



BIARRITZ - Mai 1984 -

DETOURAGE INTERACTIF DE PHOTOGRAPHIES DE PRESSE
 COMME RECHERCHE DE CHEMIN OPTIMAL DANS UN GRAPHE.

INTERACTIVE CUT-OUT OF PRESS PHOTOGRAPHS AS OPTIMAL PATH SEARCH.

Marc Bertier et Michel Habib.

Ecole des Mines, 42023 Saint-Etienne, FRANCE.

RESUME

Le **détourage** des photographies est une opération courante lors de la mise en page de documents comportant des textes et des images (presse, publicité, édition ...). Cette opération consiste à isoler l'image d'un objet du fond sur lequel il apparaît. Il s'agit donc de **segmentation d'image**.

Cependant, la **diversité quasi-infinie des photographies** à traiter, et la nécessité de laisser à **l'opérateur** la maîtrise de l'opération, démarquent le problème du détourage de celui de la segmentation automatique en imagerie médicale, en robotique, télé-détection...

La formulation du problème de segmentation comme **recherche de chemin optimal dans un graphe**, souvent utilisée, est reprise ici. Nous en proposons une présentation unifiée et synthétique, qui fait apparaître :

- que la modélisation en chemin de coût minimal dans un graphe à valuations positives est un artifice heuristique dont il faut maîtriser les effets;
- que l'algorithme A* n'a pas d'avantages décisifs, dans cette application, sur l'algorithme de Dijkstra;
- que l'on peut conjecturer qu'une modélisation en chemin maximal est "NP-difficile";
- et que d'autres heuristiques doivent être envisagées.

Nous présentons enfin la maquette d'un logiciel interactif de détourage et en particulier le mode d'interaction choisi et les techniques utilisées pour une implémentation efficace de la recherche de chemin optimal.

oOo

I. INTRODUCTION.

II. DETOURAGE, SEGMENTATION ET INTERPRETATION.

III. EXTRACTION DE CONTOUR COMME RECHERCHE DE CHEMIN OPTIMAL.

- III.1. Modélisation en chemin optimal.
- III.2. Fonctions de coût.
- III.3. Algorithmes.

IV. PRESENTATION D'UN LOGICIEL INTERACTIF DE DETOURAGE.

- IV.1. Utilisation du programme.
- IV.2. Méthodes proposées.
- IV.3. Implémentation de l'algorithme de Dijkstra.

SUMMARY

Cuting out photographs is a common task in the making up of documents including texts and images (Press, Advertising, and Edition). This operation consists in isolating the picture of an object from the background it appears upon. It does appear to deal with image segmentation.

However, the infinite variety of the photographs to be processed, and the fact the operator needs to control the operation, make this task distinct from automatic segmentation in medical imagery, robotics, teledetection etc...

The stating of the segmentation task as an optimal path search, often used, is still used herein. We propose an unified and synthetic approach in which it appears that :

- the modelisation as minimal cost path in a positively valued graph, is a heuristic trick whose effects must be controlled;
- the A* algorithm does not have crucial advantages, in this application, over Dijkstra's algorithm;
- the conjecture of the "NP-hardness" of the modelisation as maximal path can be stated;
- others heuristics need be looked at.

We present at last the model of an interactive program for cutting out photographs, particularly the interaction mode we chose and the techniques we used for an efficient implementation of the optimal path search.

oOo

I. INTRODUCTION.

II. CUT-OUT, SEGMENTATION AND INTERPRETATION.

III. CONTOUR EXTRACTION AS AN OPTIMAL PATH SEARCH.

- III.1. Modelisation as optimal path.
- III.2. Cost functions.
- III.3. Algorithms.

IV. AN INTERACTIVE PROGRAM FOR CUT-OUT.

- IV.1. The use of the program.
- IV.2. The methods we propose.
- IV.3. Dijkstra's algorithm implementation.

oOo



I. INTRODUCTION.

Un document à imprimer contient essentiellement trois types d'objets : des textes, des graphismes et des images. Dans la production des objets des deux premiers types, l'informatique s'est révélé la source d'un gain de productivité appréciable (Bureautique, photocomposition, D.A.O., "Business Graphic").

Dans le domaine des images, l'utilisation des techniques informatiques est également ancienne. Toutefois jusqu'à ces dernières années, il s'agissait d'applications (militaires, spatiales, médicales, etc...) complètement déconnectées des filières traditionnelles de production et de traitement d'images. Les progrès technologiques (matériel et logiciel) induits par ces travaux, permettent de concevoir aujourd'hui, autour de mini-ordinateurs dotés d'une périphérie spécialisée, des postes de travail informatisés pour la production de documents mêlant textes, graphiques au trait et images, et destiné aux métiers du photomontage et de la mise en page pour la presse, l'imprimerie, l'édition, la publicité, etc...

En photomontage, on appelle **détourage**, l'opération qui consiste à extraire d'une photographie, l'image d'un objet y figurant; par exemple pour l'insérer ensuite dans une autre scène ou sur un autre fond (incrustation). Cette opération doit pouvoir être réalisée sur un poste de mise en page informatisé. Il est même souhaitable d'alléger et d'assouplir cette tâche. Cela est possible, en permettant à l'opérateur et sans aucune manipulation matérielle effective, de faire des tentatives multiples, de visualiser les résultats, de revenir sur des choix précédents, etc...

Une méthode triviale consiste à faire introduire par l'opérateur, à l'aide d'un outil d'interaction adapté ("souris"), un grand nombre de points sur le contour à suivre, et à les interpoler. Elle est lourde, donne des résultats médiocres et utilise fort mal la puissance de l'ordinateur qui interpole indépendamment de l'image, donc en aveugle.

II. DETOURAGE, SEGMENTATION ET INTERPRETATION.

Tel qu'il a été défini plus haut, le détourage constitue un cas particulier de segmentation d'une image en deux zones : objet et fond. Le seul véritable critère ("critère objectif") de classement d'un pixel dans l'une ou l'autre zone, est que la lumière qu'il reçoit a été réfléchi en dernier lieu par l'objet ou non.

Dans de nombreuses applications, ce critère a un effet extrêmement direct et simple (et pour tout dire inversible) sur l'information numérique dont on dispose en un point de l'image ou en tout cas localement. Dans de tels cas, une segmentation entièrement automatique est alors praticable : segmentation des images à histogramme bimodal (ou qui ont localement un histogramme bimodal) par comparaison à un seuil (adaptatif); segmentation par suivi des points de gradient maximum, etc.../ROSENFELD et KAK 82/.

Dans les photographies du corpus très large que l'on envisage ici, la limite entre l'image d'un

objet et le fond a aussi, le plus souvent, un effet mesurable sur la luminosité et/ou la variation locale de luminosité ou de texture des points de l'image. Mais sans même invoquer le cas de photographies énigmatiques, les exceptions et les contre-exemples sont suffisamment nombreux et fréquents pour interdire un détourage entièrement automatique :

- parfois la limite objet/fond apparaît faiblement ou même pas du tout; on peut penser que ceci se produit beaucoup moins fréquemment avec des photographies en couleur qu'avec des photographies en noir et blanc, toutefois le problème se pose quand même;

- dans d'autres cas, apparaissent des variations importantes de luminosité qui ne correspondent pas à la limite de l'objet à détourer, mais, soit à des variations d'éclairage, soit aux limites d'autres objets de la scène. Ce dernier cas fait bien apparaître la nécessité absolue de laisser à un opérateur la **formulation du choix** de celui des objets qu'il désire détourer.

Dans tous les cas, ces difficultés sont résolues par l'observateur humain parce qu'il **interprète** l'image et **reconnaît** l'objet à détourer.

Développer des programmes ayant cette faculté semble encore -malgré les recherches en cours dans de nombreux laboratoires- un objectif hors d'atteinte. En tout cas, les performances à en attendre aujourd'hui et sur le type de machine envisagé, et la taille d'un tel programme sont hors de proportion avec le but visé et une telle entreprise serait à coup sur, économiquement inacceptable.

C'est pourquoi une solution interactive s'impose. Elle permet d'exploiter conjointement la capacité que possède l'opérateur d'interpréter quasi-instantanément l'image et celle d'un programme de manipuler sans difficulté particulière les contours à la résolution maximale.

Le principe retenu dans notre programme de détourage interactif, consiste à faire indiquer par l'opérateur la silhouette grossière de l'objet à détourer. Pour cela, l'opérateur devra introduire un nombre (que l'on espère petit) de points du contour. Le programme analyse alors l'image dans le voisinage de ce contour grossier pour extraire le contour exact.

PROBLEME 1.

Chercher le contour qui relie deux pixels sélectionnés successivement par l'opérateur.



**DETOURAGE INTERACTIF DE PHOTOGRAPHIES DE PRESSE
COMME RECHERCHE DE CHEMIN OPTIMAL DANS UN GRAPHE.**

Marc Bertier et Michel Habib.

INTERACTIVE CUT-OUT OF PRESS PHOTOGRAPHS AS OPTIMAL PATH SEARCH.

III. EXTRACTION DE CONTOUR COMME RECHERCHE DE CHEMIN OPTIMAL.

III.1. Modélisation en chemin optimal.

Le problème 1 se formule assez naturellement comme recherche d'un chemin optimal dans un graphe. L'idée implicitement utilisée par /MONTANARI 71/, puis proposée par /MARTELLI 72/, a été reprise souvent dans des travaux dont on trouvera la liste dans /BALLARD et BROWN 83/. Nous avons choisi la formulation ci-dessous.

LE GRAPHE orienté symétrique (appelé Grille) :

- ses sommets sont les pixels de l'image, ou d'une partie (par exemple rectangulaire de taille n par m) de l'image;
- les arcs sont donnés par la relation de voisinage que permet de définir la grille d'échantillonnage, 8- ou 4-voisinage pour une trame carrée, 6-voisinage pour une trame hexagonale.

Remarque : Martelli choisit d'associer les sommets aux "interpixels", mais cela revient sensiblement au même (Graphe dual de la grille des 4-voisins).

A tout CHEMIN ELEMENTAIRE de ce graphe on peut associer un contour possible (i.e. une liste de pixels dont chaque élément est voisin de son successeur). On cherche donc un chemin élémentaire MAXIMUM au sens où il suit AU MIEUX la limite fond/objet.

Toute la difficulté consiste à remplacer ce critère non mesurable, par un autre aisément calculable avec lequel il coïncide le plus souvent.

On peut admettre que la valeur du contraste en un point mesure la vraisemblance que le contour passe par ce point. Nous avons ainsi un critère local. Pour évaluer le contraste on peut utiliser l'une des techniques classiques /ROSENFELD et KAK 82/. Par exemple, le module du gradient fournit une valuation positive sur les sommets du graphe. On peut adapter les techniques classiques pour établir une valuation sur les arcs de la grille; le module de la projection du gradient perpendiculairement au segment qui joint deux pixels voisins, peut être évalué à l'aide des expressions suivantes :

$$v((i,j),(i,j+1)) := \left| \begin{array}{l} F(i-1,j)+F(i-1,j+1)-F(i+1,j)-F(i+1,j+1) \\ F(i-1,j)+F(i+1,j+1) \end{array} \right|$$

$$v((i,j),(i+1,j+1)) := \left| \begin{array}{l} F(i-1,j+1)+F(i,j+2)-F(i+1,j-1)-F(i+2,j) \\ F(i-1,j+1)+F(i+1,j+1) \end{array} \right|$$

où $F(x,y)$ est la luminosité au pixel de coordonnées (x,y) .

L'expression des autres valeurs associées aux arcs issus de (i,j) s'obtiennent par symétrie.

Le problème 1 se modélise alors comme suit:

PROBLEME 2.

Trouver le chemin élémentaire ch reliant 2 sommets donnés, tel que $f(ch)$ soit max

où f est une fonction des valuations des arcs de ch .

III.2. Fonctions de coût.

On peut choisir pour $f(ch)$, par exemple,

a) la somme des valuations des arcs du chemin /MONTANARI 71/.

b) une expression du type

- somme(k-valuations d'arcs)

Cette 2^e idée, utilisée très tôt, a l'avantage énorme de permettre de poser le problème en termes de plus court chemin et permet de disposer d'algorithmes performants, mais le chemin optimal ainsi obtenu dépend de k qu'on ne peut pas fixer, indépendamment de l'image. Nous suggérons donc que k puisse être fixé et modifié interactivement.

c) la moyenne des valuations des arcs du chemin.

De plus, de nombreux auteurs introduisent dans leur modélisation (a ou b) un terme permettant de contrôler la courbure du chemin. Ceci a pour effet de **supprimer certains cycles de la grille** (les cycles courts).

Enfin, certains auteurs font intervenir la longueur du chemin dans un terme soustrait de l'expression utilisée dans a.

Il semble bien qu'il n'existe pas une fonction unique qui modélise parfaitement le problème 1. La modélisation a) favorise les longs chemins; la modélisation b) les plus courts. Un autre critère de choix entre ces modélisations est l'existence d'une méthode efficace pour les résoudre.

III.3. Algorithmes.

Modélisation a.

Le problème de recherche d'un chemin élémentaire maximum dans un graphe quelconque, est connu pour être NP-difficile, même pour une valuation unitaire des arcs /GAREY et JOHNSON 79, p 213/. S'il le reste dans le cas particulier d'un graphe de type grille, la seule méthode connue pour le résoudre complètement, est une recherche exhaustive, ce qui est tout à fait impraticable dans le cas qui nous occupe.

Cependant f s'exprime ici sous une forme qui permet d'utiliser la programmation dynamique /MONTANARI 71/ (somme K de termes ne dépendant chacun que d'un arc), à condition d'avoir un graphe SANS CIRCUIT, ce qui n'est pas le cas des grilles. **Aussi Montanari construit-il implicitement à partir de la grille un graphe à K couches**; c'est dans ce graphe que la recherche de chemin maximum est effectuée :

Chaque couche a autant de sommets qu'il y a de pixels dans la portion de l'image (n,m) . Chaque sommet d'une couche est relié à ses 8 voisins dans la couche suivante.

Le nombre de couches K est le **nombre d'arcs imposé à la solution**. De plus des cycles sont autorisés dans la solution si ils ne présentent pas une courbure trop forte. La modélisation sous-jacente est une relaxation du problème 2 :

chercher le chemin ch de K arcs non nécessairement élémentaire, tel que $f(ch)$ soit maximum

Remarquons que la programmation dynamique est ici un moyen non-exponentiel, mais rapidement impraticable lorsque K augmente, pour résoudre ce problème restreint. Le nombre d'opérations est proportionnel à $K.n.m$. Ceci ne contredit pas la conjecture que nous posons plus bas.

De plus /MARTELLI 76/ remarque que dans le graphe en K couches, la programmation dynamique revient à faire un parcours en largeur et suggère plutôt un parcours de type A^* (voir ci-dessous).



**DETOURAGE INTERACTIF DE PHOTOGRAPHIES DE PRESSE
COMME RECHERCHE DE CHEMIN OPTIMAL DANS UN GRAPHE.**

Marc Bertier et Michel Habib.

INTERACTIVE CUT-OUT OF PRESS PHOTOGRAPHS AS OPTIMAL PATH SEARCH.

Une autre heuristique de type programmation dynamique suggérée par l'expression de f sous forme de somme, consiste à supprimer des arcs de la grille pour faire apparaître des couches :

- couches planes, de la source vers le but BERTIER et HABIB 83 p33/;
- couches concentriques à partir de la source;
- couches concentriques partant des 2 points à relier, d'après les méthodes qu'étudie /LAWLER 83/.

Modélisation b.

On passe à un problème de minimisation sur $-f$, et on est donc ramené, si k majore la mesure du contraste sur tous les arcs, à un problème de plus court chemin dans un graphe à valuations positives, problème pour lequel il y a abondance d'algorithmes et de références. Pour une synthèse récente, voir /GALLO et PALLOTINO 83/.

Cependant de nombreux auteurs ont suivi Martelli qui préconise l'utilisation d'une heuristique, appelée A^* . Celle-ci est utilisée en intelligence artificielle pour explorer de grands graphes de situations, où l'on calcule au fur et à mesure du parcours du graphe, les situations qui peuvent être atteintes à partir de la situation courante. Sur A^* , une littérature abondante existe: /BAGCHI et MAHANTI 83//GELPERIN 77//HART, NILSSON et RAPHAEL 68//HUYN, DECHTER et PEARL 80//MARTELLI 77//POHL 70//NILSSON 71/.

Cette méthode nous paraît peu adaptée ici car la grille est explicitement connue au départ. Une technique de plus court chemin dans un grand graphe telle que celle proposée par /LAWLER 83/ est a priori plus adéquate.

De plus A^* exploite essentiellement la donnée d'une fonction d'évaluation h minorant la valeur d'un chemin d'un point au pixel but (lorsque h est identiquement nulle, A^* n'est autre que l'algorithme célèbre de /DIJKSTRA 59/); or dans le cas de l'extraction de contour aucune fonction h ne s'impose (les différents auteurs semblent utiliser h identiquement nulle); c'est pourquoi nous avons directement implémenté l'algorithme de Dijkstra. D'ailleurs le temps de calcul obtenu étant raisonnable, il ne paraît pas indispensable d'élaguer l'arbre de recherche par d'autres idées heuristiques telles que celles suggérées par Martelli.

Modélisation c.

Cette modélisation pose le problème du calcul de la longueur d'un chemin. Pour le 4- ou le 6-voisinage, elle est équivalente au nombre d'arcs; pour le 8-voisinage, un coefficient (racine de 2) doit apparaître entre les arcs obliques et les autres.

La méthode de Montanari (programmation dynamique) ne peut être appliquée directement pour cette expression de f .

Conjecture.

Les modélisations a et b posent un problème dont nous conjecturons qu'il est NP-difficile :

Recherche d'un chemin élémentaire max sur un graphe de type grille.

En relation avec ce problème, nous avons démontré les résultats partiels suivants:

- la complexité du problème est la même pour les 4-grilles et pour les 6-grilles;
- la complexité du problème à extrémités libres et du problème à extrémités fixées est la même.

IV. PRESENTATION D'UN LOGICIEL INTERACTIF DE DETOURAGE.

IV.1. Utilisation du programme.

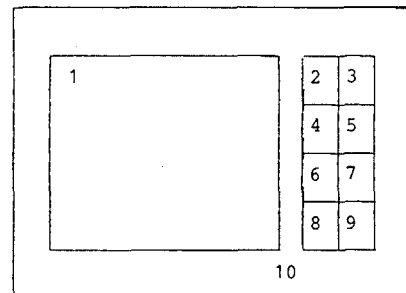
Configuration.

Le logiciel d'essai que nous avons réalisé est écrit en Pascal et il utilise la configuration décrite ci-dessous. Cette configuration n'est pas exclusivement dédiée à l'application décrite ici, et donc, sur plusieurs points, elle n'est pas adaptée au mieux.

unité centrale : DEC PDP11-40
 mémoire centrale : 128 K-octets
 disques : 80 + 2.5 + 2.5 M-octets
 mémoire d'image : LEXIDATA Solidview
 512x640x12x2
 table à numériser : BENSON format A0
 un terminal alphanumérique standard
 le système d'exploitation RSX11M version 3.2

Mode d'interaction.

Après avoir lancé l'exécution du programme depuis le terminal, l'opérateur utilise presque exclusivement la table à numériser sur laquelle un calque définit une zone représentant l'écran, et une zone de commandes comme cela apparaît sur la figure ci-dessous :



- | | |
|--|--|
| 1- déplacement du curseur | 6- annulation du dernier tronçon |
| 2- aide à l'opérateur | 7- changement de méthode et/ou de paramètres |
| 3- zoom avant | 8- bouclage sur le 1° 5-point |
| 4- annulation de tous les contours déjà tracés | 9- fin |
| 5- zoom arrière | 10- validation du curseur |

A l'aide du réticule de la table à numériser, l'opérateur demande à sa guise l'enchaînement des commandes. A chaque fois qu'il sélectionne un point dans l'image, le programme calcule le contour depuis ce pixel jusqu'au dernier point précédemment introduit. La méthode d'extraction de contour mise en oeuvre, est celle qui est sélectionnée à ce moment là.

Dans le seul cas de la commande de changement de méthode (n°7), l'interaction se poursuit sur le terminal car l'opérateur doit choisir la nouvelle méthode à mettre en oeuvre et éventuellement fixer ou modifier la valeur des paramètres associés à la méthode.

Sélection d'un pixel par l'opérateur.

Nous ne disposons sur notre mémoire d'image, ni d'une "souris", ni d'une boule roulante permettant



**DETOURAGE INTERACTIF DE PHOTOGRAPHIES DE PRESSE
COMME RECHERCHE DE CHEMIN OPTIMAL DANS UN GRAPHE.**

Marc Bertier et Michel Habib.

INTERACTIVE CUT-OUT OF PRESS PHOTOGRAPHS AS OPTIMAL PATH SEARCH.

de déplacer un curseur sur l'écran, à tout instant et indépendamment de l'exécution du programme. Nous avons donc simulé ceci par logiciel en introduisant les commandes 1 et 10 qui ont respectivement les effets suivants :

- déplacer un curseur sur l'écran de visualisation de la mémoire d'image;
- envoyer au programme les coordonnées courantes de ce curseur.

Développé au départ comme un pis aller, ce mode d'interaction s'est révélé en fin de compte agréable. Combiné à l'utilisation du zoom cablé de visualisation de la mémoire d'image, il permet aisément à l'opérateur de désigner un pixel précis dans l'image.

De plus il est très pratique de dissocier le déplacement du curseur de l'écran du déplacement du réticule de la table à numériser. Ceci permet par exemple, de faire l'extraction de contour par plusieurs méthodes successivement sans avoir à désigner à chaque fois l'extrémité à atteindre, puisque la sélection des commandes 6 (annulation du dernier tronçon) et 7 (changement de méthode), ne modifient pas la position du curseur de l'écran.

IV.2. Méthodes proposées.

Il est indispensable de proposer à l'opérateur au moins deux modes de travail :

- un mode -dit automatique- dans lequel le programme analyse l'image dans un voisinage du couple de points à relier, en appliquant l'une des heuristiques d'extraction de contour;

- un mode -dit "découpe aux ciseaux"- c'est à dire dans lequel le programme applique la méthode triviale déjà présentée en I., et interpole les points désignés par l'opérateur, par des segments de droite (un autre type d'interpolant peut éventuellement être utile). Il est impératif de laisser ainsi à l'opérateur la possibilité de résoudre lui-même les cas difficiles pour lesquels les autres méthodes fournissent un résultat inacceptable.

Nous avons testé successivement deux méthodes pour le mode automatique :

- nous avons appelé la 1^o, méthode d'extraction de contour par **déformations successives**. Elle consiste à choisir comme solution admissible du problème, le segment qui rejoint les deux points désignés par l'opérateur. On applique ensuite itérativement, des opérations élémentaires de déformation qui conservent la connexité, jusqu'à aboutir à une solution ("localement optimale") pour laquelle aucune déformation élémentaire n'améliore le critère associé (somme des modules du gradient aux points de la solution). Cette méthode est décrite de façon détaillée dans /BERTIER 83/.

- la seconde méthode que nous avons programmée est l'**algorithme de Dijkstra** pour la recherche du plus court chemin dans un graphe à valuations positives. La constante qui apparaît dans le calcul des valuations à partir du gradient est fixée par l'opérateur à qui elle permet de contrôler l'importance relative du nombre d'arcs du chemin optimal et du cumul du contraste le long de celui-ci.

IV.3. Implémentation de l'algorithme de DIJKSTRA (+ court chemin dans un graphe à valuations positives)

Algorithme.

A tout sommet on associe un état : **non-atteint, ouvert ou fermé** (1)

En tout sommet atteint N, est défini **g(N)**, le **coût du + court chemin déjà trouvé depuis la source**. Pour N fermé, g(N) est le coût optimal

En tout sommet atteint, **prec(N)** désigne le sommet précédent dans le + court chemin déjà trouvé

ouvrir(s); g(s) := 0; prec(s) := nil; FINI := faux;

Tant que pas FINI faire

N := x tel que x ouvert et pour tout y ouvert,
g(x) ≤ g(y); si égalité, priorité aux puits

fermer(N);

Si N est un puits alors FINI := vrai

sinon

Pour tous les successeurs N_i de N, faire

gchap := g(N) + c(N,N_i);

Si N_i non-atteint alors

ouvrir(N_i); g(N_i) := gchap;

prec(N_i) := N;

sinon

Si gchap < g(N_i) alors

(* N_i nécessairement ouvert *)

g(N_i) := gchap; prec(N_i) := N;

finsi;

finsi;

finpour;

finsi;

fintantque;

reconstituer le plus court chemin à l'aide des pointeurs arrières.

Analyse.

Une 1^o analyse, nous indique que c'est la recherche du nouveau sommet à fermer qui est la partie la plus coûteuse de l'algorithme, car passer en revue les successeurs de N, ne conduit jamais à reconsidérer un arc déjà pris en compte. Globalement, cette dernière opération sera en 8.n.m.log(n.m), puisque n.m est le nombre de sommets et que le nombre d'arcs lui est 8.n.m (pour le 8-voisinage).

Par contre, si à chaque fois on teste tous les sommets ouverts pour sélectionner celui à fermer (le plus petit), nous obtenons un algorithme de complexité de l'ordre de n².m².

Cependant, il existe une technique meilleure qui consiste à gérer l'ensemble des sommets ouverts en queue de priorité ("heap" : arbre binaire quasi équilibré) /JOHNSON 72/.

La queue de priorité permet de faire en O(log(k)) si elle contient k éléments, les opérations:

- insérer un nouvel élément,

- enlever l'élément ayant la valeur numérique

la plus faible.

En utilisant cette structure, lorsque le graphe est défini par ses listes d'adjacences, il faut

- pour chaque nouveau sommet fermé, O(log(n.m)) opérations,

- pour chaque voisin du sommet fermé,

O(log(n.m)) opérations,

d'où en fin de compte un algorithme en

O((9.n.m).log(n.m)); ce qui est meilleur que n².m².



**DETOURAGE INTERACTIF DE PHOTOGRAPHIES DE PRESSE
COMME RECHERCHE DE CHEMIN OPTIMAL DANS UN GRAPHE.**

Marc Bertier et Michel Habib.

INTERACTIVE CUT-OUT OF PRESS PHOTOGRAPHS AS OPTIMAL PATH SEARCH.

Remarque 1 : Dans la version classique de l'algorithme de Dijkstra, seuls 2 états, 'fermé' (dans l'ensemble S) et 'non fermé', sont définis. L'introduction d'un 3^e état permet d'exploiter une particularité des grilles : l'ensemble des sommets non fermés que l'on peut atteindre depuis l'un quelconque des sommets fermés, est un "petit" sous ensemble de S, puisqu'il correspond à sa frontière.

Remarque 2 : Pour conserver la complexité $O((n+m) \log n)$, sans disposer de la représentation explicite du graphe, la queue de priorité ne suffit pas, car les opérations à effectuer sur l'ensemble des sommets sont :

- Extraction du sommet ouvert d'évaluation minimale;
- Introduction d'un sommet d'évaluation donnée, dans l'ensemble des ouverts;
- Mise à jour de l'évaluation d'un sommet déjà ouvert;
- Test de l'état d'un sommet quelconque;
- Modification de l'état d'un sommet.

La 3^e opération ne peut être effectuée en $O(\log(n))$ avec une queue de priorité. Il est nécessaire de pourvoir l'ensemble des sommets ouverts d'une 2^e structure, par exemple celle d'arbre équilibré en hauteur (arbre AVL, /REINGOLD, NIVERGELT et DEO 77/) qui le permet.

Pour les 4^e et 5^e opérations, nous proposons, -de préférence à un arbre AVL contenant tous les sommets fermés (ou (exclusif) tous les sommets non atteints)-, d'utiliser une structure de table à 2 adresses puisque l'on en dispose d'une sous la forme de mémoire d'image. Il suffit donc de réserver dans celle-ci 2 bits à cet effet.

oOo

REMERCIEMENTS.

C'est à M. Philippe Coueignoux, maître de recherche à l'école des mines jusqu'en 82, que revient l'initiative du projet de détournement des photographies de presse. Qu'il trouve ici l'expression de nos remerciements.

Nous remercions également M. Didier Tallot qui nous a fournis les procédures de gestion de l'arbre AVL.

oOo

Bibliographie.

- Ashkar, Modestino: The Contour Extraction Problem with Biomedical Applications. CGIP 7, 1978.
- Bagchi et Mahanti: Search Algorithms Under Different kinds of Heuristics. J.A.C.M. vol 30-1, 1983, pp1-21.
- Ballard, Brown: Computer Vision. pp 131-143 Prentice-Hall, Englewood Cliffs, 1982.
- Ballard et Sklansky: A ladder-structured Decision Tree for Recognition Tumors in Chest Radiographs. IEEE Trans. Comp. pp503-513, 1976.
- Bertier: Détournement interactif de photographies d'intérêt général. Thèse de docteur-ingénieur, Ecole des Mines, Saint-Etienne 1983.
- Bertier et Habib : Contour extraction via path algorithms. Proc Int. Workshop on Graph Theoret Concepts Osnabruck 83, Trauner Verlag p29-39
- Dijkstra: A Note on Two Problems in Connexion with Graphs. Numerische Mathematik 1, 1959.
- Gallo et Pallottino: Shortest path methods, a unified approach. Netflow Mars 83 Pise Italie p28-31
- Garey et Johnson: Computers and Intractability, a Guide to the Theory of NP-completeness. Freeman, San Francisco, 1979.
- Gelperin: On the optimality of A*. Artificial Intelligence 8, 1977, pp 69-76.
- Hart, Nilsson et Raphael: A formal basis for the heuristic determination of minimum costs paths. IEEE trans syst, man and Cybern 4, 1968, pp100-107.
- Huyn, Dechter et Pearl: Probabilistic Analysis of the Complexity of A*. Artificial Intelligence 15, 1980, pp241-254.
- Johnson: On Shortest Paths and Sorting. Proc. Ann. Conf. ACM, Boston, 1972.
- Lawler, Luby et Parker: Finding Shortest Paths in Very Large Networks. Proc. Int. Workshop on Graph Theor. Concepts Osnabruck 83, Trauner Verlag
- Lester, Williams, Weintraub et Brenner: Two Graph Searching Techniques for Boundary Finding in White Blood Cell Images. Computers in Biology and Medecine, Vol 8, Pergamon, 1978.
- Martelli: Edge Detection using Heuristic Search Methods. CGIP 1 pp169-182, 1972.
- Martelli: An Application of Heuristic Search Methods to Edge and Contour Detection. Comm. ACM, vol 19 n°2 pp73-83, February 1976.
- Martelli: On the complexity of admissible search algorithms. Artificial Intelligence 8, 1977, pp1-13.
- Montanari: On optimal detection of curves in noisy images. Comm. ACM vol 14 n°5 p335-345, 1971.
- Pohl: Heuristic search viewed as path finding in a graph. Artificial Intelligence 1, 1970, pp219-236.
- Nilsson: Problem-solving Methods in Artificial Intelligence. Mc-Graw-Hill, 1971.
- Ramer: Extraction of Line Structures from Photographs of Curved Objects. CGIP 4 pp81-103, 1975.
- Reingold, Nievergelt et Deo: Combinatorial Algorithm, theory and practice. Prentice-Hall 1977.
- Rosenfeld et Kak: Digital image processing. Academic Press, 1982.

oOo