

Chapitre 1

Graphes en traitement d'images et en reconnaissance des formes

1.1 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 [25] 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.

1.1.1 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 [7] les définitions et les algorithmes classiques en théorie des graphes. On pourra également se reporter au livre de Gondran et Minoux[15].

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 [20].

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 [15]). 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 [26].

1.2 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.

1.2.1 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 [24] 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 1.1). 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 $O(n^2)$) [7]. 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.

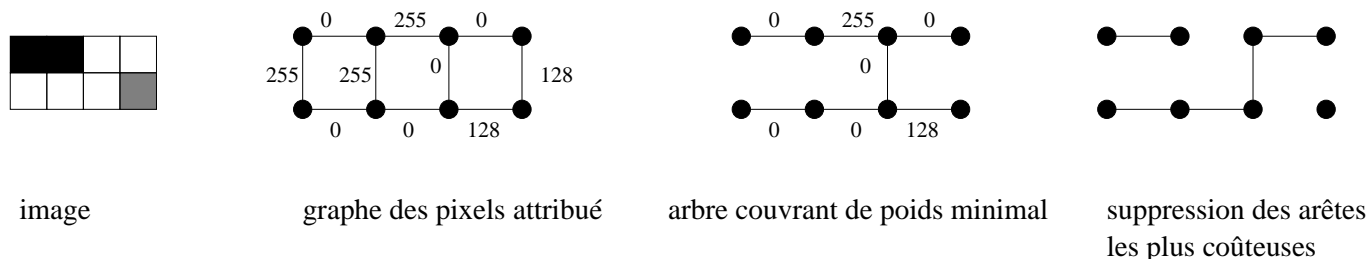


FIG. 1.1 – Segmentation par arbre couvrant de poids minimal

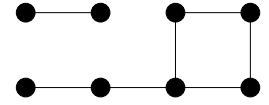
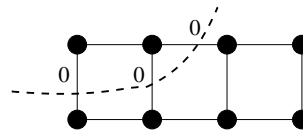
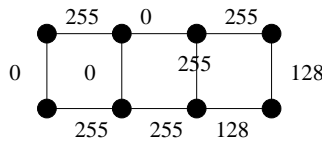
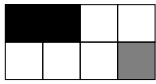
Coupe de capacité minimale

Dans le même ordre d'idée, des méthodes de segmentation recherchant la coupe de capacité minimale ont été proposées [40]. 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-connextité, 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 1.2).

La recherche de la coupe de capacité minimale se fait par l'algorithme de Ford et Fulkerson [7]. 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.



image

graphe des pixels attribué

coupe de capacité minimale

partition

FIG. 1.2 – Segmentation par coupe de capacité minimale

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 1.3, avec un graphe de 6 sommets et des arcs de poids a ou b inversement proportionnels à la distance ($b < a$).

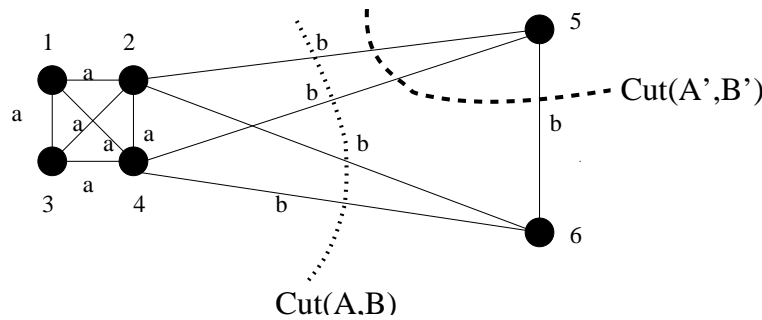


FIG. 1.3 – Influence du nombre de sommets dans la coupe : $Cut(A, B) = 4b$, $Cut(A', B') = 3b$

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 [30], 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 1.4.

Systemes 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 [32]. 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 [26] et sur des graphes polymorphiques [5].

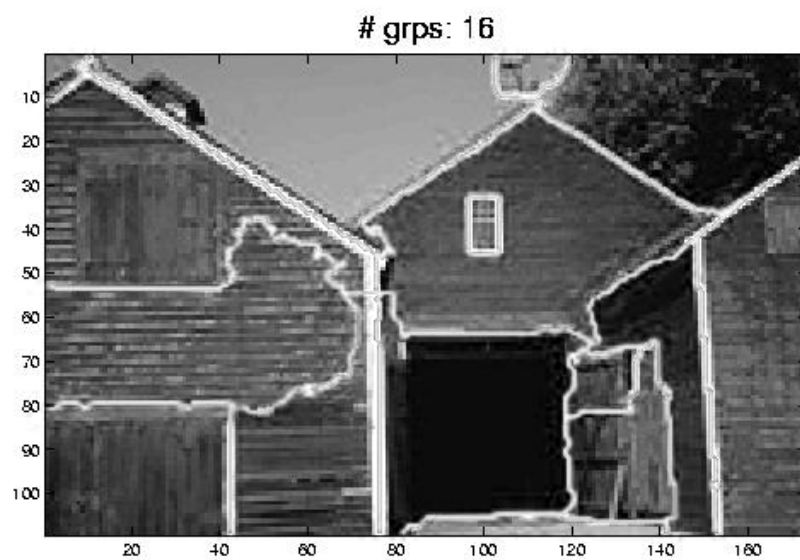
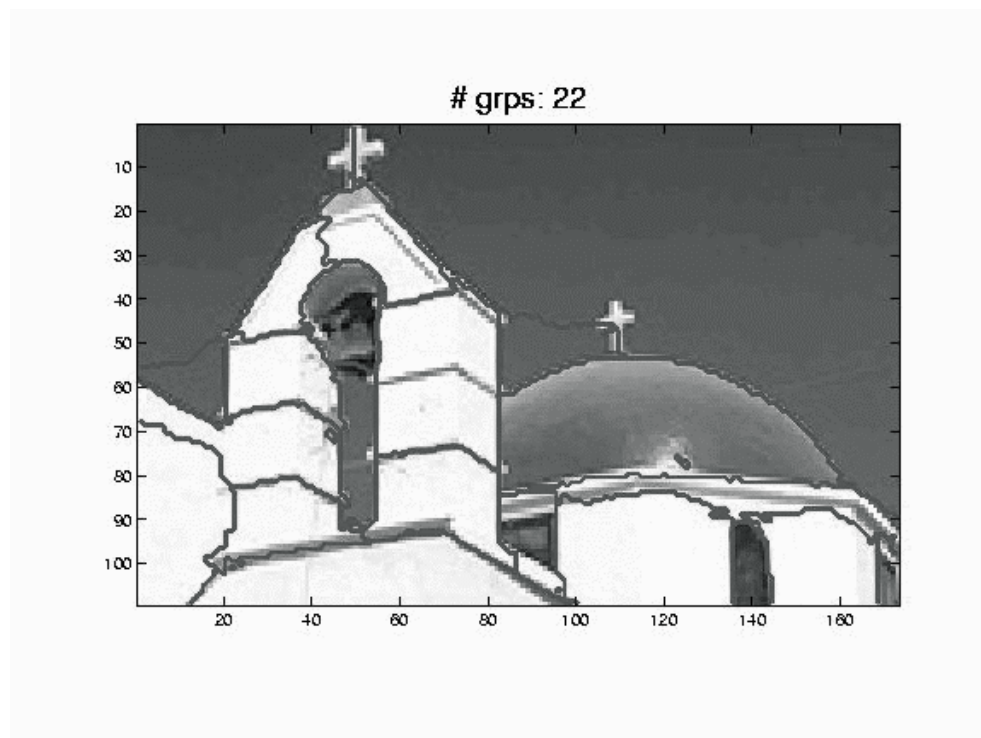


FIG. 1.4 – Résultats de segmentation par coupe normalisée (<http://www.cs.berkeley.edu/projects/vision/Grouping/manmade.html>). Les poids sont définis par $w_{ij} = \exp\left(-\frac{|F(i)-F(j)|^2}{\sigma_F^2}\right)$. Dans le premier exemple $F(i)$ est le niveau de gris du point et dans le second un vecteur d'indices texturaux.

1.2.2 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 [31], 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 [16]. 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.1.5) :

- 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).

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 :

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

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.

Les travaux de Boykov et Veksler [35] 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.

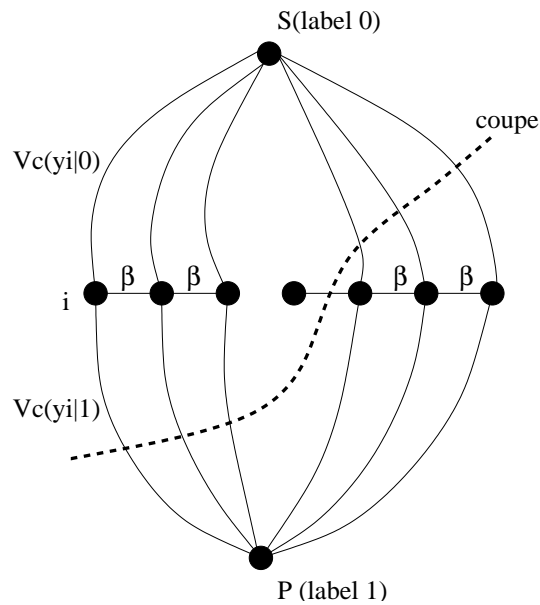


FIG. 1.5 – Définition du graphe pour la recherche du MAP dans un modèle markovien binaire

1.2.3 Graphes et reconnaissance des formes

Les graphes sont très largement utilisés pour la reconnaissance des formes [23]. 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 [18]. 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

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 .

mesurées et la complexité de la solution permet ensuite de choisir la solution. La figure 1.6 montre des exemples de bâtiments reconstruits.

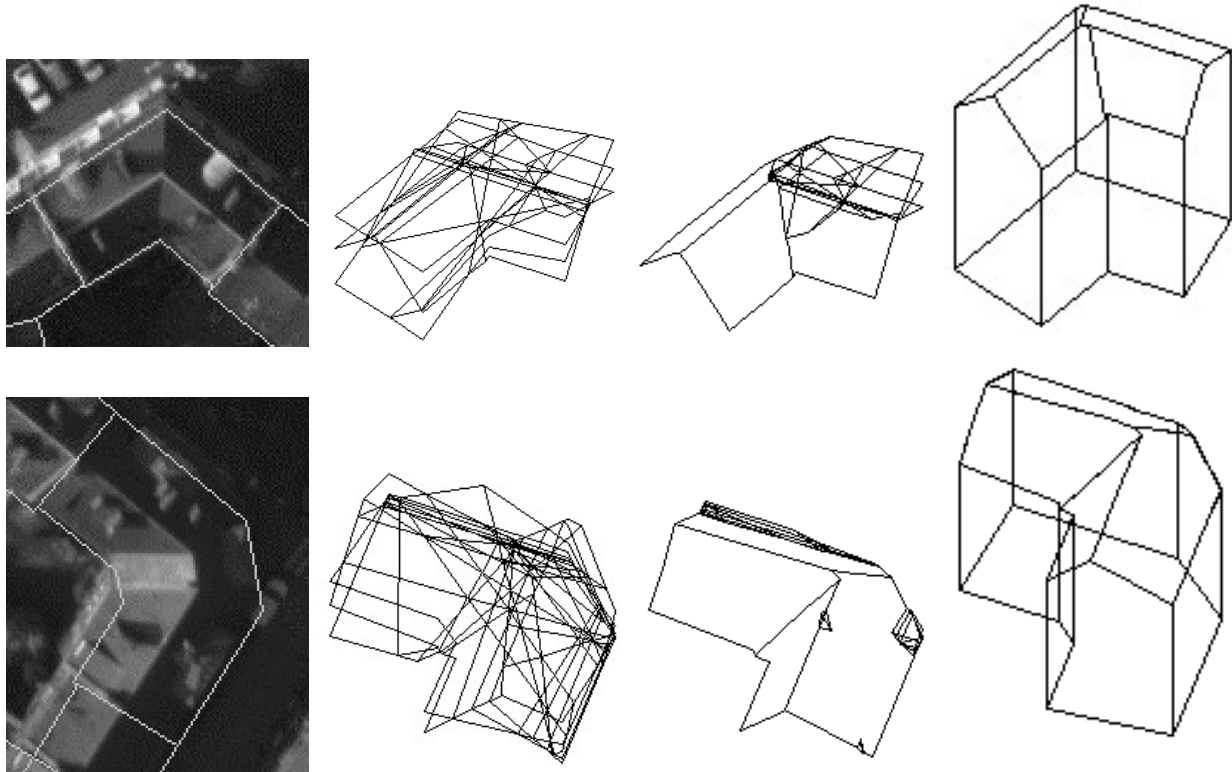


FIG. 1.6 – Détection du bâti par recherche de la clique maximale [18]. De gauche à droite : image originale et relevé cadastral associé; hypothèses de plans générées; solution obtenue après heuristique; solution finale de reconstruction automatique.

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 [3] [12].

1.3 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 [14] [39]. 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 [29] [11] [33] [28] [38] [10]. Dans ce dernier cas, on peut utiliser des algorithmes exacts (lorsque le nombre de nœuds est petit) ou des algorithmes approximatifs.

1.3.1 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 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 1.7). Un exemple pour de la mise en correspondance en stéréo-vision est donné dans [17].

Un autre algorithme est l'algorithme de Ullman [34] 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 $n < m$).

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

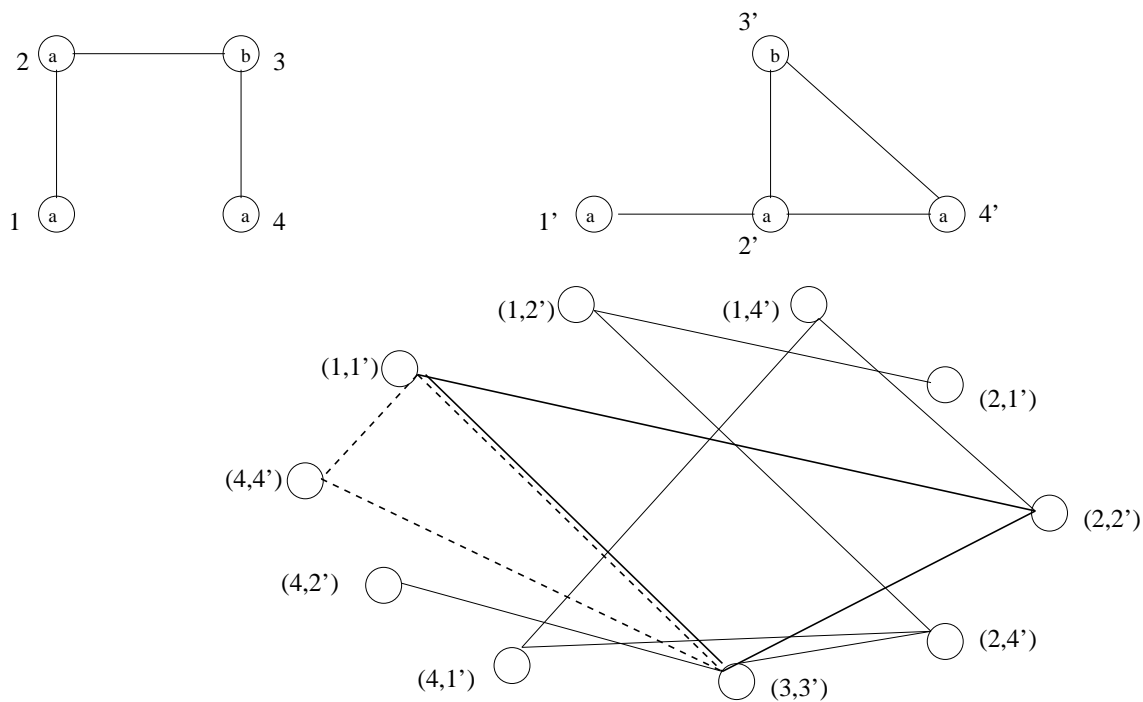


FIG. 1.7 – Deux graphes et leur graphe d’association. Les deux plus grandes cliques maximales sont indiquées en gras (trait plein et en pointillés).

1.3.2 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 distorsions 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 [23]) 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-graphes 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 [4].

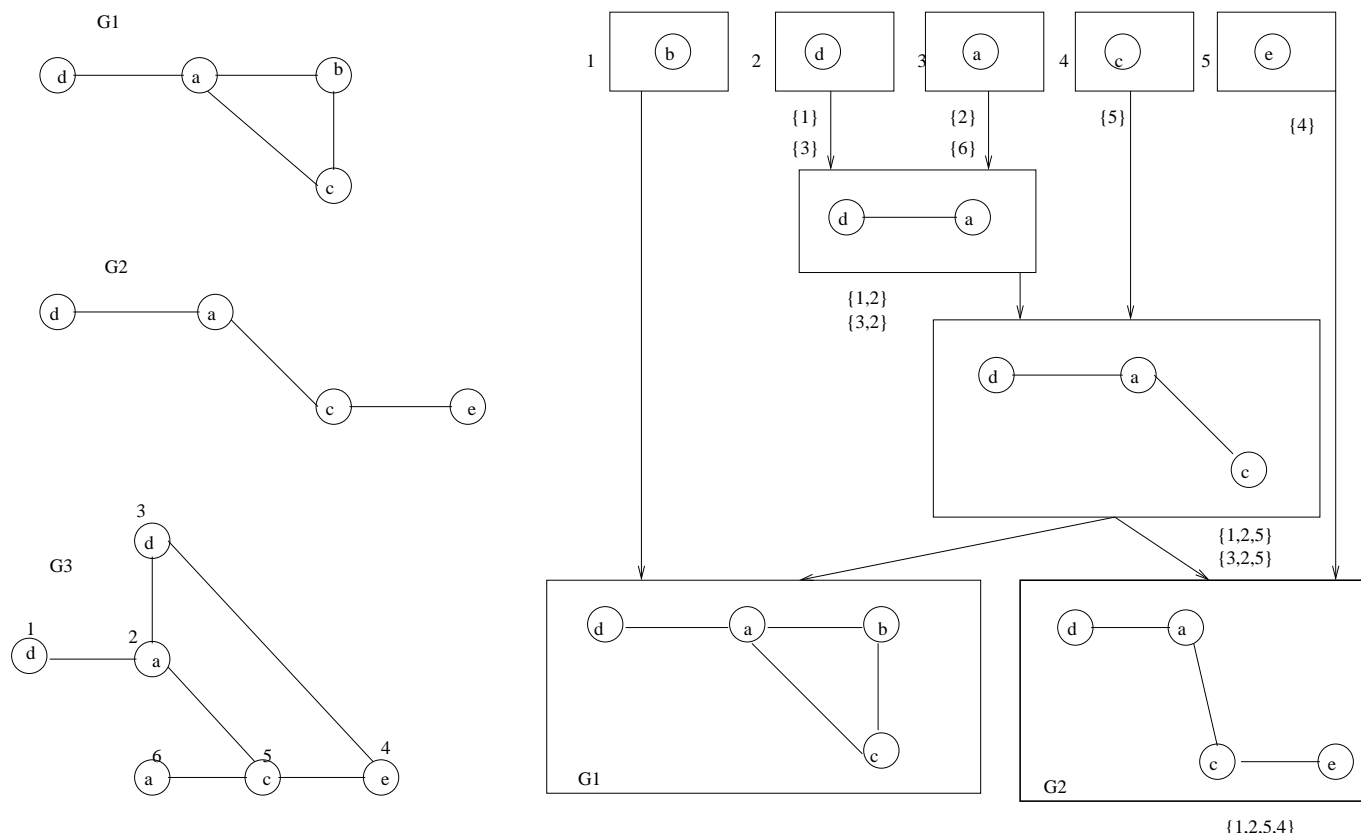


FIG. 1.8 – Exemple de décomposition en éléments communs de deux graphes G_1 et G_2 et des étapes de mise en correspondance avec un graphe G_3 . Une grande partie des opérations d'appariement n'est faite qu'une seule fois puisque le sous-graphe $\{d, a, c\}$ est commun à G_1 et G_2 (thèse de B. Messmer [22].)

Cette méthode a en particulier été adaptée dans les travaux de F. Fuchs [13] 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 1.9 montre un exemple de reconstruction obtenue.

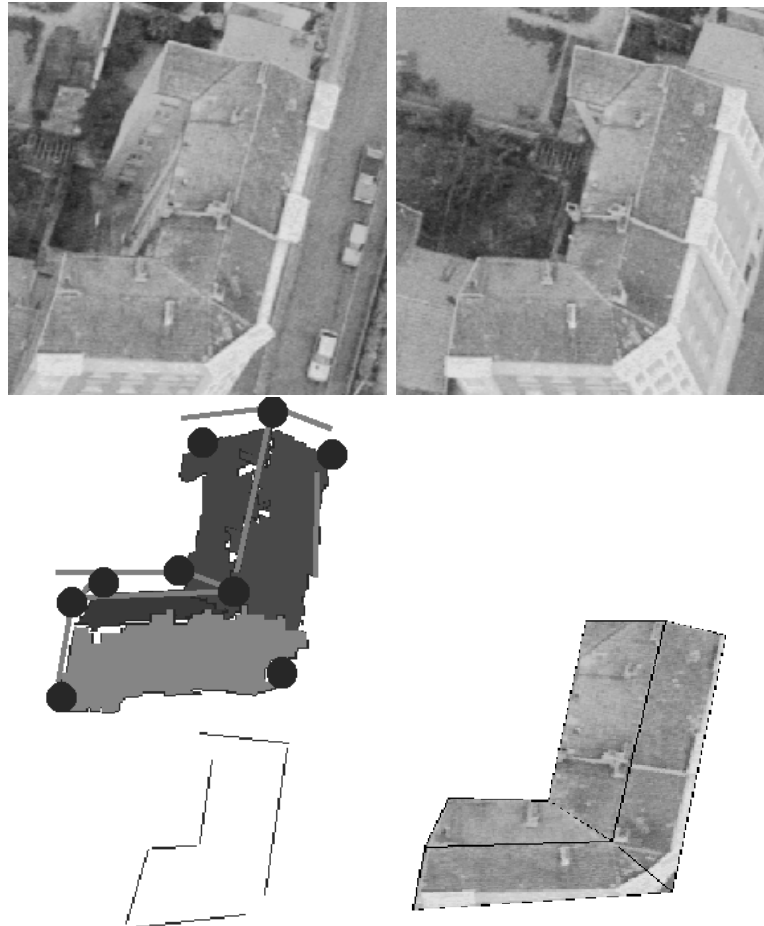


FIG. 1.9 – Exemple de reconstruction 3D par appariement entre un ensemble de graphes modèles et un graphe de données [13].

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.

1.3.3 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 [36] [9], et de J. Kittler [8]. 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 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 nœuds 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) [2].

Un exemple d'approche probabiliste [8]

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 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 [6]. 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 1.10. 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.

1.3.4 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.

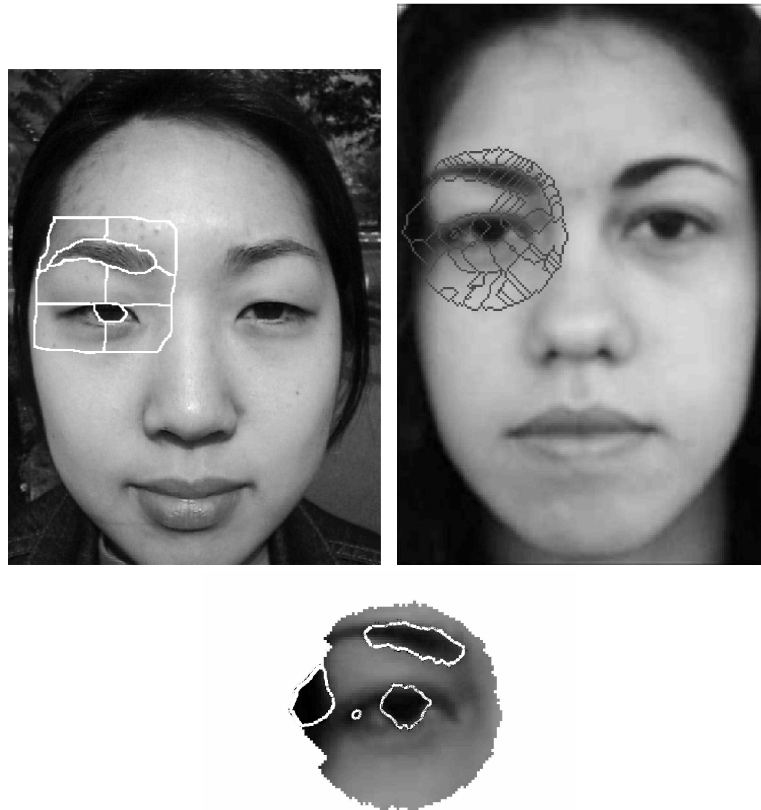


FIG. 1.10 – Graphe modèle (en haut à gauche) construit manuellement -chaque région correspond à un nœud, graphe des données construit par une sur-segmentation de l'image (à droite), résultat de la mise en correspondance (en bas) [6].

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 [27].

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 [37] [21] [19].

1.4 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.

Bibliographie

- [1] Special section on graph algorithms and computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10) :1049–1144, 2001.
- [2] E. Bengoetxea. *Mise en correspondance inexacte de graphes par algorithmes d'estimation des distributions*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, 2002.
- [3] 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.
- [4] H. Bunke and B. Messmer. Recent advances in graph matching. *International Journal of Pattern recognition and Artificial Intelligence*, 11(1) :169–203, 1997.
- [5] M. Burge, W. Burger, and W. Mayr. Recognition and learning with polymorphic structural components. *ICPR, A* :19–23, 1997.
- [6] 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.
- [7] I. Charon, A. Germa, and O. Hudry. *Méthodes d'optimisation combinatoire*. Collection Pédagogique de Télécommunication, Masson, 1996.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] K. Fu. A step towards unification of syntactic and statistic pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 200–205, 1983.

- [13] 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 structurelle*. PhD thesis, Université René Descartes - Paris V, 2001.
- [14] D. Ghahraman, A. Wong, and T. Au. Graph monomorphisms algorithms. *IEEE Transactions on Systems, man, and cybernetics*, 10(4) :189–196, 1980.
- [15] M. Gondran and M. Minoux. *Graphes et algorithmes*. Collection de la Direction des Etudes et Recherches d'Electricité de France, Eyrolles, 1979.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] W. Kropatsch and H. Bishof. *Digital Image Analysis*. Springer-Verlag, 2001.
- [21] 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.
- [22] B. Messmer. *Efficient graph matching algorithms for preprocessed model graphs*. PhD thesis, Universität Bern, 1996.
- [23] L. Miclet. *Méthodes structurelles pour la reconnaissance des formes*. CNET, Collection technique et scientifique des télécommunications, 1984.
- [24] 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.
- [25] T. Pavlidis. Representation of figures by labeled graphs. *Pattern Recognition*, 4 :5–17, 1972.
- [26] Aymeric Perchant. *Morphisme de graphes d'attributs flous pour la reconnaissance structurelle de scènes*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, 2000.
- [27] 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.
- [28] 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.

- [29] L. Shapiro and R. Haralick. Structural description and inexact matching. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 3(5) :504–519, 1981.
- [30] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8) :888–905, 2000.
- [31] sous la direction de H. Maitre. *Le traitement des images*. Hermes, Collection Traitement du Signal et des Images, 2003.
- [32] J. M. Tenenbaum. Experiments in interpretation guided segmentation. *Artificial Intelligence*, 8 :241–274, 1977.
- [33] 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.
- [34] J. Ullmann. An algorithm for subgraph isomorphism. *Journal of the association for computing machinery*, 23(1) :31–42, 1983.
- [35] O. Veksler. *Efficient graph-based energy minimization methods in computer vision*. PhD thesis, Cornell University, 1999.
- [36] R. Wilson and A. Cross and E. Hancock. Structural matching with active triangulation. *Computer Vision and Image Understanding*, pages 21–38, 1998.
- [37] 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.
- [38] 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.
- [39] E. K. Wong. Model matching in robot vision by subgraph isomorphism. *Pattern Recognition*, 25(3) :287–303, 1992.
- [40] 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.