# Chapitre 9

# Reconnaissance d'objets

## 9.1. Introduction

Dans les chapitres précédents nous avons proposé quelques méthodes pour résoudre le problème de la localisation d'un objet tri-dimensionnel à partir de données bi-dimensionnelles (chapitre 8) ou de données tri-dimensionnelles (chapitre 7). Plus précisément, toutes les méthodes que nous avons étudiées avaient ceci en commun qu'elles supposaient qu'un certain nombre de caractéristiques de l'objet étaient en correspondance terme à terme avec un nombre égal de caractéristiques issues des données. Sur la base d'appariements objets/données il a été possible de calculer une transformation entre le repère de l'objet et le repère des donnés. Le repère des données n'est autre que le repère du capteur qu'il s'agisse d'une seule caméra (données 2D) ou d'un capteur fournissant des données 3D.

Dans ce chapitre nous allons étudier une méthodologie permettant d'établir des appariements objet/données 3D/3D ou 3D/2D. Si un certain nombre de tels appariements existe on dira alors qu'une instance de cet objet se trouve dans les données, autrement dit, un objet de ce type est *vu* par le capteur ayant fourni les données. Ces appariements une fois établis on pourra alors localiser l'objet, soit calculer sa position et son orientation par rapport au repère du capteur. Le terme "reconnaissance d'objet" désignera en fin de compte l'ensemble des deux processus suivants :

***Appariement*** – établir une correspondance terme à terme entre des caractéristiques objet et des caractéristiques données ;

***Localisation*** – déterminer la position et l'orientation de l'objet dans le repère des données (du capteur).

## 9.2. Complexité

La tâche qui consiste à établir des appariements est difficile. Soit par exemple le cas où aussi bien les données que l'objet sont décrits par des caractéristiques du même type et soient $\mathcal{O}$ l'ensemble de caractéristiques objet et $\mathcal{D}$ l'ensemble de caractéristiques données.

Si $\mathcal{O}$ et $\mathcal{D}$ ont le même nombre d'éléments (ce qui sera rarement le cas !) et si $n$ est ce nombre, le nombre d'appariements possibles est :

$$n!$$

Cependant $\mathcal{O}$ et $\mathcal{D}$ ont rarement la même taille. Les données peuvent contenir des caractéristiques provenant d'autres objets. Soit $m$ la taille de $\mathcal{D}$ et $n$ la taille de $\mathcal{O}$ avec $m \geq n$. Le nombre d'appariement possible est alors :

$$C_m^n \ \ n!$$

Les caractéristiques d'un objet ne sont pas toutes présentes dans les données car :

- le capteur ne peut "voir" qu'une partie de l'objet (occlusions mutuelles entre les caractéristiques d'un même objet) ;
- l'objet peut être partiellement caché par un autre objet (occlusion mutuelle entre deux objets).

Dans ces conditions on peut espérer tout au mieux mettre en correspondance un sous-ensemble de $\mathcal{O}$ avec un sous-ensemble de $\mathcal{D}$. Le nombre d'appariements possibles devient alors beaucoup plus grand :

$$\sum_{i=1}^{n} C_n^i \ \ C_m^i \ \ i!$$

La reconnaissance est d'autant plus fiable que le nombre d'appariements est grand, voire le plus grand. Le problème à résoudre est donc le suivant : trouver le plus grand nombre d'appariements entre $\mathcal{O}$ et $\mathcal{D}$. Un tel appariement *maximal* ne peut être trouvé facilement. Même si on tient compte des propriétés des caractéristiques constituantes de $\mathcal{O}$ et $\mathcal{D}$, la complexité de ce problème est combinatoire. En pratique on se contentera d'une solution sous-optimale.

## 9.3. Appariement et isomorphisme de graphes

Une première approche possible consiste à traiter le problème d'appariement comme un problème d'isomorphisme de graphes. Un objet peut être décrit comme un ensemble de caractéristiques et des relations entre ces caractéristiques. L'objet peut alors être décrit par un graphe dans lequel un nœud représente une caractéristique et une arête représente une relation entre deux caractéristiques. Les données peuvent être décrites par un graphe de même nature. Le problème d'appariement de caractéristiqes devient alors un problème d'appariement de graphes, soit un problème d'isomorphisme de graphes.

Plus particulièrement il s'agira de résoudre le problème de trouver l'isomorphisme maximal de sous-graphes entre un sous-graphe de $G(\mathcal{O})$ et un sous-graphe de $G(\mathcal{D})$ où $G(\mathcal{O})$ est le graphe associé à l'ensemble $\mathcal{O}$. Si $n$ est le nombre de nœuds du graphe objet et $m$ est le nombre de nœuds du graphe données, $m \geq n$, la taille de l'espace des solutions est (comme auparavant) :

$$\sum_{i=1}^{n} C_n^i \ \ C_m^i \ \ i!$$

et le problème de trouver l'isomorphisme de sous-graphes maximal parmi tous ces isomorphismes de sous-graphes est un problème NP-complet [67].

Un certain nombre de travaux ont proposé des solutions au problème d'isomorphisme de graphes en général et au problème d'isomorphisme de sous-graphes maximal en particulier [71], [164]. Laurent Hérault [85] ainsi que d'autres auteurs proposent de traiter le problème d'isomorphisme de graphes comme un problème d'optimisation combinatoire – cela revient à chercher l'isomorphisme qui minimise une distance entre deux graphes ou deux sous-graphes. Si cette distance est assimilée à l'énergie potentielle d'un système physique on peut utiliser des algorithmes stochastiques qui convergent en un temps fini vers une configuration très proche de la configuration correspondant à l'énergie potentielle la plus basse.

Dans le cas de l'isomorphisme de graphes la distance a une expression simple. Dans le cas de l'isomorphisme de sous-graphes maximal, la distance a une expression plus complexe – il s'agit de la somme pondérée de plusieurs termes et en pratique à chaque cas particulier correspond une pondération différente. Des travaux de recherche sont en cours dans ce domaine.

## 9.4. Appariement et prédiction/vérification

Cette deuxième approche consiste à résoudre les problèmes d'appariement et de localisation *simultanément* dans le cadre du paradigme prédiction et vérification. Ce paradigme consiste essentiellement à faire une hypohtèse quant aux paramètres de position et d'orientation de l'objet par rapport au capteur et à vérifier que des appariements sont compatibles avec ces paramètres. Dans ses lignes les plus générales, ce paradigme peut être décrit comme suit :

- *Prédiction* :

– établir quelques appariements objet/données,

– calculer les paramètres de la transformation objet/données sur la base de ces appariements ainsi que les intervalles de confiance associés à ces paramètres,

– appliquer cette transformation à l'ensemble des caractéristiques objet de façon à pouvoir prédire lesquelles parmi ces caractéristiques sont susceptibles d'être vues par le capteur et quelle est leur position et orientation dans le repère du capteur.

- *Vérification* :

– pour chaque caractéristique objet ainsi prédite, sélectionner les caractéristiques données les plus proches et les plus ressemblantes. Former ainsi des nouveaux appariements objet/données tout en rejetant les appariements qui s'avèrent incorrects,

– fusionner chaque nouvel appariement avec les appariements précédents afin de mettre à jour les paramètres de la transformation objet/données ainsi que les intervalles de confiance associés.

La figure 9.1. illustre cette approche par un exemple lorsque les caractéristiques sont des points et lorsque les données sont 3D. L'objet à reconnaître (à droite) est un parallélipipède et les données (à gauche) contiennent une instance de cet objet caché partiellement par un autre objet, ainsi que du bruit. Dans ce cas il suffit de 3 appariements (en haut) pour estimer une transformation objet/données. Le choix de ces appariements initiaux est crucial. Ils ne peuvent être validés ... qu'en vérifiant que d'autres appariements produisent les mêmes paramètres de la transformation objet/données. Les points de l'objet sont transformés dans le repère des données (au milieu) sauf les points cachés par l'objet lui-même. Dans ce cas l'objet a 8 points mais 7 points seulement sont prédits visibles dans les données. La position de ces 7 points est prédite dans les données. A chacune de ces positions est également associé une tolérance qui dans ce cas est une sphère de rayon $r$.

Soit $\mathbf{m}_i$ un point de l'objet et soit $\hat{\mathbf{m}}_i$ l'image de $\mathbf{m}_i$ par rotation et translation :

$$\hat{\mathbf{m}}_i = \mathbf{R}\ \mathbf{m}_i + \mathbf{t}$$

Les points données $\mathbf{c}_i$ qui satisfont à :

$$||\hat{\mathbf{m}}_i - \mathbf{c}_i|| \leq r$$

sont des appariements potentiels de $\mathbf{m}_i$. Pour chaque point $\hat{\mathbf{m}}_i$ on sélectionne un point $\mathbf{c}_i$ parmi tous les candidats possibles. La figure 9.1. (en bas) montre 4 appariements parmi lesquels 3 seulement sont corrects (1, 3 et 4). En effet, l'appariement 2 correspond à un point de l'objet prédit visible mais caché en réalité par un autre objet. De ce fait le point a été incorrectement apparié avec un point des données appartenant à un autre objet. Ce genre de faux appariements arrive souvent lorsqu'on a affaire à des scènes complexes.

En pratique nous verrons comment on peut éviter ce genre de problème en effectuant à la fois une vérification *locale* de chaque appariement, une vérification *globale* d'un ensemble d'appariements ainsi que la mise en œuvre d'une structure algorithmique que recherche le meilleur ensemble d'appariements dans l'espace formé par tous les appariements.

Abordons d'un point de vue plus formel les phases de prédiction et de vérification pour les deux cas de reconnaissance auxquels on s'intéresse : 3D/3D et 3D/2D.

### 9.4.1. Prédiction et vérification 3D/3D

Dans ce cas aussi bien l'objet que les données sont tri-dimensionnels. Au chapitre 7 nous avons calculé la rotation et translation optimale entre deux ensembles 3D exprimés dans deux repères différents pour des points, des droites et des plans.

La phase de prédiction consiste donc dans ce cas à calculer les paramètres de rotation et translation utilisant un petit nombre d'appariements, le plus petit nombre étant 3 (pour les points et les plans) et 2 (pour les droites). Soient $\mathbf{R}$ et $\mathbf{t}$ la rotation et translation ainsi calculées.

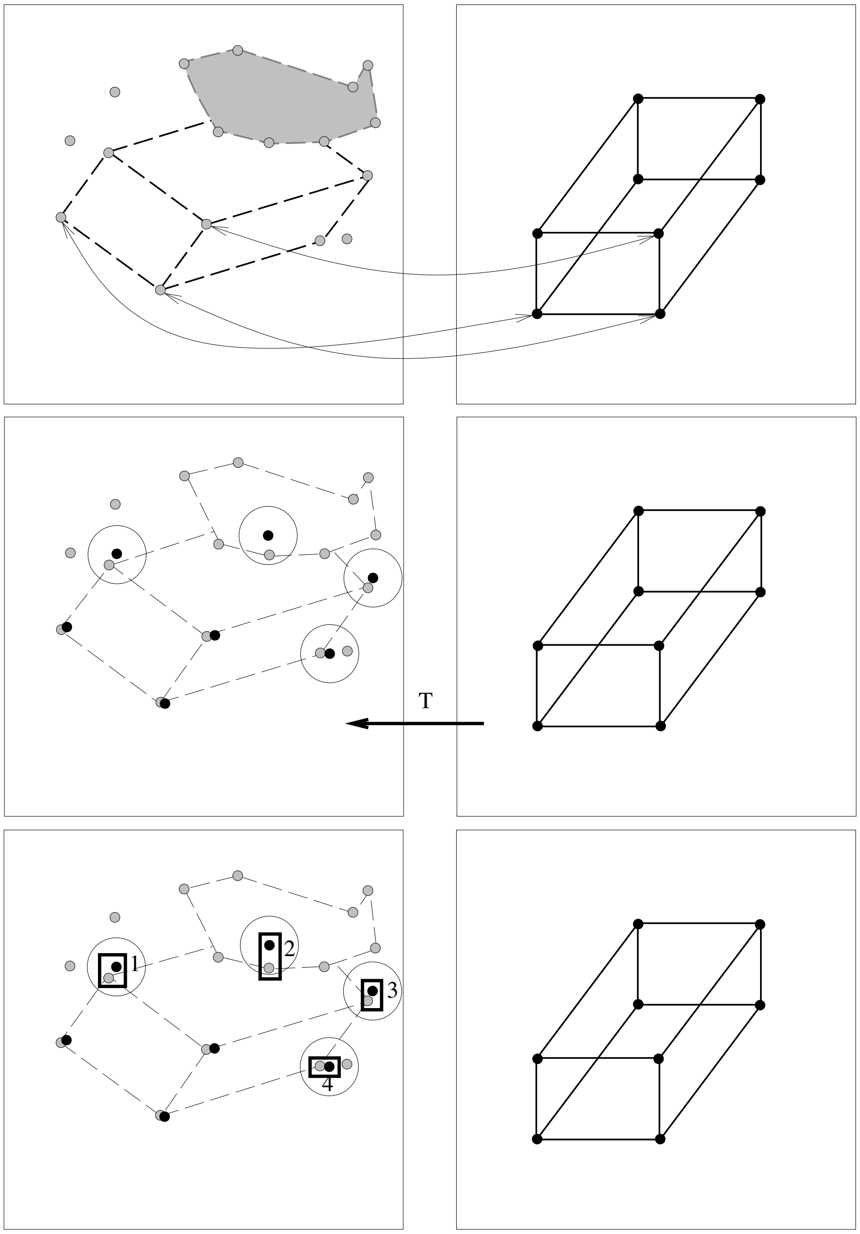


**Figure 9.1.** *Le principe de l'approche prédiction/vérification.*

La phase de vérification consiste à transformer d'autres caractéristiques de l'objet dans le repère des données en utilisant la transformation précédemment calculée et à confirmer ou rejeter la prédiction sur la base d'un certain nombre de mesures. Soit $p$ le nombre d'appariements initial et soit $k$ le nombre d'appariements à confirmer ou à rejeter. Si $n$ est le nombre de caractéristiques de l'objet on a nécessairement $p \leq k \leq n$. Soit par exemple le cas où les caractéristiques sont des segments de droite. Deux vérification sont possibles: une vérification individuelle (ou locale) de chaque appariement et une vérification globale d'un ensemble d'appariements.

### *9.4.1.1. Vérification individuelle*

Soient $\mathbf{v}_i$ et $\mathbf{p}_i$ les deux vecteurs décrivant la droite support du segment objet et $\mathbf{v}'_i$ et $\mathbf{p}'_i$ les vecteurs décrivant la droite support correspondante dans les données. Soit $\mathbf{m}_i$ le point milieu du segment objet et $\mathbf{m}'_i$ le point milieu du segment données. Soit $l$ la longueur du segment objet et $l'$ la longueur du segment données. Le segment objet est apparié avec le segment données si les critères suivants sont respectés (voir les équations [7.4] et [7.5] ainsi que la figure 9.2.) :

$$\begin{aligned}
||\mathbf{v}'_i - \mathbf{R}\ \mathbf{v}_i||^2 &\leq \epsilon_1 \\
||\mathbf{p}'_i - \mathbf{R}\ \mathbf{p}_i - \mathbf{t} + ((\mathbf{R}\ \mathbf{v}_i) \cdot \mathbf{t})\mathbf{R}\ \mathbf{v}_i||^2 &\leq \epsilon_2 \\
||\mathbf{m}'_i - \mathbf{R}\ \mathbf{m}_i - \mathbf{t}||^2 &\leq \epsilon_3 \\
(l'_i - l_i)^2 &\leq \epsilon_4
\end{aligned}$$

Lorsque $\epsilon_1$, $\epsilon_2$, $\epsilon_3$ et $\epsilon_4$ sont petits ces équations garantissent une très bonne ressemblance entre le segment "données" et le segment "objet". Si ces critères sont satisfaits, on peut alors utiliser ce nouvel appariement pour incrémenter le nombre d'appariements et mettre à jour les paramètres de la transformation objet-données.

### *9.4.1.2. Vérification globale*

Soient maintenant $k$ appariements, chaque appariement individuel satisfaisant aux critères qu'on vient de décrire. La qualité de cet ensemble d'appari-

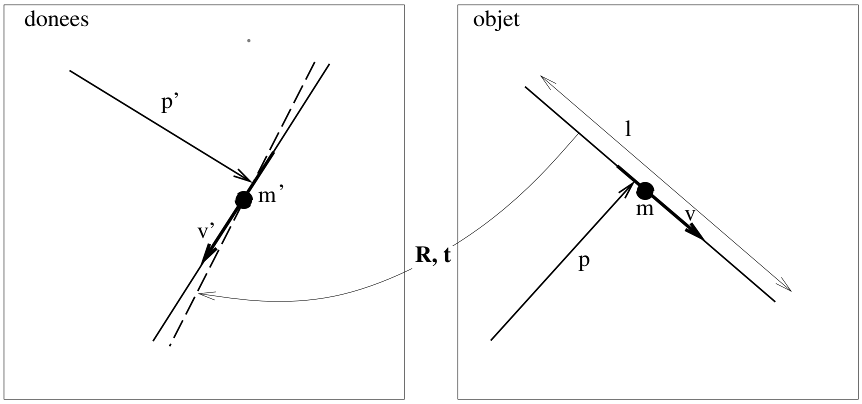


**Figure 9.2.** *Un segment de droite est représenté par un vecteur directeur $\mathbf{v}$, un vecteur de position $\mathbf{p}$, son point milieu $\mathbf{m}$ et sa longueur $l$. Moyennant la connaissance de la rotation et la translation on peut comparer un segment de l'objet avec un segment des données.*

ments peut être évalué en sommant sur les appariements :

$$
\begin{aligned}
q_1 &= \frac{1}{k}\sum_{i=1}^{k} \|\mathbf{v}'_i - \mathbf{R}^{\,k}\mathbf{v}_i\|^2 \\
q_2 &= \frac{1}{k}\sum_{i=1}^{k} \|\mathbf{p}'_i - \mathbf{R}^{\,k}\mathbf{p}_i - \mathbf{t}^k + ((\mathbf{R}^{\,k}\mathbf{v}_i)\cdot\mathbf{t}^k)\mathbf{R}^{\,k}\mathbf{v}_i\|^2 \\
q_3 &= \frac{1}{k}\sum_{i=1}^{k} \|\mathbf{m}'_i - \mathbf{R}^{\,k}\mathbf{m}_i - \mathbf{t}^k\|^2 \\
q_4 &= \frac{1}{k}\sum_{i=1}^{k} (l'_i - l_i)^2
\end{aligned}
$$

Dans ces équations $\mathbf{R}^{\,k}$ et $\mathbf{t}^k$ désignent la rotation et la translation correspondants à $k$ appariements, différentes de la rotation et de la translation calculées lors de l'étape de prédiction.

### 9.4.2. Prédiction et vérification 3D/2D

Lorsque les données sont 2D (une image) on peut obtenir une solution pour la rotation et la translation avec des points et/ou des segments de droites. Il faut un minimum de 3 points pour déterminer les paramètres de position et d'orientation de l'objet par rapport au capteur (la caméra) mais la solution n'est pas unique. Chaque solution ainsi obtenue constituera une prédiction qu'il faudra vérifier. De même, il faut au minimum 3 segments de droite pour déterminer la rotation et la translation mais là encore la solution n'est pas unique. Si l'on dispose d'un nombre de correspondances plus élevé la solution devient unique (voir chapitre 8) et on peut alors associer une seule prédiction à un ensemble de correspondances point/point ou droite/droite.

Même si une prédiction correspond à un nombre élevé de correspondances, rien ne prouve que la solution est correcte du point de vue de la reconnaissance d'objet (elle est correcte d'un point de vue numérique). Il faut donc envisager une méthode de vérification de la prédiction. Comme auparavant la vérification consiste à mesurer la distance entre un point de l'objet et son homologue dans l'image.

Soient $\mathbf{R}$ et $\mathbf{t}$ les paramètres associés avec une prédiction et soit $\mathbf{m}_i/\mathbf{m}'_i$ une correspondance point-objet/point-image n'appartenant pas à l'ensemble d'appariements associés avec la prédiction. Les coordonnées de la projection de $\mathbf{m}_i$ dans l'image sont données par :

$$\begin{pmatrix} su_i \\ sv_i \\ s \end{pmatrix} = \begin{pmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ 0^t & 1 \end{pmatrix} \begin{pmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{pmatrix}$$

où $\alpha_u$, $\alpha_v$, $u_0$ et $v_0$ sont les paramètres intrinsèques de la caméra. L'appariement $\mathbf{m}_i/\mathbf{m}'_i$ est vérifié si l'inégalité :

$$(u_i - u'_i)^2 + (v_i - v'_i)^2 \leq \epsilon$$

est vérifiée, $\epsilon$ étant petit. Pour vérifier $k$ appariements il suffit de sommer la grandeur qu'on vient de calculer :

$$\sum_{i=1}^{k} ((u_i - u'_i)^2 + (v_i - v'_i)^2)$$

et de s'assurer qu'elle est inférieure à un seuil.

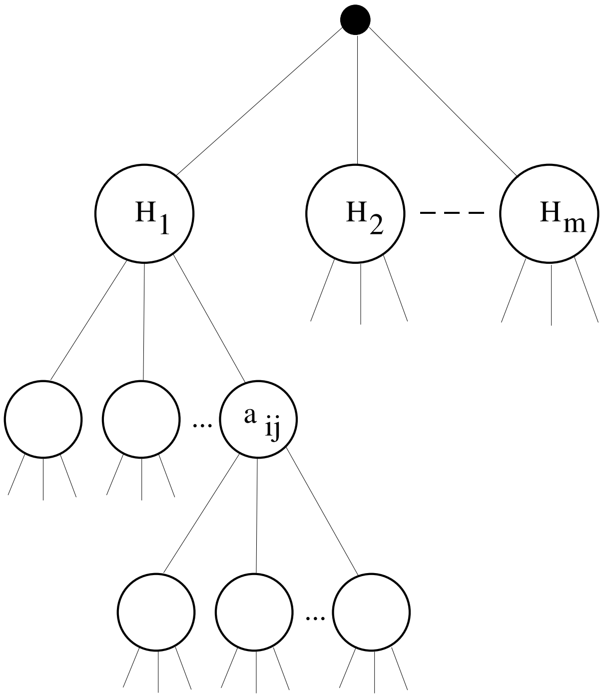


**Figure 9.3.** *La structure de l'arbre de recherche pour le paradigme prédiction/vérification.*

### 9.5. Arbre de recherche

La façon la plus répandue de réaliser en pratique le paradigme prédiction/vérification est par une recherche en *profondeur d'abord* d'une structure d'arbre (depth-first tree search) [10], [75], [76], [119], [21], [59], [113], [72], [87], [95]. La structure de cet arbre est celle de la figure 9.3.

Le premier niveau de cet arbre, juste après sa racine, est constitué de nœuds-hypothèses, soit $H_1$, $H_2$, ... $H_m$. Chaque hypothèse comporte en fait le nombre minimum d'appariements permettant de calculer une transformation objet-données. Puisqu'une telle hypothèse détermine entièrement une position et une orientation de l'objet par rapport au capteur, on peut lui associer "une vue" de l'objet, une vue étant définie par un ensemble de caractéristiques de l'objet susceptibles d'être visibles pour une orientation donnée. Ce sont ces caractéristiques visibles sous une hypothèse qui vont servir à la vérification de cette hypothèse.

Les niveaux suivants de l'arbre sont constitués de nœuds tels que chaque nœud représente un seul appariement. Un nœud $a_{ij}$ décrit une caractéristique $m_i$ de l'objet mise en correspondance avec une caractéristique $c_j$ des données. Si une caractéristique objet n'a pas de correspondant dans les données, soit parce qu'elle n'est pas visible, soit parce que les données sont défaillantes on notera dans l'arbre un nœud "nul".

A partir d'une hypothèse comportant $p$ appariements on considère une nouvelle caractéristique objet, $m_{p+1}$ ainsi que l'ensemble des caractéristiques données qui peuvent lui correspondre, $c^1_{p+1}$, ... $c^k_{p+1}$, figure 9.4. Soit le premier de ces appariements $m_{p+1}/c^1_{p+1}$ grâce à qui on met à jour les paramètres de la transformation associée à l'hypothèse en cours ainsi que les intervalles de tolérance associés avec ces paramètres. Si cet appariement est accepté on monte dans l'arbre au niveau $p+2$ et ainsi de suite. Un chemin dans un arbre comporte un nœud par niveau. Un chemin représente donc une solution, soit un ensemble d'appariements :

$$m_1/c_1 - \ldots - nul - \ldots - m_i/c_i - \ldots$$

Lorsqu'un chemin n'est pas satisfaisant, la recherche fait un retour en arrière jusqu'au dernier point de choix (backtracking). Les points de choix apparaissent dès lors qu'il y a plusieurs interprétations possibles pour la même caractéristique objet.

Les nœuds "nuls" permettent de "sauter" une caractéristique objet soit parce que cette caractéristique n'a pas de correspondant, soit parce que le fait de lui attribuer un correspondant est trop pénalisant. Par exemple, sur la figure 9.1. le fait d'accepter l'appariement 2 (qui est faux) peut pénaliser fortement le coût de la reconnaissance. Dans cet exemple, la solution 1-nul-3-4 est peut-être meilleure que la solution 1-2-3-4.

La question cruciale qui se pose avec ce type de méthodes est de savoir sur quelles bases décide-t-on que le nombre d'appariements d'un chemin de l'arbre est une bonne solution. Une condition nécessaire est que les critères numériques développés auparavant (sections 9.4.1. et 9.4.2.) soient satisfaits. Une autre condition est que le nombre d'appariements soint le plus grand possible, voire le plus grand. C'est cette deuxième condition que nous allons analyser maintenant.

Soit $H$ une hypothèse en cours de vérification et soit $n_H$ le nombre de

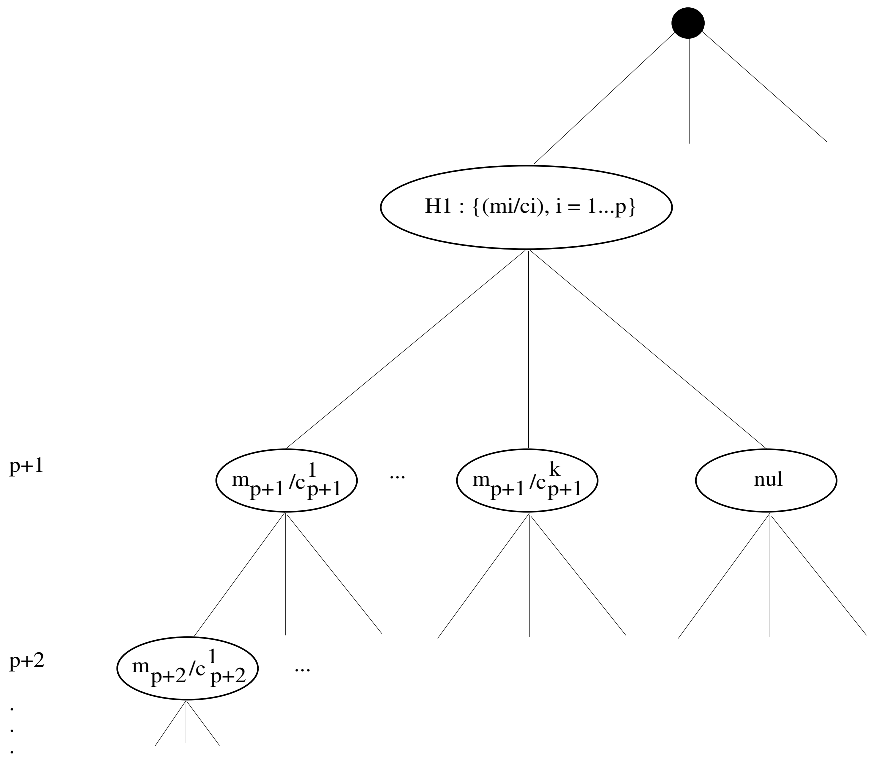


**Figure 9.4.** *A partir d'une hypothèse comportant p appariements, chaque niveau dans l'arbre de recherche correspond à une* $(p+i)^{ème}$ *caractéristique de l'objet en cours de vérification.*

caractéristiques objet prédites visibles grâce aux paramètres de cette hypothèse. Si $n$ est le nombre total de caractéristiques de l'objet on a $n_H < n$. En pratique ces $n_H$ caractéristiques ne seront pas toutes vérifiables dans les données pour des raisons que nous avons déjà discutées. On sera donc amené à accepter une solution telle que le nombre d'appariements $k$ qui lui est associé est inférieur à $n_H$. Une solution sera acceptable si :

- $k$ est supérieur à $m_H$, le nombre *minimum* de caractéristiques qui doivent être présentes pour accepter l'hypothèse $H$ et $k$ doit donc satisfaire à : $m_H \leq k \leq n_H$ et

- parmi ces $k$ appariements il y a des caractéristiques objet "clés", c'est à dire des caractéristiques objet dont la présence dans l'image est indispensable.

Par la suite nous analysons comment on peut tenir compte de ces deux critères pour accélérer la recherche.

### 9.5.1. Abandon d'un chemin

A partir du premier critère qu'on vient d'établir on peut envisager une stratégie très simple pour accélérer la recherche dans l'arbre. L'idée est de pouvoir prédire le nombre d'appariements d'un chemin et d'abandonner le parcours de ce chemin s'il est peu probable d'améliorer la solution courante.

Remarquons d'abord qu'à chaque hypothèse correspond un sous-arbre. Soit donc un tel sous-arbre dont le nœud de départ est un nœud-hypothèse – l'hypothèse $H$. Si cette hypothèse comporte $p$ appariements et si le nombre total de caractéristiques prédites visibles par cette hypthèse est de $n_H$, la profondeur du sous-arbre est alors précisément $n_H - p$. Supposons par ailleurs que $p$ est inférieur à $m_H$.

Soit $i$ la profondeur du nœud courant du sous-arbre, $p < i < n_H$ et supposons que l'on a déjà apparié $s_i$ caractéristiques. $s_i$ est inférieur ou égal à $i$ car le chemin courant jusqu'à la profondeur $i$ peut comporter des nœuds nuls (pas d'appariement). Le nombre de nœuds restant à parcourir le long de ce chemin est $n_H - i$ et on peut donc espérer, tout au mieux, trouver $n_H - i$ appariements. On peut donc estimer la taille de la meilleure solution le long de ce chemin, soit $[s_i + (n_H - i)]$ appariements. Si $s_i + (n_H - i) < m_H$ alors cette solution ne sera pas satisfaisante et ceci peut être prédit à la profondeur $i$ sans perdre le temps d'aller jusqu'au bout du chemin. En conclusion :

*Si, à un stade quelconque du parcours d'un sous-arbre associé à une hypothèse $H$, le nombre d'appariements ($s_i$) de la solution courante augmenté du nombre de caractéristiques qui restent à vérifier est inférieur à $m_H$, il faut alors abandonner le chemin courant et revenir au dernier point de choix car cette solution ne sera pas satisfaisante.*

Ce concept d'abandon de parcours de chemin est illustré sur la figure 9.5. Le même concept peut être utilisé dès qu'on estime qu'on ne peut pas améliorer la meilleure solution courante. Soit en effet $k_{max}$ le nombre d'appariements de la meilleure solution disponible à un moment donné au cours de la recherche dans l'arbre. On abandonne la recherche si on se rend compte que, lors du parcours d'un chemin, on ne pourra pas améliorer cette solution. On a donc :

*Si, à un stade quelconque du parcours de l'arbre le nombre d'appariements ($s_i$) de la solution courante augmenté du nombre de caractéristiques qui restent à vérifier est inférieur à $k_{max}$, il faut alors abandonner le chemin courant et revenir au dernier point de choix car cette solution ne sera pas meilleure que la meilleure solution disponible.*

### 9.5.2. Abandon d'une hypothèse

La recherche sera d'autant plus rapide que l'arbre aura un petit nombre de nœuds. Le nombre de nœuds d'un arbre dépend crucialement du nombre de nœuds de ses premiers niveaux. Le premier niveau étant constitué d'hypothèses, moins il y aura d'hypothèses prédites, plus la recherche sera efficace.

Soit une hypothèse courante $H$ à qui sont associés des appariements ainsi que des paramètres de position et orientation. Ces derniers paramètres caractérisent géométriquement l'hypothèse. Si ces paramètres coïncident avec les paramètres d'une hypothèse déjà traitée, fut-elle bonne ou mauvaise, il y a de fortes chances que l'hypothèse $H$ qu'on s'apprête à vérifier produise une solution équivalente. En effet plusieurs ensembles d'appariements différents peuvent produire la même solution géométrique : ceci est dû aux symétries inhérentes à beaucoup d'objets.

### 9.5.3. Ordonnancement des nœuds

L'ordre dans lequel on considère les caractéristiques de l'objet peut avoir une influence considérable sur la taille de l'espace de recherche. On peut par

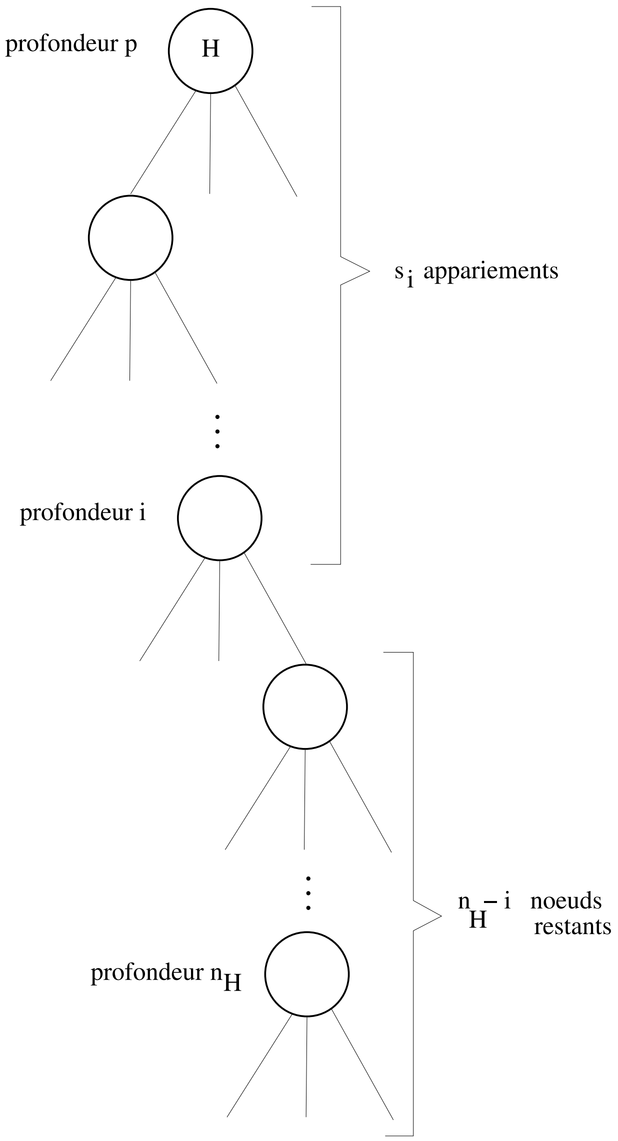


**Figure 9.5.** *A tout moment on peut abandonner le parcours d'un chemin si les nœuds restants ne sont pas en nombre suffisant pour constituer une solution ou pour améliorer la meilleure solution.*

exemple classifier les caractéristiques d'un objet de la plus discriminante à la plus ambiguë. Dans le cas de segments de droite, cela revient à classer les segments par leur longueur, du plus long au plus court. En effet, un segment long a plus de chances d'être vu dans l'image et a intrinsèquement moins de correspondants. Cette dernière propriété est due au phénomène d'obstruction. Un segment de l'objet, pouvant être partiellement caché, peut potentiellement correspondre à des segments donnés qui lui sont supérieurs ou égaux en longueur.

Supposons qu'on divise les segments de l'objet en deux classes : segments longs et segments courts et supposons que les segments longs constituent les premiers niveaux de l'arbre, juste au-dessus de la racine et les segments courts consituent les derniers niveaux de l'arbre. Soit $i$ la profondeur de l'arbre à la jonction entre les segments longs et les segments courts. Si l'on a constaté que les segments longs sont présents dans les données (disons à 75%) cela veut dire qu'on est peut-être sur le bon chemin. Si la plupart de ces segments sont absents des données (disons à 50%) cela veut dire qu'on devrait abandonner cette hypothèse.

## 9.6. Un exemple : 3DPO

Le système 3DPO (Three Dimensional Part Orientation) a été développé à SRI International en 1983 par l'un des auteurs en collaboration avec R. Bolles et M. J. Hannah [22], [24], [21], [88]. Il s'agit d'une méthode pour reconnaître et localiser un objet 3D à partir de données 3D "denses" obtenues grâce à un capteur stéréoscopique actif, chapitre 5, section 5.9. et figure 5.18. 3DPO détermine la position et l'orientation d'un objet dans le repère du capteur ainsi que la position relative de cet objet par rapport à d'autres objets : quel objet se trouve au-dessus de la pile et est-il partiellement caché par d'autres objets ?

Pour répondre à cette question 3DPO procède en trois étapes. La première étape consiste en des traitements bas niveau des données afin d'extraire des caractéristiques propices pour la reconnaissance. La deuxième étape consiste en une procédure prédiction/vérification d'appariement entre les données et le modèle. La troisième étape détermine la configuration globale de la scène analysée en comparant les données avec une image synthétique produite à partir du modèle de l'objet et des paramètres de position et d'orientation précédemment calculés.

La figure 9.6. montre une image 3D d'un ensemble d'objets à reconnaître et

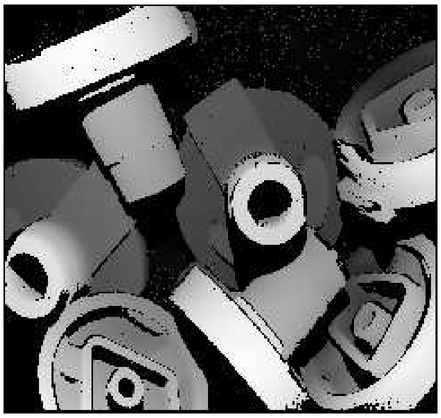

**Figure 9.6.** *Une image de profondeur obtenue grâce à un capteur stéréoscopique actif. Les pixels sombres codent des points éloignés et les pixels clairs codent des points proches du capteur.*

à localiser. Dans cette image les niveaux de gris codent la hauteur en chaque pixel, les pixels sombres correspondant à des points éloignés et les pixels clairs à des points proches du capteur. Cette image a été obtenue *tranche par tranche* ou ligne par ligne en déplaçant le plan de lumière du capteur transversalement par rapport à la scène.

Cette image est tout d'abord traitée comme une image 2D afin d'extraire les contours. A chaque point de contour on associe ensuite ses coordonnées 3D ce qui permet finalement d'extraire des arêtes 3D. La figure 9.7. montre les contours 2D et les contours 3D. Ces contours sont segmentés de façon à obtenir des arêtes circulaires et des arêtes droites, une arête étant définie comme l'intersection de deux surfaces (deux plans ou un plan et un cylindre). La figure 9.8. montre une arête droite et une arête circulaire. Plus précisément, une arête droite est définie par :

- longueur,
- position,
- orientation de ses facettes

et une arête circulaire est définie par :

- rayon,
- longueur,

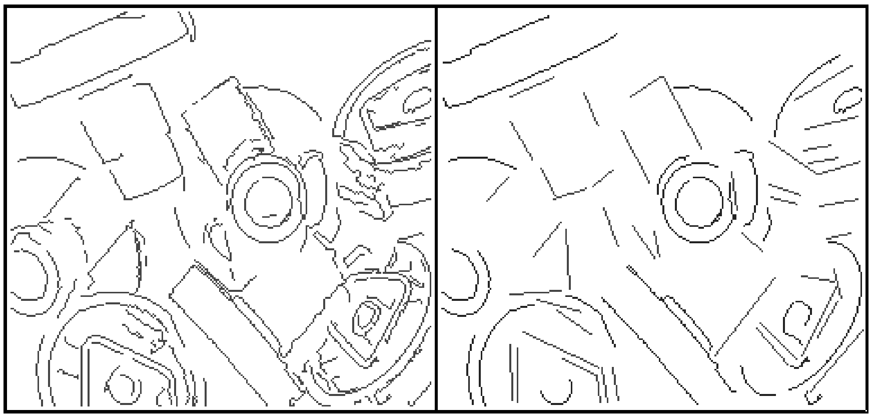

**Figure 9.7.** *Les contours 2D (gauche) et 3D (droite) extraits d'une image de profondeur.*

- position du centre,
- orientation du plan contenant l'arête.

La figure 9.9. montre des cercles et des droites correspondant à des arêtes circulaires et droites extraites le long des contours. Un système de modélisation géométrique a été utilisé pour décrire l'objet à identifier. Ce modèle contient des informations de type géométrique et topologique (ou relationnelles). Le modèle géométrique est tout simplement constitué d'une liste d'arête droites et circulaires telles qu'on vient de les décrire. Le modèle topologique décrit la manière dont ces arêtes sont reliées entre elles. La figure 9.10. montre le modèle de l'objet à reconnaître.

L'étape de reconnaissance est basée sur le paradigme prédiction/vérification qu'on vient de décrire. On peut remarquer qu'un appariement arête-modèle/arête-image définit 5 parmi les 6 paramètres de position et orientation associés avec une hypothèse. Dans le cas d'une arête droite la translation le long de l'arête n'est pas définie et dans le cas d'une arête circulaire la rotation autour de l'axe du cylindre correspondant n'est pas défini (figure 9.8.). A partir d'une telle hypothèse la vérification est matérialisée par une recherche en profondeur dans un arbre afin de confirmer ou d'infirmer une hypothèse. La figure 9.11. montre le résultat de la reconnaissance et de la localisation des 7 objets présents dans la scène. A partir de ces 7 objets prédits présents dans la scène, du modèle géométrique associé et de la position et l'orientation de chaque objet,

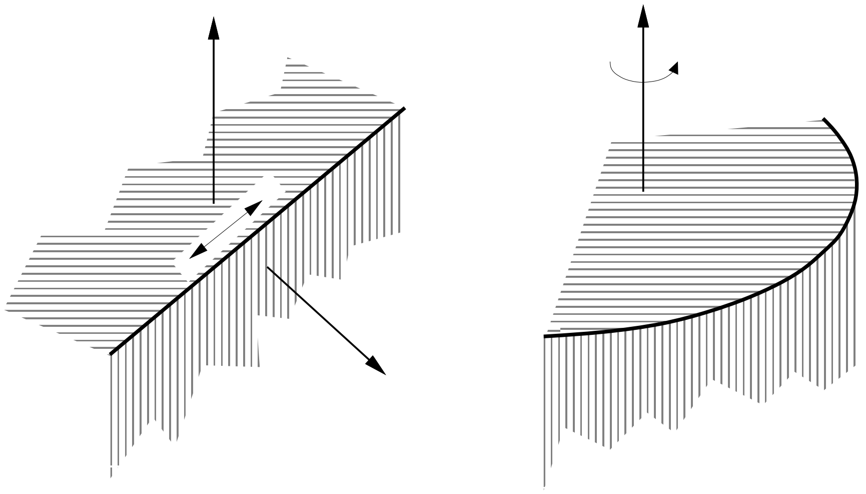

**Figure 9.8.** *Les deux caractéristiques extraites des images de profondeur par le système 3DPO.*

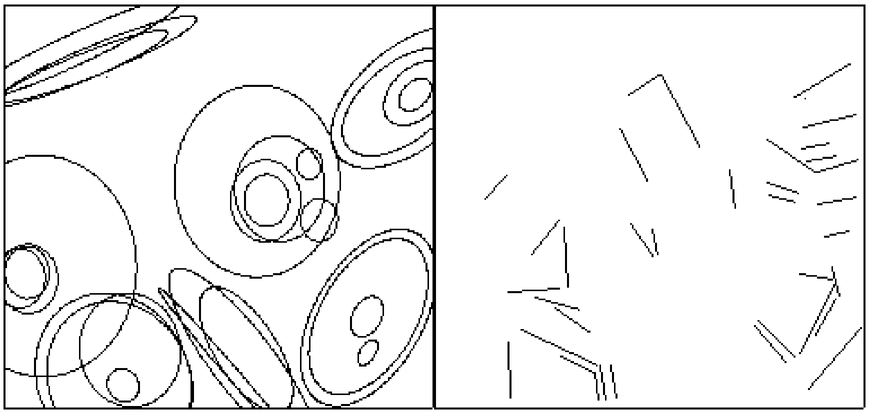

**Figure 9.9.** *Les arêtes circulaires et droites détectées dans l'image de profondeur.*

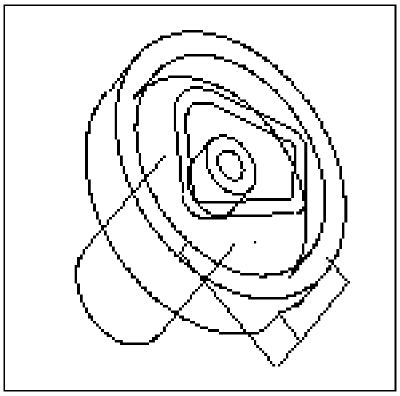

**Figure 9.10.** *Une vue "fil de fer" du modèle géométrique de l'objet à reconnaître. Toutes les arêtes de cet objet sont modélisées par des arcs de cercle et des segments de droite.*

on construit une image de profondeur *synthétique* qui est comparée à l'image de profondeur de la scène. Cette comparaison permet de décider pour chaque objet quels sont les objets qui le cachent. La figure 9.12. montre l'image synthétique ainsi obtenue.

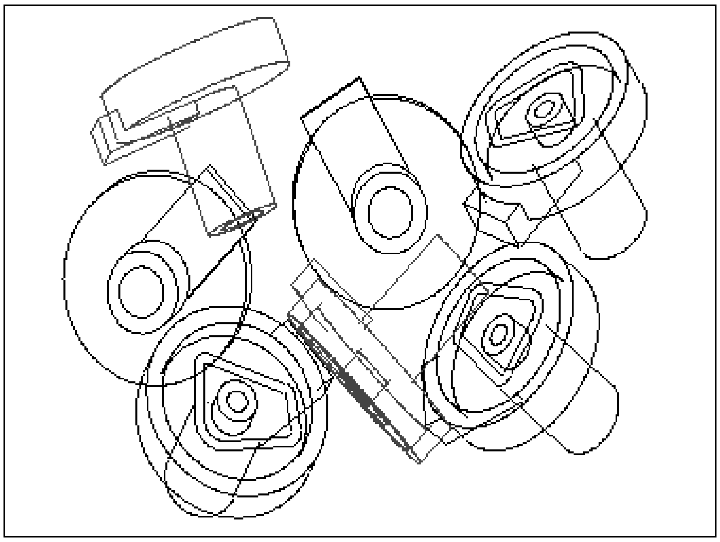

**Figure 9.11.** *Résultat de la localisation de 7 objets identiques dans la scène observée. La précision de la localisation dépend du nombre d'arêtes visibles de chaque objet ainsi que de la précision avec laquelle chaque arête est détectée dans l'image.*

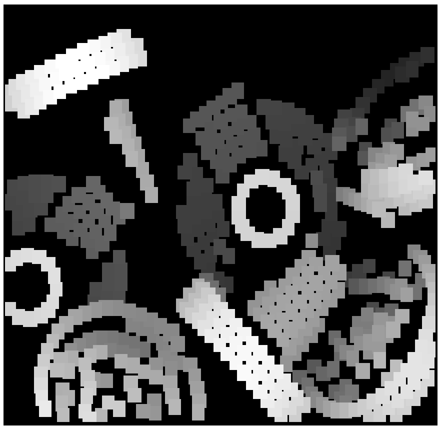

**Figure 9.12.** *Image synthétique obtenue en utilisant la position et l'orientation précédemment calculées et en éliminant les points d'un objet cachés par les points d'un autre objet.*