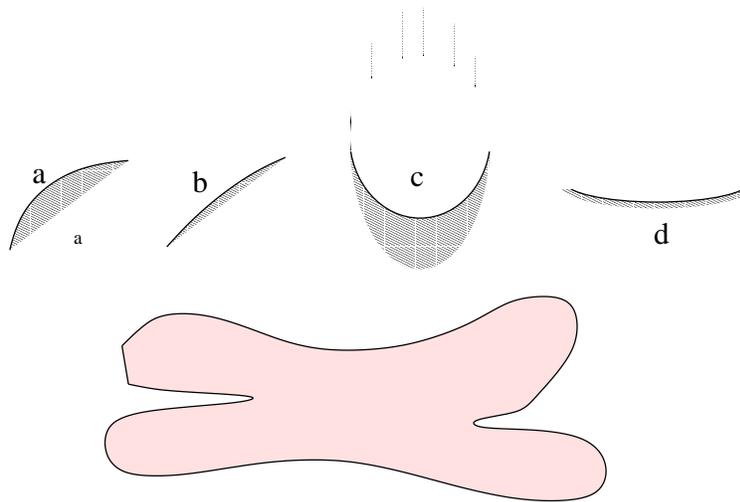
$$\begin{aligned}
 X &:: AA \\
 A &:: SA \quad AS \quad BR \quad BR \quad LB \\
 L &:: Bc \\
 R &:: cB \\
 S &:: bS \quad Sb \quad bd \\
 B &:: bB \quad Bb \quad a
 \end{aligned}$$


FIG. 4.7 – Les éléments terminaux de la grammaire de Ledley et un chromosome de type X.

4.4.2 Les descriptions des objets tri-dimensionnels

Des grammaires ont également été proposées pour décrire des polyèdres à 3 dimensions (voir figure 4.9), et en particulier les règles qui président à la visibilité des arêtes en fonction de la position de l'observateur par rapport à la facette (voir 3.1). Ce sont ces grammaires qui permettent des décompilations des aspects complexes [28, 46].

4.5 Les grammaires pour les images

L'un des problèmes importants qui se posent lorsqu'on cherche à exprimer par des grammaires les niveaux de gris d'une image est la transformation de la représentation matricielle par une représentation ordonnées [5, 63]. On peut imaginer plusieurs types de balayages (voir chapitre sur la couleur de [6]) : balayage de type zig-zag (comme dans la DCT), balayage lexicographique (les lignes d'abord puis les colonnes, etc.).

No	Kind of feature "fork" type	Binary vector								Disposition of faces having corresponding vertex "fork" type
		1	2	3	4	5	6	7	8	
1		1	0	1	0	0	1	0	0	1,3 - FF; 1,6 - UF; 3,6 - LF;
2		1	0	0	1	0	1	0	0	1,4 - RF; 1,6 - UF; 4,6 - LF;
3		1	0	0	1	0	0	1	0	1,4 - RF; 1,7 - UF; 4,7 - LF;
4		0	1	0	1	0	0	1	0	2,4 - FF; 2,7 - UF; 4,7 - LF;
5		0	1	0	0	1	0	1	0	2,5 - LF; 2,7 - RF; 5,7 - UF;
6		0	1	0	0	1	0	0	1	2,5 - LF; 2,8 - RF; 5,8 - UF;
7		0	0	1	0	1	0	0	1	3,5 - FF; 3,8 - RF; 5,8 - UF;
8		0	0	1	0	0	1	0	1	3,6 - LF; 3,8 - RF; 6,8 - UF;
Legend: FF - front face UF - upper face LF - left face RF - right face										

FIG. 4.8 – La grammaire de Gotchev pour décrire les objets à 3D : ici les liaisons en « fourchette » [28].

Line drawings			
Descriptions of drawings	$P_1 P_2, P_2 P_3, P_3 P_4, P_4 P_1, ::$ $P_2 P_3, P_3 P_6, P_6 P_3, P_3 P_2, ::$ $P_4 P_3, P_3 P_6, P_6 P_7, P_7 P_4, ::$	$P_1 P_2, P_2 P_3, P_3 P_1, ::$ $P_1 P_3, P_3 P_6, P_6 P_1, P_3 P_2, ::$ $P_2 P_6, P_6 P_4, P_4 P_3, P_3 P_2, ::$	$P_1 P_2, P_2 P_3, P_3 P_4, P_4 P_5, P_5 P_1, ::$ $P_2 P_6, P_6 P_7, P_7 P_3, P_3 P_2, ::$ $P_4 P_3, P_3 P_7, P_7 P_8, P_8 P_4, ::$ $P_5 P_4, P_4 P_8, P_8 P_3, P_3 P_5, ::$
What kind of 2D geometrical figures are the separate faces ?	1 - quadrangle 2 - quadrangle 3 - quadrangle	1 - triangle 2 - quadrangle 3 - quadrangle	1 - pentagon 2 - quadrangle 3 - quadrangle 4 - quadrangle
What are the common edges of the faces ?	$P_2 P_3 - 1, 2$ $P_3 P_4 - 1, 3$ $P_6 P_3 - 2, 3$	$P_2 P_3 - 1, 3$ $P_3 P_1 - 1, 2$ $P_3 P_4 - 2, 3$	$P_2 P_3 - 1, 2; P_7 P_3 - 2, 3;$ $P_3 P_4 - 1, 3; P_8 P_4 - 3, 4;$ $P_4 P_5 - 1, 4;$
What are the tree-face vertexes ?	P_3	P_3	$P_3; P_4$
How are the faces positioned with respect to one another ?	1 - UF 2 - RF 3 - FF	1 - UF 2 - LF 3 - RF	1 - UF 2 - RF 3 - FF 4 - LF
What 2D geometrical figure is the base ?	Quadrangle	Triangle	Pentagon
What is the 3D figure ?	Quadrangular prism	Triangular prism	Pentagonal prism

FIG. 4.9 – Exemple de décomposition et d'analyse à partir de la grammaire de Gotchev pour décrire les objets à 3D [28].

4.5.1 Grammaires et détection

Des grammaires très simples ont été proposées pour permettre des détections très rapides de primitives simples : bord, cotours, points isolés, dans des fenêtres de taille 3×3 (cf figure 4.10). Elles s'appliquent à des images binaires ou dont les niveaux sont repérés par rapport à un seuil en clairs ou sombres. Les grammaires mises en œuvre sont simplistes et peuvent être remplacées par des tables de transcodage (LUT) [3, 75].

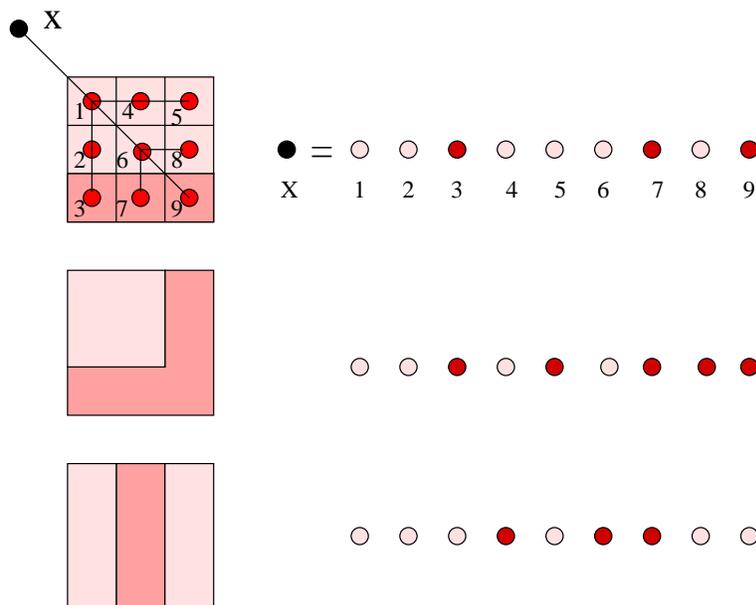


FIG. 4.10 – Détection de motifs simples dans une image par la méthode de Basu et Lu (1987). Le mode de balayage indiqué en haut à gauche fournit les « phrases » représentées dans les lignes de droite, pour des détections d'un contour horizontal (en haut), d'un coin (au milieu) ou d'une ligne verticale (à droite).

4.5.2 Grammaires et textures

Les travaux les plus importants sur les grammaires ont été accomplis par Lu et Fu sur les grammaires de textures. Leur motivation était de créer des textures plus que de les analyser. Ils ont choisi pour cela deux niveaux de représentation, tous deux exprimés par des grammaires. Un premier niveau permet de décrire la primitive de texture (cf [6]), le second niveau de décrire l'agencement entre les primitives. L'usage de grammaires stochastiques permet de prendre en compte l'aspect aléatoire de la texture à l'un comme à l'autre des niveaux.

Le niveau de la primitive est généralement décrit sur une fenêtre de taille $n \times n$ pixels (n de l'ordre de 5 ou 9). Le balayage de la fenêtre se fait selon un schéma un peu différent de celui de Basu (cf. figure 4.10) car on entre par la gauche et le centre de la fenêtre. Les règles indiquent pour chaque état du signal quelle valeur placer dans le nouveau pixel, cette valeur pouvant être choisie aléatoirement parmi des configurations possibles.

Au macro niveau, les cellules de $n \times n$ pixels sont elles-mêmes associées selon les règles

d'une seconde grammaire qui indique comment s'agencent entre elles les primitives.

4.6 Des usages plus avancés

On veut citer ici ce qui a été fait par Mangin (1995) pour décrire les sillons du cerveau pour permettre une reconnaissance des structures cérébrales dans les images d'IRM anatomique. La grammaire, stochastique, exprime les diverses relations entre les sillons, leurs raccords, la présence de ponts dans les sillons et la position des terminaisons.

D'autres grammaires ont été développées dans des travaux de l'IGN (Fuchs 2002) pour expliquer la construction des toits de maison à partir de pans de toits : ainsi on construit des toits à deux pans ou à quatre pans, ou des toits à pans coupés, etc.

Enfin des grammaires stochastiques permettent de décrire la mise en page des articles dans les documents scannés : relation entre titres, noms d'auteurs, résumés, paragraphes, chapitres et marges [36], des règles de typographie précisant les positions respectives des légendes et des figures par exemple, ou des séparations entre paragraphes.

$S \rightarrow t h g B u$	$B =$ corps de texte, $t =$ marge haute, $h =$ entête, g saut d'entête de corps, $u =$ marge inférieure
$B \rightarrow l C g C r$	$C =$ colonne
$C \rightarrow i g C$	$i =$ ligne de texte

TAB. 4.2 – Exemples de règles utilisées par la grammaire de Kannungo et Mao [36]

(... à suivre ... évidemment !)

Bibliographie

- [1] Special section on graph algorithms and computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10) :1049–1144, 2001.
- [2] S. Basu. Image segmentation by semantic method. *Pattern Recogn.*, 20 :497–511, 1987.
- [3] S. Basu and K.S. Fu. Image segmentation by syntactic method. *Pattern Recogn.*, 20 :33–44, 1987.
- [4] E. Bengoetxea. *Mise en correspondance inexacte de graphes par algorithmes d'estimation des distributions*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, 2002.
- [5] G. Biswas and R.C. Dubes. Some experiments in two-dimensional grammatical inference. *Pattern Recogn. Letters*, 2 :173–177, 1984.
- [6] I. Bloch, Y. Gousseau, H. Maître, B. Pesquet, M. Roux, F. Schmitt, M. Sigelle, and F. Tupin. Analyse de images, polycopié de la brique anim. Technical Report 2002TSI05, Ecole Nationale Supérieure des Télécommunications, jun 2002.
- [7] H. Bunke. Attributed programmed graph grammars and their application to schematic diagram interpretation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(6) :574–582, 1982.
- [8] H. Bunke. Hybrid methods in pattern recognition. In Devijver and Kittler, editors, *Pattern Recognition*, Berlin, 1987. Springer Verlag - NATO - ASI series.
- [9] H. Bunke, K. Grebner, and G. Sagerer. Syntactic analysis of noisy input strings with an application to the analysis of heart-volume curves. In *7 int. Conf. Pattern Recogn.*, pages 1145–1147, Montreal, Ca, 1984.
- [10] H. Bunke and B. Messmer. Recent advances in graph matching. *International Journal of Pattern recognition and Artificial Intelligence*, 11(1) :169–203, 1997.
- [11] M. Burge, W. Burger, and W. Mayr. Recognition and learning with polymorphic structural components. *ICPR*, A :19–23, 1997.
- [12] R. Cesar, E. Bengoetxea, and I. Bloch. Structural matching in computer vision using probabilistic relaxation. *International Conference on Pattern Recognition*, 17(8) :749–764, 1995.
- [13] I. Charon, A. Germa, and O. Hudry. *Méthodes d'optimisation combinatoire*. Collection Pédagogique de Télécommunication, Masson, 1996.
- [14] W. J. Christmas, J. Kittler, and M. Petrou. Structural matching in computer vision using probabilistic relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8) :749–764, 1995.
- [15] A. Cornuéjols, L. Miclet, and Y. Kodratoff. *Apprentissage artificiel : Concepts et algorithmes*. Eyrolles, Paris, France, 2002.

- [16] A. Cross and E. Hancock. Graph matching with a dual step em algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11) :1236–1253, 1998.
- [17] P. Daniel. *Peut-on extraire le relief d'une seule image ?* PhD thesis, IRIT, Université Paul Sabatier de Toulouse,, 2000.
- [18] C.R. Dyer. Scene reconstruction from multiple views. In L.S. Davis, editor, *Foundations of image understanding*, pages 469–489. Kluwer, Boston.
- [19] M. A. Eshera and K. Fu. An image understanding system using attributed symbolic representation and inexact graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(5) :604–618, 1986.
- [20] O. Faugeras and K. Price. Semantic description of aerial images using stochastic labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(6) :633–642, 1981.
- [21] K. Fu. A step towards unification of syntactic and statistic pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 200–205, 1983.
- [22] K. S. Fu. *Syntactic pattern recognition and applications*. Prentice Hall, New York, USA, 1982.
- [23] K. S. Fu and T. L. Booth. Grammatical inference - part i. *IEEE trans on S. M. C.*, SMC-5(1) :95–111, 1975.
- [24] K. S. Fu and T. L. Booth. Grammatical inference - part ii. *IEEE trans on S. M. C.*, SMC-5 :873–885, 1975.
- [25] F. Fuchs. *Contribution à la reconstruction du bâti en milieu urbain, à l'aide d'images aériennes stéréoscopiques à grande échelle. Etude d'une approche structurale*. PhD thesis, Université René Descartes - Paris V, 2001.
- [26] D. Ghahraman, A. Wong, and T. Au. Graph monomorphisms algorithms. *IEEE Transactions on Systems, man, and cybernetics*, 10(4) :189–196, 1980.
- [27] M. Gondran and M. Minoux. *Graphes et algorithmes*. Collection de la Direction des Etudes et Recherches d'Electricité de France, Eyrolles, 1979.
- [28] G.V. Gotchev. Computer linguistic analysis of line drawings. *Pattern Recogn.*, 17 :433–440, 1984.
- [29] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *J. R. Statist. Soc. B*, 51(2) :271–279, 1989.
- [30] C. Hernandez and F. Schmitt. Multi-stereo 3d object reconstruction. In *3DPVT - 1st International Symposium on 3D Data Processing Visualization and Transmission*, Padova, Italy, jun 2002.
- [31] R. Horaud and T. Skordas. Stereo correspondance through feature grouping and maximal cliques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(11) :1168–1180, 1989.
- [32] B.K.P. Horn and M. J. Brooks. *Shape from Shading*. The MIT Press, Cambridge, Ma, USA, 1989.
- [33] B.P. K. Horn and M.J. Brooks. The variational approach to shape from texture. In B.K.P. Horn and M. J. Brooks, editors, *Shape from Shading*, pages 174–214. The MIT Press, 1989.
- [34] K. Ikeuchi and B.P.K. Horn. Numerical shape from shading and occluding boundaries. *Artificial intelligence*, 17(1-3) :141–184, 1981.

- [35] H. Jibrini. *Reconstruction automatique des bâtiments en modèles polyédriques 3D à partir de données cadastrales vectorisées 2D et d'un couple d'images aériennes à haute résolution*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, 2002.
- [36] T. Kanungo and S. Mao. Language models for style-directed layout analysis of documents. *IEEE trans on Image Processing*, 12(5) :583–597, 2003.
- [37] J.J. Koenderink. *Solid Shape*. Mit Press, Cambridge, Ma, 1990.
- [38] J.J. Koenderink and J. van Doorn. Photometric invariants related to solid shape. *Optica Acta*, 27(7) :981–996, 1980.
- [39] C. Kotropoulos, A. Tefas, and I. Pitas. Frontal face authentication using morphological elastic graph matching. *IEEE Transactions on Image Processing*, 9(4) :555–560, APR 2000.
- [40] W. Kropatsch and H. Bishof. *Digital Image Analysis*. Springer-Verlag, 2001.
- [41] K. Kukulakos and S. Seitz. A theory of shape by space carving. *Intern. J. of Comp. Vision*, 38(3) :199–218, 2000.
- [42] M. Lades, C. Vorbrügge and J. Buhmann, J. Lange, C. Malsburg, R. Würz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42(3) :300–310, JAN 1993.
- [43] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE T-PAMI*, 16 :150–162, 1994.
- [44] A. Laurentini. How far 3d shapes can be understood from 2d silhouettes. *IEEE T-PAMI*, 17 :188–195, 1995.
- [45] R.S. Ledley. *Computerized Medical Imaging And Graphics (Formerly Computerized Radiology)*. 1988.
- [46] W.C. Lin and K.S. Fu. 3d-plex grammars. *Information Sciences*, 34 :1–24, 1984.
- [47] P.L. Lions, E. Rouy, and A. Tourin. Shape from shading, viscosity and edges. *Numerische Mathematik*, 64 :323–353, 1992.
- [48] H.R. Lu and K.S. Fu. A general approach to inference of context-free programmed grammars. *T-SMC*, 14 :191–202, 1984.
- [49] B. Messmer. *Efficient graph matching algorithms for preprocessed model graphs*. PhD thesis, Universität Bern, 1996.
- [50] L. Miclet. *Méthodes structurelles pour la reconnaissance des formes*. CNET, Collection technique et scientifique des télécommunications, 1984.
- [51] L. Miclet. *Méthodes structurelles pour la reconnaissance des formes*. Eyrolles, Paris, France, 1984.
- [52] O. J. Morris, J. Lee, and A. Constantinidès. Graph theory for image analysis : an approach based on the shortest spanning tree. *IEE Proceedings*, 133(2) :193–229, 1986.
- [53] T. Pavlidis. Representation of figures by labeled graphs. *Pattern Recognition*, 4 :5–17, 1972.
- [54] A. P. Pentland. Local shading analysis. In B.K.P. Horn and M. J. Brooks, editors, *Shape from Shading*, pages 444–487. The MIT Press, 1989.
- [55] Aymeric Perchant. *Morphisme de graphes d'attributs flous pour la reconnaissance structurelle de scènes*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, 2000.

- [56] M. Richetin and F. Vernadat. Efficient regular grammatical inference for pattern recognition. *Pattern Recogn.*, 17(2) :245–250, 1984.
- [57] M. Roux. *Recalage d'images multi-sources. Application au recalage d'une image SPOT et d'une carte*. PhD thesis, Ecole Nationale Supérieure des Télécommunications (92 E 020), September 1992.
- [58] E. Rouy and A. Tourin. A viscosity solution approach to shape from shading. *SIAM J. Num. Anal.*, 29(3) :867–884, 1992.
- [59] A. Sanfeliu and K. Fu. A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 13(3) :353–362, 1983.
- [60] Francis Schmitt. *Projet sculpteur - modélisation numérique 3d d'objets muséologiques à partir de séquences d'images*, oct 2002.
- [61] L. Shapiro and R. Haralick. Structural description and inexact matching. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 3(5) :504–519, 1981.
- [62] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8) :888–905, 2000.
- [63] Q.Y. Shi. Semantic-syntax-directed translation and its application to image processing. *Information Sciences*, 32 :75–90, 1984.
- [64] sous la direction de H. Maitre. *Le traitement des images*. Hermes, Collection Traitement du Signal et des Images, 2003.
- [65] J.V. Stone and S.D. Isard. Adaptive scale filtering : A general method for obtaining shape from texture. *IEEE T-PAMI*, 17 :713–718, 1995.
- [66] B.J. Super and A.C. Bovik. Shape from texture using local spectral moments. *IEEE T-PAMI*, 17, 1995.
- [67] R. Szeliski. Rapid octree construction from image sequences. *Image Understanding*, 58 :23–32, 1993.
- [68] J. M. Tenenbaum. Experiments in interpretation guided segmentation. *Artificial Intelligence*, 8 :241–274, 1977.
- [69] P.S. Tsai and M. Shah. Shape from shading using linear approximation. *Image and Vision Computing*, 12(8) :487–494, 1994.
- [70] W. Tsai and K. Fu. Subgraph error-correcting isomorphisms for syntactic pattern recognition. *IEEE transactions on Systems, Man and Cybernetics*, 13(1) :48–60, 1983.
- [71] W .H. Tsai and K. S. Fu. Attributed grammar. a tool for combining syntactic and statistical approaches to pattern recognition. *IEEE trans on S. M. C.*, SMC-10 :873–885, 1980.
- [72] J. Ullmann. An algorithm for subgraph isomorphism. *Journal of the association for computing machinery*, 23(1) :31–42, 1983.
- [73] O. Veksler. *Efficient graph-based energy minimization methods in computer vision*. PhD thesis, Cornell University, 1999.
- [74] R. A. Wagner and M. J. Fisher. The string to string correction problem. *J.A.C.M.*, 21(1) :168–173, 1974.
- [75] P.S.P. Wang. An application of array grammars to clustering analysis for syntactic patterns. *Pattern Recogn.*, 17 :441–451, 1984.

- [76] R. Wilson and A. Cross and E. Hancock. Structural matching with active triangulation. *Computer Vision and Image Understanding*, pages 21–38, 1998.
- [77] L. Wiskott, J. Fellous, N. Kruger, and C. von-der Malsburg. Face recognition by elastic bunch graph matching. *Intelligent biometric techniques in fingerprint and face recognition*, 355 :357–396, 1999.
- [78] A. Wong and M. You. Entropy and distance of random graphs with application to structural pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(5) :599–609, 1985.
- [79] E. K. Wong. Model matching in robot vision by subgraph isomorphism. *Pattern Recognition*, 25(3) :287–303, 1992.
- [80] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering : theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11) :1101–1113, 1993.