Chapitre 4

Calculs de visibilité dynamique

Dans ce chapitre nous montrons comment maintenir la vue autour d'un point se déplaçant dans la scène. Nous donnons deux algorithmes qui ont chacun leur spécificité : le premier s'applique plutôt pour des petits déplacements successifs, le deuxième pour un long déplacement le long d'une trajectoire connue à l'avance.

Nous montrons ensuite comment maintenir le complexe de visibilité quand les objets se déplacent dans la scène et comment maintenir la vue autour de points quand les points de vue et les objets de la scène sont tous en mouvement.

— Arrêtez-vous Lagaffe !! On ne voit rien !!
— Ben ! On n'y verra pas plus si on s'arrête, malin !...
Un gaffeur sachant gaffer.

4.1 Maintien d'une vue autour d'un point mobile

Nous étudions dans cette section comment, après avoir calculé une vue autour d'un point p_v, nous pouvons maintenir cette vue quand le point de vue se déplace dans la scène.

4.1.1 Changements de visibilité

Soit une scène de polygones. Si nous étendons dans l'espace libre de cette scène les arêtes de son graphe de visibilité, nous obtenons une partition de la scène en régions de visibilité constante : tous les points situés dans une même région ont la même vue. (figure 4.1).

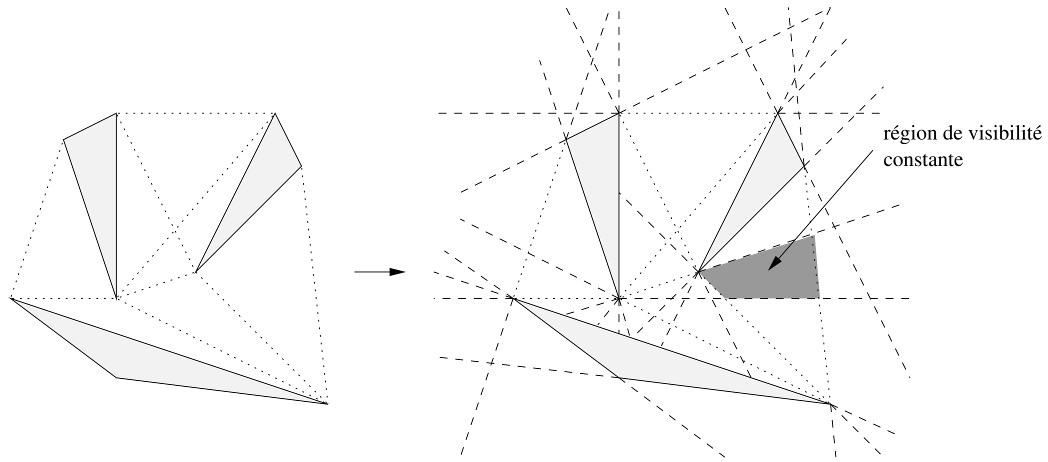

FIG. 4.1 – *Découpage en régions de visibilité constante.*

Plus précisément, nous disons que deux points p_v et p'_v ont une même vue si la structure combinatoire de leur polygone de visibilité est la même, c'est-à-dire si les côtés transversaux d'un polygone de visibilité et ceux correspondants de l'autre polygone de visibilité sont portés par les mêmes côtés d'obstacle, et si aux côtés radiaux du type (p_v, p) de l'un correspondent les côtés radiaux (p'_v, p) de l'autre ; de plus les extrémités des côtés sont donnés par les mêmes sommets de la scène ou par les points d'intersection avec les mêmes droite correspondantes. Seules les données numériques (coordonnées des sommets) peuvent être différentes, mais elles sont déduites automatiquement de la structure combinatoire du polygone de visibilité.

Même si la vue autour d'un point ne change pas réellement quand celui-ci traverse une arête non étendue du graphe de visibilité, de tels changements sont intéressants (ils peuvent signaler le passage d'une porte par exemple) et surtout ils correspondent à un changement de situation dans l'espace dual. Ces changements seront appelés aussi abusivement changements de visibilité.

L'idée générale derrière les algorithmes de maintien de vue que nous exposons, est de calculer les frontières de la région de visibilité où se trouve le point, de calculer le côté de sortie de cette région quand le point se déplace et de mettre à jour les frontières de la nouvelle région de visibilité.

Le complexe de visibilité permet d'accéder directement au découpage induit par les arêtes du graphe de visibilité. Dans le chapitre précédent, nous calculons une vue en cherchant la liste des arêtes du complexe traversées par la courbe duale p_v^* du point de vue p_v. Soit e une des arêtes traversées et soit p le point (visible depuis p_v) de la scène correspondant. Les deux sommets v_g et v_d délimitant cette arête déterminent deux arêtes du graphe de visibilité incidentes à p dont les droites support v_g^* et v_d^* encadrent (p_v, p).

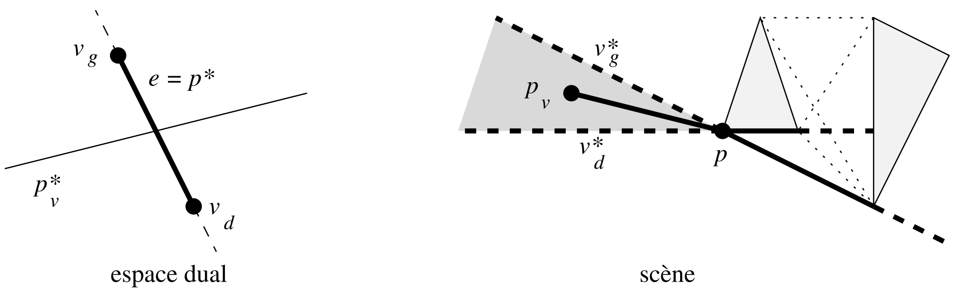

FIG. 4.2 – *Limites de changement de visibilité.*

Ces arêtes, une fois étendues, déterminent deux limites de changement de visibilité pour p_v (figure 4.2) : si le point de vue sort du coin délimité par v_g^* et v_d^*, alors sa vue change. Le coin délimité par v_g^* et v_d^* est en fait l'intersection des deux demi-plans délimités par v_g^* (resp. v_d^*) et contenant p_v. À chaque arête du complexe traversée par la courbe duale p_v sont alors associés deux demi-plans, et la région de visibilité constante associée au point de vue p_v est constituée par l'intersection de tous ces demi-plans.

De façon plus détaillée, considérons pour chaque point p de la vue l'arête e du complexe traversée par p_v^* et associée à p. Chacune des deux droites associées aux sommets délimitant e peut être classée inférieure ou supérieure : si p_v est au dessus de la droite v^*, alors la droite est dite inférieure ; si p_v est en dessous de la droite v^*, alors la droite est dite supérieure. Nous obtenons alors deux enveloppes de droites qui délimitent la région de visibilité : l'enveloppe inférieure, composée des droites inférieures, et l'enveloppe supérieure, composée des droites supérieures.

Dans l'espace dual, si le sommet du complexe v correspondant à la droite v^* est situé au-dessus (resp. en dessous) de la courbe duale du point de vue, alors la droite appartiennent à l'enveloppe supérieure (resp. inférieure), comme illustré par l'exemple de la figure 4.3.

Pour maintenir la vue, deux stratégies différentes peuvent être mises en œuvre, selon que l'on effectue des petits déplacement successifs, ou que l'on effectue un grand déplacement le long d'une trajectoire.

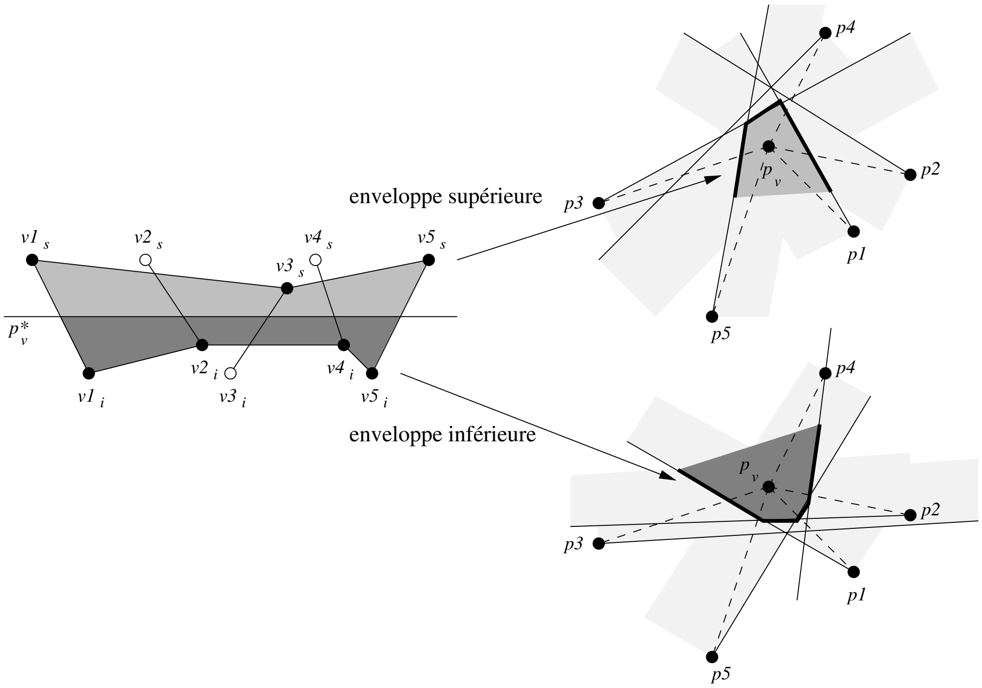

FIG. 4.3 – *Limites supérieure et inférieure de visibilité.*

4.1.2 Maintien de la vue lors d'un petit déplacement

Supposons que le point de vue se déplace en ligne droite de p_v à p'_v. Les deux enveloppes de droites délimitant la visibilité se coupent en deux points. En testant la position de p'_v par rapport à la droite passant par ces deux points (droite qui sépare en quelque sorte les deux enveloppes), nous savons par quelle enveloppe le point risque de sortir. Nous vérifions ensuite si le point de vue est vraiment sorti de cette enveloppe grâce à une recherche dichotomique utilisant la convexité de l'enveloppe.

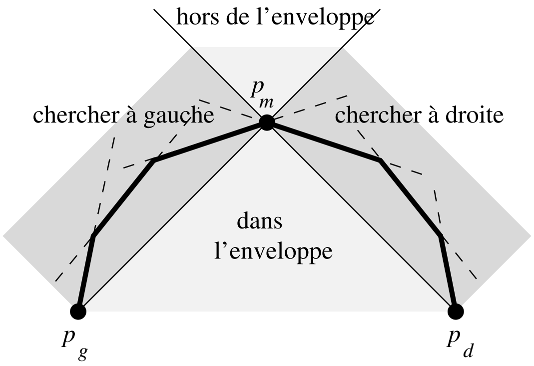

FIG. 4.4 – *Tester la sortie du point de vue de la région de visibilité.*

En effet, considérons le sommet milieu p_m de l'enveloppe et les droites passant par ce sommet et les points extrêmes p_g et p_d de l'enveloppe (figure 4.4). Selon la position de p'_v par rapport à ces deux droites, nous savons si le point est dans l'enveloppe, hors de l'enveloppe, ou s'il faut poursuivre la vérification dans la partie gauche (resp. droite) de l'enveloppe. Cette vérification prend un temps logarithmique $O(\log v)$.

Nous pouvons de plus utiliser cette vérification pour trouver le cas échéant la droite de sortie : si (p_v, p'_v) coupe le segment $[p_g, p_m]$, alors p'_v sort par la partie gauche de l'enveloppe. Si (p_v, p'_v) coupe le segment $[p_m, p_d]$, alors p'_v sort par la partie droite. Nous pouvons donc trouver la droite de sortie aussi en temps $O(\log v)$.

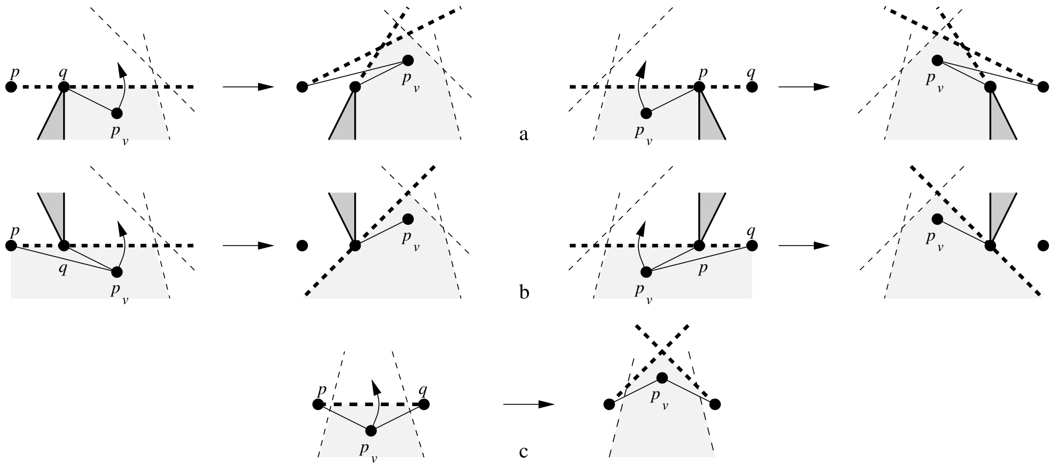

FIG. 4.5 – *Mise à jour de la vue et de l'enveloppe supérieure.*

Les modifications à effectuer pour mettre à jour la vue et l'enveloppe lors du franchissement d'une arête étendue du graphe de visibilité sont locales : si un sommet de l'arête devient visible (figure 4.5a), il est ajouté à la vue et deux nouvelles limites de visibilité sont insérées dans l'enveloppe de sortie ; si un sommet de l'arête devient invisible (figure 4.5b), il est enlevé de la vue et seule la nouvelle limite de visibilité associée à l'autre sommet de l'arête est insérée dans l'enveloppe de sortie ; enfin si le point de vue passe par l'arête même (non étendue) du graphe de visibilité (figure 4.5c), les deux sommets de l'arête restent visibles et les nouvelles limites de visibilité associées aux sommets remplacent dans l'enveloppe de sortie les anciennes limites.

Les modifications correspondantes – lors du franchissement de l'arête étendue (p, q) du graphe de visibilité – dans le complexe de visibilité concernent la liste des arêtes du complexe traversées par la courbe duale du point de vue. Ces modifications sont aussi locales. Si p (resp. q) reste visible, alors l'arête correspondante dans la liste doit être remplacée par l'arête suivante (ou précédente) portée par la même courbe duale p^* (resp. q^*) (figure 4.6c). Si p (resp. q) devient invisible, alors l'arête correspondante est enlevée de la liste (figure 4.6b). Si p (resp. q) devient visible, alors l'arête correspondante doit être ajouté dans la liste (figure 4.6a). Notons que si les deux arêtes associées aux deux points sont coupées par p_v^*, alors elles sont consécutives dans la liste d'arêtes coupées par p_v^*.

La mise à jour de la vue (et de la liste des arêtes du complexes coupées par p_v^*) se fait en temps constant grâce à l'utilisation par exemple d'une liste doublement chaînée. Le maintien de l'enveloppe de droites s'effectue en utilisant des algorithmes de maintien dynamique d'enveloppe convexe.

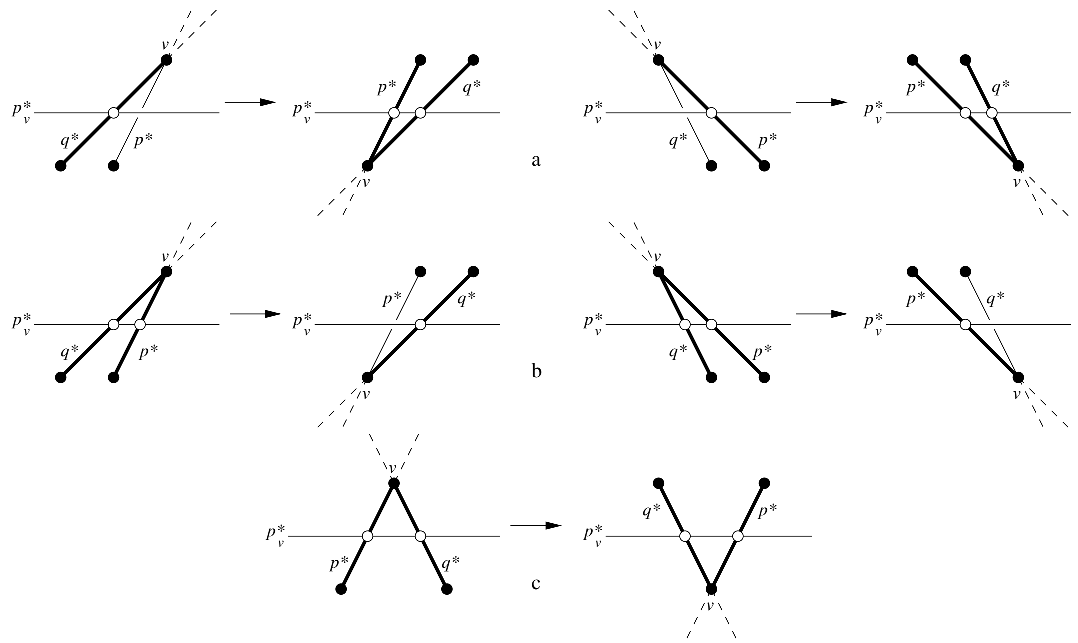

FIG. 4.6 – *Cas de mise à jour de la vue dans le complexe correspondant aux cas de la figure précédente.*

Proposition 4.1 *Après un calcul initial effectué en temps $O(v_0 \log v_0)$ (v_0 taille de la vue initiale), la vue autour d'un point peut être maintenue lors du déplacement du point de vue en temps $O(\log^2 v)$, ou en temps randomisé $O(\log v)$, à chaque changement élémentaire de visibilité, v désignant la taille de la vue à l'étape considérée.*

Démonstration. Vérifier si le point sort de l'enveloppe convexe et calculer le cas échéant l'arête de sortie se fait en temps $O(\log v)$. Le maintien dynamique de l'enveloppe s'effectue en temps $O(\log^2 v)$ en utilisant l'algorithme de maintien d'enveloppe convexe d'Overmars et van Leeuwen [OvL81], ou en temps randomisé $O(\log v)$ en utilisant l'algorithme de Dobrindt et Yvinec [DY93]. La construction initiale de l'enveloppe prend alors un temps $O(v_0 \log v_0)$, et un espace $O(v_0)$ (resp. $O(v_0 \log v_0)$ dans le cas randomisé). □

Cet algorithme est plutôt adapté à des petits déplacements, successifs, où peu de changements de visibilité interviennent et où l'important est d'avoir rapidement la vue finale. Nous avons supposé que le point de vue s'était déplacé en ligne droite. Nous pouvons, sans modification de l'algorithme, considérer d'autres mouvements pas trop « tordus » , c'est-à-dire qui ne faussent pas les test de localisation par rapport aux droites de l'enveloppe.

4.1.3 Maintien de la vue le long d'une trajectoire

Si nous considèrons maintenant que le problème est :

étant donnée une trajectoire sur laquelle se déplace le point de vue, calculer tous *les changements de visibilité qui interviennent le long de cette trajectoire,*

alors une autre stratégie peut être utilisée.

Nous supposons que le point de vue se déplace de p_v à p'_v sur une trajectoire telle que nous puissions savoir en temps constant si une droite coupe cette trajectoire et calculer alors (en temps constant) le point d'intersection. Nous supposons aussi que nous pouvons comparer en temps constant deux points sur cette trajectoire, c'est-à-dire savoir quel est le point situé avant l'autre sur la trajectoire.

Après avoir calculé la vue autour de p_v, nous considérons pour chaque point de la vue les deux droites associées qui délimitent les frontières de visibilité. Nous calculons les points d'intersection (s'ils existent) entre ces droites et la trajectoire du point de vue. Nous mettons ces points dans une queue de priorité en fonction de leur ordre de position sur la trajectoire et passons à chaque fois le premier point de la queue. À chaque passage d'un point d'intersection, nous mettons à jour la vue en y insérant ou supprimant un sommet. Lors de cette mise à jour, de nouvelles droites frontières de visibilité peuvent apparaître. Nous calculons alors leur intersection éventuelle avec la trajectoire et mettons à jour la liste ordonnée des points d'intersection (figure 4.7).

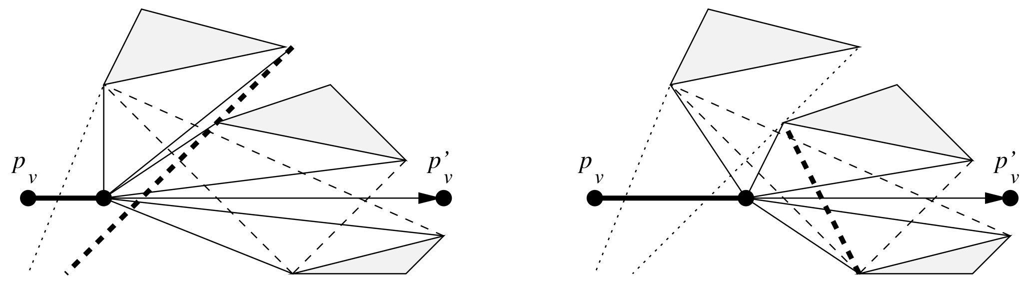

FIG. 4.7 – *Passage d'un changement de visibilité.*

Proposition 4.2 *Soit un point de vue se déplaçant suivant une trajectoire déterminée à l'avance. Après un calcul initial en* $O(v_0 \log v_0)$*, la vue autour de ce point peut être maintenue en temps* $O(\log v)$ *à chaque changement de visibilité le long de cette trajectoire. Les* k *changements de visibilité sont donc calculés en temps* $O(v_0 \log v_0 + k \log(v_0 + \frac{k+1}{2}))$.

Démonstration. Le maintien de la liste des points d'intersection a un coût logarithmique à chaque étape, et la vue augmente d'au plus un sommet à chaque mise à jour. □

Le premier algorithme est plutôt adapté à des petits déplacement successifs. En effet, le calcul de l'enveloppe de droites (en $O(v_0 \log v_0)$) n'est fait qu'une fois au début, et ensuite chaque changement de visibilité ne coûte qu'un temps $O(\log^2 v)$ (ou $O(\log v)$ randomisé), quels que soient les déplacements successifs. Le deuxième algorithme doit recalculer et trier (en temps $O(v \log v)$) les points d'intersection entre la trajectoire et les limites de visibilité à chaque changement de trajectoire.

En revanche, sur un long déplacement donné, le coût plus faible de mise à jour à chaque changement de visibilité du deuxième algorithme ($O(\log v)$) devient plus intéressant que celui du premier algorithme ($O(\log^2 v)$), surtout que les algorithmes de maintien dynamique d'enveloppe convexe sont sensiblement plus complexes que le simple maintien d'une queue de priorité.

4.2 Maintien dynamique du complexe de visibilité

Nous étudions dans cette section le maintien du complexe de visibilité quand les objets de la scène sont en mouvement. Nous étudions d'abord la mise à jour du complexe lors d'un changement élémentaire de visibilité, puis nous montrons comment gérer ces changements élémentaires pour maintenir le complexe.

4.2.1 Changements élémentaires de visibilité

Quand les coordonnées des sommets des polygones de la scène changent, les données numériques du complexe – coordonnées des sommets (c'est-à-dire pente des arêtes du graphe de visibilité) par exemple – sont modifiées. Cependant, les informations les plus importantes contenues dans le complexe sont les informations concernant les relations de visibilité entre les objets de la scène : une fois ces relations connues, les calculs numériques sont immédiats. Or ces informations de visibilité ne dépendent que de la topologie du complexe. Nous ne devons prendre en compte les changements dans la scène que lorsqu'ils produisent des modifications dans les relations de visibilité entre objets, modifications qui se traduisent par une modification de la topologie du complexe de visibilité.

Toute modification de la structure du complexe dans une scène dynamique se ramène à une succession de *changements élémentaires de visibilité* – changements de visibilité les plus simples – successifs, qui ne produisent qu'une modification topologique locale dans la structure du complexe.

Ces changements élémentaires de visibilité surviennent au voisinage de trois sommets p_1, p_2 et p_3 de la scène reliés successivement par les arêtes (p_1, p_2) et (p_2, p_3) du graphe de visibilité. Les changements se produisent quand, lors du déplacement, les trois points deviennent alignés puis changent de position relative : le sommet p_2 initialement au-dessus (resp. en dessous) de la droite (p_1, p_3) passe en dessous (resp. au-dessus) de cette droite (figure 4.8). La visibilité entre les points extrêmes p_1 et p_3 est alors modifiée : s'ils étaient mutuellement visibles, alors ils deviennent cachés par l'obstacle en p_2 ; s'ils étaient cachés par cet obstacle, ils deviennent alors mutuellement visibles.

La figure 4.9 montre les modifications qui interviennent dans la topologie du squelette du complexe lors des changements élémentaires de visibilité de la figure 4.8. Dans le premier cas, les courbes duales p_1^* et p_3^* qui se coupaient en projection (resp. en le sommet v_{13} du complexe) en dessous de p_2^* glissent et se coupent maintenant au-dessus de p_2^* en le sommet v_{13} du complexe (resp. en projection).

Considérons maintenant plus en détail tous les éléments du complexe au voisinage de ces trois courbes. Différentes modifications interviennent lors de la mise à jour.

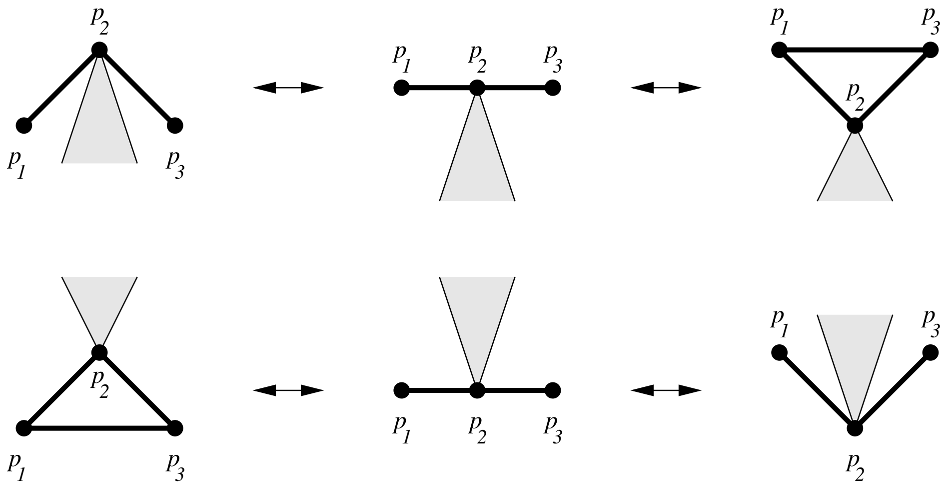

FIG. 4.8 – *Changements élémentaires de visibilité au voisinage de trois points.*

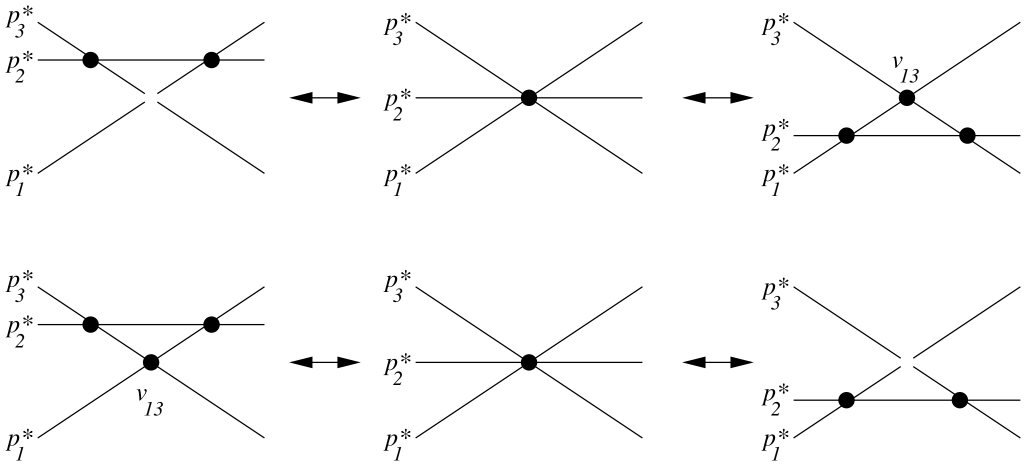

FIG. 4.9 – *Changements élémentaires de topologie du squelette du complexe.*

La figure 4.10 montre les changements de base communs à toutes les configurations géométriques correspondant aux deux situations de la figure 4.8. Dans la première situation, l'arête e_2 du complexe est détruite ; la face f, les trois arêtes qui la bordent et le sommet v_{13} sont créés ; les relations d'incidences entre les arêtes et les sommets v_{12} et v_{23} sont modifiées ; enfin les relations d'incidences entre les arêtes et les faces sont elles aussi modifiées.

Ce schéma commun doit cependant être précisé et complété, car les modifications complètes sont différentes pour chacune des 50 configurations géométriques possibles autour des trois points, montrées dans la figure 4.11. Ainsi, le type des arêtes créées dépend de la configuration géométrique. De plus d'autres faces incidentes aux arêtes portées par les courbes duales p_1^*, p_2^* et p_3^* que celles indiquées dans la figure 4.10 sont affectées par la suppression (resp. création) de l'arête e_2 et de celles bordant la face f. Le nombre et la position de ces faces sont eux aussi différents pour chacune des

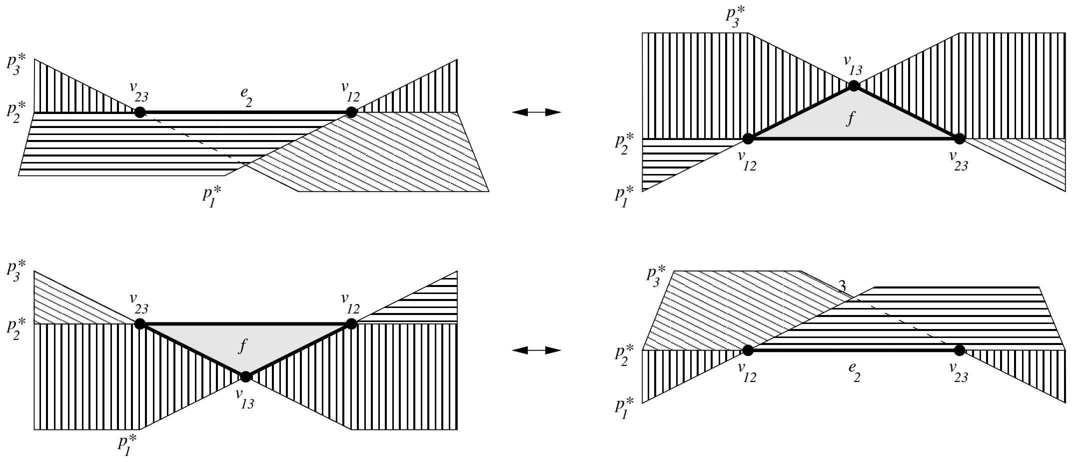

FIG. 4.10 – *Modifications de base dans le complexe lors d'un changement de visibilité.*

configurations géométriques.

Cependant, toutes ces configurations ne sont pas entièrement indépendantes. Nous avons regroupé ces configurations dans la figure 4.11 par famille : chaque ligne comporte des configurations proches de la première configuration de la ligne. Les différences au sein d'une famille se traduisent principalement par l'absence de quelques faces par rapport à la configuration de base.

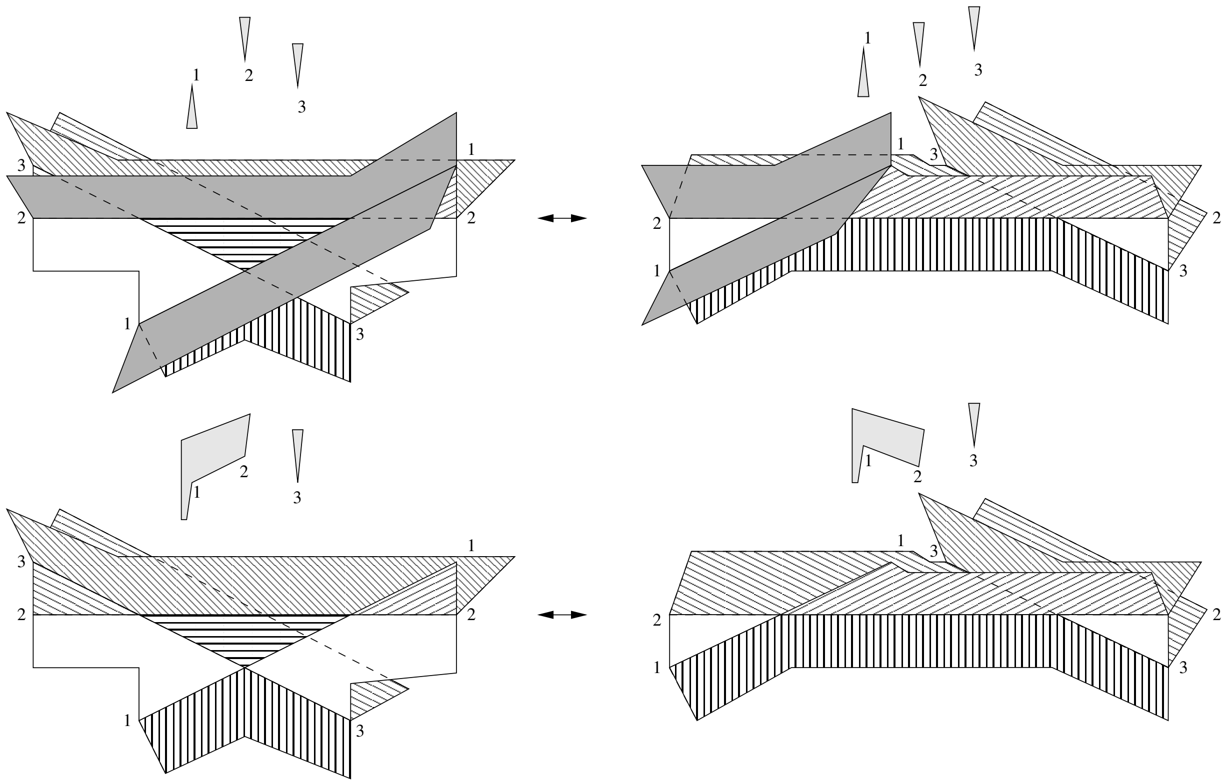

FIG. 4.12 – *Exemple de différences entre deux configurations.*

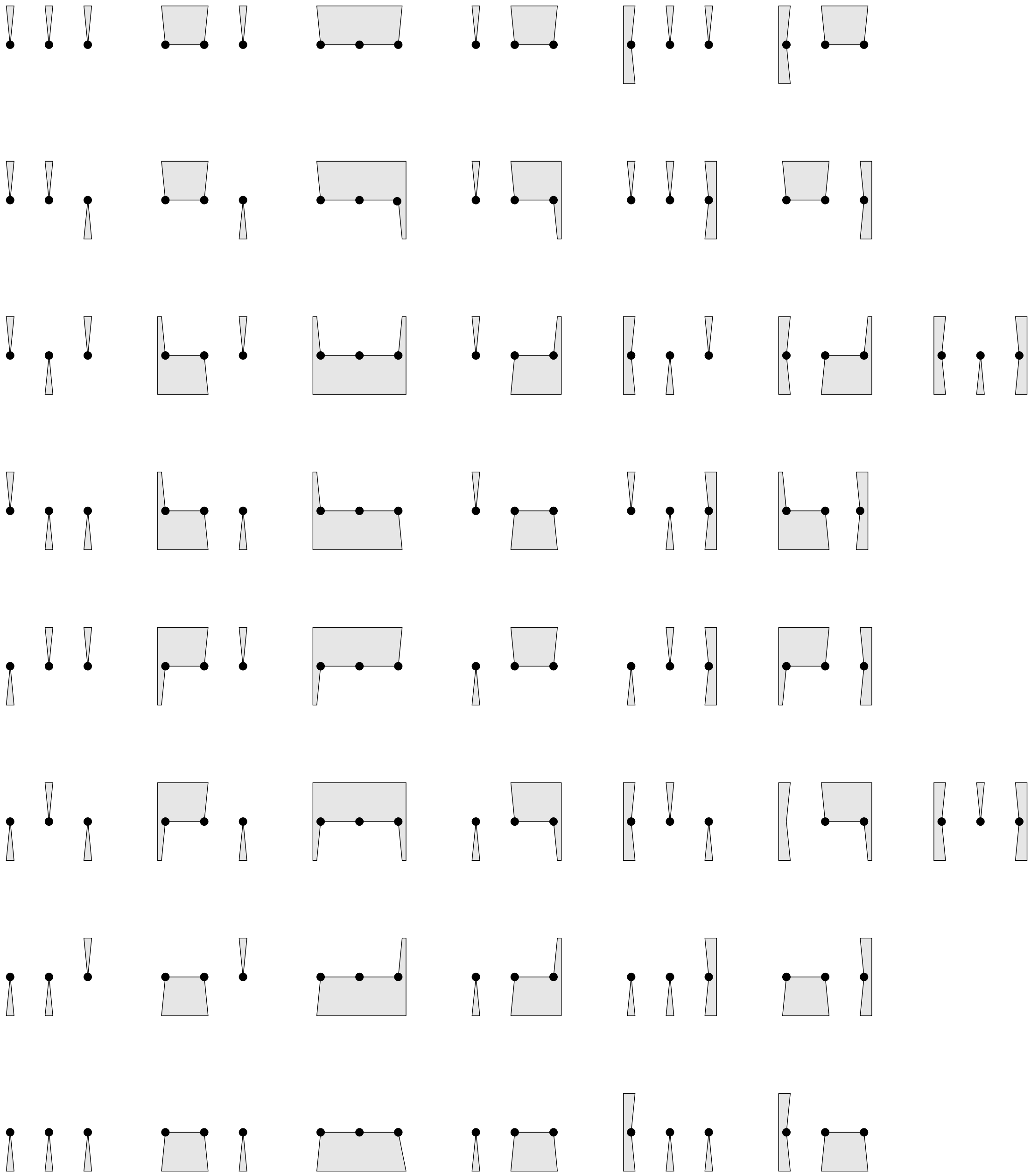

Fig. 4.11 – *Les 50 configurations géométriques au voisinage de trois points.*

Ainsi, dans la figure 4.12 nous avons indiqué en gris uniforme les deux faces manquantes entre deux configurations d'une même famille choisies en exemple.

Ces changements élémentaires servent d'opération de base pour maintenir le complexe : tout changement dans la structure du complexe se produisant lors du déplacement des points se ramène à une suite de changements élémentaires.

4.2.2 Maintien du complexe pour une scène mouvante

Nous supposons que les sommets des polygones ont une trajectoire connue à l'avance, déterminée par la position $p_i(t)$ du point p_i, où t varie de 0 à 1. Nous cherchons à maintenir le complexe depuis $t = 0$ jusqu'à $t = 1$. Nous pouvons pour cela maintenir la liste des changements élémentaires de visibilité à venir.

Les changements élémentaires se produisent quand, dans le complexe, une arête devient réduite à un sommet. Cependant, dans le cas où les trois points p_1, p_2 et p_3 sont mutuellement visibles, trois arêtes deviennent dégénérées lors de ce changement. Nous décidons alors de ne considérer que les arêtes $e = (v_g, v_d)$ qui sont arête haute (resp. basse) de leurs sommets extrêmes v_g et v_d. Ainsi une seule arête est associée à un changement élémentaire de visibilité, que les points extrêmes p_1 et p_3 soient mutuellement visibles ou non. Pour chacune de ces arêtes e, nous calculons une « date de mort », qui est la date t_e où les droites associées aux sommets $v_g(t_e)$ et $v_d(t_e)$ de l'arête seront confondues et où le changement de visibilité aura lieu.

Il suffit alors d'effectuer les changements associés aux arêtes dans l'ordre de leur date de mort. À chaque changement, il faut mettre à jour les arêtes à considérer ainsi que leur date de mort. Cependant, comme le changement est local, ces modifications sont elles aussi locales.

La figure 4.13 montre un changement de visibilité autour de l'arête e_{2m}. Cette arête est détruite et parmi les trois arêtes créées, seule e_{2m} doit être considérée. Les arêtes e_{3g} et e_{1d} ont (si elles sont à considérer) une nouvelle date de mort qu'il faut calculer. Les arêtes e_{1g} et e_{3d} ont un statut et une (éventuelle) date de mort inchangés. Enfin, les arêtes e_{2g} et e_{2d} changent de statut : si elle étaient à considérer, elles ne le sont plus ; si elles n'étaient pas à considérer, il faut vérifier si elles le deviennent et calculer le cas échéant leur date de mort.

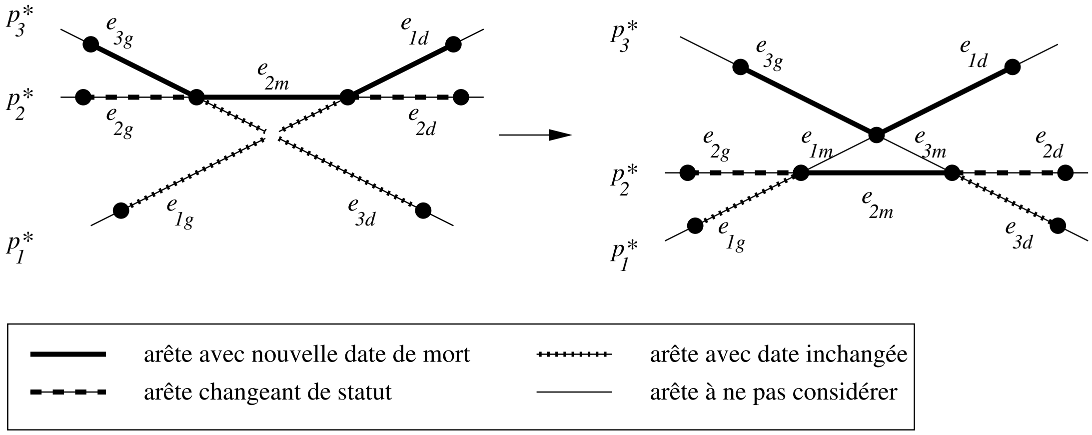

FIG. 4.13 – *Mise à jour des dates de mort : arêtes à considérer.*

Grâce à l'utilisation d'une queue de priorité, nous pouvons effectuer les changements élémentaires de visibilité dans l'ordre et maintenir le complexe de visibilité lors du mouvement des objets :

Proposition 4.3 *Après un calcul initial en temps $m_0 \log m_0$, où m_0 est la taille du* *complexe à $t = 0$, le complexe de visibilité peut être maintenu de $t = 0$ à $t = 1$ en temps $O(\log n)$ à chaque changement élémentaire de visibilité.*

Démonstration. Il y a $O(m)$ arêtes à considérer à chaque instant. L'initialisation de la queue de priorité prend un temps $O(m_0 \log m_0)$. À chaque traitement d'un changement de visibilité, les changements à effectuer sont locaux et se font en temps constant. La mise à jour de la queue de priorité prend un temps $O(\log m) = O(\log n)$. □

Dans le pire cas, il peut y avoir effectivement $O(n^2)$ arêtes à considérer (n triangles presque alignés).

4.2.3 Maintien d'une vue dans une scène mouvante

Nous cherchons maintenant à maintenir la vue autour d'un point mobile dans une scène où les objets sont aussi sujets à modifications. Si nous voulions réutiliser les algorithmes de maintien de vue dynamique du chapitre précédent, il faudrait remettre à jour les structures de données utilisées lors de ce maintien (enveloppes convexes par exemple) à chaque changement de topologie dans le complexe.

Il est plus simple d'inclure les changements de visibilité du point de vue avec les changements de visibilité entre objets de la scène. Nous avons vu que le calcul d'une vue consistait à calculer les arêtes du complexe de visibilité coupée par la courbe duale du point de vue, et que la vue changeait quand le courbe duale passait par l'un des sommets des arêtes coupées. Nous calculons donc pour chaque arête de la coupe les dates t où le point de vue $p(t)$ sera aligné avec les points correspondant à chaque sommet de l'arête. Ces dates sont rajoutées dans la queue de priorité avec les dates de changement élémentaires de visibilité.

La mise à jour lors d'un changement dans la vue du point $p(t)$ s'effectue comme décrit dans la section précédente. Nous devons juste ensuite calculer les dates de passage avec les nouvelles arêtes coupées par $p^*(t)$. La mise à jour et le calcul des nouvelles dates se fait en temps constant. L'insertion des nouvelles dates dans la queue de priorité se fait en temps logarithmique.

Lors d'un changement élémentaire de visibilité, en plus de la mise à jour du complexe, il faut vérifier si les arêtes modifiées ne font pas partie des arêtes coupées par le point de vue. Le cas échéant, il faut recalculer les dates de franchissement de ces arêtes. Vérifier si une arête est dans la coupe du point de vue peut se faire en temps constant en maintenant un tableau de taille n indexé par les points de la scène dont chaque case contient un pointeur sur l'arête de la coupe associée au point indice (le pointeur est NULL s'il n'y a pas d'arête associée dans la coupe). Ce surplus de mise à jour se fait aussi en temps constant, et la mise à jour de la queue de priorité en temps logarithmique.

Le maintien d'une vue dans une scène dynamique se fait en temps $O(\log n)$ à chaque changement élémentaire de visibilité dans la scène ou à chaque changement élémentaire dans la vue.

Si maintenant nous voulons maintenir la vue autour de n_p points, en ne tenant pas compte de la visibilité entre ces points de vue, alors avec un espace mémoire supplémentaire en $O(n_p n)$, nous pouvons maintenir les visibilités en temps $O(log(n^2 + nn_p))$ à chaque changement.