Chapitre 1

Visibilité dans le plan

Dans ce chapitre nous indiquons où se situent nos travaux – utilisation de la cohérence spatiale et temporelle dans les calculs de visibilité – par rapport aux problèmes de visibilité dans le plan. Nous exposons ensuite les structures de données – graphe et complexe de visibilité – au centre de ces travaux et les techniques algorithmiques – balayage et dualité – principales que nous utilisons.

— Seurhăne, ma Seurhăne ! Ne vois tu rien venir ?
Astérix chez Rahăzade.

1.1 Nature des problèmes de visibilité dans le plan

Les problèmes de visibilité font partie des problèmes fondamentaux de la géométrie algorithmique et de synthèse d'image. Dans les problèmes de planification de trajectoire par exemple, les objets d'une scène sont des obstacles à éviter et le calcul d'une trajectoire fait intervenir les relations de visibilité entre les objets de la scène.

Si nous cherchons plutôt à visualiser une scène, seuls les objets visibles doivent apparaître à l'écran : il vaut mieux alors éviter de faire des calculs coûteux sur les objets cachés car le temps passé dans ces calculs n'aura pas d'incidences sur l'affichage de la scène.

Nous retrouvons aussi des problèmes de visibilité aussi au cœur des problèmes de surveillance ou d'éclairage.

Une grande partie des problèmes de planification de trajectoire concerne divers véhicules qui se déplacent au sol, c'est-à-dire sur un plan. Bien que l'image de synthèse concerne principalement des scènes dans l'espace 3D, certains problèmes de visualisation ont une base 2D : les calculs effectués pour afficher ce que voit un piéton se déplaçant dans une ville font intervenir le plan (2D) de la ville. Les problèmes de surveillance sont eux-aussi principalement des problèmes 2D. L'étude des relations de visibilité dans le plan n'est donc pas limitée à un seul intérêt théorique.

Les relations de visibilité dans le plan ont d'abord été étudiées sur des objets simples, notamment sur des ensembles de segments disjoints. Certains algorithmes ont même été adaptés à partir d'algorithmes opérant sur des ensembles de points (calcul de graphe de visibilité d'Edelsbrunner et Guibas [EG86] par exemple). Le cas de scènes polygonales a été aussi abordé, mais beaucoup d'algorithmes ne considèrent en fait que des scènes composées d'un seul polygone, où l'espace libre de la scène est limité à l'intérieur du polygone (calculs de visibilité de Chazelle et Guibas [CG89] ou de Heffernan et Mitchell [HM90] par exemple).

Notre travail porte sur des scènes polygonales, c'est-à-dire des scènes composées de polygones simples ayant deux à deux une intersection nulle.

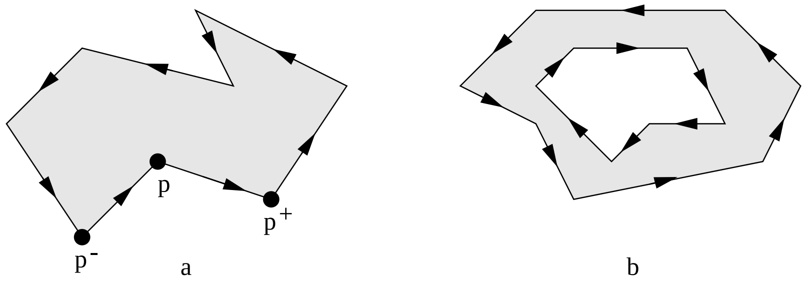

FIG. 1.1 – *Définition des polygones.* **a.** *Orientation des sommets.* **b.** *Polygone avec trou.*

Nous orientons les sommets d'un polygone de telle façon que le polygone soit locale-ment à gauche de deux sommets consécutifs. Pour un sommet p nous notons p^+ (resp. p^-) le sommet suivant (resp. précédent) sur le polygone (figure 1.1a). Les polygones ont deux à deux une intersection nulle, mais un polygone peut en contenir un autre qui, s'il est correctement orienté, forme un trou (figure 1.1b). Ces trous peuvent à leur tour contenir d'autres polygones...

Un polygone peut représenter un mur délimitant un bâtiment, mur dont les pans peuvent avoir des caractéristiques différentes (couleur, matière, réflectance, etc.). Ces pans devront être considérés individuellement lors de l'affichage de la scène par exemple. Il est alors plus facile de visualiser les polygones si nous avons directement la liste des côtés visibles à afficher plutôt que de la calculer à partir de la liste des polygones. Un côté de polygone peut aussi représenter une fenêtre, et nous voulons alors calculer la partie de la scène visible depuis ce côté seulement et non pas la partie visible depuis tout le polygone.

Par conséquent, les objets élémentaires que nous considérons dans les relations de visibilité ne sont pas les polygones eux-mêmes, mais les côtés de polygone. Pour nous, un polygone est un objet composite composé d'éléments ayant chacun leurs propriétés propres. De plus, ce point de vue permet de ne pas se préoccuper de la convexité ou non des polygones dans les algorithmes que nous étudions.

Les divers problèmes de visibilité étudiés peuvent être regroupés en deux catégories générales : calculs globaux et calculs locaux de visibilité.

1.1.1 Calculs globaux de visibilité

Les calculs globaux de visibilité concernent les calculs où les données du problème sont constituées par la scène elle-même : le programme ne prend en entrée que les coordonnées des objets définissant la scène, effectue des calculs de visibilité sur les objets de la scène et renvoie les résultats du calcul global demandé.

Une grande catégorie de ces problèmes concerne les problèmes de gardiennage, appelés *problèmes de galerie d'art* en référence à la surveillance des œuvres d'art dans les musées. L'énoncé type de ces problèmes est :

> *étant donné un polygone simple de n côtés, combien faut-il mettre de gardes aux sommets du polygone pour que tout point de la frontière intérieure du polygone soit visible d'au moins un garde, et comment placer ces gardes ?*

La réponse à cette question a été donnée par Chvátal [Chv75] qui a montré que $\lfloor n/3 \rfloor$ gardes étaient toujours suffisants et parfois nécessaires. Avis et Toussaint [AT81] ont donné un algorithme de placement qui s'exécute en temps $O(n \log n)$.

Divers variantes de ce problème ont été étudiées : spécialisation pour des polygones particuliers (en étoile, orthogonaux...), gardes mobiles le long d'une arête, surveiller l'extérieur du polygone, etc. Le livre d'O'Rourke [O'R87] décrit en détails les divers problèmes de galerie d'art.

Des problèmes similaires se posent dans des calculs d'éclairage, où des sommets, des côtés de polygone, représentent des lampes illuminant l'intérieur du polygone.

Le calcul du graphe de visibilité, qui consiste à calculer toutes les paires de sommets de la scène mutuellement visibles, est un autre problème de calcul global de visibilité très étudié (il permet ensuite de faire des calculs de planification de trajectoire). Nous étudions ce calcul en détail dans la section 1.4.

1.1.2 Calculs locaux de visibilité

Les calculs locaux concernent plutôt le calcul de réponses à des requêtes locales de visibilité. La scène fait partie du contexte, et les données du problème sont constituées des données supplémentaires qui définissent la requête.

Les requêtes les plus courantes concernent le calcul des objets vus depuis un point de vue, que ce soit dans le contexte de calculs discrets comme dans le lancer de rayons ou dans le contexte de calculs continus comme dans le calcul d'une vue autour d'un point.

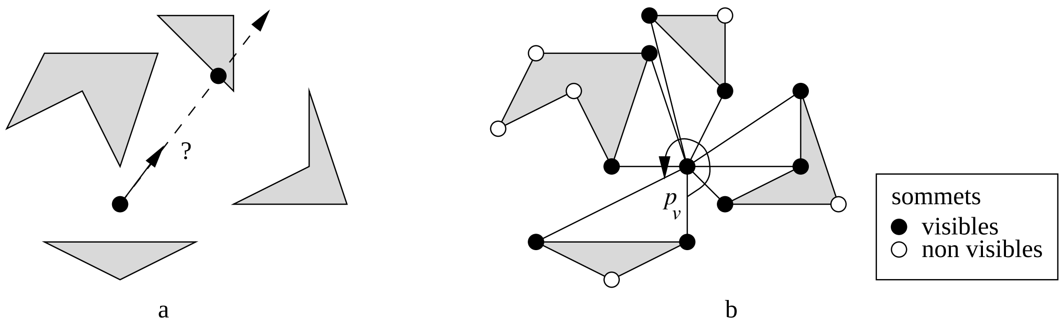

FIG. 1.2 – *Requêtes locales de calcul de visibilité.* **a.** *Calcul discret : lancer de rayons.* **b.** *Calcul continu : calcul de vue.*

L'énoncé d'une requête de lancer de rayons est :

> *soit un point et une direction de vision, quel est le premier objet de la scène intersecté par la demi-droite issue du point dans la direction donnée ?*(figure 1.2a)

Le lancer de rayons est utilisé en synthèse d'image pour visualiser une scène. L'image à calculer est déterminée par un point de vue et un écran de projection : c'est l'image de la scène obtenue sur l'écran quand elle est regardée à partir du point de vue. L'image est discrétisée en pixels (éléments de couleur uniforme) et pour la calculer, un rayon issu du point de vue et passant par le centre de chaque pixel est lancé dans la scène. La couleur résultante d'un pixel est celle du point de l'objet de la scène coupé par le rayon correspondant.

Ce calcul est discret : il ne concerne la visibilité que d'un rayon. Le calcul doit être refait à chaque fois que la direction de vision est modifiée, même si le point de vue reste le même.

La vue – ou polygone de visibilité – autour d'un point permet d'avoir une description continue dans l'espace objet de la visibilité autour du point de vue :

> *la vue autour d'un point p_v est l'ensemble des sommets de polygone de la scène vus depuis ce point et classés par ordre angulaire.* (figure 1.2b)

La vue code les limites de visibilité autour du point de vue : entre deux sommets consécutifs de la vue, l'objet vu depuis p_v est le même. Cette description est une description continue de la visibilité depuis ce point : elle ne fait intervenir aucune discrétisation arbitraire.

Les calculs continus permettent parfois de résoudre plus efficacement des problèmes d'abord résolus par une série de calculs discrets. Par exemple, le calcul d'une image en lancer de rayons nécessite de lancer un rayon par tous les pixels de l'image alors que l'image résultante peut directement être obtenue par un calcul de vue : il suffit de déterminer la projection sur l'écran des limites de visibilité.

Nous venons de décrire des calculs statiques, où tous les objets concernés sont immobiles. Les calculs de visibilité ont aussi une grande importance dans des situations dynamiques, où les données du problème (objets de la scène, point de vue) varient au cours du temps.

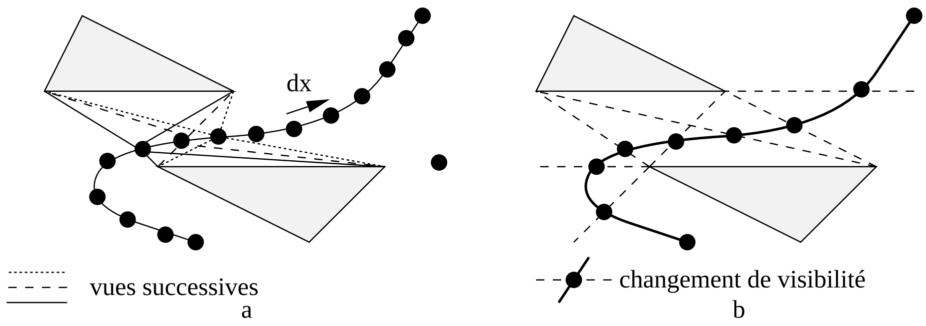

FIG. 1.3 – *Calcul de vue le long d'une trajectoire.* **a.** *Calcul discret : recalcul à chaque déplacement dx.* **b.** *Calcul continu : calcul des limites de visibilité.*

La simulation d'une personne en mouvement nécessite de pouvoir calculer la vue autour d'un point se déplaçant le long d'une trajectoire. Là aussi deux approches peuvent être envisagées pour ce calcul. Dans l'approche discrète, la vue est entièrement recalculée à chaque déplacement élémentaire dx du point de vue (figure 1.3a).

Cependant, pour accélérer les calculs il est préférable d'adopter une approche continue, c'est-à-dire de calculer les limites dans lesquelles l'ensemble d'objets vus reste le même et de faire, si possible, lors d'un changement de visibilité une mise à jour incrémentale moins chère qu'un recalcul complet (figure 1.3b).

L'utilisation de l'approche continue pour résoudre un problème de visibilité permet en général une résolution plus souple et plus économique. En effet, l'approche continue utilise la cohérence spatiale de la scène – comme dans le cas du calcul d'une vue – ou la cohérence temporelle du problème – comme dans le cas du calcul d'une vue le long d'une trajectoire – pour déterminer rapidement si la visibilité a changé et le cas échéant pour mettre à jour la visibilité sans devoir la recalculer entièrement. Les travaux que nous présentons dans cette thèse sont centrés sur l'utilisation de l'approche continue – utilisation de la cohérence spatiale et temporelle – pour répondre à des requêtes de calculs de visibilité statique et dynamique.

La cohérence spatiale ou temporelle ne peut cependant être pleinement utilisée que si certains calculs de visibilité ont déjà été effectués auparavant, notamment lors d'une phase de prétraitement.

1.1.3 Prétraitement

Dans la plupart des cas, le résultat d'une requête locale – calcul des objets vus depuis un point par exemple – ne fait intervenir qu'une petite partie des objets de la scène. Si ce calcul est fait en considérant tous les objets de la scène, alors le temps de calcul ne dépend que de la taille de la scène, quelle que soit la taille du résultat. Or tous ces calculs sont inutiles si au final un seul objet est vu ! Pour accélérer les calculs et obtenir des algorithmes *sensibles à la sortie*, c'est-à-dire dont le temps de calcul dépend de la taille du résultat calculé, il ne faut tenir compte si possible que des objets vus. Ceci n'est en général possible que si une certaine connaissance de la scène – répartition des objets, relations de visibilité, etc. – a déjà été acquise. Cette connaissance est construite lors d'une étape de prétraitement, où des calculs préliminaires sont effectués une fois pour toute avant toute requête. Ce prétraitement construit en général certaines structures de données qui permettent ensuite d'améliorer le temps de calcul des requêtes.

Le prétraitement s'adresse aux calculs locaux de visibilité, cependant un calcul global de visibilité peut servir d'étape de prétraitement pour ces calculs locaux.

Nous nous intéressons dans cette thèse à la construction et à l'utilisation de structures globales pour accélérer les calculs de visibilité.

1.2 Partitions dans le plan

Une grande catégorie de prétraitements est basée sur des schémas de découpage du plan dont le but est de subdiviser le plan en régions contenant moins d'objets. Lors d'un calcul de visibilité, une première série de tests de visibilité est effectuée sur les régions issues du découpage. Les régions non visibles, et tous les objets qu'elles contiennent, sont éliminés du calcul. Après ce premier calcul, qui ne doit pas être trop coûteux, le nombre d'objets restant à considérer est en général plus petit que le nombre total d'objets et prend moins de temps à traiter.

1.2.1 Boîtes englobantes

Une première méthode pour découper le plan est de regrouper les objets en groupes, et d'utiliser pour la phase de sélection la boîte englobante de ces groupes objets. Une boîte englobante est une région de l'espace, généralement un polygone, qui contient tous les objets du groupe. Ces boîtes englobantes peuvent avoir des côtés parallèles aux axes, être constituées par l'enveloppe convexe des objets, etc. La première étape d'un calcul de lancer de rayons par exemple, consiste à tester l'intersection entre le rayon et les boîtes englobantes. Ensuite un calcul normal de lancer de rayons est effectué, non pas sur tous les objets de la scène, mais seulement sur ceux contenus dans les boîtes englobantes coupées successivement par le rayon (figure 1.4).

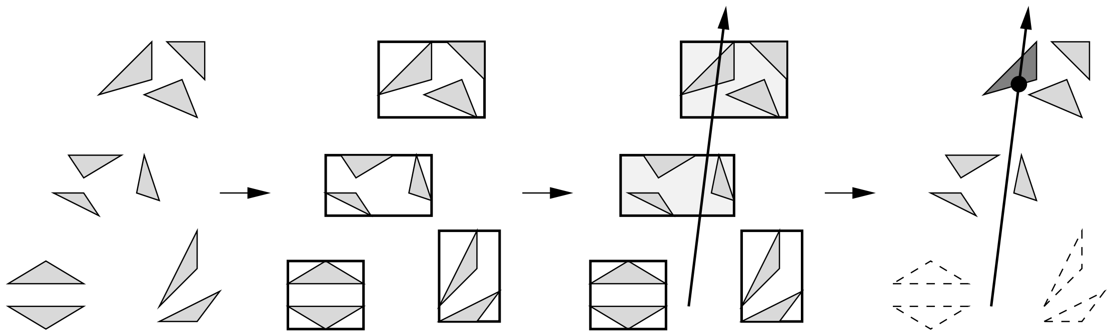

FIG. 1.4 – *Utilisation des boîtes englobantes dans le lancer de rayons.*

Les boîtes englobantes sont bien adaptées aux calculs discrets de visibilité. Cependant leur utilité dans des calculs continus est très limitée. De plus, la construction automatique de ces boîtes, ou plus précisément le regroupement optimal des objets en agrégats, n'est pas une tâche facile et fait toujours l'objet de recherches.

1.2.2 Grilles, grilles récursives

D'autres types de méthodes de découpage sont basées sur la subdivision de l'espace en régions et permettent de se déplacer d'une région à une autre sans considérer toute la scène.

La façon la plus simple de découper automatiquement la scène est d'utiliser une grille régulière. À chaque rectangle de la grille est associée la liste des objets qu'il contient. Lors d'un calcul de lancer de rayons, ne sont considérés que les objets contenus dans les rectangles de la grille coupés successivement par le rayon (figure 1.5). La structure de grille permet d'avoir directement les rectangles coupés par le rayon sans avoir à considérer les autres rectangles : il est en effet possible de passer d'un rectangle à un autre rectangle incident en temps constant.

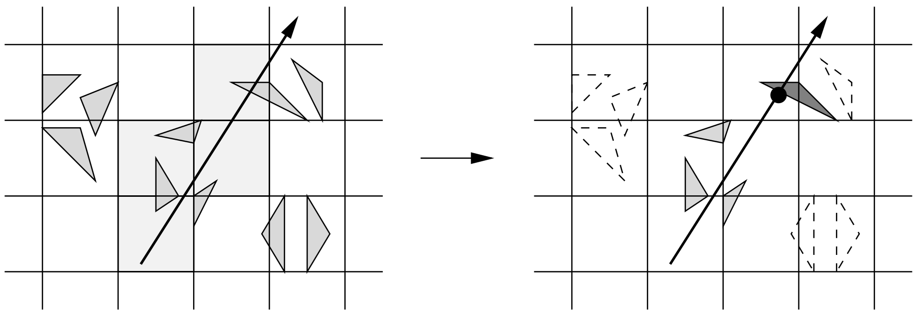

FIG. 1.5 – *Utilisation d'une grille dans le lancer de rayons.*

Dans le cas où les objets de la scène ne sont pas uniformément répartis, il est plus utile d'utiliser des grilles adaptatives qui tiennent compte de la répartition des objets et qui évitent le découpage de régions vides. Ces grilles sont souvent définies de façon récursive.

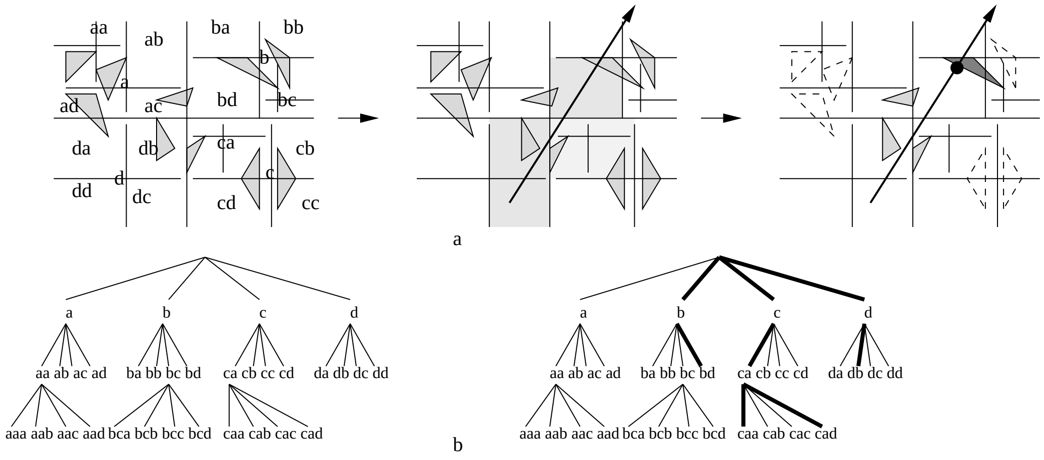

FIG. 1.6 – *Utilisation d'arbre quadtree dans le lancer de rayons.* **a.** *Découpage de la scène.* **b.** *Arbre correspondant.*

Dans les arbres *quadtree* par exemple, une région est subdivisée en quatre, et chacune des sous-régions est elle même subdivisée à son tour en quatre... La subdivision s'arrête quand une région devient vide (figure 1.6a). Ces subdivisions sont représentées sous la forme d'un arbre : chaque région est un nœud de l'arbre et a pour fils ses quatre sous-régions (figure 1.6b). De plus, à chaque région est associée la liste des objets qu'elle contient. Plusieurs méthodes peuvent être employées pour effectuer la subdivision : subdivision régulière, subdivision cherchant à répartir uniformément les objets dans les sous-régions... La structure d'arbre peut être utilisée pour rechercher en temps logarithmique les régions traversées par un rayon dans une requête de lancer de rayons.

Il existe d'autres types de grilles récursives, comme les *kd*-arbres qui subdivisent les régions en deux sous-régions selon une direction alternativement parallèle à l'axe des abscisses et à l'axe des ordonnées ; cependant le principe général d'utilisation reste le même (pour plus de détails, voir le survol de Samet [Sam84]).

Ces grilles ont l'avantage par rapport aux boîtes englobantes d'introduire une utilisation de la cohérence spatiale grâce à la possibilité de considérer pour une région donnée ses régions incidentes. Les grilles sont utiles dans le cadre de calculs discrets locaux ou dans le cadre de calculs plus globaux (comme déterminer l'ordre d'affichage des objets dans l'algorithme du peintre). En revanche, elles restent peu adaptées aux calculs continus de visibilité car elles permettent difficilement de calculer des limites de visibilité.

1.2.3 Triangulations

Découper la scène en régions ne signifie pas uniquement découper la scène en régions contenant des groupes d'objets. Le découpage de l'espace libre, le plan auquel on a ôté les objets de la scène, permet aussi d'accélérer les calculs. Un des découpages les plus connus est la triangulation, qui consiste à découper l'espace libre d'une scène polygonale en triangles dont les sommets sont les sommets des polygones de la scène (figure 1.7).

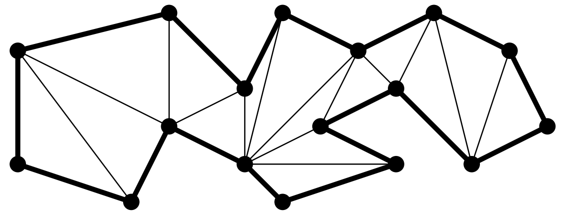

FIG. 1.7 – *Triangulation d'un polygone.*

Les triangulations servent notamment à résoudre des problèmes de visibilité quand l'espace libre est l'intérieur d'un polygone simple, comme par exemple des problèmes de lancer de rayons, de plus courts chemins ou d'illumination par un néon (voir par exemple [GHL+87]).

Bien qu'il existe un algorithme pour trianguler un polygone simple en temps linéaire, celui-ci est assez compliqué. D'autres découpages utilisant des formes polygonales plus grandes et générant moins de régions ont été étudiés pour résoudre ces problèmes, comme par exemple les profils de visibilité structurés d'Heffernan et Mitchel [HM90] ou les décompositions en trapèzes de Chiang, Preparata et Tamassia [CPT96].

Ces découpages introduisent une plus grande utilisation de la cohérence spatiale : les frontières des régions découpées s'appuient sur les côtés des objets. Cependant cela ne suffit pas toujours à réduire la complexité des calculs : un rayon peut traverser un nombre important de régions avant de heurter un des côtés du polygone. Pour éviter ces sauts de complexité, les auteurs greffent sur les structures de découpage des structures de recherche purement combinatoire (fractional cascading, $BB(\alpha)$-trees, etc.) et relativement compliquées afin d'obtenir des complexités intéressantes même dans les cas les pires.

1.3 Travaux présentés dans ce mémoire

Le schéma heuristique [1] de la figure 1.8 fait le point sur les problèmes de visibilité dans le plan. Il indique les relations entre ces problèmes, les structures de données, les algorithmes, etc., et souligne (en gras) les problèmes abordés dans cette thèse.

Plus précisément, nous cherchons dans ce travail de thèse à utiliser la cohérence spatiale et temporelle pour résoudre des problèmes de visibilité. Pour cela nous voulons utiliser des structures de données purement géométriques qui ne requièrent pas l'utilisation supplémentaire de structures combinatoires de recherche compliquées (pas plus compliquées qu'un arbre de recherche équilibré par exemple) pour obtenir de bonnes performances.

[1] Tony Buzan, *Une tête bien faite*, Éditions d'Organisation.

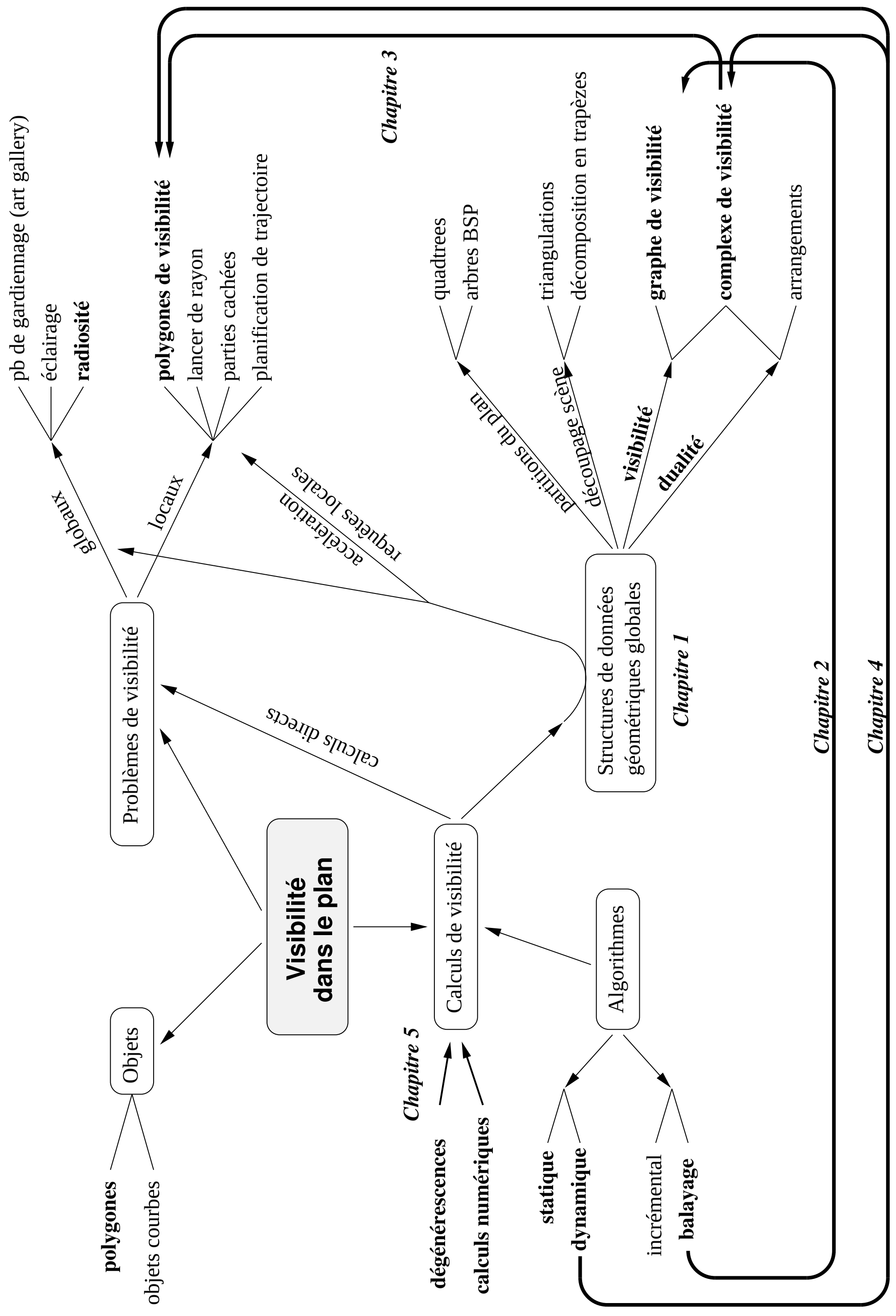

FIG. 1.8 – *Schéma heuristique décrivant les problèmes de visibilité dans le plan et indiquant les sujets traités dans ce mémoire.*

Le reste de ce chapitre 1 est consacré à l'étude des structures de données que nous allons utiliser : le graphe et le complexe de visibilité. Cette étude nous permettra aussi d'introduire deux outils de géométrie algorithme que nous utilisons presque constamment dans les algorithmes que nous étudions : le calcul par balayage et la dualité.

Dans le chapitre 2 nous décrivons un algorithme optimal pour construire ces structures de données.

Dans le chapitre 3 nous montrons comment utiliser le complexe de visibilité pour calculer des polygones de visibilité :

- le polygone de visibilité d'un point, qui représente la vue autour de ce point ;
- le polygone de visibilité d'un segment, qui représente la zone de la scène éclairée par le segment considéré comme un néon.

Dans le chapitre 4 nous abordons des problèmes de visibilité dynamique : nous étudions comment maintenir la vue d'un point qui se déplace et comment maintenir le complexe de visibilité quand les objets de la scène sont en mouvement. Les algorithmes de maintien de vue que nous donnons améliorent les temps de calculs des algorithmes déjà existants.

Tous les algorithmes que nous présentons sont développés dans le cadre de scènes d'objets en *position générale*, c'est-à-dire où les sommets des polygones ne sont pas alignés. Nous supposons donc que les scènes ne contiennent pas de configurations dégénérées telles que celles données en exemple dans la figure 1.9

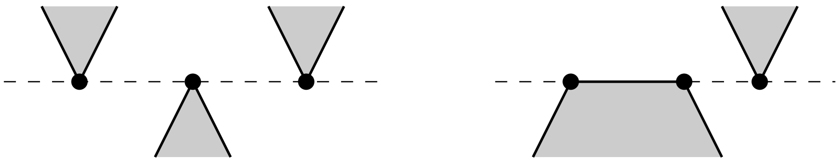

FIG. 1.9 – *Configurations dégénérées où des sommets de polygones sont alignés.*

Nous supposons aussi que tous les calculs effectués dans ces algorithmes sont faits de façon exacte. Les problèmes de dégénérescence et d'imprécision des calculs numériques ne sont pas oubliés pour autant : ces problèmes sont importants en pratique et font l'objet du chapitre 5 de ce mémoire.

Enfin, nous faisons dans le dernier chapitre un bilan des travaux présentés et indiquons des directions de recherches permettant de poursuivre et compléter ces travaux.

1.4 Structures de visibilité

Nous étudions ici des structures de données directement liées à la visibilité – graphe et complexe de visibilité – qui d'une certaine façon codent les relations de visibilité entre objets. Pour mener cette étude, nous utilisons notamment des relations de dualité entre droites et points qui permettent de mieux appréhender ces structures.

1.4.1 Graphe de visibilité

Une des structures globales de visibilité les plus étudiées est le *graphe de visibilité*.

Étant donné une scène composée de polygones simples disjoints, le graphe de visibilité de cette scène est le graphe $(\mathcal{V}, \mathcal{E})$ tel que

- $\mathcal{V}$, ensemble des sommets du graphe, est l'ensemble des sommets des polygones ;
- $\mathcal{E}$, ensemble des arêtes du graphe, est l'ensemble des paires $e = (p, q)$ de sommets de $\mathcal{V}$ tels que le segment $[p, q]$ a une intersection nulle avec l'intérieur des polygones.

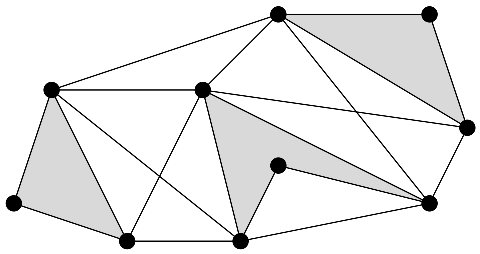

FIG. 1.10 – *Graphe de visibilité d'une scène.*

Les arêtes du graphe de visibilité relient donc les sommets de la scène mutuellement visibles. La figure 1.10 montre la représentation géométrique du graphe de visibilité, où les segments correspondant aux arêtes du graphe ont été tracés dans la scène. Le graphe de visibilité contient au moins les arêtes correspondant aux côtés des polygones et aux segments de l'enveloppe convexe de la scène. Ce graphe n'est pas orienté, mais les arêtes incidentes à un sommet sont classées dans l'ordre dans lequel sont répartis les segments correspondants dans la scène.

Le nombre de sommets de ce graphe est le nombre total n de sommets des polygones. Son nombre d'arêtes, noté m, varie en revanche d'une taille linéaire ($m = O(n)$) à une taille quadratique ($m = O(n^2)$). La taille minimale est obtenue quand chaque polygone ne voit qu'un seul polygone et qu'il n'y a que quatre arêtes de visibilité entre ces paires de polygones (figure 1.11a) ; la taille est alors $m = n + 4(n_p - 1)$, où n_p est le nombre de polygones. La taille maximale est obtenue quand tous les sommets sont mutuellement visibles (figure 1.11b) ; le graphe de visibilité est alors le graphe complet K_n de taille maximale $m = n(n-1)/2$.

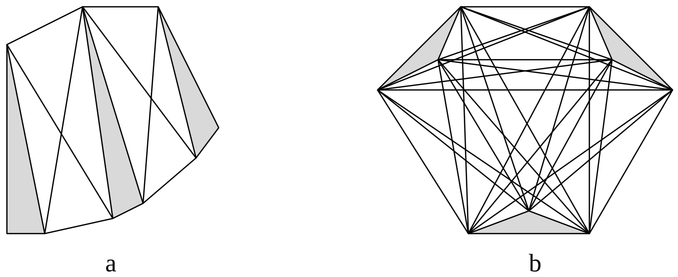

FIG. 1.11 – *Graphes de visibilité.* **a.** *De taille minimale.* **b.** *De taille maximale.*

Les graphes de visibilité servent par exemple en planification de trajectoire : dans une scène de polygones, le plus court chemin évitant les obstacles entre deux points p

et q emprunte les arêtes du graphe de visibilité (figure 1.12). Le calcul de ce chemin se ramène alors à un calcul de plus court chemin dans le graphe de visibilité, où le poids des arêtes du graphe est la longueur de l'arête géométrique (voir [LPW79]).

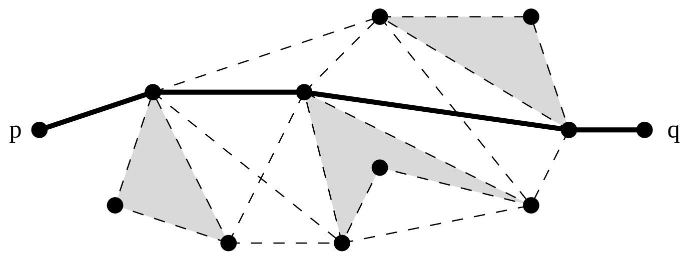

FIG. 1.12 – *Plus court chemin entre deux points en présence d'obstacles.*

L'algorithme naïf pour calculer le graphe de visibilité consiste à examiner toutes les paires (p, q) de sommets et à vérifier si les segments $[p, q]$ correspondants ne sont pas coupés par un côté de polygone. Pour chaque paire, la vérification prend un temps $O(n)$, soit un temps total de calcul en $O(n^3)$. Même si par chance il suffit de tester $O(1)$ côtés pour les segments ne correspondant pas à une arête du graphe de visibilité, il faut tester les n côtés avant de s'apercevoir qu'un segment correspond à une arête de visibilité, soit un coût total de $O(n^2 + (n-1)m)$. Que m soit linéaire ou quadratique, le temps de calcul est toujours n fois trop long par rapport au temps optimal $O(n \log n + m)$.

Grâce au balayage d'un espace de configuration annexe (un arrangement de droites), Edelsbrunner et Guibas [EG86] ont réduit le coût du calcul de graphe de visibilité d'une scène de n segments à un coût en $O(n^2)$, qui n'est optimal que pour le calcul de graphes denses. Overmars et Welzl [OW88] ont transposé ces techniques dans la scène de segments, ce qui leur a permis d'obtenir ensuite un algorithme sensible à la sortie, de complexité $O(m \log n)$. Ces algorithmes utilisent des notions et des techniques (dualité, balayage) au centre de nos travaux ; nous les détaillons dans les parties qui suivent, en les adaptant en général au cadre de scènes polygonales.

Entre-temps, Gosh et Mount [GM91] ont élaboré le premier algorithme de calcul de graphe de visibilité de polygones en temps optimal $O(n \log n + m)$. Ils utilisent un calcul incrémental (qui utilise donc un espace mémoire en $O(m)$) assez compliqué et reconnaissent eux-mêmes que leur algorithme est inutilisable en pratique, car les constantes impliquées dans le temps de calcul sont assez importantes. Nous ne détaillons pas cet algorithme dans ce mémoire.

Dans le cadre de scènes composées d'objets courbes convexes, Pocchiola et Vegter [PV96b] ont élaboré une nouvelle structure de données qui étend la notion de graphe de visibilité : le *complexe de visibilité*. Cette structure leur permet de calculer le graphe de tangence de la scène en temps optimal $O(n \log n + m)$, d'abord par un algorithme incrémental [PV96b], puis par un algorithme de balayage [PV96a].

En reprenant les concepts de complexe de visibilité, de dualité et de balayage dans le cadre de scènes polygonales, nous avons élaboré un nouvel algorithme de calcul de graphe de visibilité, le premier qui soit optimal en temps ($O(n \log n + m)$) et en espace de travail ($O(n)$). Nous l'exposons en détail dans le chapitre suivant.

Nous exposons jusqu'à la fin de ce chapitre les techniques et les structures de données que nous utilisons ensuite dans nos travaux.

1.4.2 Dualité et problèmes de visibilité

L'idée derrière l'utilisation de la dualité en géométrie est celle de l'adage « si quelque chose ne marche pas, essayez quelque chose de différent ». Il arrive que certains (ensembles d') objets et certaines propriétés géométriques soient difficiles à manipuler directement. L'utilisation d'une transformation géométrique peut permettre de transformer l'espace de départ en un autre espace, où les objets étudiés sont transposés en d'autres objets, où les propriétés de départ sont transposées en d'autres propriétés, le tout étant plus facile à manipuler. Cela permet de résoudre le problème dans ce nouvel espace – l'espace dual – puis de retransposer si nécessaire la solution dans l'espace de départ.

Il y a par exemple une correspondance naturelle entre les hyperplans et les points. La transformation la plus connue entre hyperplans et points est l'inversion de centre O, notée *, qui dans $\mathbb{R}^n$ associe à l'hyperplan H d'équation $\sum_{i=1}^{n} a_i x_i + 1 = 0$ le point de coordonnées $H^*(a_1, \ldots, a_n)$. Cette bijection a la propriété d'inverser la relation d'inclusion : en notant H^+ et H^- les deux demi-espaces délimités par l'hyperplan H (H^+ contenant O), alors pour tout point p

$$p \in H \text{ (resp. } H^+\text{, resp. } H^-\text{)} \iff H^* \in p^* \text{ (resp. } p^{*+}\text{, resp. } p^{*-}\text{).}$$

Cette transformation permet alors de manipuler un ensemble de points, plus facile à manier que l'ensemble d'hyperplans de départ.

D'une manière générale, on appelle *dualité* toute bijection entre points et hyperplans qui inverse la relation d'inclusion. Nous détaillons maintenant l'utilité et l'utilisation de relation de dualité dans le cadre de problèmes de visibilité dans le plan.

Quand nous travaillons sur des problèmes de visibilité dans le plan, nous manipulons en fait des droites orientées qui représentent les directions de vision, et les objets vus sont ceux coupés par ces droites. Pour accélérer les calculs de visibilité, nous voulons classer les droites en différents ensembles qui les regroupent en fonction des objets qu'elles coupent. Ainsi, pour savoir par exemple quel objet est vu dans telle direction, il suffit d'identifier à quel ensemble appartient la droite correspondante et de consulter les propriétés de visibilité de cet ensemble, propriétés communes à toutes les droites de l'ensemble.

Il est difficile de manipuler directement des ensembles de droites : comment peut-on les représenter, tester l'appartenance d'une droite à un ensemble, ou définir les frontières communes entre ces ensembles ? L'utilisation de relations de dualité nous permet de transformer ces ensembles de droites en ensembles de points, plus faciles à appréhender.

Nous considérons l'espace des droites orientées. Ainsi, les deux droites passant par un point p et de vecteur directeur respectif $\vec{v}$ et $-\vec{v}$, bien que géométriquement confondues, sont considérées comme étant deux droites distinctes.

Il y a plusieurs façons d'associer un point à une droite. Deux transformations souvent utilisées sont (figure 1.13a) :

- $d : y = ax + b \rightarrow d^* \ (a, b)$ (dualité (a, b))
- $d : y \cos\theta - x \sin\theta - u = 0 \rightarrow d^* \ (\theta, u)$ (dualité (θ, u))

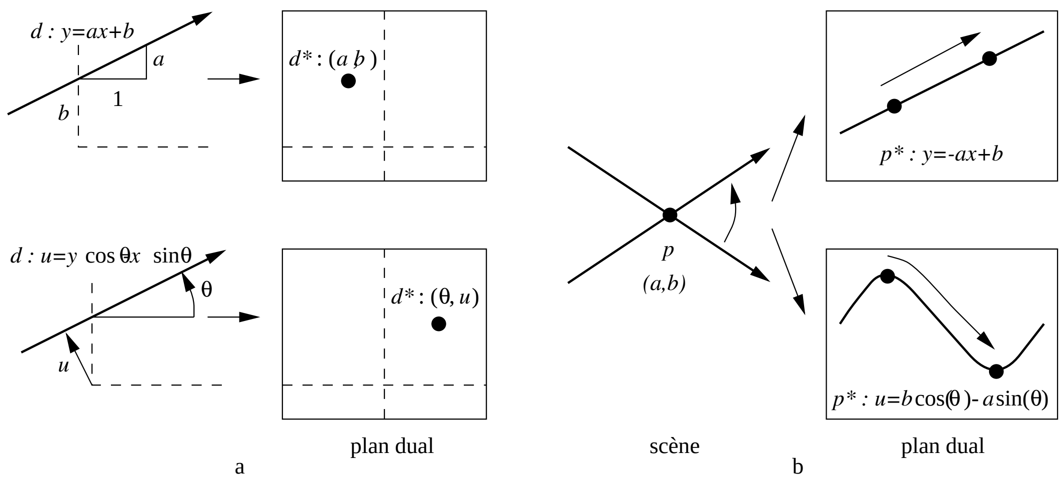

FIG. 1.13 – *Relations de dualité.* **a.** *Point dual d'une droite.* **b.** *Courbe duale d'un point.*

Ces deux relations ont beaucoup de propriétés communes, et leurs différences ne sont pas fondamentales (ce point est précisé plus loin). Dans la suite, la notation * désigne indifféremment l'une ou l'autre des deux relations quand les propriétés évoquées sont les mêmes ; s'il y a des différences, nous préciserons alors la relation concernée.

L'axe des abscisses du plan dual utilisé représente la pente (ou l'angle) des droites. Si d_1 et d_2 sont deux droites parallèles telles que d_1 soit au-dessus de d_2, alors les points duaux d_1^* et d_2^* ont même abscisse et d_1^* est au-dessus de d_2^*.

Pour résoudre les problèmes de visibilité que nous étudions, nous voulons, étant donné un côté de polygone (p, q), pouvoir considérer les droites qui coupent ce segment. Or une droite de direction θ coupe le segment (p, q) si et seulement si elle est située entre les droites de même direction θ et passant respectivement par p et q. Les droites passant par un sommet représentent des limites de visibilité, il est donc intéressant de considérer l'ensemble des droites passant par un sommet donné $p = (a, b)$. À chaque droite passant par p est associé un point dans l'espace dual. Quand la droite tourne autour de p, le point dual décrit une courbe duale p^* dans l'espace dual : la droite d'équation $y = -ax + b$ (dualité (a, b)) ou la sinusoïde d'équation $u(\theta) = b\cos\theta - a\sin\theta$ (dualité (θ, u)) (figure 1.13b).

Nous retrouvons avec ces courbes duales des propriétés d'inversion des relations d'inclusion. En effet, la courbe duale p^* d'un point p représente dans le plan dual le graphe d'une fonction qui sépare le plan en deux demi-espaces : p^{*+} (resp. p^{*-}) est le demi-espace ouvert situé au-dessus (resp. en dessous) de la courbe p^*. Nous avons alors les relations (figure 1.14a)

$$p \in d \text{ (resp. } d^+\text{, resp. } d^-\text{)} \iff d^* \in p^* \text{ (resp. } p^{*-}\text{, resp. } p^{*+}\text{).}$$

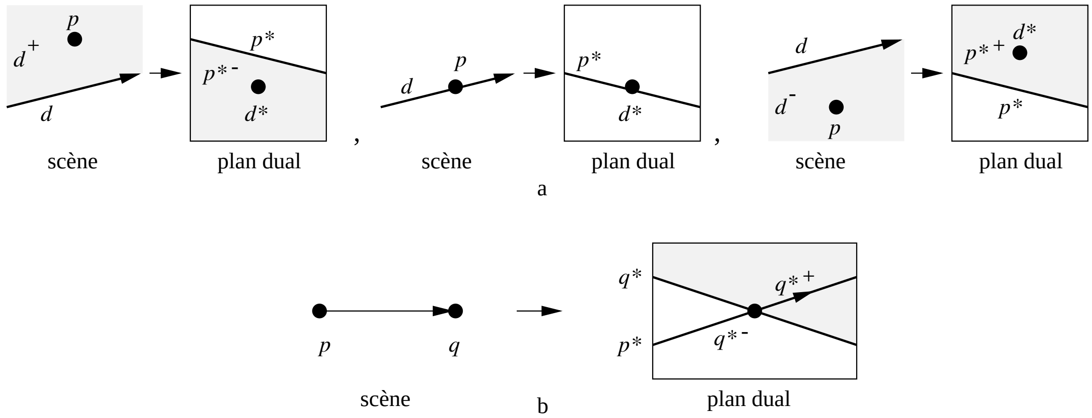

FIG. 1.14 – *Propriétés de la dualité.* **a.** *Inversion de la relation d'inclusion.* **b.** *Position relative et intersection dans le plan dual.*

Ces courbes duales ont d'autres propriétés. Si nous considérons la droite orientée (p, q), alors dans l'espace dual les courbes duales p^* et q^* se coupent en le point dual $(p, q)^*$, de telle façon que p^* passe de q^{*-} à q^{*+} (p^* traverse q^* de bas en haut) (figure 1.14b). Cette propriété permet d'obtenir directement dans l'espace dual des informations sur la position relative des objets par rapport à une direction de vision.

Reprenons notre côté de polygone (pq). Les deux courbes duales découpent l'espace dual en trois parties : au-dessus, entre et en dessous des deux courbes. Alors une droite d coupe le segment (pq) si et seulement si son point dual d^* est dans l'espace situé entre les deux courbes p^* et q^* (figure 1.15a). Si le point est au-dessus (resp. en dessous) de cette région, alors le segment est en dessous (resp. au-dessus) de la droite. Cette partition de l'espace dual correspond à une partition de l'ensemble des droites en fonction de leur position relative par rapport au segment.

Considérons maintenant une scène composée de polygones. Nous appliquons le découpage de l'espace dual précédent pour chaque côté de polygone. Nous obtenons alors un découpage de l'espace dual par les courbes duales des sommets des polygones : l'*arrangement dual* de la scène (figure 1.15b).

Cet arrangement dual est composé de faces, d'arêtes (portions de courbes comprises entre deux sommets) et de sommets (intersection de deux courbes). Les faces représentent les ensembles de droites intersectant une même liste d'objets (dans l'ordre de la liste).

Plus précisément, à chaque droite d nous associons une *étiquette*, constituée des objets intersectés dans l'ordre par la droite. Si la droite coupe le segment ouvert O_i, alors nous mettons O_i dans l'étiquette. Si la droite passe par un sommet p, alors nous mettons $p^{\pm\pm}$ dans l'étiquette : le premier signe de $p^{\pm\pm}$ est $+$ (resp. $-$) si le côté (p^-, p) du polygone est au-dessus (resp. en dessous) de la droite ; le deuxième signe de $p^{\pm\pm}$ est $+$ (resp. $-$) si le côté (p, p^+) du polygone est au-dessus (resp. en dessous) de la droite. La relation

$d \sim d'$ si et seulement si d et d' ont la même étiquette et on peut déplacer continûment d vers d' en gardant l'étiquette constante,

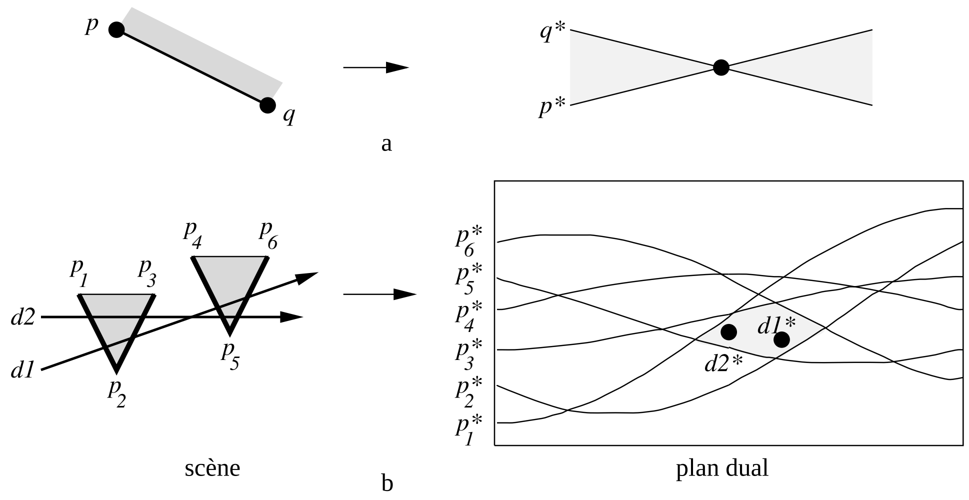

FIG. 1.15 – *Classification des droites grâce à la dualité.* **a.** *Zone d'un côté de polygone.* **b.** *Arrangement dual de la scène : droites correspondant à une face.*

est une relation d'équivalence. Les éléments de l'arrangement dual représentent les classes d'équivalence de cette relation.

Les deux relations de dualité évoquées jusqu'à maintenant sont très liées. Nous avons vu un certain nombre de propriétés communes, et elles sont liées de plus par la correspondance $a = \tan\theta$ et $b = u/\cos\theta$. Cependant la dualité (a, b) paraît plus limitée : elle ne permet de ne considérer que les droites orientées de direction comprise dans l'intervalle $]-\pi/2, \pi/2[$. Mais ces limitations ne sont pas importantes ici, car nous ne nous servons que des informations topologiques de l'arrangement dual, et les tests numériques sont tous retranposés sur les points et les droites de la scène.

De plus, la partie de l'arrangement dual de courbes (dualité (θ, u)) compris dans l'intervalle $]\theta_0 - \pi/2, \theta_0 + \pi/2[$ d'une scène et l'arrangement dual de droites (dualité (a, b)) de la scène à laquelle une rotation d'angle θ_0 a été appliquée sont topologiquement identiques (figure 1.16, où nous avons colorié des faces pour mieux faire apparaître les correspondances). Enfin, pour avoir toutes les informations de visibilité, il suffit de se restreindre à un intervalle de directions de longueur π, car les droites de direction θ et $\theta + \pi$ sont géométriquement identiques. Cela se traduit par des relations de symétrie dans l'arrangement dual de courbes (dualité (θ, u)). Pour visualiser un intervalle de directions de longueur 2π avec les arrangements de droites, il suffit de « coller » à l'arrangement son symétrique par rapport à l'axe des abscisses.

Nous utilisons ici dans nos représentations de l'espace dual la formulation (a, b), car elle est plus pratique pour nos illustrations. Cependant nous ne nous servons que des informations topologiques, et non pas géométriques, des droites duales. Nous n'excluons donc aucune droite (verticale) dans l'espace dual.

La formulation (θ, u) est plus pratique à utiliser dans le cadre de scènes composées d'objets courbes, où les courbes duales correspondent aux droites tangentes aux objets. Pocchiola [Poc90] utilise cette dualité pour construire l'arrangement dual d'un scène d'objets convexes courbes.

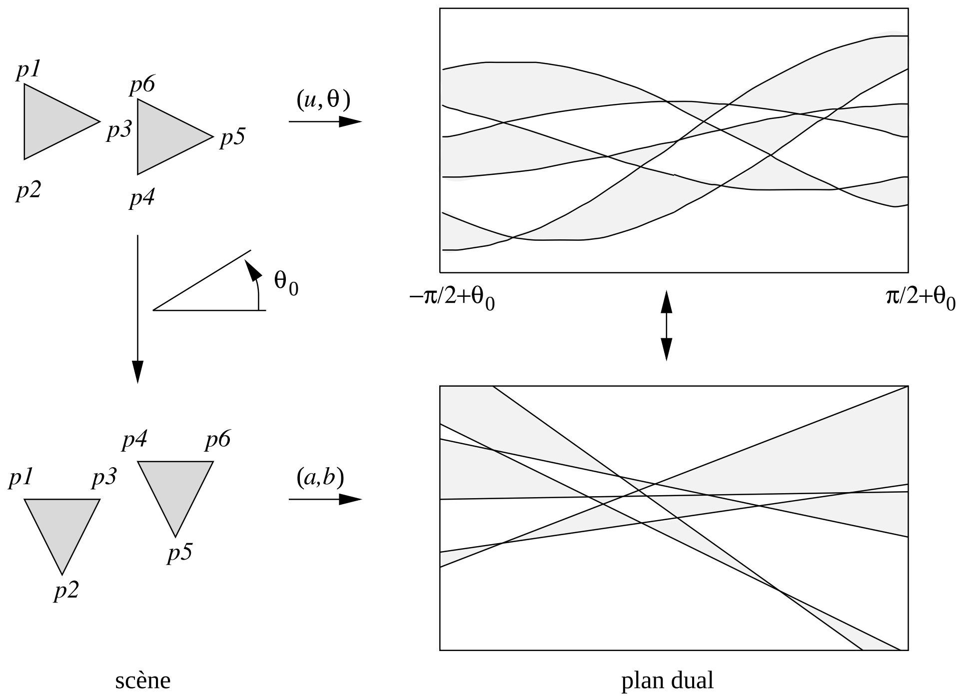

FIG. 1.16 – *Correspondance entre les deux relations de dualité.*

Pour savoir quels sont les objets traversés par une droite, il suffit de localiser l'élément de l'arrangement qui contient le point dual. Il existe deux tactiques pour construire cet arrangement de droites : construction incrémentale et balayage.

La construction incrémentale consiste à introduire les droites une par une. Supposons que nous avons déjà construit l'arrangement de $n-1$ droites. Nous introduisons la n^e droite, en cherchant les faces de l'arrangement traversées par cette droite, faces qui seront coupées en deux. Pour cela, il faut d'abord chercher la face semi-infinie gauche où commence cette droite. Ensuite, pour trouver le point de sortie de la face courante traversée, il suffit de parcourir les arêtes bordant la face – dans le sens trigonométrique direct par exemple – depuis l'arête d'entrée (figure 1.17).

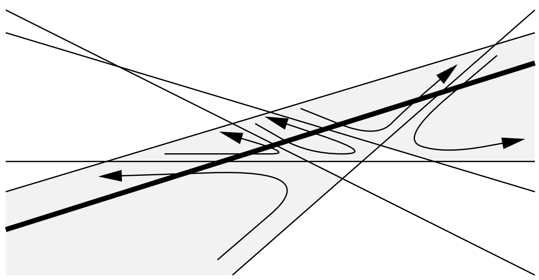

FIG. 1.17 – *Construction incrémentale d'un arrangement : faces traversées par la nouvelle droite insérée.*

Le théorème de la zone ([EOS86],[CGL85]) indique que le nombre total d'arêtes bordant les faces traversées par la droite est linéaire en fonction du nombre de droites de l'arrangement. La construction incrémentale d'un arrangement de n droites s'effectue donc en temps optimal $O(n^2)$.

Le balayage de l'arrangement consiste à parcourir les éléments de l'arrangement comme s'ils étaient balayés de la gauche vers la droite par une ligne. Le balayage est modélisé par la liste $(e_{i_k})_{1\leq k\leq n}$ des arêtes « coupées » par la ligne de balayage, où il y a pour chaque sommet de la scène exactement une arête portée par la courbe duale de ce point (figure 1.18a).

Le processus de balayage est en fait discret : le balayage progresse en passant à chaque fois un sommet incident à deux arêtes consécutives e_{i_k} et $e_{i_{k+1}}$ de la coupe. La coupe est alors mise à jour en remplaçant ces deux arêtes par les arêtes suivantes e'_{i_k} et $e'_{i_{k+1}}$ respectivement sur les droites support de i_{k+1} et i_k (figure1.18b).

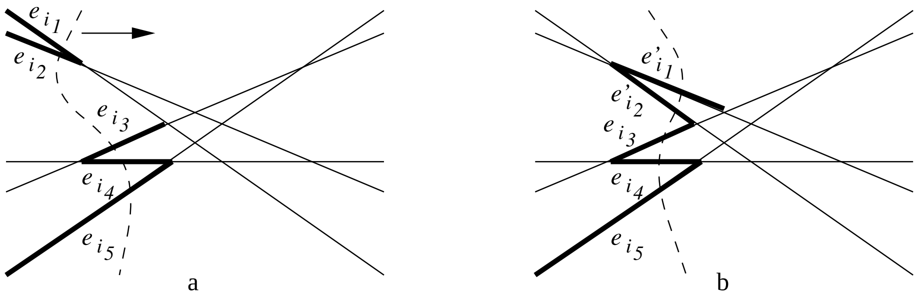

FIG. 1.18 – *Balayage d'un arrangement de droites.* **a.** *Coupe de balayage.* **b.** *Progression du balayage.*

La façon la plus directe d'effectuer ce balayage est de parcourir les sommets de l'arrangement dans l'ordre selon les abscisses croissantes. Cela revient à prendre une ligne de balayage verticale « droite ». Le balayage nécessite alors l'utilisation d'une queue de priorité et s'effectue en temps $O(n^2 \log n)$. Cependant ce type de balayage est trop strict et un balayage *topologique* plus souple suffit. Le balayage topologique parcourt les sommets de façon cohérente : il ne passe un sommet $v = d_i \cap d_j$ qu'après avoir déjà visité les sommets précédents sur d_i et d_j.

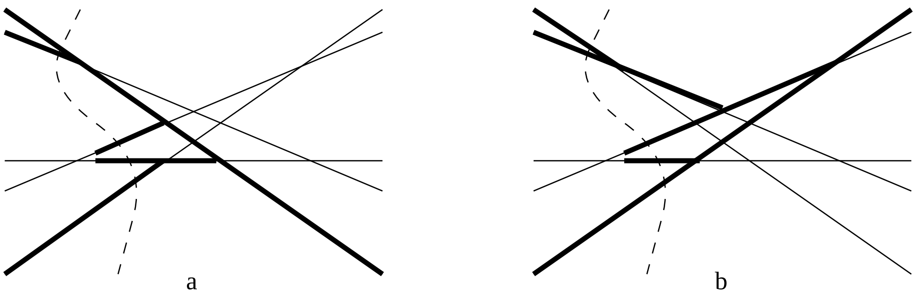

FIG. 1.19 – *Arbres d'horizon.* **a.** *Arbre supérieur.* **b.** *Arbre inférieur.*

Pour effectuer ce balayage, Edelsbrunner et Guibas [EG86] ont créé une structure de données auxiliaire : les *arbres d'horizon*. Étant donné une coupe de balayage, l'arbre d'horizon supérieur (resp. inférieur) est construit en étendant chaque arête de la coupe vers la droite jusqu'à ce qu'elle rencontre une droite de l'arrangement de pente plus petite (figure 1.19) (remarque : Edelsbrunner et Guibas [EG86] utilisent la dualité $d : y = ax + b \rightarrow d^*(a, b)$, qui inverse certaines relations par rapport à la dualité que nous utilisons. Nous avons retransposé ici la description de l'algorithme avec la dualité que nous utilisons). Alors le balayage peut progresser en passant un sommet v si et seulement si v est présent dans les deux arbres d'horizon : v est alors intersection de deux arêtes consécutives de la coupe.

Pour mettre à jour l'arbre d'horizon supérieur après le passage d'un sommet $v = e_{i_k} \cap e_{i_{k+1}}$, il faut trouver la nouvelle extrémité droite du segment porté par la droite i_{k+1}. Pour cela, il suffit de parcourir les arêtes de la baie formée par les arêtes successives de l'arbre d'horizon et commençant par l'arête $e_{i_{k-1}}$ de la coupe située juste au-dessus (figure 1.20).

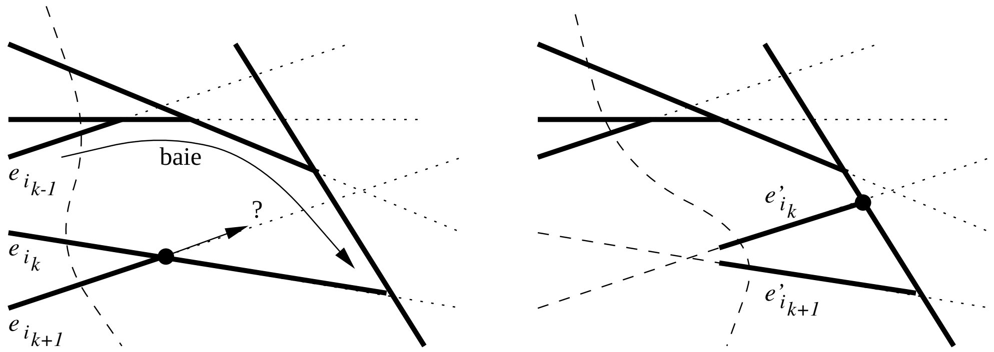

FIG. 1.20 – *Mise à jour de l'arbre d'horizon lors du passage d'un sommet.*

La mise à jour à chaque passage coûte *a priori* un temps $O(n)$, car une baie peut comporter $O(n)$ arêtes, soit pour les $O(n^2)$ passages un temps total $O(n^3)$. Cependant, grâce à un schéma de comptage global, les auteurs montrent dans [EG89] qu'au total seulement $O(n^2)$ arêtes sont balayées dans les baies. Le balayage topologique s'effectue alors en temps optimal $O(n^2)$.

Nous avons présenté cet arrangement dual comme permettant de résoudre des problèmes liés à la visibilité. Il permet en effet de calculer le graphe de visibilité de la scène, comme nous allons le voir ci-après, et de faire des calculs de vues, comme nous le verrons dans le chapitre 3.

Cependant cet arrangement de droites permet de résoudre d'autres problèmes géométriques, comme calculer le triangle d'aire minimale formé par 3 points parmi n points donnés, calculer une droite coupant le plus possible de segments parmi n segments donnés, tester la dégénérescence de $d+1$ points dans $\mathbb{R}^d$, etc. (voir par exemple [EG89] et [CGL85] pour d'autres applications).

1.4.3 Balayage de graphe de visibilité : arbres d'horizon et arbres de rotation

Les sommets de l'arrangement dual correspondent aux segments reliant deux sommets de la scène. En particulier, parmi ces sommets il y a ceux qui correspondent aux arêtes du graphe de visibilité. Puisque le balayage parcourt les sommets de l'arrangement un par un, il faut en profiter pour déterminer si le sommet visité correspond à une arête de visibilité ou non. Grâce à la cohérence du balayage, cette vérification peut se faire en temps constant.

En effet, la cohérence du balayage implique que lorsque le sommet v de l'arrangement correspondant à la droite (p, q) est visité, parmi les droites issues de q et passant par un autre sommet de la scène, toutes celles de pente inférieure à (p, q) ont déjà été parcourues et toutes celles de pente supérieure n'ont pas encore été parcourues (figure 1.21a).

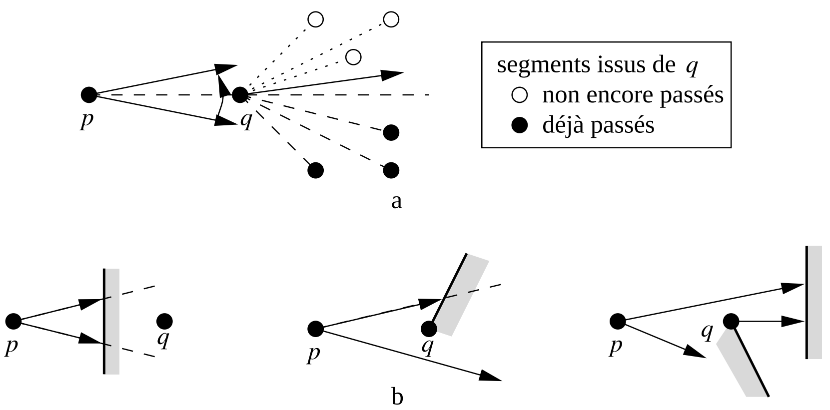

FIG. 1.21 – *Balayage de l'arrangement.* **a.** *Cohérence du balayage.* **b.** *Mise à jour de l'objet vu lors du passage d'un segment* (p, q).

À chaque sommet p de la scène est alors associé l'objet vu depuis p par tout rayon appartenant dans l'espace dual à l'arête e de la coupe associée à p. Lors du passage du sommet v correspondant au segment (p, q), la mise à jour de l'objet vu depuis p connaissant celui vu depuis q se fait alors en temps constant (la figure 1.21(b) montre les situations à considérer) et vérifier si (p, q) est une arête du graphe de visibilité est immédiat.

Overmars et Welzl [OW88] ont remarqué qu'un seul arbre d'horizon pouvait suffire pour effectuer le balayage : il suffit alors de toujours passer le sommet en bas à gauche de l'arbre d'horizon supérieur, sommet qui peut être mis à jour en temps constant à chaque passage. Ils ont alors transposé les arbres d'horizon de l'espace dual dans la scène, où ils deviennent des *arbres de rotation* : comme déplacer un point sur une droite d dans le plan dual revient à faire tourner dans la scène une droite passant par le point d^*, balayer l'arrangement de droites dual revient à faire un balayage *rotationnel* dans la scène, c'est-à-dire à faire pivoter des droites autour des sommets de la scène.

Overmars et Welzl innovent ensuite pour tenir compte directement des relations de visibilité : au lieu d'étendre une demi-droite, ils étendent un demi-rayon libre maximal, qui s'arrête donc sur le premier objet rencontré.

L'arbre de rotation est alors défini pour une direction θ : à chaque sommet p de la scène, est associé le premier sommet vu depuis p en tournant dans le sens trigonométrique direct depuis la direction θ (figure 1.22). Un point fictif d'abscisse plus grande que celles des points de la scène et d'ordonnée infinie est ajouté à la scène en tant que sommet susceptible d'être vu.

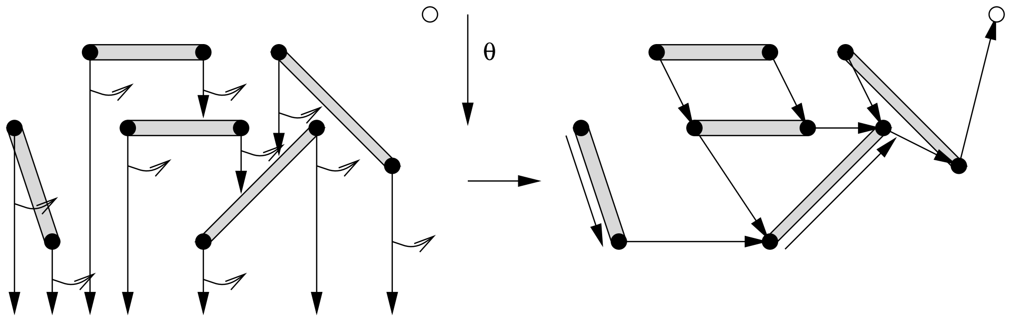

FIG. 1.22 – *Arbre de rotation d'une scène de segments.*

Ce graphe est un sous-graphe de taille n du graphe de visibilité. Le maintien de cet arbre quand θ varie de $-\pi/2$ à $\pi/2$ permet de trouver toutes les arêtes du graphe de visibilité. Le maintien se fait, comme pour les arbres d'horizon, de façon discrète : l'arête de plus petite pente est passée à chaque fois, ce qui assure en plus la cohérence du balayage.

L'étude des propriétés géométriques du graphe induites par le balayage droit conduit les auteurs à considérer des chaînes d'arêtes, suites maximales d'arêtes consécutives telles que les segments de la scène incidents aux sommets d'une chaînes soient au-dessus de la chaîne.

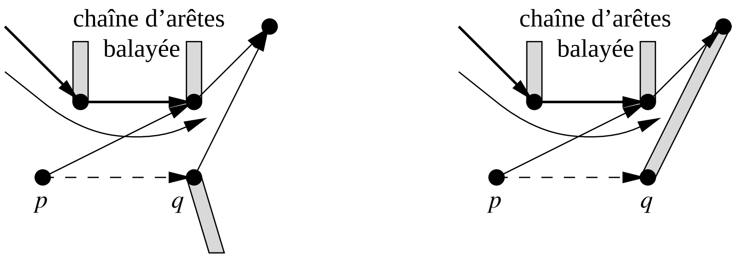

FIG. 1.23 – *Mise à jour de l'arbre de rotation lors du passage d'une arête.*

Lors du passage de l'arête (p, q), la nouvelle extrémité r de l'arête issue de p est le point de tangence des droites passant par p et d'une chaîne d'arêtes déterminée (figure 1.23). Ce point est trouvé en balayant les arêtes de la chaîne.

L'utilisation de la queue de priorité coûte un temps $O(m \log n)$. Le coût du balayage des arêtes lors des mises à jour pourrait être dans le pire cas $O(mn)$. Les auteurs annoncent (sans preuve détaillée) que le coût total n'est que de $O(\alpha(n)m)$. En fait le chapitre suivant montre implicitement que cette complexité de balayage n'est au total qu'en $O(m)$.

1.4.4 Complexe de visibilité

L'arrangement dual permet d'avoir une structure de données discrète qui classifie les droites selon les objets intersectés. Cependant, comme les droites traversent les obstacles, manipuler ces ensembles de droites a l'inconvénient de prendre en compte des événements entre objets non visibles ; les complexités obtenues ne dépendent alors que de n, même si le résultat obtenu est très petit. Pour ne tenir compte que des événements entre objets visibles et obtenir des algorithmes sensibles à la sortie, Pocchiola et Vegter ne manipulent plus des droites, mais des *segments libres maximaux*. Les segments libres maximaux (que nous appellerons aussi rayons) sont des segments de droite de longueur maximale dans l'espace libre de la scène. Ils peuvent être obtenus à partir d'un point p et une direction u : à partir de p un segment est étendu dans la direction u (vers l'avant et vers l'arrière) jusqu'à ce que ses extrémités rencontrent l'intérieur d'un objet. L'étiquette d'un rayon est définie de façon similaire à celle des droites, et est en fait constituée des objets de départ et d'arrivée du rayon et des points situés entre ces objets et par lesquels passe le rayon. Deux rayons sont équivalents si on peut passer continûment de l'un à l'autre en gardant l'étiquette constante. Le complexe de visibilité représente alors les classes d'équivalences des segments libres maximaux selon cette relation d'équivalence.

Une première version du complexe de visibilité a été donnée par Vegter [Veg90] pour des scènes de segments, sous le nom de *diagramme de visibilité*. Le complexe est ensuite pleinement étudié par Pocchiola et Vegter [PV96b] dans le cadre de scènes d'objets courbes convexes. Nous présentons ici (de manière moins formelle et moins combinatoire) le complexe de visibilité dans le cadre de scènes polygonales.

Le complexe de visibilité comporte trois sortes d'éléments : des faces, des arêtes et des sommets.

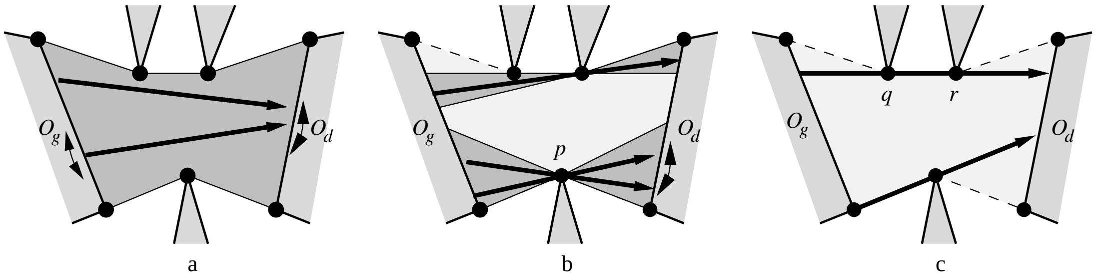

Fig. 1.24 – *Éléments du complexe de visibilité.* **a.** *Face.* **b.** *Arêtes.* **c.** *Sommets.*

Les faces sont des éléments de dimension deux : elles représentent les rayons d'étiquette (O_g, O_d), c'est-à-dire les rayons qui partent de O_g et arrivent à O_d sans toucher d'autre objet. Un tel rayon a donc deux degrés de liberté (figure 1.24a).

Les arêtes sont des éléments de dimension un : elle représentent les rayons d'étiquette $(O_g, p^{\pm\pm}, O_d)$, c'est-à-dire les rayons allant de O_g à O_d en passant par un sommet p. Ces rayons n'ont plus qu'un seul degré de liberté : ils doivent passer par p (figure 1.24b).

Les sommets sont des éléments de dimension zéro : ils représentent les rayons d'étiquette $(O_g, q^{\pm\pm}, r^{\pm\pm}, O_d)$, c'est-à-dire le rayon allant de O_g à O_d en passant par q et r. Ces rayons sont fixés : ils doivent passer par deux sommets (figure 1.24c).

De la même façon qu'il était plus facile de représenter les classes de droites dans l'espace dual grâce à l'arrangement de droites dual, nous représentons aussi les éléments du complexe de visibilité en utilisant cet espace dual. En effet, un segment libre maximal est inclus dans une droite support; nous représentons alors un tel segment par le point dual de sa droite support. L'espace dual nous permet de mieux visualiser les relations entre les divers éléments du complexe (figure 1.25).

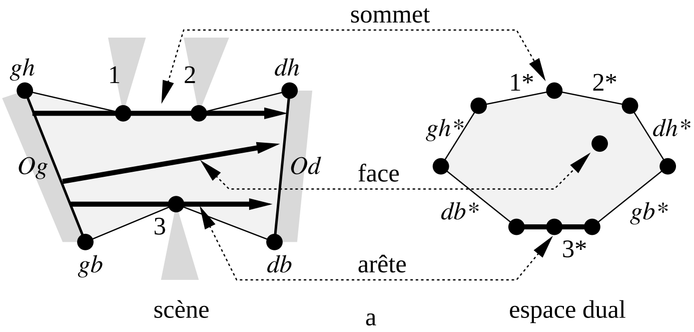

FIG. 1.25 – *Éléments du complexe de visibilité : visualisation dans l'espace dual.*

Un sommet représente un rayon passant par deux sommets p et q. Dans l'espace dual c'est donc un point, intersection des deux courbes duales p^* et q^* associées aux deux sommets. Une arête représente des rayons de visibilité constante passant par un sommet p, qui sera dit sommet associé à l'arête. Dans l'espace dual, une arête est donc une portion de la courbe duale p^* associée au sommet, délimitée par deux sommets du complexe. Une arête est incidente à deux sommets, appelés respectivement sommet gauche et sommet droit (figure 1.26a). Un sommet est incident à quatre arêtes, appelées arêtes gauche/droite-haute/basse (figure 1.26b).

Les arêtes sont classées en six catégories, en fonction de la géométrie locale du polygone autour du sommet p associé (figure 1.27) : arête λ-convexe, λ-concave, μ-convexe, μ-concave, ν-gauche et ν-droit. Selon sa catégorie, une arête peut être incidente à une, deux ou trois faces. D'une manière générale, nous désignons quatre faces incidentes à une arête par face gauche/droite-haute/basse, même si certaines de ces faces sont nulles ou confondues.

Considérons maintenant une face du complexe, associée à (O_g, O_d). Nous voyons qu'il existe un seul rayon de pente minimale (resp. maximale) dans cette face. Ces deux rayons correspondent à deux sommets extrêmes bordant la face, dits sommets gauche et droit de la face. Si nous prenons maintenant un rayon appartenant à cette face, alors nous pouvons le translater vers le bas (resp. vers le haut) en gardant la même visibilité, jusqu'à ce que nous rencontrions un sommet qui provoque, si nous continuons le déplacement, un changement de visibilité. Les deux rayons extrêmes appartiennent chacun à une arête bordant la face. En fait, la face est bordée par deux chaînes d'arêtes, une chaîne supérieure et une chaîne inférieure, toutes deux délimitées par les deux sommets extrêmes de la face. La face délimite dans la scène une région géométrique,

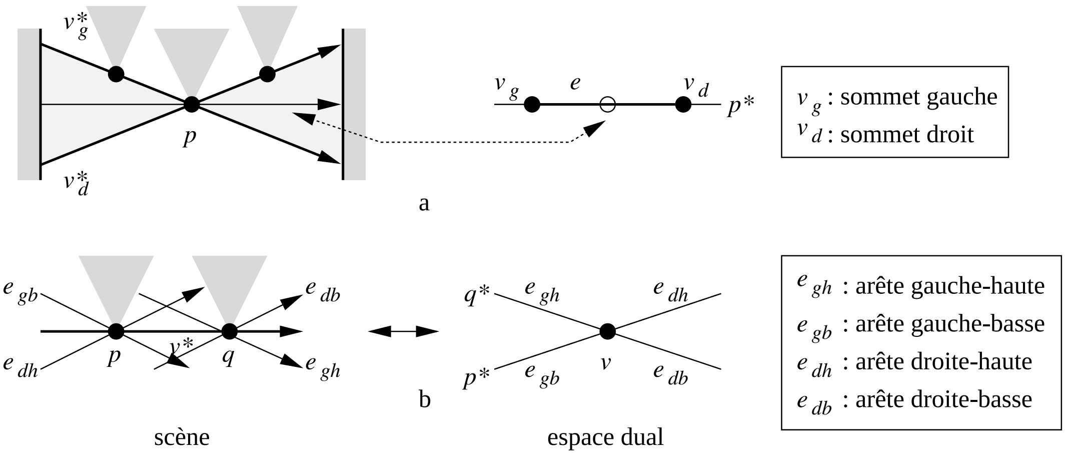

FIG. 1.26 – *Relations d'incidences entre* **a.** *arête et sommets,* **b.** *sommet et arêtes, dans le complexe de visibilité.*

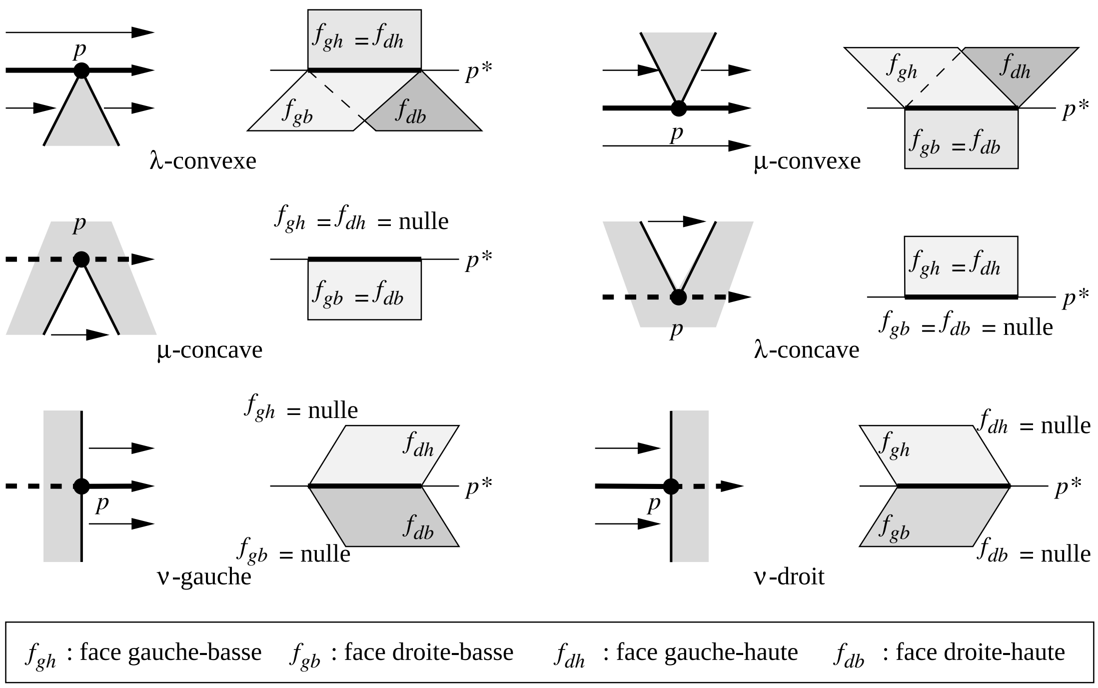

FIG. 1.27 – *Les différents types d'arêtes du complexe.*

limitée par les segments de droite associés aux sommets des chaînes d'arêtes (sommets extrêmes exclus). La figure 1.28 montre les éléments du complexe qui délimitent une face.

Si nous traçons la face dans l'espace dual en utilisant la dualité (a, b), nous voyons que c'est un polygone convexe. Cette caractéristique est due à une propriété plus générale des faces : si nous prenons dans l'espace dual deux points dans une face, alors la courbe duale passant par ces deux points est, entre ces deux points, entière-

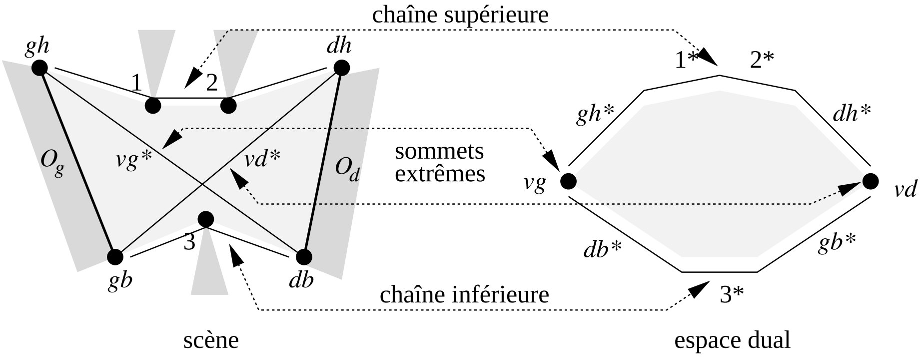

FIG. 1.28 – *Éléments délimitant une face du complexe.*

ment contenue dans la face. Cela signifie dans la scène que si nous prenons deux rayons r_1 et r_2 dans la face et notons $p = r_1 \cap r_2$ le point d'intersection des deux droites support des rayons, alors si nous déplaçons r_1 vers r_2 en faisant pivoter sa droite support autour de p, alors il ne quitte pas la face. En effet, les deux frontières de la face (les chaînes d'arêtes) sont convexes par visibilité. Cela signifie notamment que dans la scène, chaque sommet du bord haut de la face est visible des sommets du bord bas de la face et réciproquement.

Les arêtes peuvent encore être classées en deux catégories. En effet, une arête bordant une face (O_g, O_d) peut soit avoir aussi la visibilité (O_g, O_d), soit passer par l'une des extrémités de O_g ou O_d. Dans ce dernier cas, la face peut être bordée par une série d'arêtes consécutives portées par la même courbe duale (figure 1.29).

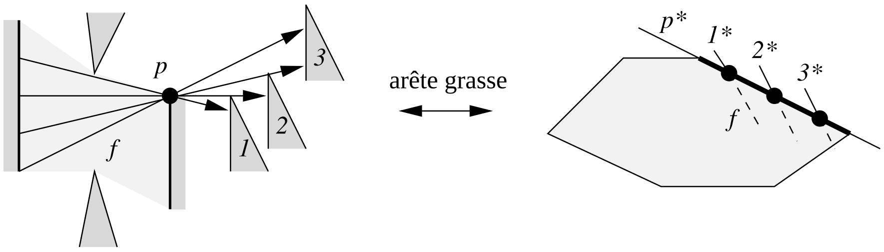

FIG. 1.29 – *Arêtes grasses.*

De telles arêtes sont dites *grasses* (terminologie empruntée à Pocchiola et Vegter [PV96b]), car une portion de courbe duale est décomposée en sous-arêtes bordant une même face. Chacune des deux chaînes d'arêtes d'une face peut avoir deux séries d'arêtes grasses : une en début de chaîne (arête grasse gauche) et une en fin de chaîne (arête grasse droite).

Contrairement à l'arrangement de courbes dual, le complexe de visibilité n'a pas une structure plane. En effet, des segments libres maximaux différents peuvent être contenus dans une même droite support. Ils sont représentés alors dans le plan dual par un même point dual. Nous plongeons ce plan dans $\mathbb{R}^3$, où ces points duaux sont « décollés » (figure 1.30).

La figure 1.31 montre la structure non plane du complexe au voisinage de l'un de

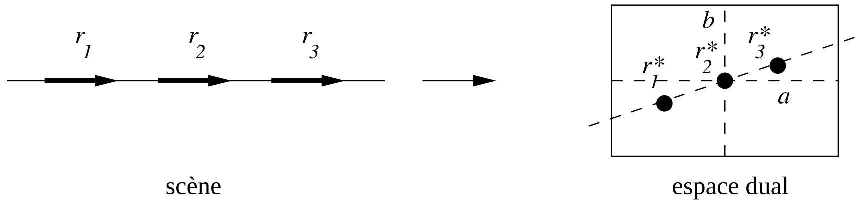

FIG. 1.30 – *Utilisation d'une dimension supplémentaire pour représenter les rayons confondus en projection dans le plan dual.*

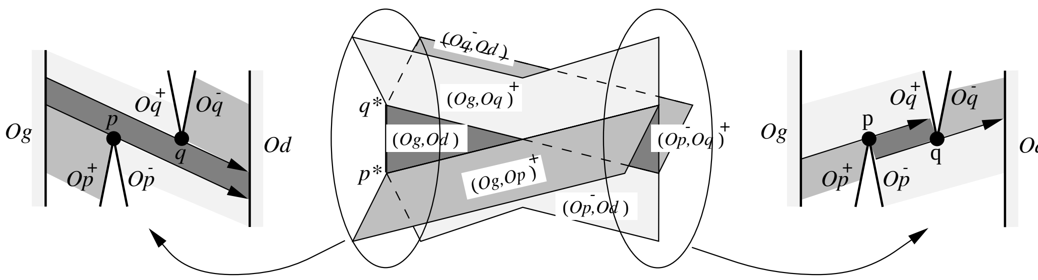

FIG. 1.31 – *Non planarité du complexe au voisinage d'un sommet.*

ses sommets et montre les deux situations dans la scène correspondant à la coupe du complexe avant (resp. après) le sommet. En particulier, deux courbes duales qui se coupent en projection dans le plan dual ne se coupent pas forcément en un sommet dans le complexe, comme c'était le cas dans l'arrangement dual : elles se coupent seulement si les deux points associés sont mutuellement visibles.

La configuration du complexe autour d'un sommet (nombre de faces incidentes, relations d'incidence) correspondant à l'arête de visibilité $[p, q]$ dépend alors de la configuration géométrique de la scène autour des points p et q. La figure 1.32 montre les 17 configurations géométriques possibles autour de ces sommets.

1.4.5 Relations avec le diagramme de visibilité

Vegter [Veg90] a introduit le diagramme de visibilité pour des scènes composées de segments de droite. Ce diagramme de visibilité est bien le complexe de visibilité de la scène, mais avec une représentation des données légèrement différente. Vegter met l'accent sur les rayons issus d'un même segment s.

Il définit d'abord pour chaque segment s une subdivision $Vis(s, S)$ (S désigne la scène) qui représente la classification en classe d'équivalences des rayons issus de s. $Vis(s, S)$ est donc l'ensemble des faces du complexe de visibilité qui ont dans leur étiquette pour objet gauche le segment s. Ces faces sont donc comprises dans l'espace dual entre les deux courbes duales p_1^* et p_2^* des deux extrémités p_1 et p_2 du segment s (figure 1.33a). Parmi les sommets de $Vis(s, S)$, ceux situés sur p_1^* (resp. p_2^*) ne sont incidents qu'à trois arêtes. Les arêtes ne sont incidentes qu'à la face (resp. qu'aux deux faces) qu'elles bordent dans $Vis(s, S)$. Le diagramme de visibilité est alors constitué

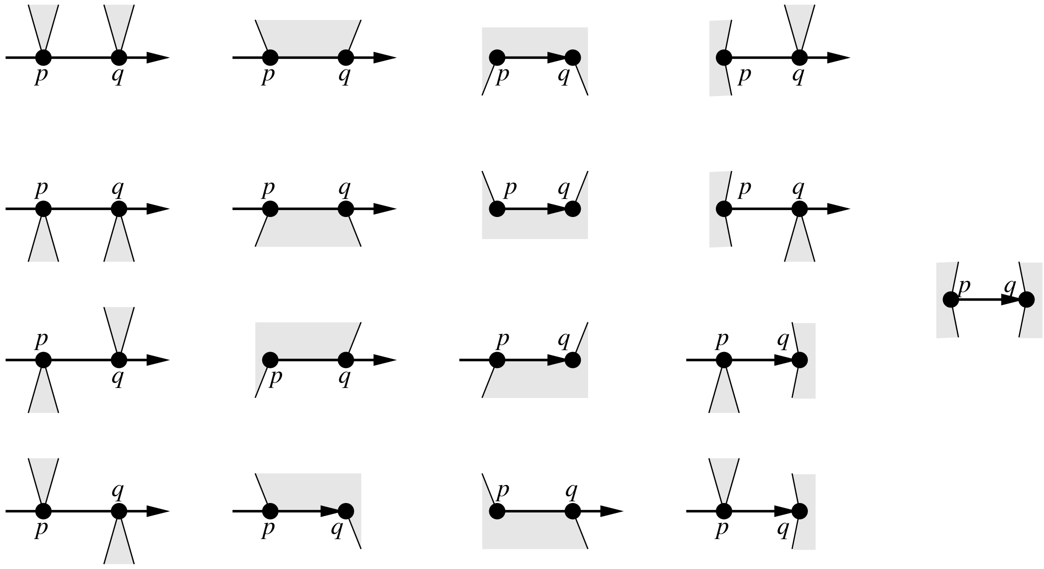

FIG. 1.32 – *Différentes configuration géométriques correspondant à un sommet du complexe.*

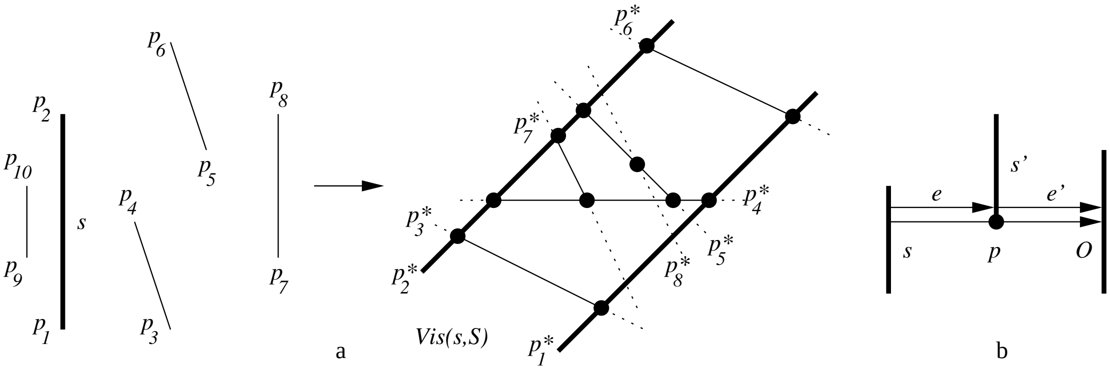

FIG. 1.33 – *Diagramme de visibilité.* **a.** *Décomposition $Vis(s, S)$ du diagramme de visibilité.* **b.** *Arêtes partenaires e et e'.*

de l'ensemble de ces subdivisions $Vis(s, S)$. Ces subdivisions sont reliées entre elles grâce aux arêtes partenaires. Un rayon issu du segment s et passant par le sommet p d'un segment s' avant d'être arrêté par un segment O peut-être associé soit à l'arête e bordant les faces $f(s, s')$ et $f(s, O)$ de $Vis(s, S)$, soit à l'arête e' bordant la face $f(s', O)$ de $Vis(s', S)$ (figure 1.33b). Les deux arêtes e et e' sont alors dites partenaires et ont chacune un pointeur sur l'autre.

Les calculs sont plutôt considérés dans une subdivision $Vis(s, S)$, avec occasionnellement un passage d'une subdivision à une autre par l'intermédiaire d'arêtes partenaires. Le diagramme de visibilité et le complexe de visibilité sont équivalents. La représentation du complexe que nous utilisons, adaptée de celle de Pocchiola et Vegter [PV96b], est un peu plus uniforme.

Nous montrons dans le chapitre suivant comment calculer de façon optimale le complexe de visibilité d'une scène polygonale.