Aides informatiques à la réalisation de dessins animés

Jean-Claude Moissinac

ECOLE NATIONALE SUPERIEURE
DES MINES DE SAINT-ETIENNE

N° d'ordre: 44 I.I.

THESE

présentée par

Jean-Claude MOISSINAC

pour obtenir

LE TITRE DE DOCTEUR-INGENIEUR
EN INFORMATIQUE

AIDES INFORMATIQUES A LA REALISATION DE DESSINS ANIMES

Soutenue à Saint-Etienne, le 21 Décembre 1984,
devant la Commission d'Examen

MM.	R. MAHL	Président
	P. BAUDELAIRE	Examinateurs
	M. GANGNET	
	M. HABIB	
	J. ROUXEL	

ECOLE NATIONALE SUPERIEURE
DES MINES DE SAINT-ETIENNE

N° d'ordre: 44 I.I.

THESE

présentée par

Jean-Claude MOISSINAC

pour obtenir

LE TITRE DE DOCTEUR-INGENIEUR
EN INFORMATIQUE

AIDES INFORMATIQUES A LA REALISATION DE DESSINS ANIMES

Soutenue à Saint-Etienne, le 21 Décembre 1984,
devant la Commission d'Examen

MM.	R. MAHL	Président
	P. BAUDELAIRE	Examinateurs
	M. GANGNET	
	M. HABIB	
	J. ROUXEL	

ECOLE NATIONALE SUPERIEURE DES MINES DE SAINT ETIENNE

Directeur : M. M. MERMET
Directeur des Etudes et de la formation : M. J. CHEVALIER
Secrétaire Général : Melle M. CLERGUE

PROFESSEURS DE 1ère CATEGORIE

MM.	COINDE	Alexandre	Gestion
	GOUX	Claude	Métallurgie
	LEVY	Jacques	Métallurgie
	LOWYS	Jean-Pierre	Physique
	MATHON	Albert	Gestion
	RIEU	Jean	Mécanique - Résistance des Matériaux
	SOUSTELLE	Michel	Chimie
	FORMERY	Philippe	Mathématiques Appliquées

PROFESSEURS DE 2ème CATEGORIE

MM.	HABIB	Michel	Informatique
	PERRIN	Michel	Géologie
	VERCHERY	Georges	Matériaux
	TOUCHARD	Bernard	Physique Industrielle

DIRECTEUR DE RECHERCHE

M.	LESBATS	Pierre	Métallurgie

MAITRES DE RECHERCHE

MM.	BISCONDI	Michel	Métallurgie
	DAVOINE	Philippe	Géologie
Mle	FOURDEUX	Angeline	Métallurgie
MM.	GUILHOT	Bernard	Chimie
	KOBYLANSKI	André	Métallurgie
	LALAUZE	René	Chimie
	LANCELOT	Francis	Chimie
	LE COZE	Jean	Métallurgie
	PLA	Jean Marie	Mathématiques
	THEVENOT	François	Chimie
	TRAN MINH	Canh	Chimie

PERSONNALITES HABILITEES A DIRIGER LES TRAVAUX DE RECHERCHE

MM.	COURNIL	Michel	Chimie
	DRIVER	Julian	Métallurgie
	MAGNIN	Thierry	Métallurgie
	THOMAS	Gérard	Chimie

PROFESSEUR A L'U.E.R. DE SCIENCES DE SAINT ETIENNE

M.	VERGNAUD	Jean Marie	Chimie des Matériaux

REMERCIEMENTS

Je tiens à remercier Monsieur Robert MAHL qui m'a fait l'honneur d'accepter de présider le jury de cette thèse.

Les conseils de Monsieur Patrick BAUDELAIRE m'ont aidé à orienter mes travaux. Je lui suis très reconnaissant d'avoir accepté de juger ce travail.

Je remercie vivement Monsieur Jacques ROUXEL pour les discussions fructueuses que nous avons eues pendant toute la durée de mes travaux et pour avoir accepté de participer à ce jury.

Je tiens à remercier Monsieur Michel HABIB pour l'intérêt qu'il a manifesté à l'égard de ces recherches. Ses conseils et ses suggestions ont toujours été extrêmement précieux.

Monsieur Michel GANGNET m'a initié à la recherche et m'a constamment encouragé dans la réalisation de ce travail. Je lui suis très redevable pour les conseils qu'il a su me prodiguer ainsi que pour ses nombreuses et pertinentes suggestions. Qu'il trouve ici l'expression de mes chaleureux remerciements.

J'exprime aussi mon amitié à tout le personnel du département informatique de l'Ecole des Mines, avec qui de fréquents échanges de points de vue ont été extrêmement profitables. Je remercie tout particulièrement les membres de l'équipe "Communications Visuelles", Sabine COQUILLART, Djamchid GHAZANFARPOUR et Dominique MICHELUCCI pour leur collaboration amicale.

Je profite de l'occasion pour exprimer ma reconnaissance envers les personnes du centre de calcul qui participent au bon fonctionnement de la configuration matérielle.

Je remercie enfin Mesdames AVONDO et VINCENT et Messieurs BROSSARD, DARLES et LOUBET qui ont participé avec beaucoup de soins et de gentillesse à la réalisation de cet ouvrage. La présentation de ce rapport doit beaucoup au logiciel GROFF (Girardot's Ridiculous Output File Formatter) développé par Jean-Jacques GIRARDOT.

PLAN

INTRODUCTION 1

Chapitre 1 :
REALISATION D'UN DESSIN ANIME. 1

1. Processus de production et principaux objets manipulés.
   1.1. Le storyboard.
   1.2. Les dessins de base.
   1.3. La bande son.
   1.4. Le lay-out.
   1.5. Les dessins et les images.
   1.6. La feuille de prise de vue.
   1.7. Les documents administratifs.

2. Analyse globale. 21

3. Etapes possibles pour l'informatisation. 22

4. Informatiser. 27

Chapitre 2 :
SAISIE DES DESSINS. 1

1. Quel problèmes cherchons-nous à résoudre? 1

2. Représentations numériques des images.. 2
   2.1. Représentation par table de points.
   2.2. Repr'senation géométrique.

3. Saisie manuelle. 8

4. Saisie automatique. 14

5. Obtenir un vectogramme à partir d'une saisie optique. . .17
   5.1. Méthodes de codage.
   5.2. Le code chaîné par balayage: principe et exemples.
   5.3. L'implémentation.
   5.4. Vectorisation.
   5.5. Remarques.
   5.6. Conclusion.

6. Filtrage d'un vectogramme. 41
7. Problèmes latents. 48
8. Compléments pour la comparaison des méthodes de saisie.

PLAN

Chapitre 3 :
DU DESSIN AU TOURNAGE 1

1. Introduction. 1

2. Test d'animation. 4
   2.1. Le film de test.
   2.2. L'enregistrement vidéographique.
   2.3. Les images numériques.

3. Structuration des dessins. 14
   3.1. Introduction.
   3.2. Description d'une carte planaire.
   3.3. Eléments quantitatifs.

4. Vérification des dessins. 25
   4.1. Les contrôles nécessaires.
   4.2. Fermetures automatiques.
   4.3. Marquage ou élimination des petits bords.
   4.4. Fermetures intéractives.
   4.5. Résultats des tests.

5. Affichage d'une carte planaire sur surface à balayage.
   5.1. Introduction.
   5.2. L'algorithme et ses paramètres.

6. Gouachage. 49
   6.1. Résultats à atteidre-Méthodes utilisées.
   6.2. Gouachage d'un vectogramme.

7. Tournage. 55

Chapitre 4 :
COMPARAISON ENTRE LES IMAGES D'UNE SEQUENCES 1

1. Introduction. 2

2. Situation du problème. 3
   2.1. Hypothèses de représentation des données.
   2.2. Difficultés.

3. Structuration. 6
   3.1. Cartes planaires.
   3.2. Principe des méthodes de comparaison de cartes.

4. Implémentations. 11
   4.1. Construction élémentaire.
   4.2. Implémentation structurée.

5. Autre utilisation de la comparaison.
6. Conclusion.

Chapitre 5 :
CONCLUSION 1

ANNEXE. . . . . .

CONFIGURATION MATERIELLE.

BIBLIOGRAPHIE.

INTRODUCTION

Nous exposons ici les motivations de cette étude et nous énonçons rapidemment les problèmes que cette thèse contribue à résoudre.

La circulation de l'information devient une préoccupation prioritaire des sociétés modernes. C'est pourquoi, la production audiovisuelle destinée aux télévisions et aux entreprises doit considérablement se développer dans les prochaines années. Une concurrence sévère naît dès maintenant de cette situation. Elle oppose les entreprises artisanales, qui jusqu'à ce jour occupaient le marché, et les sociétés industrielles qui se créent pour répondre à la demande prévisible. La vie et le savoir-faire d'un secteur d'activité sont menacé. Les petites entreprises, qui constituent le potentiel de production de dessin animé de notre pays, doivent s'adapter à une nouvelle situation internationnale.

En effet, la Grande-Bretagne, le Japon et les USA ont pris une place importante dans le domaine de la série télévisée de dessins animés. Ils ont conquis le marché en proposant leurs séries à la fois plus vite et moins cher. Contrairement à une idée répandue, ils n'utilisent pas encore systématiquement les techniques informatiques. Ils ont porté l'économie sur les frais de personnels et la simplicité de réalisation en mettant en oeuvre des méthodes classiques d'industrialisation d'une fabrication: parcellisation du travail, normes de qualité souvent respectées à leur niveau minimum...

Mais la force commerciale et l'expérience acquises leur permettent désormais de proposer des productions de qualité croissante et d'intégrer progressivement des techniques modernes.

dessin animé. Cette base de travail a été établie en collaboration étroite avec des professionnels du domaine. Nous l'avons complétée par une esquisse des apports actuels ou potentiels de l'informatique. Nous terminons en abordant les problèmes techniques que ce travail contribue à résoudre. Les principales références qui ont servi a établir ce chapitre sont: [BRI84], [MAR70], [BCT84].

Cette description ne concerne que le dessin animé classique : le cartoon. Nous n'avons pas explicité ici les mille et une façons de faire du cinéma d'animation. Citons pour mémoire l'animation de papier découpé, de marionnettes, la pixilation.

Face à la demande fortement croissante de dessins animés, destinés en particulier aux programmes de télévision, il devient nécessaire d'améliorer la productivité dans ce domaine. Outre des méthodes industrielles classiques, comme la parcellisation des tâches, des apports de nature différente peuvent être espérés de l'informatique: gestion de l'information (flux, mises à jours, cohérences entre documents), automatisation de tâches répétitives, outils interactifs de définition numérique d'images et de mouvements.

Des recherches tendent à faire de l'ordinateur un outil nouveau de création d'images animées. Notamment, des études sont faites pour définir des objets avec des caractéristiques: élasticité, poids,... et les manipuler en temps réel sur un écran d'ordinateur (formes du langage Logo, [LUC84]).

D'autres études ont plutôt porté sur l'imitation du processus traditionnel de production de dessin animé. Des travaux ont été entrepris dans ce sens depuis plus de dix ans. Les plus grands studios: Hanna-Barbera, Walt Disney ont investi dans ces recherches. Aujourd'hui les systèmes informatiques d'aide à la production de dessins animés commencent seulement à sortir des laboratoires pour être utilisés en production [WAL81] [DIS84].

La fabrication des dessins animés est une activité créatrice nécessitant un grand travail manuel. Les trois points clés qui déterminent la concurrence entre les studios sont le coût, les délais et les volumes de production. Il sera intéressant de gagner sur chacun de ces points.

Le chapitre 1 présente le processus classique de production d'un dessin animé, ainsi que les apports possibles de l'informatique pour chaque phase du processus. Les chapitres 2 et 3 rendent compte de solutions que nous proposons à l'informatisation de certaines étapes et des développements techniques approfondis auxquels elles ont donné lieu. Une méthode pour exploiter la cohérence entre les dessins successifs d'un dessin animé est exposée au chapitre 4.

CHAPITRE 1

REALISATION D'UN DESSIN ANIME

Ce chapitre présente la description du processus de fabrication d'un dessin animé qui a servi de guide à mon travail. Il définit le vocabulaire utilisé par la suite, en respectant autant que possible la terminologie usuelle du

dessin animé. Cette base de travail a été établie en collaboration étroite avec des professionnels du domaine. Nous l'avons complétée par une esquisse des apports actuels ou potentiels de l'informatique. Nous terminons en abordant les problèmes techniques que ce travail contribue à résoudre. Les principales références qui ont servi a établir ce chapitre sont: [BRI84], [MAR74], [BCT84].

Cette description ne concerne que le dessin animé classique : le cartoon. Nous n'avons pas explicité ici les mille et une façons de faire du cinéma d'animation. Citons pour mémoire l'animation de papier découpé, de marionnettes, la pixilation.

Face à la demande fortement croissante de dessins animés, destinés en particulier aux programmes de télévision, il devient nécessaire d'améliorer la productivité dans ce domaine. Outre des méthodes industrielles classiques, comme la parcellisation des tâches, des apports de nature différente peuvent être espérés de l'informatique: gestion de l'information (flux, mises à jours, cohérences entre documents), automatisation de tâches répétitives, outils interactifs de définition numérique d'images et de mouvements.

Des recherches tendent à faire de l'ordinateur un outil nouveau de création d'images animées. Notamment, des études sont faites pour définir des objets avec des caractéristiques: élasticité, poids,... et les manipuler en temps réel sur un écran d'ordinateur (formes du langage Logo, [LUC84]).

D'autres études ont plutôt porté sur l'imitation du processus traditionnel de production de dessin animé. Des travaux ont été entrepris dans ce sens depuis plus de dix ans. Les plus grands studios: Hanna-Barbera, Walt Disney ont investi dans ces recherches. Aujourd'hui les systèmes informatiques d'aide à la production de dessins animés commencent seulement à sortir des laboratoires pour être utilisés en production [WAL81] [DIS84].

La fabrication des dessins animés est une activité créatrice nécessitant un grand travail manuel. Les trois points clés qui déterminent la concurrence entre les studios sont le coût, les délais et les volumes de production. Il sera intéressant de gagner sur chacun de ces points.

De nombreuses informations doivent circuler dans les équipes de réalisation, une analyse soignée de ces flux d'information devra être faite pour arriver à une optimisation informatique de la communication interne au studio.

Les coûts et les délais du processus traditionnel de fabrication sont bien maîtrisés. Les personnels compétents sont répertoriés; les spécialités nécessaires sont connues. L'informatisation doit être faite avec précaution afin de contrôler les surcoûts de nouvelles méthodes de travail: machines coûteuses ou fragiles, formation du personnel, recrutement de personnel spécialisé.

1.1. PROCESSUS DE PRODUCTION ET PRINCIPAUX OBJETS MANIPULES

La figure 1.2 présente un synoptique du processus de production d'un dessin animé. On peut lui reprocher un caractère exagérément linéaire: en pratique, de fréquents retours en arrière permettent d'affiner la conception.

Il faut noter qu'il s'agit d'une présentation des tâches; certaines d'entre elles peuvent être assurées par plusieurs personnes ou plusieurs équipes, d'autres seront regroupées sous la responsabilité d'une seule personne.

CONCEVOIR - DECRIRE L'HISTOIRE

Synopsis | Graphisme de base
Scénario | Commentaire
Storyboard | Dialogue

METTRE EN SCENE

Lay-out / Bande son
Modèles: couleurs, expressions

ANIMER

Dessins-clés
Dessins intermédiaires
Feuille de prise de vue
Film de test
Nettoyage (cleaning)

PRODUIRE LES DECORS

Décors

TRACER - GOUACHER - VERIFIER

Cellulos

FILMER

"Rushes"

MONTER - SONORISER

Le Film

Phases de fabrication d'un dessin animé

Figure 1.2.

Une vision simplifiée du processus est la suivante. Un sujet est travaillé en vue d'en faire un dessin animé. Dans cette phase de conception, le travail qui suivra est décrit et préparé de façon très approfondie. Ensuite, les dessins à assembler pour obtenir l'image finale sont réalisés: il s'agit des dessins composant l'action et des décors qui lui serviront de fond. Les dessins doivent être ensuite repris sur un support transparent, permettant leur assemblage lors de la prise de vue, et mis en couleur: traçage, gouachage. Pour terminer le film est tourné, des vérifications sont faites et le produit final naît de l'assemblage avec la bande son. Une description plus détaillée et plus exacte suit.

1.1.1. LE STORYBOARD

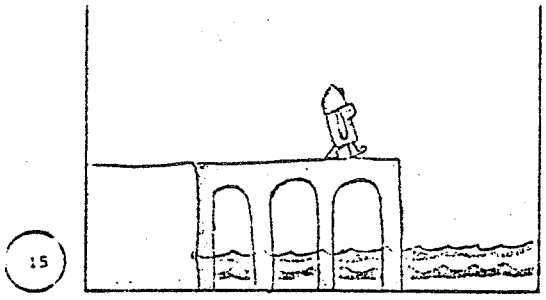
> On le trouve à scénarimage dans le dictionnaire. Il s'agit de la description préparatoire du film. Sa présentation ressemble à une bande dessinée commentée. Il indique les éléments nécessaires à une vue d'ensemble du film. Les commentaires portent sur la bande son, les mouvements de caméra, les effets spéciaux. Il sert de base de discussion pour la conception d'ensemble, puis, une fois accepté, devient le document de référence commun pour les intervenants.

Le storyboard prend des formes diverses, plus ou moins élaborées. Les dessins du storyboard sont souvent des croquis. Le storyboard peut être consulté par l'ensemble des intervenants. A chaque plan du film correspond au moins un dessin en présentant l'essentiel. Un équivalent informatique apporterait des facilités d'édition.

Le storyboard est réalisé par un dessinateur spécialisé qui le construit d'après les indications des concepteurs: réalisateur, scénariste, commanditaires.

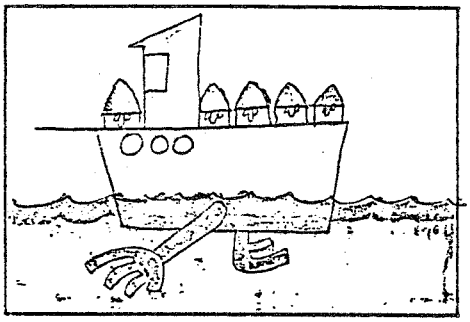
L'affichage de ces dessins à un rythme choisi pourrait constituer une maquette du film définitif. Ce rythme devrait pouvoir être fixé en rapport avec la bande son ou une maquette de celle-ci. La manipulation informatique de cette maquette offrirait des facilités évidentes: test de plusieurs versions,corrections. L'étude technique de ces possibilités reste à faire. Une version sur papier reste irremplaçable à ce jour: affichage dans le bureau, version personnelle annotée.

Figure 1.3.
Une portion de storyboard



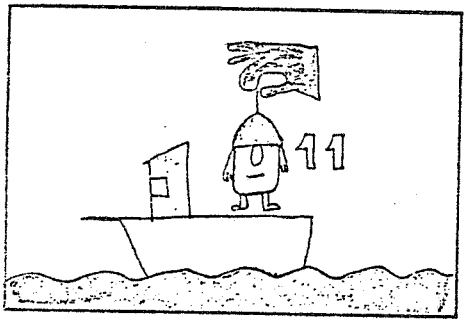
15

COM. C'est à lui aussi qu'on devra le fameux demi-pont, pour ceux qui ne veulent traverser que la moitié des rivières.



16

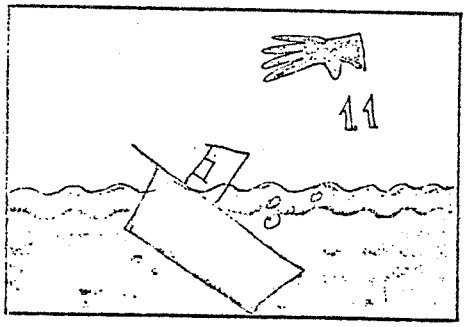
COM. En attendant, pour traverser, ils avaient un bateau.



17

ACT. La main amène des Matics un à un et les laisse tomber dans le bateau. Le nombre augmente au fur et à mesure.

COM. L'ennui, c'est qu'on ne pouvait y tenir qu'à dix ... Sinon



18

ACT. Au onzième, le bateau coule.

COM. Il aurait fallu compter.

Cependant, l'adaptation informatique de ce document n'est pas prioritaire: sa conception constitue l'essentiel de son prix; il subit peu de mises à jour; le volume de données correspondant est relativement faible; le support papier actuellement utilisé est simple et rapide d'emploi.

Pour mémoire, je cite ici deux autres descriptions du film jouant un rôle moins important dans la communication entre les intervenants:

- synopsis: description brève du film, il en énonce l'idée directrice, il constitue l'ébauche du scénario et du storyboard, une photocopie en est mise à la disposition de l'ensemble des intervenants;
- scénario: rédaction détaillée des diverses scènes dont le film sera composé.

1.1.2. LES DESSINS DE BASE

Il s'agit des dessins faits dès la conception pour orienter le style des développements ultérieurs. Ils constituent une partie des modèles et sont souvent désignés par ce terme.

1.1.3. LA BANDE SON

Elle est composée d'un ensemble de sons dont le rôle peut être différent: musique, dialogues, récit, bruitages.

Elle est nommée ainsi car elle est généralement stockée sur une bande magnétique. Suivant les cas, elle sera travaillée en pré ou post-synchronisation par rapport aux images; par exemple, on peut soigneusement étudier un enregistrement des dialogues pour synchroniser les images sur celui-ci ou enregistrer après coup les dialogues en faisant appel à des doubleurs professionnels.

Dans la suite, nous nous bornerons à supposer que des repères temporels peuvent être issus d'une bande son et associés sur le lay-out, la feuille de timing et la feuille de prise de vues (cf.parag.1.5.2) à des références à des évènements visuels.

Des recherches ambitieuses ont été entreprises pour le travail sur la bande son [LUC83],[GRA84]. Ces recherches tirent parti de la transcription numérique des sons pour fournir des outils puissants de création, de mixage et de synchronisation avec les images. Nous avons laissé celles-ci hors du champ de notre étude.

1.1.4. LE LAY-OUT

Il s'agit de l'ensemble des documents préalables au travail d'animation proprement dit. Il se compose de graphes techniques et de dessins sur papier qui constituent une description très approfondie du film. Chaque plan est décrit par sa durée, les acteurs qui y interviennent, les principaux événements, les principales lignes du décor, l'équipe qui en prendra la charge. Les événements sont représentés graphiquement avec plus de détails que dans le storyboard: des indications schématiques préfigurent le mouvement et les trajets des acteurs.

La responsabilité de ce travail incombe à un des concepteurs du film qui assurera le suivi de sa production: le réalisateur ou son premier assistant. Il est analogue au "découpage technique" pour les films en prise de vue réelle.

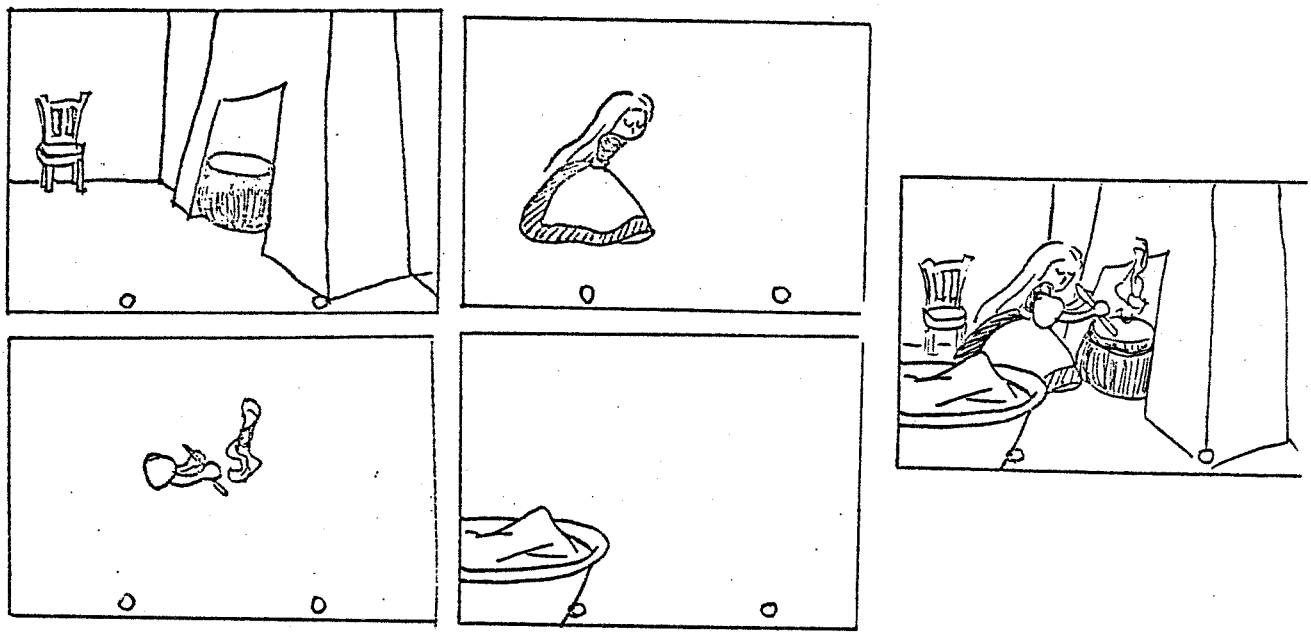
Lors de la réalisation de ce document les cadrages sont choisis, ainsi que la mise en scène de l'action.

Les principaux documents qui le composent sont : des dessins, une feuille de timing, des indications dessinées pour la mise en scène: placement des personnages par rapport au décor et au cadre, mouvement du cadre sur un décor.

En fonction de sa complexité, la décomposition éventuelle d'un acteur en plusieurs parties est décidée et notée sur le lay-out; l'animation de chaque partie sera préparée par des dessins différents sur des feuilles différentes (cf. 1.5.2.).

Le lay-out autorise la réalisation d'une bande son en donnant au responsable de celle-ci des repères approximatifs; en retour, celle-ci pourra servir à établir un minutage précis des plans et des mouvements à l'intérieur de chaque plan .

La feuille de timing est un élément important du lay-out. En fonction des données du storyboard, des dialogues, de la musique, du commentaire ,prévus ou existants, ce document fixe les durées exactes des actions élémentaires du récit, la synchronisation précise avec la bande son. Il indique également les mouvements de caméra



Exemple de décomposition

Figure 1.4.

et leur durée, les fondus enchaînés et autres effets spéciaux.

A partir de ces documents, le travail est réparti par équipe, chacune recevant une copie du lay-out des plans qu'elle réalise. Un équivalent informatique devrait donc être en permanence accessible; l'accès d'une équipe à la partie qui la concerne pourrait être optimisé.

Comme pour le storyboard, l'adaptation informatique de ces documents n'est pas prioritaire. De plus, on rencontrera couramment des problèmes du type suivant: l'animateur utilise le lay-out comme support à sa préparation des dessins clés (définis plus loin); il souhaite donc afficher simultanément le lay-out et les dessins clés qu'il réalise. L'existence du lay-out sur papier règle les problèmes de sa consultation en parallèle au travail sur d'autres objets manipulés numériquement: dessins, feuille de prise de vue.

Réduite à l'imitation du document traditionnel, l'informatisation de ce document n'apporterait guère que des facilités d'édition. La réalisation et la représentation de ce document sur ordinateur peut prendre une forme différente . Par exemple:

le réalisateur peut travailler interactivement une séquence sur un écran graphique d'ordinateur. Chaque acteur de la séquence est figuré par un tracé simplifié au trait. Le réalisateur peut jouer avec le cadre et avec les trajectoires des acteurs dans celui-ci. Régulièrement, il peut tester en temps réel l'état de son travail; il voit alors se produire à l'écran les mouvements qu'il a prévu au rythme qu'il a prévu. Suivant le résultat, soit il continue à modifier, soit il sauve son travail sur un support magnétique pour l'utiliser ou le modifier ultérieurement.

Cela revient un peu à une répétition théâtrale avec des acteurs en costume de ville.

L'enregistrement du résultat pourrait, comme pour le storyboard, constituer une maquette du produit fini; voire même, si la mise en couleur est effectuée, constituer un produit fini peu élaboré, où les seuls mouvements sont des déplacements et des changements de taille d'objets rigides.

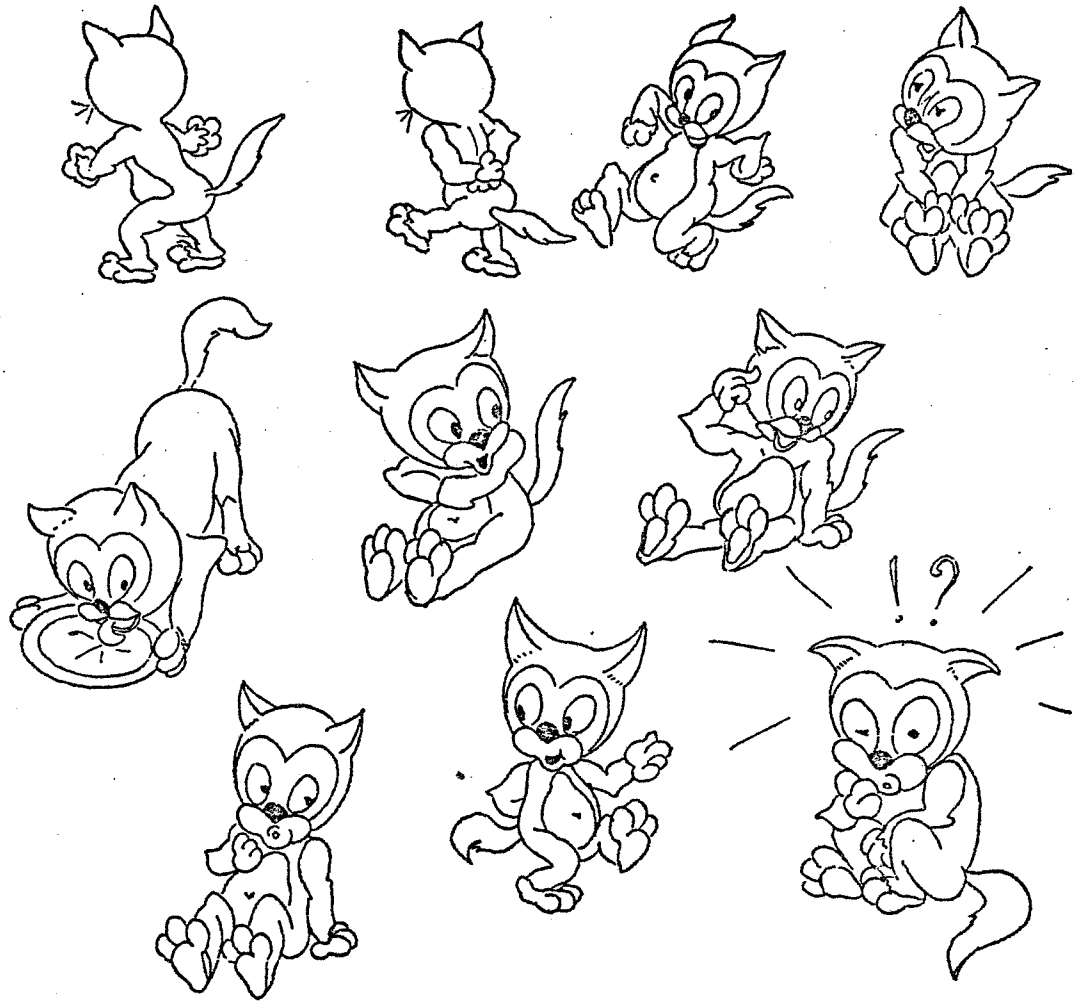
1.1.5. LES DESSINS ET LES IMAGES

Nous avons déjà parlé des dessins de base et, comme nous l'avons vu, le storyboard et le lay-out sont en partie composés de dessins. Les autres images manipulées peuvent être réparties en plusieurs classes: les modèles, les dessins clés, les dessins intermédiaires, les décors, les cellulos, les images filmées.

1.1.5.1. Les modèles

> Un dessinateur constitue une sorte de catalogue des attitudes et expressions caractéristiques des acteurs et autres éléments importants du film . Le modèle porte aussi mention des constantes obligatoires de chaque acteur: proportions, costume, expressions. Le coloriste attribue à chaque acteur les couleurs qui le caractériseront.

Ainsi, le modèle présentera sur une grande page de croquis les principales attitudes d'un acteur. Il s'agit souvent de dessins au trait en noir et blanc, afin de pouvoir être photocopiés et diffusés à l'ensemble des intervenants.



Une page de modèle d'attitudes

Figure 1.5.

Ces dessins servent tout au long du film de référence aux intervenants. Ainsi l'animateur s'inspirera d'une attitude de la page de modèle comme point de départ de son interprétation d'un mouvement.

L'adaptation informatique des modèles n'est pas prioritaire pour les raisons invoquées au sujet du storyboard. De plus, les dessinateurs les réalisent habituellement sur papier.

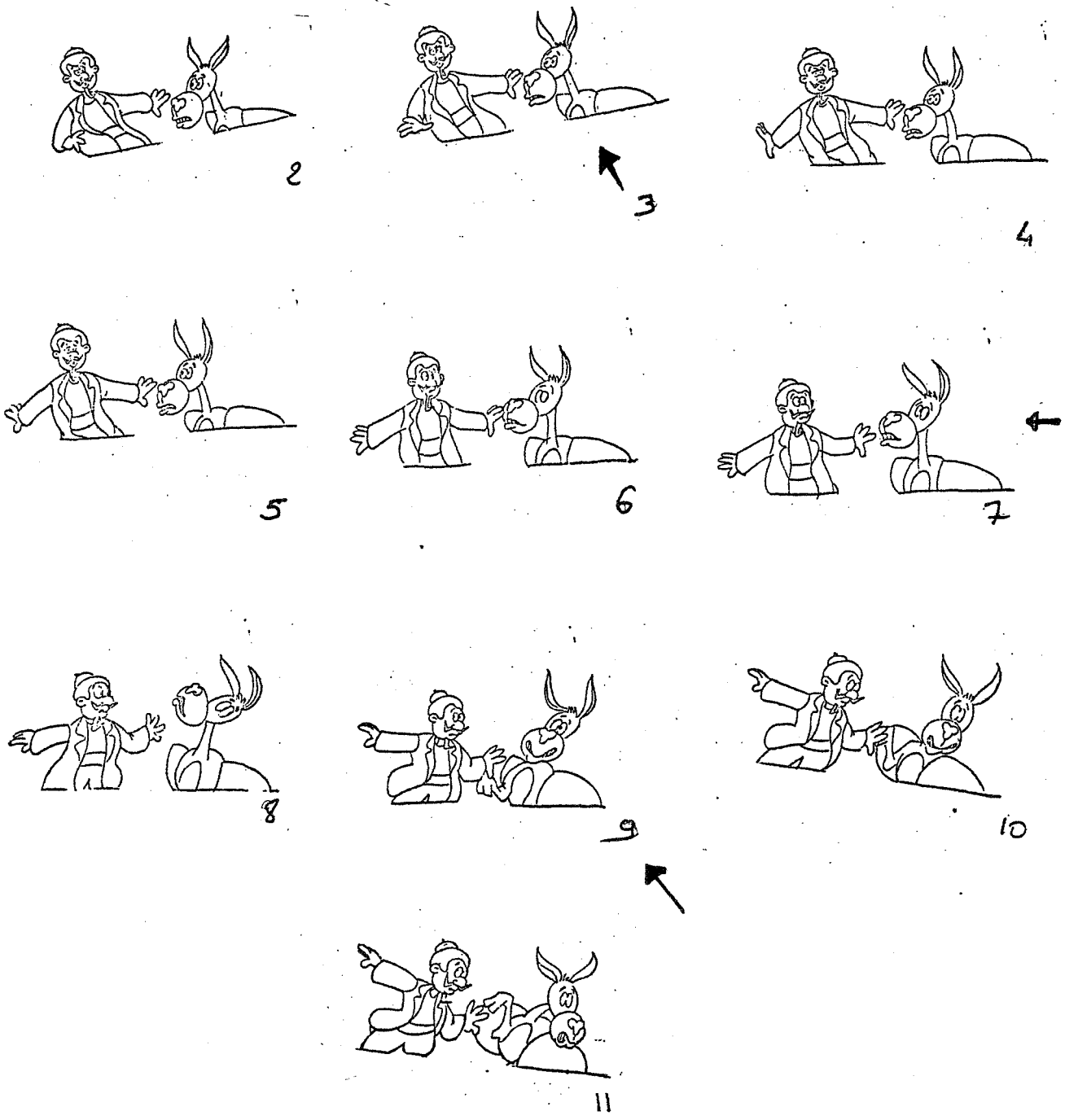
1.1.5.2. Les dessins clés

Ils constituent la première étape dans la construction du mouvement des acteurs par les animateurs. Un dessin clé représente au trait chaque temps fort de chaque mouvement (figure 1.6.). En général, cela correspond à environ un cinquième des dessins à produire (2 à 3 dessins par seconde du film et par acteur).

En principe, les animateurs les plus expérimentés effectuent ce travail. L'animateur peut choisir de briser un mouvement difficile, de le décomposer sur plusieurs dessins clés ou de décomposer un acteur en parties fixes et mobiles.

Figure 1.6.

Dessins clés et intermédiaires



les dessins clés sont marqués d'une flèche

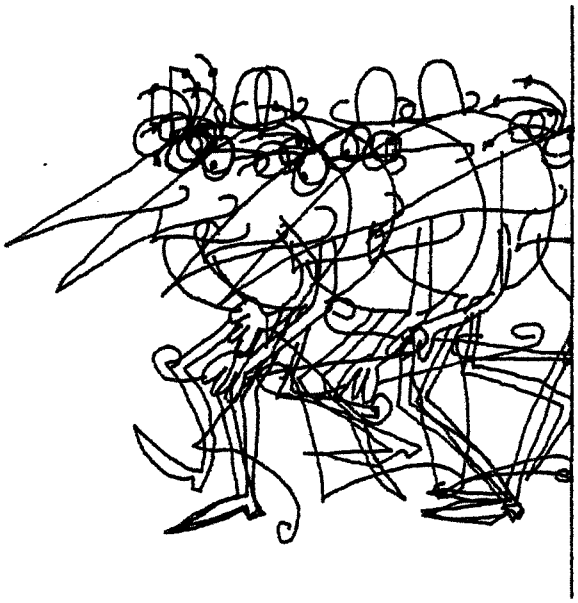
Ces dessins sont effectués au crayon sur papier mince perforé; les perforations servent à caler les dessins les uns par rapport aux autres. Elles permettent en particulier d'empiler de façon précise deux dessins d'un mouvement sur une table lumineuse afin d'en dessiner une étape intermédiaire en s'inspirant par transparence du début et de la fin du mouvement. Elles auront aussi un rôle primordial dans la composition de l'image lors de la prise de vue: calage des dessins entre eux et par rapport aux décors.

Chaque dessin clé porte une référence du plan auquel il appartient et un numéro qui correspond à peu près à son instant d'utilisation dans ce plan. Ces références sont utilisées au fur et à mesure pour actualiser la feuille de prise de vue.

1.1.5.3. Les dessins intermédiaires

> Les dessins intermédiaires sont l'ensemble des dessins qui vont s'intercaler entre les dessins clés pour compléter la décomposition du mouvement.

Les dessins clés servent de guide pour réaliser tous les dessins intermédiaires qui vont compléter le mouvement. Des assistants animateurs effectuent ce travail. Ils donnent au mouvement une partie de son expression. Il ne s'agit pas d'un simple intervallage mécanique, comme beaucoup d'informaticiens l'ont espéré (quelques références sur les recherches en matière de dessin animé: [BUR83], [REE81], [COU80]).



Les phases d'un mouvement superposées

Figure 1.7.

Chaque dessin intermédiaire est référencé comme chaque dessin clé: plan et instant d'apparition.

Après la feuille de prise de vue, et pour des raisons analogues, la prise en compte informatique des dessins constitue une priorité. Elle pose d'importants problèmes en raison du volume de données à stocker et à traiter. Le paragraphe 4.1 explicite ces problèmes; le chapitre 2 propose des solutions.

1.1.5.4. Les décors

Les décors vont principalement constituer le fond de l'image. Certaines scènes nécessitent des éléments de décor devant l'action, voire intercalés à celle-ci (fig. 1.4.). Un décor est éventuellement plus grand que les cadrages dont il sera l'objet.

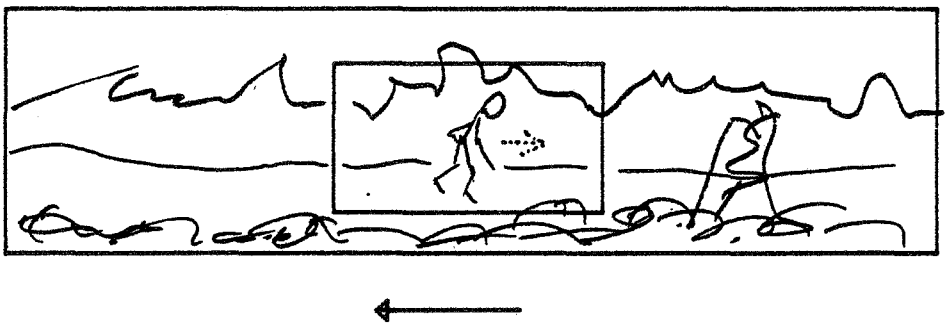
La réalisation des décors est confiée aux décorateurs et assistants décorateurs.

Il y en a un et un seul par plan du film (sauf lorsqu'on réutilise un même décor pour plusieurs plans du film); c'est en gros ce qui peut définir le plan: partie du film où une action se déroule continûment sur un même décor. Nous verrons plus loin que l'utilisation de décors réalisés à l'aide de logiciels de synthèse d'images tridimensionnelles remet en cause cette définition.

L'animation figurera sur un support transparent (paragraphe suivant: les cellulos) et laissera voir le décor autour des acteurs (figure 1.8.); le décor n'est donc dessiné qu'une fois, ce qui autorise plus de finesse dans sa représentation que dans celle de l'animation où les plages de couleur unie sont presque de rigueur.

Il s'agira souvent de dessin, de peinture ou d'aquarelle. On utilisera aussi des photos ou l'incrustation de l'animation dans un film (Mary Poppins).

Les décors doivent être conçus en fonction de leur utilisation au tournage. Lorsque des mouvements de caméra sont prévus sur un décor, ce dernier pourra largement excéder la taille du cadre de prise de vue. Par exemple, lorsqu'un personnage court parallèlement à l'écran et qu'on le suit, le décor doit être réalisé sur une longue bande qu'on fera défiler face à la caméra (figure 1.9.).



Décor plus grand que le cadre:
dans cet exemple, un personnage marche perpendiculairement à l'axe de la caméra; la caméra est supposée le suivre; l'effet est obtenu en déplaçant le décor par rapport au cadre filmé par la caméra.

Figure 1.9.

Nous verrons dans les paragraphes suivants les alternatives à l'adaptation informatique des décors.

Leur fabrication se déroule parallèlement à celle de l'animation proprement dite (dessins clés et intermédiaires), et est effectuée par des personnes différentes.

1.1.5.5. Les cellulos

Les cellulos sont des feuilles d'acétate transparent sur lesquelles on dessine l'action, ainsi seules les parties des cellulos où figurent des acteurs masquent le décor.

Traçage:

Les dessins clés et intermédiaires sont copiés sur les cellulos. Cette opération est assurée par des personnes (traceurs) qui reprennent chaque dessin à l'encre de chine ou par une sorte de photocopie (XEROXAGE). Si la deuxième solution est retenue, les dessins doivent être auparavant nettoyés .

Gouachage:

Ensuite, la mise en couleur s'effectue par gouachage du dos de ces cellulos, ainsi le gouacheur ne risque pas de cacher des traits, qui doivent rester apparents, par des bavures de gouache. En général les frontières entre couleurs sont cernées par le dessin au trait; ce n'est pas toujours le cas (ombre...). Un problème pratique est posé par le séchage des cellulos gouachés.

Il faut prendre garde à la stabilité des couleurs des gouaches utilisées d'une image à l'autre.

Signalons que la transparence des cellulos ne permet guère d'empiler plus de cinq couches sans dégradation des couleurs à la prise de vue.

Vérification:

La manipulation d'un grand nombre de dessins ne se fait pas sans erreurs. Une fastidieuse étape de vérification est nécessaire avant de transmettre les séquences de dessins pour le tournage. Il faut contrôler que tous les dessins y sont, correctement gouachés, correctement repérés.

La prise en compte informatique des dessins n'a de sens qu'accompagnée de l'informatisation du traçage et du gouachage. Elle peut être vue de deux façons:

-> traçage et gouachage automatiques des cellulos suivis d'un tournage traditionnel,
-> simulation informatique des cellulos incluant la phase de tournage

1.1.5.6. Les images filmées

Il s'agit des images obtenues par assemblage du décor et des différents cellulos correspondants à une image du film. Elles n'auront d'existence que sur le film lui-même. Il faut cependant les décrire et les préparer auparavant de façon précise. Leur description est l'objet de la feuille de prise de vue. Leur préparation constitue le travail de l'opérateur de banc-titre lors du tournage: il empile le décor et les dessins, régle les éclairages, positionne la caméra, comme le lui indique la feuille de prise de vue, puis il filme l'image.

1.1.6. LA FEUILLE DE PRISE DE VUE

Elle rassemble les données nécessaires à la prise de vue image par image:

- éléments à rassembler pour reconstituer l'image complète: références à des décors, à des cellulos.
- mouvements de caméra
- indications diverses : éclairages, ...

Exemple de feuille de prise de vue

aaa

FILM LM 26 | PLAN 17 | PAGE I/II

BANC TITRE

Rectification
Voir (A) (B) & (C)
EDF 26 PP (10)

TRAV. AVANT
(A) (B) EN
20 IMAGES

ANIMATION

BANC TITRE

ANIMATION

Coupé

Des annotations sur des groupes d'images indiquent les mouvements de caméra requis.

La feuille de prise de vue se présente sous forme de tableaux. Chaque image du film est décrite sur une ligne. Chaque colonne correspond à un niveau de l'empilement des dessins, voir plus loin: description de la prise de vue. Une colonne indique le décor. Une colonne plus large permet des annotations: effets spéciaux,éclairages.

La présentation de ce document s'apparente à ce qu'on peut faire sur ordinateur avec les tableurs électroniques comme le célèbre programme Visicalc, sans nécessiter la capacité de calcul d'un tel programme.

L'adaptation informatique de ce document apportera de telles facilités qu'elle constitue une priorité dans l'effort d'amélioration des techniques de fabrication d'un dessin animé. En effet, entre sa première version et son utilisation finale au tournage, ce document, d'un volume très important puisque chaque image y est décrite, subit de très nombreuses mises à jour. Les intervenants bénéficieront des facilités d'édition et de certains contrôles automatiques sur la cohérence de leur travail: par exemple, par vérification automatique de la relation entre la durée prévue d'une scène et le nombre d'images qui sont censées la composer.

1.1.7. LES DOCUMENTS ADMINISTRATIFS

La réalisation d'un film nécessite évidemment l'établissement de feuilles de payes, de comptes, d'état du stock de fournitures...

Certains de ces documents sont liés à un projet unique, d'autres servent à la gestion globale du studio de dessin animé. Ces documents se partagent des informations; il faut donc prendre garde à maintenir la cohérence entre les différents documents. Il s'agit d'un rôle que l'ordinateur sait bien tenir.

L'établissement de comptes sur un film doit respecter des règles spécifiques et prendre en compte le recours constant à des personnels temporaires.

La réalisation d'un programme de gestion d'un studio doit donc intégrer un grand nombre de composantes et supporter divers niveaux d'interrogation. Nous ignorons si un tel programme a déjà été commercialisé.

1.2. ANALYSE GLOBALE

Les chiffres qui suivent vont permettre d'orienter les efforts sur certains points sensibles; ils ont été obtenus à partir de la réalisation d'une petite série pour la télévision (26x6 minutes) ; les proportions sont données par rapport à l'ensemble de la réalisation; la post-production -sonorisation, montage- étant exclue.

Nous nous devons de signaler le caractère précaire de pareille évaluation. Ces chiffres ne sont en fait représentatifs que de quelques productions en France en 1983. Certains coûts peuvent être liés à des investissements importants ou à l'intervention de prestataires de services, rendus nécessaires par la petite taille des sociétés françaises de dessin animé. On peut par exemple s'étonner de la part consacrée à la prise de vue. Ces chiffres ne sont là que comme complément aux opinions des personnes consultées.

Environ le tiers du temps et le tiers du budget sont consacrés à la conception et l'ordonnancement. L'animation occupe le quart du temps et un peu plus de 10% du budget. La finition, traçage et gouachage, encore un quart du temps et 10% du budget. La prise de vue à peine 10% du temps, mais plus du quart du budget.

L'essentiel du gain de temps devrait donc être réalisé sur les parts les plus créatives du travail: conception et animation. Un gain appréciable pourrait également être obtenu sur la finition et le tournage. Il devrait être relativement aisé de réduire grâce à des outils informatiques les opérations répétitives qui ralentissent ces deux étapes.

Les gains en coût doivent être préférentiellement recherchés d'une part sur la conception et l'ordonnancement, d'autre part sur la prise de vue.

Dans cette optique deux voies semblent se dessiner. L'une, résolument industrielle, consiste à acquérir un ensemble d'équipements analogues à ceux proposés par le New-York Institute of Technology ou ceux étudiés pour les studios Hanna-Barbera; cela nécessite de lourds investissements, sans doute justifiés par l'ampleur prévisible du marché et le potentiel créatif français en matière de dessin animé.

L'autre voie, plus artisanale, consiste à tirer au mieux parti des petits moyens techniques financièrement à la portée des structures françaises actuelles de production de dessin animé. Des expériences ont montré que des produits originaux et très concurrentiels peuvent naître de cette voie.

Pour l'une ou l'autre voie, trois axes d'études se dégagent:

- améliorer les conditions de la conception,
- limiter les opérations répétitives par l'automatisation,
- améliorer les conditions d'encadrement et de suivi.

Des résultats à plus long terme peuvent être attendus des recherches tendant à renouveler les méthodes de création [LUC84].

1.3. ETAPES POSSIBLES

1.3.1. REMARQUES GENERALES

Divers matériels actuellement commercialisés prétendent à une diminution des coûts quasi-immédiate de la prise de vue et certains prennent même en charge la finition (traçage, gouachage). Leur évaluation n'a pas été possible dans le cadre de cette étude; les plus complets de ces systèmes ont une diffusion très restreinte, voire limitée à la firme qui les a conçus. Les principales indications disponibles ont été publiées dans les compte-rendus de la conférence annuelle Siggraph : [SIG79]...

Le travail sur la conception et l'ordonnancement doit commencer par une analyse soignée des flux d'informations dans un studio industriel de dessin animé. La feuille de prise de vue apparaît immédiatement comme un élément pivot de l'activité des studios de dessin animé.

Les éléments existants, matériels ou logiciels, font appel à des techniques naissantes ou pas encore stabilisées; les problèmes critiques sont alors la fiabilité, et souvent le manque d'ouverture et de documentation. Ainsi des matériels qui, théoriquement réunis, devraient répondre à certains besoins s'avèrent en pratique impossibles à assembler dans des conditions convenables de production, ou nécessiter l'adjonction d'intermédiaires coûteux: telle commande de prise de vue image par image n'est pas pilotable par ordinateur; tel

matériel de test d'animation ne permet pas d'imprimer la feuille de prise de vue qu'il contient sous forme de fichier, tel matériel est utilisable à l'aide de programmes différents qui produisent des résultats incompatibles entre eux.

1.3.2. APPORTS DE L'INFORMATIQUE

Conception, Ordonnancement

La conception (scénario, storyboard, dialogue, commentaire, graphisme de base) et l' ordonnancement (embauche, planning, comptabilité) prennent une part importante des délais et des coûts. Il serait donc intéressant d'obtenir un gain temporel et financier, par exemple en informatisant la gestion de l'information.

La production d'un dessin animé nécessite la manipulation, la mise à jour, la consultation d'un grand nombre d'informations qui font référence les unes aux autres implicitement ou explicitement. Nous proposons pour des développements ultérieurs la notion d'une base de données spécialisée comprenant la gestion de la totalité des informations manipulées : feuille de prise de vue, dessins, paye...

Des documents de travail permettent la coordination des équipes, le suivi du travail et la gestion de l'important stock de dessins (décors, animation, modèles). Ils ont une forme quasi-universelle définie par l'usage. A chaque projet sont associés des documents spécifiques qui partagent certaines données avec des documents d'administration générale du studio de production (comptabilité, paye).

La constitution d'une installation opérationnelle conduira sans doute à organiser les données en vue de leur exploitation sur un réseau local d'ordinateurs constituant un ensemble de postes de travail spécialisés (postes de saisies de dessins, poste de prise de vue, poste de test...). La spécialisation des postes permet de n'allouer à chacun que les ressources qui lui sont strictement nécessaires; le réseau local permet à chaque poste de mettre l'information qu'il crée à la disposition de tout autre poste qui en a besoin. Au vu des développements techniques des chapitres suivants, nous verrons en conclusion des caractéristiques souhaitables d'une telle installation.

Ce travail passe par l'informatisation de la feuille de prise de vue.

Bien que formé au travail sur la créativité, je ne développerai pas ce thème, les apports possibles relevant sans doute plus d'aides structurelles à la créativité des intervenants que d'outils informatiques.

Animation

La réalisation de l'animation est une des activités créatives principales de la production d'un dessin animé. Nous pensons que, avant longtemps, rien ne remplacera l'expérience et le talent des animateurs qui conçoivent l'animation avec du papier et un crayon. Ils utilisent un support courant qui ne pose pas de problème technique ou d'entretien, qu'ils maîtrisent bien.

D'autres méthodes ont été étudiées; nous avons cité plusieurs références sur ce sujet au paragraphe 1.5.3.. Nous proposons d'aider les animateurs par des outils de contrôle et de documentation de l'animation, plutôt que par des outils de réalisation de l'animation elle-même. Nous pensons que les travaux les plus intéressants sur ce dernier thème n'aboutiront que dans plusieurs années [LUC84][DIS84]; ils proposeront alors de nouvelles façons de créer une animation, mais ne remplaceront probablement pas les techniques traditionnelles.

L'animateur doit contrôler le rythme et la décomposition du mouvement qu'il dessine. Pour cela, il utilise deux méthodes : flip-book et film de test (linetest).

- Flip-book: le test du mouvement se fait en feuilletant rapidement la pile de dessins; l'animateur utilise ainsi de façon sommaire la persistance rétinienne.
- Linetest: un test plus complet du rythme et du mouvement d'une séquence est obtenu grâce au film de test: les dessins préparatoires au crayon noir sur papier sont filmés image par image dans des conditions analogues au tournage définitif. Dans le chapitre 3 figurent des alternatives à ces techniques traditionnelles.

En parallèle à son travail sur le dessin des mouvements, l'animateur doit mettre à jour la feuille de prise de vue. Elle servira à indiquer à l'opérateur de banc-titre, lors du tournage définitif, quels dessins doivent être utilisés pour composer une image, combien de fois les utiliser, dans quel ordre. Ce travail sera grandement facilité par une gestion informatique de ce document.

Les chiffres ci-dessus montrent que le résultat brut de ces nouvelles techniques sera plus un gain de temps qu'un gain financier. Cependant la réduction des délais de production est un argument commercial puissant; de plus elle permet de réduire la durée des investissements financiers à faire pour réaliser un dessin animé. L'évaluation de la rentabilité financière des investissements nécessaires est délicate sur des dispositifs exprérimentaux trop restreints. Il serait dangereux de chercher à utiliser ceux-ci directement pour des réalisations commerciales importantes; il est nécessaire de les tester dans des conditions proches de leur utilisation prévisible.

Finition

Pour la finition (traçage et gouachage), plusieurs systèmes ont été commercialisés. Ils permettent d'obtenir par des moyens informatiques des images vidéo. Ils partent des dessins préparatoires de l'animation. On obtient une représentation numérique de ces dessins à l'aide d'une saisie caméra. Un opérateur met ces dessins en couleur l'un après l'autre; il dispose pour cela d'outils informatiques rapides de propagation des couleurs. Les dessins sont ensuite assemblés avec les décors. Les images obtenues sont ensuite enregistrées automatiquement une à une; ces systèmes intègrent donc la prise de vue.

Les chapitres 2,3,4 concernent cette étape de la production. Chaque fois que cela sera nécessaire, nous situerons l'apport des solutions techniques proposées dans cette étude par rapport aux solutions retenues par ces systèmes.

De l'informatisation de la finition, il faut attendre essentiellement des gains de temps de production.

Prise de vue

Cette phase de la production, relativement courte coûte cher. Les solutions à mettre en oeuvre doivent donc réduire les coûts; ces solutions peuvent impliquer des investissements importants. Il faut être prudent sur l'évaluation de ceux-ci. Notons qu'un équipement de prise de vue (banc-titre) haut de gamme vaut dans les 600 Kf.

Nous présentons les différents aspects de ce problème au paragraphe suivant de ce chapitre: méthode de prise de vue, stockage et manipulation des dessins et des décors, utilisation de la feuille de prise de vue.

La méthode de prise de vue dépend des moyens techniques et financiers dont on dispose et du support final retenu (film, bande vidéographique, disque optique numérique, console informatique).

L'utilisation de la feuille de prise de vue peut nécessiter la présence d'un opérateur ou se faire automatiquement. Dans le premier cas, il suffit de disposer de celle-ci sur papier. Dans le second cas, il sera nécessaire d'avoir une représentation informatique de celle-ci.

Le problème de la manipulation des cellulos lors de la prise de vue dépend de leur support : classique (acétate), magnétique (vidéo), numérique. Les trois composantes à prendre en compte sont :

- support des décors,
- support des cellulos,
- support final.

Ce problème peut se ramener à deux questions techniques:

- passage d'un support à un autre,
- incrustation d'un élément dans un autre de même support.

Etape	Apport	Méthode
Tournage	Coût	automatisation cf.ch.3.parag.7
Finition	Temps	traçage: ch.2 gouachage: ch.3
Animation	Temps; Confort Coût Anim.couleur	test; tableur intervallage automatique
Conception	Coût; Temps	l'avenir le dira
Ordonnancement	Temps; Confort	bureautique

Il faut noter que l'informatisation du processus de production de dessin animé aura certainement des retombées sur le processus créatif qui l'accompagne. En effet, si l'on produit plus vite, le concepteur disposera de moins de temps pour affiner sa conception au cours de la réalisation.

Les expériences de production avec des outils informatiques ont montré que ceux-ci ne s'adaptent bien qu'à certains styles graphiques et narratifs. Il en sera sans doute de même des outils futurs comme il en était ainsi des outils traditionnels.

1.4. INFORMATISER

Nous allons étudier ici plus en détail les étapes possibles pour une informatisation progressive de l'ensemble de la production d'un dessin animé. Nous verrons comment chaque objet peut être simulé ou géré par des outils informatiques. Nous verrons que certaines des étapes proposées peuvent être réalisées indépendamment; une coordination est tout de même indispensable pour que les résultats obtenus séparément puissent être réunis.

1.4.1. QU'INFORMATISER D'ABORD?

Nous allons envisager l'informatisation en remontant le processus de réalisation d'un dessin animé de la fin vers le début. Ceci pour trois raisons: parce que les activités créatives et d'organisation du début du processus soit soulèvent des problèmes très délicats de rapport entre la technique et la création, soit font appel à des techniques d'informatique de gestion tout à fait classiques qui sortent donc du champ d'une recherche. Ensuite parce que nous avions des idées concernant les autres étapes qui méritaient un important travail de validation. De plus, l'obtention de l'animation sur un support diffusable est une des étapes les plus coûteuses de la chaîne de production.

La conception et l'ordonnancement représentent près de la moitié du budget. Il sera intéressant d'étudier comment la productivité à ce niveau peut être améliorée, notamment par des outils informatiques: traitement de texte, pense-bête, tableurs. L'ordonnancement peut être facilité dès aujourd'hui avec des outils du prêt à porter informatique: progiciels de comptabilité générale et de paye. Cette étude est du ressort de sociétés de service en informatique ou des studios de dessin animé eux-mêmes.

La prise de vue représente un quart du budget. Il s'agit d'une étape très minutieuse. Son automatisation devrait permettre un gain appréciable de productivité et de coût. Nous commencerons notre étude par elle.

Notons que pour une série de 26x26mn, s'il fallait 3mn pour mettre en place (composer) chaque image pour la prise de vue, il faudrait environ 6 ans de tournage 24 heures sur 24. En pratique, plusieurs ateliers de tournage fonctionnent en parallèle.
Cette même série comprendrait:

* plus de 8000 décors si on met un décor par plan de 7 secondes

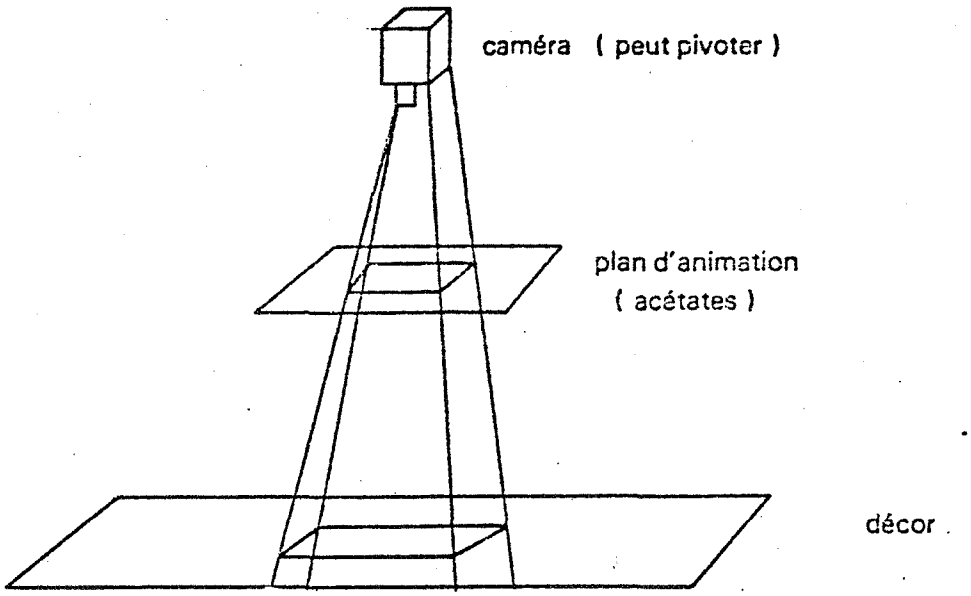
* plus de 480 000 images à composer à la prise de vue (12i/s)

* environ 1,5 millions de cellulos à raison de trois par image

On perçoit aisément l'importance d'une bonne gestion de production et, plus généralement, la nécessité d'une organisation très méthodique du travail.

1.4.2. PRISE DE VUE

La prise de vue est traditionellement faite sur un banc-titre. Il s'agit d'un support de caméra qui permet de placer les dessins et les décors sur des plans perpendiculaires à l'axe de la caméra (figure 1.11.). Celle-ci est mobile suivant cet axe. Les plans horizontaux sont mobiles à la fois suivant cet axe et dans le plan perpendiculaire à celui-ci (translations et rotations). La combinaison de ces mouvements permet notamment des effets de travelling et de zoom.



Banc-titre

Figure 1.11.

Un opérateur, guidé par la feuille de prise de vue, compose les images en plaçant les documents référencés par celle-ci sur les plans et dans l'ordre d'empilement convenable. Il s'occupe également de régler les éclairages et les mouvements de caméra.

La célèbre multiplane de Walt Disney se distinguait des banc-titres normaux par un grand nombre de plans et une grande liberté de mouvements de chacun par rapport à la caméra.

1.4.2.1. Informatisation du banc-titre

Il existe dans le commerce des banc-titres gérés par ordinateur (de 150 à 1700 kf). Ils permettent à la fois un gain de productivité de l'opérateur de banc-titre et un gain dans la qualité et les raffinements de son travail. L'opérateur continue à manipuler les dessins; il est assisté par l'ordinateur pour les mouvements relatifs de la caméra et des dessins.

Les logiciels permettant la description de ces mouvements pourraient être intégrés à un système plus global. Certains permettent par exemple à l'opérateur de préparer une translation de la caméra sur son axe (zoom), en indiquant l'ampleur du mouvement, sa durée en nombre d'images. La dynamique est quelquefois précisée parmi quelques options: déplacement linéaire ou amorti en début et en fin, d'autres fois elle peut même être précisée sous forme d'une courbe tracée sur une table à numériser. Ainsi ces logiciels, étudiés pour répondre aux besoins des professionnels, proposent nombre de fonctionnalités qu'il serait bon d'intégrer dans un studio informatisé.

1.4.2.2. Informatisation de la prise de vue

Pour tirer le meilleur parti de l'informatique pour effectuer la prise de vue, il faut créer une représentation informatique des images à manipuler et disposer de matériel de transfert de ces images numériques sur un support de diffusion: vidéo, film. Nous étudierons donc ces questions avec l'étude de l'informatisation des images.

1.4.2.3. Informatisation des images

L'automatisation de la prise de vue passe par l'informatisation des images, à moins de concevoir des processus mécaniques complexes de manipulation des cellulos sur acétate.

Le rôle d'un opérateur sur les cellulos simulés informatiquement deviendrait très pauvre: désigner des fichiers à utiliser et leur appliquer des opérateurs, en fonction de la feuille de prise de vue.

Toute l'information nécessaire étant contenue dans la feuille de prise de vue, l'informatisation de celle-ci, qui ne pose pas de problème informatique difficile, doit

être effectuée.

L'utilisation de cellulos informatiques suppose plusieurs opérations:

* description des dessins au système informatique: cette opération sera détaillée au chapitre 2 ; à chaque dessin est attribué son "nom" dans la feuille de prise de vue,
* description des décors; idem pour les noms,
* coloriage des dessins, assemblage des dessins et des décors: nous abordons ces problèmes au chapitre 3,
* prise de vue.

Nous verrons par la suite que les décors peuvent être traités soit dans, soit hors du système informatique.

La prise de vue peut être envisagée de plusieurs façons:

* enregistrement image par image sur système vidéo adapté
* report image par image sur film 16 ou 35 mm
* stockage sur un support informatique permettant la diffusion

Les systèmes informatiques restituent usuellement les images sous forme de signal vidéo. Si ce signal respecte les normes en vigueur en télévision professionnelle, il n'y a pas d'obstacle à son enregistrement. Citons pour mémoire des systèmes vidéo permettant l'enregistrement image par image : magnétoscope 1 pouce (Sony; environ 1000 Kf), prototype Thomson, système Anivid de pilotage de magnétoscope 3/4 de pouce de montage (environ 100 Kf), disque magnétique à enregistrement analogique.

Deux techniques peuvent être utilisées pour le report sur film:

* filmer un écran spécialement conçu (Matrix environ 300Kf...); des films réalisés ainsi ont montré que cette méthode donne des résultats acceptables.
* dessiner sur pellicule (Com Benson,...); cette méthode donne des résultats d'une exceptionnelle qualité, mais à des coûts élevés et à condition de connaître les images sous une forme compatible avec ce mode de tracé (voir chapitre 2).

Dans la suite nous nommerons ces trois techniques vidéo, report sur film, traçage sur pellicule.

Les décors

Ils peuvent être issus de diverses sources et différer par leur support:

* support classique: papier, toile,
* support vidéo : caméra pour filmer directement un décor réel ou une maquette, bande magnétique, vidéodisque,
* support film : pour incrustation de dessin animé dans un film (Mary Poppins),
* image numérique

Les supports classiques fournissent un stockage dense et laissent une large part de leur utilisation à des techniques désormais bien maîtrisées.

Le stockage d'images numériques avec la haute définition nécessaire pour un décor est volumineux. Il sera bon d'utiliser un codage pour réduire ce volume de données informatiques. Le codage par plage semble bien adapté à des décors dessinés.

Un épisode d'une série de 26 mn comprend environ 300 plans et peut compter donc 300 décors; si on leur attribue une définition de 768x540 pixels (norme télé professionnelle) sur 24 bits par pixel, on arrive à environ 1 Méga-octet de stockage par décor, de l'ordre de 10 fois moins pour une image codée par plage. On pourra éviter le stockage par une numérisation des décors simultanée à leur utilisation. Des supports magnétiques de stockage, disponibles actuellement, à lecture et écriture assez rapides (cartouches) ont une capacité de 10 Mo, soit environ 10 décors non codés par cartouche. Nous proposons plutôt un groupement des dessins et du décor d'une séquence sur un support unique.

Une cartouche contiendra par exemple 3 séquences de 6 secondes, comportant donc 72 images différentes, chacune composée de 3 dessins de base; chaque dessin pourra être stocké sur 10 Kilo-octets. Ces chiffres sont tout à fait compatibles avec les estimations établies lors des expérimentations décrites par la suite (Notons qu'une cartouche coûte en Septembre 84 de l'ordre de 1 Kf).

On a trois composantes du problème :

support du décor
support des parties mobiles
support final

On se ramène essentiellement à deux problèmes techniques:

->passage d'un mode de représentation à un autre
->incrustation entre deux images représentées dans un même mode

passage d'un mode de représentation des images à un autre:

passer de-> / à ↓	classiq.	vidéo	film	numériq.
classiq.	possible	éviter	av.réserve	av.rése
vidéo	possible	éviter	possible	possible
film	possible	av.réserve	possible	possible
numériq.	possible	possible	possible	possible

incrustation d'un élément dans une image ayant même support:

classique : facile, manuel, lourd
vidéo : facile, qualité vidéo, manipulation globale d'images prédéfinies correctement, perte de qualité
film : processus délicat sur du matériel de précision
numérique : facile, problèmes d'aliassage

Pratiquement chaque type d'incrustation est possible. Chaque type de transfert aussi moyennant un intermédiaire (qui peut être coûteux : film->classique->digital). Il est donc bon de recenser les avantages qu'il peut y avoir à utiliser un support plutôt qu'un autre d'une part pour les décors, d'autre part pour les dessins. Les résultats observés à ce jour montrent que la réponse est très liée à l'application.

Actuellement, les systèmes d'image informatique produisent directement des images vidéo. Nous verrons plus loin que certains choix sur la représentation informatique des images permettent de s'adapter à d'autres périphériques d'affichage d'image trés différents. Les décors restent un problème: leur incrustation sur film est très délicate; leur digitalisation induit une perte de

qualité et nécessite d'importants stockages; seule leur incrustation en vidéo semble bien résolue.

Prise de vue vidéo

Il est simple de transférer à l'aide d'une caméra un décor d'un support classique à un support vidéo. On se ramène alors au cas du décor sur support vidéo.

Un décor sur support film peut être passé en support vidéo par un télécinéma. Un décor sur support vidéo peut être utilisé grâce à l'ensemble des techniques vidéographiques bien connues; l'incrustation en particulier. Les parties animées de l'image issues de l'ordinateur seront traduites en signal vidéo et incrustées sur les décors. Il faudra disposer d'un système de repérage précis des décors sur le support vidéo notamment dans le cas où le décor est mobile.

Cette méthode peut être mise en oeuvre sur un banc-titre automatisé (simplifié puisqu'il ne sert qu'à obtenir une représentation vidéo du décor) sans autre manutention que le changement de décor. Puis l'incrustation des parties mobiles dans ces décors sera quasi-automatique. De plus des mouvements de caméra pourront alors être facilement maîtrisés sur ces décors, en corrélation avec les mouvements de l'animation.

Un décor digital peut être traduit en signal vidéo, on se ramène alors au cas précédent. Il peut être généré par un programme (par exemple calcul de perspective sur une maquette numérique). Ceci permet en particulier d'avoir facilement des décors mieux définis et filtrés que les parties mobiles.

Un décor digital peut être aussi utilisé comme tel. Les parties mobiles y seront intégrées sous forme numérique, puis l'image résultante traduite en vidéo sera enregistrée. Cette méthode permet de garder les avantages de la digitalisation jusque sur l'image résultante avant son enregistrement (en particulier retouche interactive).

Les dessins, les cellulos

Nous abordons en détail au cours des développements techniques les questions liées à la fourniture à un système informatique d'une description des dessins (saisie: chapitre 3), posées par les représentations informatiques de cette description (codage,stockage: chapitre 3), induites par les opérations sur ces représentations (vérification, gouachage, prise de vue: chapitre 4).

4.2.4. La feuille de prise de vue

Son informatisation peut être menée indépendamment de l'informatisation de tout le reste de la prise de vue. Il ne faudra pourtant pas bloquer l'accès des informations qu'elle contient, en effet celles-ci peuvent permettre une automatisation complète de la prise de vue, autant pour les tests que pour le tournage final.

Dans un premier temps, il peut s'agir seulement d'adapter les méthodes connues pour la réalisation de tableurs en donnant à l'utilisateur une imitation informatique de la feuille de prise de vue sur papier. L'avantage résidera dans la facilité de correction et de mise à jour, ainsi que dans les contrôles de cohérence; des facilités pour la gestion d'animations préfabriquées [BAU84] peuvent être envisagées.

Lors du couplage avec le stockage informatique des images, il paraît souhaitable que la portion de feuille de prise de vue de chaque séquence soit stockée sur le même support que les images de cette même séquence.

BIBLIOGRAPHIE DU CHAPITRE

[BAU84] [BCT84] [BRI84] [BUR83] [COU80] [LUC83] [LUC84] [MAR70] [REE81] [SIG79] [WAL81]

CHAPITRE 2

SAISIE DES DESSINS

2.1. QUELS PROBLEMES CHERCHONS-NOUS A RESOUDRE?

Parmi les voies d'études présentées au chapitre précédent, nous avons approfondi les travaux concernant l'informatisation des images et étudié les conséquences de celle-ci sur le traçage, le gouachage et le tournage. Nous avons laissé hors de notre étude l'informatisation de la feuille de prise de vue. Nous mentionnerons dans la conclusion de ce travail les choix que les résultats obtenus suggèrent sur la conception d'une chaîne de production de dessin animé.

Nous approfondissons dans ce chapitre et les deux suivants des solutions à certains problèmes posés par les données graphiques à manipuler: nature et qualité des données graphiques, stockage, coloriage, restitution.

Nous avons choisi de commencer l'informatisation au traçage, tel qu'il a été défini au chapitre 1. L'animation a été préparée sous forme de séries de dessins faits au crayon sur du papier mince perforé. Nous voulons passer de ces dessins au document diffusable, en couleur, animé, à un rythme de l'ordre de 25 dessins par seconde. La télévision française diffuse 25 images par seconde; les télévisions nord-américaines diffusent 30 images par seconde; la plupart des dessins animés répètent chaque dessin deux fois: ils nécessitent donc en France la production de 12 dessins par seconde.

Il nous faut connaître les dessins en noir et blanc sous forme numérique; puis il faut les mettre en couleur. Pour finir, on assemblera les dessins de base avec les décors pour composer l'image définitive et l'enregistrer sur le support de diffusion (film, cassette vidéo...).

> Dans ce chapitre nous nous attacherons à voir comment on produit une représentation numérique des dessins. Nous comparerons les types de saisie proposés et les représentations numériques obtenues, entre elles et avec les saisies actuellement utilisées par les systèmes existants.

2.2. REPRESENTATIONS NUMERIQUES DES IMAGES

Dans la suite, nous employons le mot dessin pour désigner des images composées de traits noirs sur fond blanc. Le mot image est pris dans son sens le plus général; les dessins sont donc des images particulières.

Il nous faut une représentation numérique des dessins. Nous devons donc choisir entre les deux formes principales que peut prendre celle-ci: géométrique ou point par point.

2.2.1. REPRESENTATION POINT PAR POINT DES IMAGES : PICTOGRAMME

> La représentation point par point fournit l'image sous forme de trame régulière de points; à chacun de ces points a été attribuée une couleur. La résolution ou définition de l'image est donnée par trois valeurs: le nombre de points par ligne, le nombre de points par colonne, le nombre de valeurs possibles pour chaque point. Nous reprendrons la terminologie introduite par Baudelaire [BAU84]] et nommerons par la suite cette représentation pictogramme.

Avec une résolution d'au moins 500 points sur 500 et un grand choix de nuances de couleur, le pictogramme peut offrir une grande qualité du rendu graphique: textures, nuances, et une complète compatibilité avec des images naturelles numérisées par caméra. Son principal défaut est le volume de données nécessaire pour son stockage et sa manipulation.

Une image destinée à une télévision avec une définition de 768x576 pixels représentés sur 8 bits chacun, soit 256 couleurs différentes dans l'image, minimum requis si on veut faire de l'anticrénelage entre les traits noirs et les plages de couleur [STE79], occupe environ 400 kilo-octets. Si on veut coder pour réduire ce volume, on doit supposer qu'il existe certaines cohérences dans l'image. On perd alors la liberté de travail sur l'image que donne le pictogramme .

En pratique de telles cohérences existent toujours dans les dessins destinés à l'animation de série. Elles sont notamment utilisées dans le système décrit par Wallace [WAL81]; dans ce système, le stockage et les traitements sont effectués sur un codage par plage des images. La photo A (cf.planche 1 p.55) présente une image 512x640 codée sur 30 kilo-octets par un codage par plage élémentaire: chaque plage horizontale comprend au maximum 256 pixels; elle est représentée sur un octet par sa longueur et un octet pour sa couleur. Nous montrons par la suite que ces mêmes cohérences permettent de représenter ces dessins par des vectogrammes et que cette dernière représentation offre des facilités de traitement que n'offre pas le pictogramme.

Certains logiciels et matériels, comme les palettes électroniques, permettent à un graphiste d'agir sur un pictogramme avec des outils dont l'ergonomie a été étudiée pour se rapprocher de l'utilisation de pinceaux et de crayons [SMI81].

La difficulté des manipulations géométriques sur cette représentation fait qu'on stocke généralement les images à la résolution de l'écran prévu pour l'affichage. Les données stockées sont alors peu compatibles avec l'affichage sur un autre support. Ceci implique que l'ensemble des outils de visualisation utilisés dans la chaîne de production soient du même type, à moins de mettre en oeuvre des procédures complexes de traduction des images pour chaque outil.

Le passage de tout mode de représentation informatique d'une image à une représentation par un pictogramme est possible et même généralement facile. Le pictogramme est la plus répandue des représentations internes aux dispositifs d'affichage.

Si l'image est destinée à une diffusion par les télévisions européennes, elle doit comporter 576 lignes de 768 points; il faut 480x640 points pour les télévisions nord-américaines.

Aujourd'hui peu de mémoires d'image commercialisées offrent cette résolution. Le format le plus fréquent est de 512 lignes; il reste donc quelques lignes non influencées par la mémoire d'image lors de l'affichage vidéo; elles restent généralement noires (il faut noter que sur les 575 lignes imposées par la norme, toutes ne sont pas réellement visibles; le nombre de lignes utiles est environ 530).

Le pictogramme permet la définition aisée de chaque point de l'image; il permet donc notamment facilement la représentation de traits d'épaisseur variable. Nous revenons sur l'intérêt de cette facilité au paragraphe 5.4 de ce chapitre. Par ailleurs, le problème du remplissage d'une zone d'un pictogramme par une couleur est abordé à la fin du chapitre 3.

Pour l'aide à la réalisation de dessin animé de type cartoon, les systèmes commerciaux actuels utilisent le pictogramme, dans une de ses représentations: explicite, codée par plage. Au cours de cette étude, nous présenterons ce que nous savons de la façon d'opérer de ces systèmes et comparerons ces méthodes à celles que nous proposons.

2.2.2. REPRESENTATION GEOMETRIQUE DES IMAGES : VECTOGRAMME

> La représentation géométrique fournit l'image comme une collection de formes géométriques plus ou moins organisées : points, ligne, polygones,cercle...Il s'agit d'une représentation structurée de l'image. Ces formes sont définies par leurs caractéristiques géométriques : coordonnées des extrémités, centre...Une image est ainsi décrite sous forme d'une liste d'objets géométriques. Nous désignerons cette représentation par le mot vectogramme [BAU84].

Il est par exemple possible de conserver une image sous forme d'une liste de polygones; chacun peut être associé à la couleur qui servira à le remplir. Mais plus la représentation de l'image est structurée, plus elle impose de contraintes pour la méthode de saisie. C'est pourquoi,

dans la suite, on utilisera le terme vectogramme pour désigner des listes de lignes brisées composées de segments de droite.

Le vectogramme permet une représentation précise des formes avec une grande compacité des données numériques nécessaires à son stockage. On peut lui reprocher une certaine pauvreté dans le rendu graphique des traits. Des travaux ont été faits pour améliorer cet aspect; le principe est d'attacher un mode de tracé à chaque segment [COU80] : tracé "aérographe", épaisseur, ...

Le problème des épaisseurs de traits

Il est d'usage d'améliorer la lisibilité des dessins animés en marquant les frontières de couleur par un liseré d'une couleur différente, dont l'épaisseur est variable. Dans les productions de qualité, il n'est pas rare de rencontrer des liserés de couleurs différentes et même quelques zones sans liseré. Cela complique le travail des traceurs (cf.ch 1) et augmente donc le coût de la production.

Le dessin animé de série nécessite une homogénéisation du travail des dizaines d'intervenants. Les caractéristiques des dessins sont donc normalisées : modèle de crayon utilisé, qualité du papier... Ainsi les nuances exprimées par les épaisseurs de traits ne font pas partie des caractéristiques courantes de l'animation de série. Dans ce type de production, le liseré est généralement noir et d'épaisseur assez uniforme.

Au cours de cette étude, nous avons indiqué des solutions pour conserver l'information sur les épaisseurs des traits dans la représentation numérique des dessins. Nous présentons notamment dans ce chapitre, au paragraphe 5 une méthode de saisie optique. Elle fournit des vectogrammes à partir des dessins originaux. Elle permet de conserver l'information sur les épaisseurs de traits à la précision des organes de saisies.

Vectogramme et tournage

Des outils de transfert de représentations informatiques d'images ont été présentés au chapitre 1: COM ou film d'écran pour produire du film, mémoire d'image et magnétoscope ou disque magnétique pour produire de la

vidéo. La seule représentation acceptée en entrée par l'ensemble de ces outils est le vectogramme.

Certains logiciels destinés aux graphistes exploitent déjà l'idée de stocker une image sous forme de vectogramme. Au lieu de stocker le pictogramme obtenu par le graphiste à la fin de sa session de travail, ils mémorisent l'historique des actions effectuées. Cet historique utilise les données fournies par une table à numériser; ces données sont donc conservées à la résolution des organes de saisie et non à celle de l'écran de travail, elles peuvent être ensuite utilisées pour produire des diapositives à haute résolution (4000x4000).

Nous avons choisi de faire porter l'essentiel des développements ultérieurs de cette thèse sur la représentation des images par vectogramme. Nous avons voulu étudier et, autant que possible, évaluer l'utilisation de cette représentation par rapport aux méthodes utilisant des pictogrammes [WAL81][STE79]. Le vectogramme est compact, facile à manipuler, facile à restituer sur divers supports; il faut évaluer le coût de ces avantages lors des opérations les plus importantes pour notre application: saisie, coloriage, restitution. Les solutions techniques proposées dans la suite permettent sans problème d'utiliser des décors sous forme soit de pictogramme, soit de vectogramme.

La suite de ce chapitre est consacrée aux problèmes de la saisie numérique des dessins préparés au crayon noir sur papier blanc. Deux types de saisie ont été distingués. Les saisies manuelles sur table à numériser nécessitent un opérateur, qui va suivre les traits du dessin un à un à l'aide d'un outil spécial. Les saisies automatiques: caméra, scanner permettent d'obtenir une représentation numérique d'un dessin uniquement en le plaçant sur un dispositif de lecture optique.

	PICTOGRAMME	VECTOGRAMME
Structuration	faible	forte
Analogies langages info	machine	structuré
Transfo. géométriq.	difficiles, coûteuses	faciles
Stockage	volumineux, sauf codage	compact
Rendu graphique	libre	dépend du logiciel
Compat. imag. caméra	oui	non, sauf suivi de contours
Changement représent.	difficile vers vecto.	facile vers picto.
Epais. traits	facile	difficile
Tournage	report vidéo	tous supports

2.3. SAISIE MANUELLE

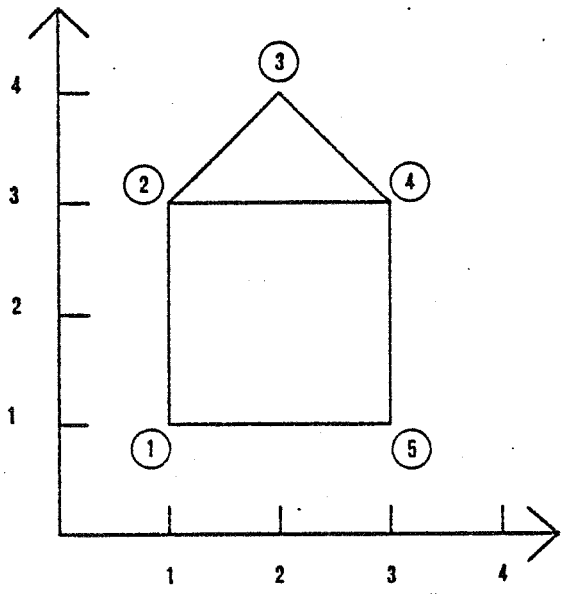
> Nous nous inspirons ici d'une des méthodes traditionnelles de traçage. A partir de dessins de préparation faits sur papier, des personnes (traceurs) recopient à l'encre de chine chaque dessin sur une feuille d'acétate transparent (cellulo). La saisie numérique se fera de façon analogue en suivant les traits du dessin à l'aide d'un outil spécial: table à numériser, stylo optique.

L'électronique de contrôle du stylo optique permet de connaître la position sur un écran de la tête d'une tige (stylo) qu'un opérateur déplace. La résolution de l'affichage (vidéo) limite la résolution des données obtenues. Des postes de travail spéciaux ont été conçus pour cet outil; ils comportent un écran presque horizontal pour limiter la fatigue de l'opérateur de saisie qui, sinon, se trouve le bras levé en permanence. Cet outil a l'avantage de l'identité entre le support de saisie et celui de visualisation; cet avantage s'avère important pour les applications où on souhaite réaliser les dessins sur le système informatique, sans préparation préalable. Pour nous, la définition des données obtenues est insuffisante et la méthode est peu adaptée à la copie de dessins déjà préparés sur papier.

Nous disposions pour nos tests d'une table à numériser. Il s'agit d'une surface sensible à la position d'un capteur que la main déplace ("stylo", curseur, réticule). L'électronique qui gère l'ensemble émet vers un processeur externe des triplets (marqueur,x,y) où :

marqueur est un indicateur de la nature des données suivantes. Nous avons utilisé ici les deux valeurs marqueur = plume haute et marqueur = plume basse indiquant si l'opérateur souhaite que les points transmis par le numériseur fassent partie d'un tracé ou non.

x, y sont les coordonnées du point lu dans le repère de la table. A titre d'exemple, la table à numériser utilisée dans notre travail délivre des coordonnées utiles en dixièmes de millimètres.



Ligne 1: points 1,2,3,4,5,1
Ligne 2: points 2,4

3831 points
2085 arêtes

Figure 2.1.

Il est alors possible de saisir sur une table à numériser les dessins préparés classiquement. Cette méthode de saisie entraîne des frais en personnel d'exploitation. Elle est moins coûteuse en installation et maintenance et permet avec peu de moyens d'envisager les apports de l'informatique pour les étapes suivantes: traçage, gouachage, tournage. La table à numériser est un outil s'adaptant facilement à d'autres applications. De plus, contrairement à d'autres méthodes de saisie envisagées dans cette étude, elle ne pose pas de problèmes d'ingénierie pour sa mise en oeuvre. Les logiciels associés sont simples et peu gourmands en place mémoire. Cela permet d'envisager un poste de saisie peu coûteux : un micro-ordinateur gérant une table à numériser et un dispositif de stockage statique à accès rapide, comme un disque dur. Un tel poste individuel ne pose pas les problèmes que posent les systèmes multi-utilisateurs (cf.Saisie manuelle et qualité graphique).

Saisie manuelle et contours de zones

La numérisation manuelle résout partiellement, le problème rencontré par tous les sytèmes d'aide informatique à la réalisation de dessins animés basés sur l'utilisation de dessins préparés sur papier. Il s'agit du problème des surfaces non fermées: le gouachage informatique des dessins impose de connaître les limites des zones à remplir, or les zones que nous percevons comme distinctes ne sont parfois séparées par aucun trait sur le dessin.

Lors de saisies sur table à numériser, il est possible de renvoyer une partie de la responsabilité des fermetures sur l'opérateur de saisie pour tous les cas où le défaut est perceptible sur l'écran de contrôle. De plus, il est possible de doter le poste de saisie d'outils logiciels de détection des ouvertures. Nous verrons au chapitre 3 que tous les problèmes ne disparaissent pas même avec un opérateur parfaitement soigneux, si les données sont conservées à la résolution fournie par la table à numériser (vectogramme); si on passe directement à la représentation par pictogramme, toute ouverture entre deux zones est en principe perceptible sur l'écran d'affichage; en pratique, une ouverture d'un pixel le long d'un trait noir est difficilement perceptible.

Style graphique se prêtant mal au traitement automatique des zones de couleur; il y a très peu de zones fermées par un contour, alors que des zones sont visuellement distinctes: yeux, oreilles, dents... (d'après P.Barletta "Au-delà de minuit",INA-AAA 83)

Figure 2.2.

Saisie manuelle et qualité graphique

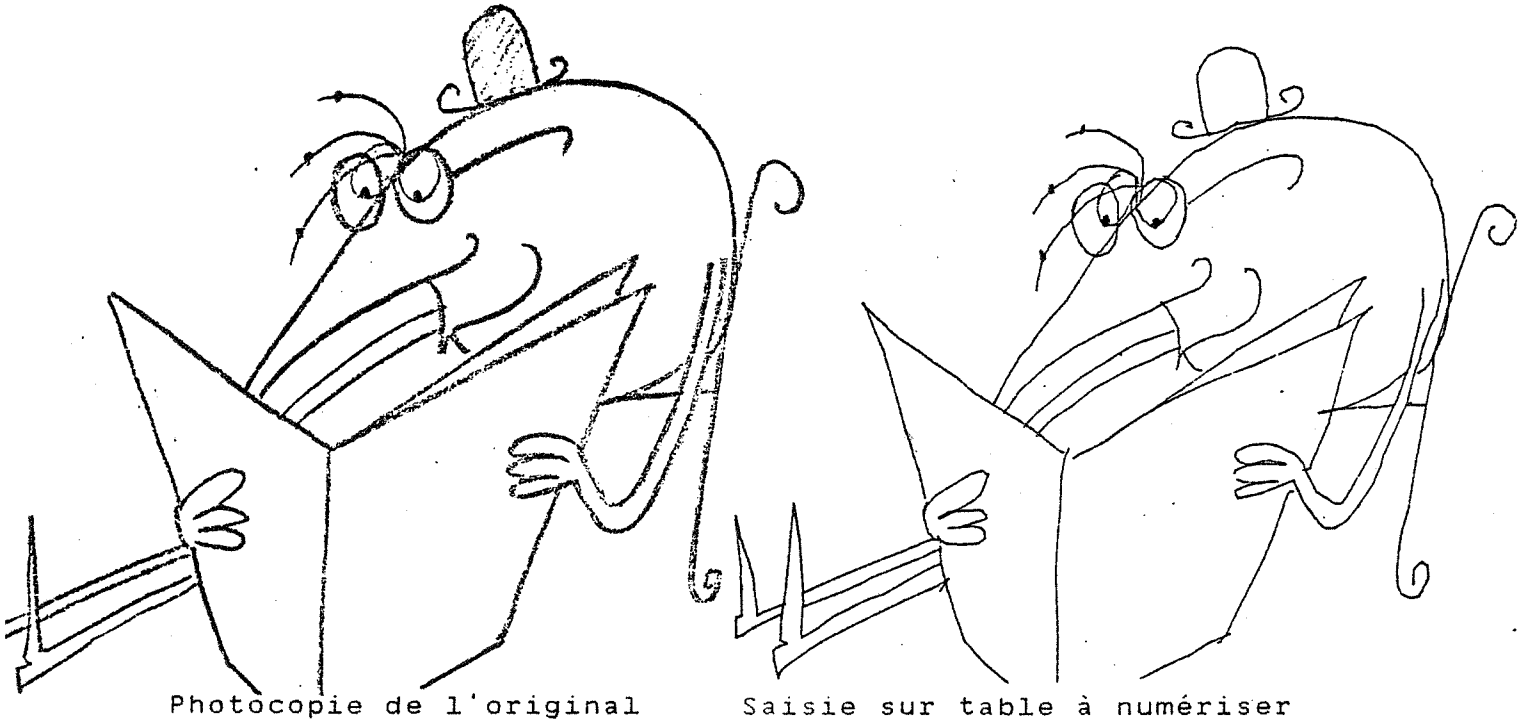
La qualité graphique des données saisies dépend de plusieurs facteurs. L'adaptation de l'opérateur à ce mode de saisie intervient : contrairement aux supports classiques, la surface de visualisation du dessin tracé -l'écran- est différente de la surface sur laquelle le dessin est tracé -la table. Il semble que la plupart des personnes s'habituent rapidement à ce découplage.

La saisie manuelle proposée ne résout pas un des problèmes du traçage traditionnel: il est difficile de repasser fidèlement sur les traits d'un dessin déjà tracé et de ne pas oublier des traits. Des asservissements optiques du stylo de saisie ont été étudiés pour aider au suivi du tracé; nous manquons de références à ce sujet.

La vitesse d'acquisition des points par l'électronique de commande doit être suffisante par rapport à la vitesse de déplacement du crayon par l'opérateur. Si cette condition n'est pas remplie, des lignes brisées disgracieuses remplaceront les belles courbes tracées. Le même résultat sera obtenu si la transmission vers le processeur externe n'est pas assez rapide et que des points sont perdus au cours de la transmission. Ce défaut est évité si l'interface avec l'ordinateur dispose d'une mémoire tampon importante. Une telle mémoire est indispensable lorsque l'ordinateur est muni d'un système multi-utilisateur.

Cette interface pourrait également être munie d'une intelligence lui permettant de transformer les données saisies moyennant un certain nombres de règles paramétrables: changement de repère, distance minimum entre deux points consécutifs, angle minimum entre deux arêtes consécutives [GAN84], calage des points sur une grille.

De plus si l'interface fournit elle-même l'écho du dessin à un écran de contrôle, l'influence de la vitesse du transfert vers l'ordinateur hôte disparaît. On a donc avantage à disposer d'un terminal de visualisation capable de gérer localement une table à numériser.



Photocopie de l'original Saisie sur table à numériser

Problèmes d'épaisseur de traits et de fidélité de la saisie

Figure 2.3.

La figure 2.3 est la reproduction d'un dessin original accompagnée d'une saisie sur table à numériser dessinée sur un traceur format A4. Cette saisie comprend 1000 points et environ le même nombre d'arêtes. La reproduction de l'original n'est pas exempte de défauts dûs à la méthode utilisée: photocopie d'un dessin au crayon. Elle permet cependant d'illustrer la perte de qualité graphique dûe à la saisie sur table à numériser : une telle saisie ne peut rendre compte de l'épaisseur des traits de l'original et le dessinateur de saisie parvient plus ou moins à suivre le tracé original; le choix de cette méthode de saisie a donc une incidence directe sur la qualité graphique du dessin animé produit.

Une saisie prend de 2 (fig.2.3 ch.2) à 12 minutes (fig.2.16 ch.2) et comprend généralement de 500 à 5000 points. Au moins autant que pour les méthodes classiques, on aura intérêt à décomposer les dessins en plusieurs parties: il est plus rapide de saisir 5 dessins simples qu'un seul regroupant les mêmes traits, l'opérateur repérant mieux où il en est de sa saisie.

Le peu d'intérêt du travail de saisie condamne cette méthode à terme. Pour l'instant, elle se justifie par le faible coût d'acquisition du matériel nécessaire et peut permettre ainsi à des équipes artisanales de bénéficier des apports de l'informatique. Notons par ailleurs que les méthodes d'intervallage automatique sont liées à cette méthode de saisie. Nous montrons dans ce chapitre que des représentations de dessins de type vectogramme peuvent être obtenues à partir de saisies optiques (cf.∫o.4) et abordons au chapitre 4 paragraphe 5 une technique pour obtenir semi-automatiquement la structuration des dessins nécessaires aux méthodes d'intervallage automatique [BUR76][REE81].

2.4. SAISIE AUTOMATIQUE

Nous envisageons ici les méthodes de saisie numérique des dessins se rapprochant de la méthode suivante:

Les dessins sont passés dans une machine de type photocopieuse qui copie le dessin sur cellulo (terme professionnel: Xéroxage). Cette opération nécessite de disposer de dessins sources très propres. Une étape de nettoyage est nécessaire pour enlever les taches parasites issues de la photocopie. Actuellement un opérateur assure manuellement le calage de chaque dessin sur des ergots permettant un positionnement rigoureux des dessins par rapport à l'optique de saisie. L'automatisation complète de ce processus soulève de nombreux problèmes mécaniques: qualité du papier, précision...

La saisie optique peut faire appel à plusieurs technologies.

Saisie par caméra

> La saisie par caméra consiste à placer les dessins convenablement éclairés face à une caméra vidéo; le signal vidéo est numérisé pour fournir une représentation de l'image par un pictogramme, en général avec plusieurs bits par pixel, la combinaison des bits indiquant le niveau de gris du pixel.

Les saisies de ce genre offrant une bonne définition spatiale -au moins 500x500 pixels- sont actuellement coûteuses car elles sont construites avec une optique et une électronique de grande qualité. Elles nécessitent un banc de saisie bien étudié pour avoir de bonnes conditions d'éclairement et de calage des dessins. Aucun système du marché ne permet une numérisation automatique d'une pile de dessins. Les caméras vidéo de qualité sont à la fois coûteuses et fragiles.

Au crédit de cette installation, il faut noter qu'elle permet la saisie numériques d'images quelconques : décor, photos...Elle est donc exploitable avec d'autres programmes de production d'image et pourra ainsi être mieux amortie par des structures dont la production de dessins animés n'est pas le seul but: trucage d'images fixes, stockage d'images, archivage sur disque optique numérique.

La façon dont les systèmes actuels exploitent ce type de saisie nous est connue principalement par les descriptions de Stern [STE79] et de Wallace [WAL81]. Un opérateur installe les dessins l'un après l'autre sous la caméra vidéo. Ceux-ci sont numérisés à raison de huit bits par pixel, puis traités pour ne conserver que quatre bits par pixel. Plusieurs bits sont gardés afin de réduire l'effet de crénelage dû à la résolution de saisie (de l'ordre de 500x500). Par la suite les zones blanches de l'image seront remplies avec des teintes; à proximité des traits, un mélange sera effectué entre le niveau de gris du pixel et la teinte (ch.3 parag.6: Gouachage).

Il faut noter que des systèmes assurent une numérisation temps réel (25 images par seconde fournies par une source vidéo) avec une définition de l'ordre de 500x500 pixels, représentés chacun sur 8 bits. La saisie automatique rapide d'une pile de dessins ne pose donc pas de problème de numérisation, mais:

- des problèmes mécaniques de placement des dessins face à la caméra vidéo, qui ne relèvent pas de solutions théoriques,
- des problèmes de volume de données à traiter ou à stocker; notamment le temps de transfert de ces données vers un stockage statique ne peut être négligé: de l'ordre de 30 secondes par image sur notre configuration (cf. annexe 1).

Ce type de saisie a donné lieu à de nombreux travaux pour en corriger les défauts. La plupart des systèmes de numérisation d'image par caméra ont intégré les résultats obtenus pour la correction des distorsions géométriques et la compensation des différences d'éclairement sur la surface de saisie. On trouvera des indications intéressantes dans [ODG81] [STE79].

Nous avons effectué des essais avec une numérisation par caméra fournissant 512x512 pixels chacun sur 1 bit. Le stockage d'une telle image nécessite 32 kilo-octets dans une représentation explicite et environ 6 kilo-octets une fois codé par plages. Les essais élémentaires de codage par plage effectués sur notre configuration demandaient environ 30 secondes par image.

Scanner

Il existe un autre système de numérisation : les numériseurs à barre de diodes, utilisés notamment pour les appareils de transmission de fac-similé par téléphone (télécopie) . Une barre de diodes balaye le document; un récepteur photosensible couplé à chaque diode permet de connaître la valeur noir ou blanc, de chaque point. Certains dispositifs peuvent reconnaître plusieurs niveaux de gris par point. Les problèmes de distorsion sont grandement limités.

La norme des appareils de télécopie fixe la définition d'une telle numérisation à 1768x2200 pixels tout ou rien (noir/blanc). La numérisation d'un document prend de l'ordre de 30 secondes avec les appareils disponibles actuellement. Pour l'instant les documents à numériser doivent être placés un par un sur le système. Il est cependant envisageable qu'une pile de documents puisse être placée en entrée et numérisée sans intervention humaine, à condition que ces documents respectent les contraintes mécaniques imposées par le système d'entraînement de type photocopieur (épaisseur du papier). On retrouve ici les problèmes de calage mentionnés au début de ce paragraphe à propos de la méthode reprographique de traçage.

L'image tout ou rien obtenue peut être ramenée par filtrage à une image en niveaux de gris similaire à celles obtenues par saisie caméra: des points blancs partout où il n'y a pas de traits, des points à divers niveaux de gris pour représenter les traits Par exemple, une méthode élémentaire consiste à moyenner les valeurs de chaque groupe carré de 4 points voisins, pour passer d'une image 2200x1768 à une image 550x442 à 5 niveaux de gris.

Il faudra prévoir une étape ultérieure de nettoyage et de contrôle des dessins numérisés, pour remédier aux erreurs possibles de tout système de reprographie: mauvaise manipulation, mauvais réglage, document impropre à la copie.

Ces deux méthodes: caméra et barre de diodes fournissent un pictogramme représentant le dessin original. Nous étudions au paragraphe 5 une méthode permettant d'obtenir un vectogramme équivalent.

2.5. OBTENIR UN VECTOGRAMME A PARTIR D'UNE SAISIE OPTIQUE

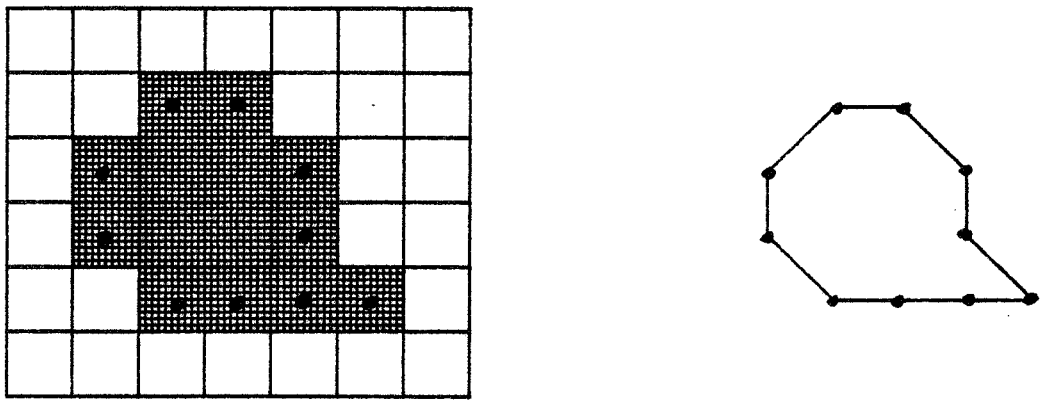
Nous supposons ici qu'un numériseur produise ligne par ligne le pictogramme binaire (noir/blanc) du dessin saisi. Nous voulons au cours de la saisie calculer un vectogramme donnant une représentation exacte du pictogramme d'origine.

Le vectogramme fournit une représentation exacte du pictogramme d'origine s'il est possible de reconstruire ce dernier à partir de la connaissance du premier.

L'image tout ou rien obtenue est codée pour obtenir un stockage plus dense. Il faut choisir un codage assurant un bon compromis entre le taux de compression, la facilité de codage et de décodage et la richesse de l'information directement apparente dans le code. Nous proposons un code proche de celui de Freeman [FRE74], mais compatible avec le balayage de trame. L'information apparente dans ce code est la description sous forme chaînée des contours des zones noires de l'image.

Pour cela, nous détaillons une méthode générale d'analyse et de codage d'images binaires au cours d'un balayage de celles-ci de haut en bas et de gauche à droite. La généralisation à des images quelconques est envisagée. Des suggestions d'implémentation tant matérielle que logicielle sont proposées. Cette méthode semble bien s'adapter à des dessins du type de ceux qu'on rencontre en dessin animé (voir illustrations de cette thèse: fig.2.2,2.3,2.14,2.16...).

La représentation obtenue est du type vectogramme puisqu'il s'agit de listes de petits segments codés. Les traits du dessin sont des zones de l'image; puisque le code décrit les contours de ces zones, l'information sur les épaisseurs des traits est préservée par cette représentation.



Une forme et son contour sous forme de chaînons

Figure 2.4.

Nous sommes partis des travaux de Cederberg [CED79][CED81], en nous efforçant de clarifier les fondements de la méthode; cette clarification a permis une nette amélioration de l'algorithme de codage, au point de faire apparaître une structure simple pour sa réalisation matérielle, ainsi que pour celle du décodage.

Cette méthode permet l'extraction des contours et le calcul de caractéristiques attachées à chaque contour: périmètre, surface. Les contours peuvent être aisément fournis sous une forme proche du code de Freeman [FRE74],[LUC81] mais compatible avec le balayage ligne à ligne; ce code peut donc être assimilé à un vectogramme. Ce codage,très compact, est assez comparable à un codage par plage avec exploitation de la cohérence verticale, et permet dans certains cas d'implémentation un accès simple aux contours.

Pour notre application, trois aspects de ce codage sont essentiels: implémentation matérielle simple, compactage des données stockées, connaissance des contours de l'image sous une forme de type vectogramme.

2.5.1. METHODES DE CODAGE

Le codage d'images binaires a déjà de nombreuses applications pour le stockage, le traitement et la transmission de celles-ci. Dans ce type d'images on trouve notamment le courrier et les imprimés, les schémas (électronique, bâtiment...) et certains relevés géographiques: courbes de niveau, cartes météorologiques.

Le but principal du codage est de réduire le nombre de bits nécessaires pour représenter l'image et donc le nombre de bits à stocker, à traiter ou à transmettre. Son efficacité peut se mesurer suivant plusieurs critères; le plus usuel est le taux de compression: rapport du nombre de bits de code d'une image à son nombre de pixels. Le deuxième critère est son adaptation à l'application visée.Il faut citer aussi: l'importance de l'erreur introduite par le codage, la complexité du codage et du décodage, la sensibilité du code à des erreurs de transmission.

La plupart des codages d'image binaire utilisent le fait que l'information sur l'image est complétement décrite par la donnée des limites entre les zones noires et les zones blanches. Les principales exceptions sont le code quad-tree [KLI76] et certains codes pour la télécopie où une reconnaissance des symboles présents dans l'image est effectuée [PRA80].

Des méthodes de codage par des arcs de courbe de ces frontiéres ont été proposées [COU81]. Une autre méthode répandue, directement liée à la représentation des images numériques sous forme de tableaux de pixels est le code par chaînage (Freeman [FRE74]): pour chaque frontière, on donne les coordonnées d'un point appartenant à cette frontière, puis on avance de proche en proche le long de la frontière jusqu'à revenir au point de départ. Chaque point a huit voisins sur la grille, il suffit donc de trois bits pour indiquer où est le point suivant du parcours. Ce codage, très commode en analyse et en traitement, nécessite au moment de son élaboration et au décodage un accès aléatoire aux points de l'image.

D'autres méthodes ont tenu compte de la contrainte courante d'accès à l'image par balayage de haut en bas et, sur chaque ligne, de gauche à droite. Il s'agit en particulier du codage par plage [YAS80] et de ses variantes, notamment bidimensionnelles [YAS80]. Pour chaque ligne d'un codage par plage on mémorise le début de chaque zone noire et sa longueur. Un tel code permet facilement le remplissage convenable des zones noires en un balayage de trame au cours du décodage et ne nécessite de mémoire d'image ni au codage ni au décodage.

Le code présenté ici combine les avantages du codage par chaînage et ceux du code par plage. Il est adapté au balayage et ne nécessite pas de mémoire d'image pour le codage et le décodage; il facilite nombre de traitements et d'analyses. En particulier, il me semble intéressant de

l'utiliser en télécopie, où pour gagner en compression on a l'habitude de combiner méthodes de compression et d'analyse.

Nous présentons et illustrons d'abord le principe du codage des contours au cours du balayage. Puis nous donnons les règles et des exemples d'implémentation logicielle ainsi que des indications pour une implémentation matérielle.

2.5.2. LE CODE CHAINE PAR BALAYAGE: PRINCIPE ET EXEMPLES

2.5.2.1. Principe

Pour parvenir à suivre les contours pendant le balayage, il faut les découper en portions que l'on saura suivre de haut en bas et de gauche à droite. Pour cela, nous devons détecter les points de naissance des contours: nous en donnons une définition exacte ci-après; il s'agit essentiellement des minimums locaux au sens du balayage. A partir de tels points, deux branches descendantes (parois) naissent. Nous pouvons suivre chacune d'elles et la coder suivant un code analogue au code par chaînage (fig. 1). Nous terminons une chaîne décrivant une branche dès qu'elle atteint un point de mort: il s'agit essentiellement des maximums locaux.

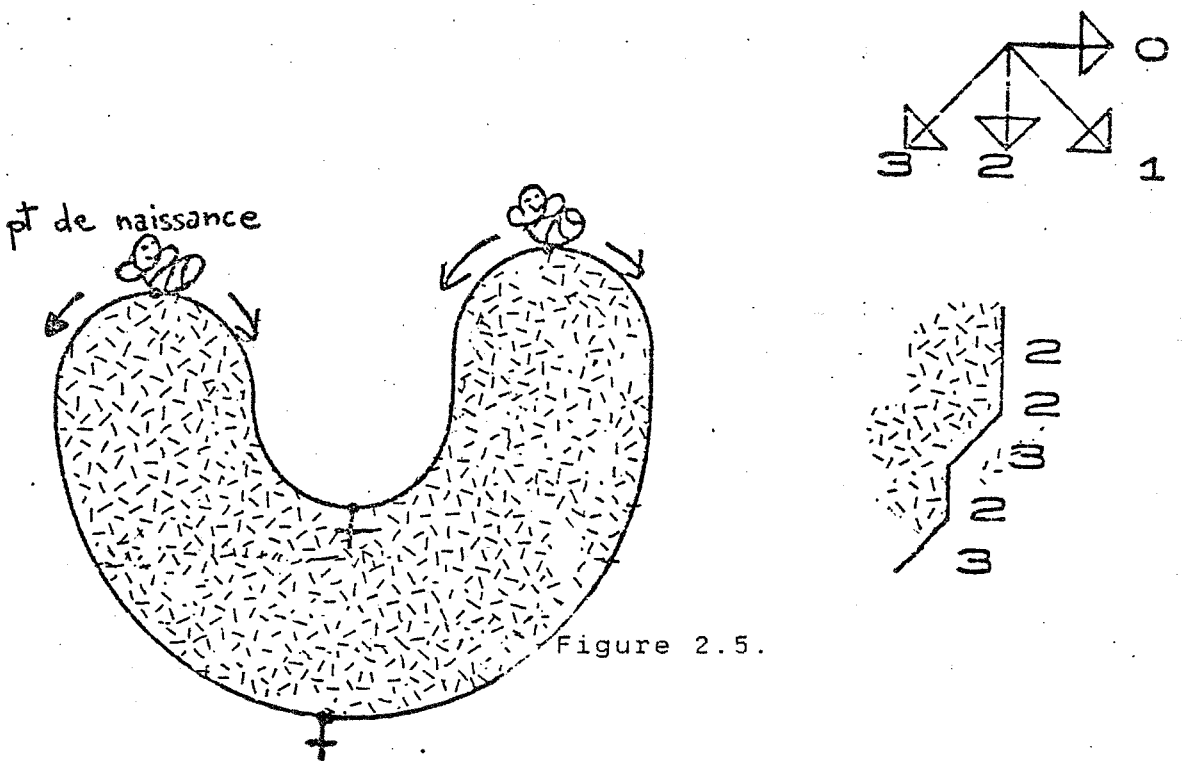


Figure 2.5.

Des traitements de ce type permettent d'obtenir des descriptions mathématiques des contours, sous forme de liste d'arcs de courbe [COU81], [BLO82].

Le balayage ligne à ligne introduit un ordre entre les pixels de l'image:

Pixel(i,j)>Pixel(k,l) si et seulement si
(i>k ou (i=k et j>l))

Les pixels de contour sont les pixels noirs dont au moins un des voisins est blanc; le voisinage utilisé correspond à la 8-connexité sur une trame carrée: deux points sont voisins s'ils ont en commun un coin ou un côté. Ces définitions permettent d'établir un ordre entre les pixels appartenant à une même paroi:

si Pixel(i,j) et Pixel(k,l) appartiennent à la même paroi, Pixel(k,l) précède Pixel(i,j)

si et seulement si Pixel(i,j)>Pixel(k,l).

Cet ordre permet de poser les définitions suivantes:

-point de naissance : point de contour sans précédent;

-point de mort : point de contour sans successeur ;

-paroi : liste des points de contour d'un point de naissance à un point de mort;

Un lien est une indication de direction pour passer d'un pixel à un de ses voisins supérieurs à lui. Il y a quatre types de liens possibles (cf. fig. 2.5). En pratique, une paroi sera représentée par son point de naissance et une liste de liens permettant de parcourir les points de contour de la paroi.

Sur la grille usuellement utilisée pour représenter une image numérisée, nous devons parvenir à établir les chaînages entre les points de contour et reconnaître les points de naissance et de mort. Pour chaque point à étudier, la première information importante est de savoir s'il appartient au fond ou à une tache. Il suffit de regarder sa valeur.

Pour savoir si un point appartenant à une tache est sur son contour ou non, nous devons observer si un au moins de ses huit voisins est de la couleur du fond. Pour savoir

s'il s'agit d'un point de naissance, nous devons observer son voisin de gauche et ses trois voisins du dessus. Pour savoir s'il s'agit d'un point de mort nous devons observer son voisin de droite et ses trois voisins du dessous (fig. 2.6.). Pour étudier l'image au niveau d'un point nous avons donc besoin d'observer une zone de 3x3 points centrée sur le point étudié.

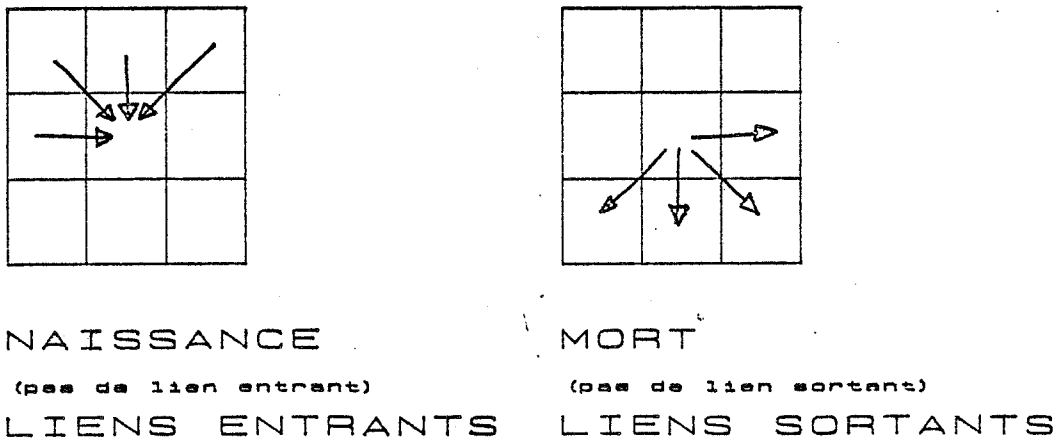


Figure 2.6.

Il y a 512 types de zones 3X3 possibles. Pour chacune, nous pouvons noter dans une table les caractéristiques du point central qu'elle recouvre: point du fond, point interne à une tache, point de contour, point de naissance, point de mort, chaînage(s) du point avec son(ses) précédent(s) et son (ses) successeur(s) La table peut être réduite à 256 entrées si la couleur du point central de la zone est préalablement testée.

Méthode de construction d'un index à partir du voisinage d'un point:

```
+-+-+-+
|0|1|2|   les pixels entourant le pixel testé sont
+-+-+-+
|3| |4|   numérotés un entier est constitué avec
+-+-+-+
|5|6|7|   les bits associés dans l'ordre 76543210
+-+-+-+
```

On a alors une table à 256 entrées associant chaque zone 3X3 à ses caractéristiques; nous donnons ci-après des échantillons de cette table. Sa construction est aisée, aussi nous ne l'avons pas incluse dans le texte (voir

aussi § 5.3.4.2).

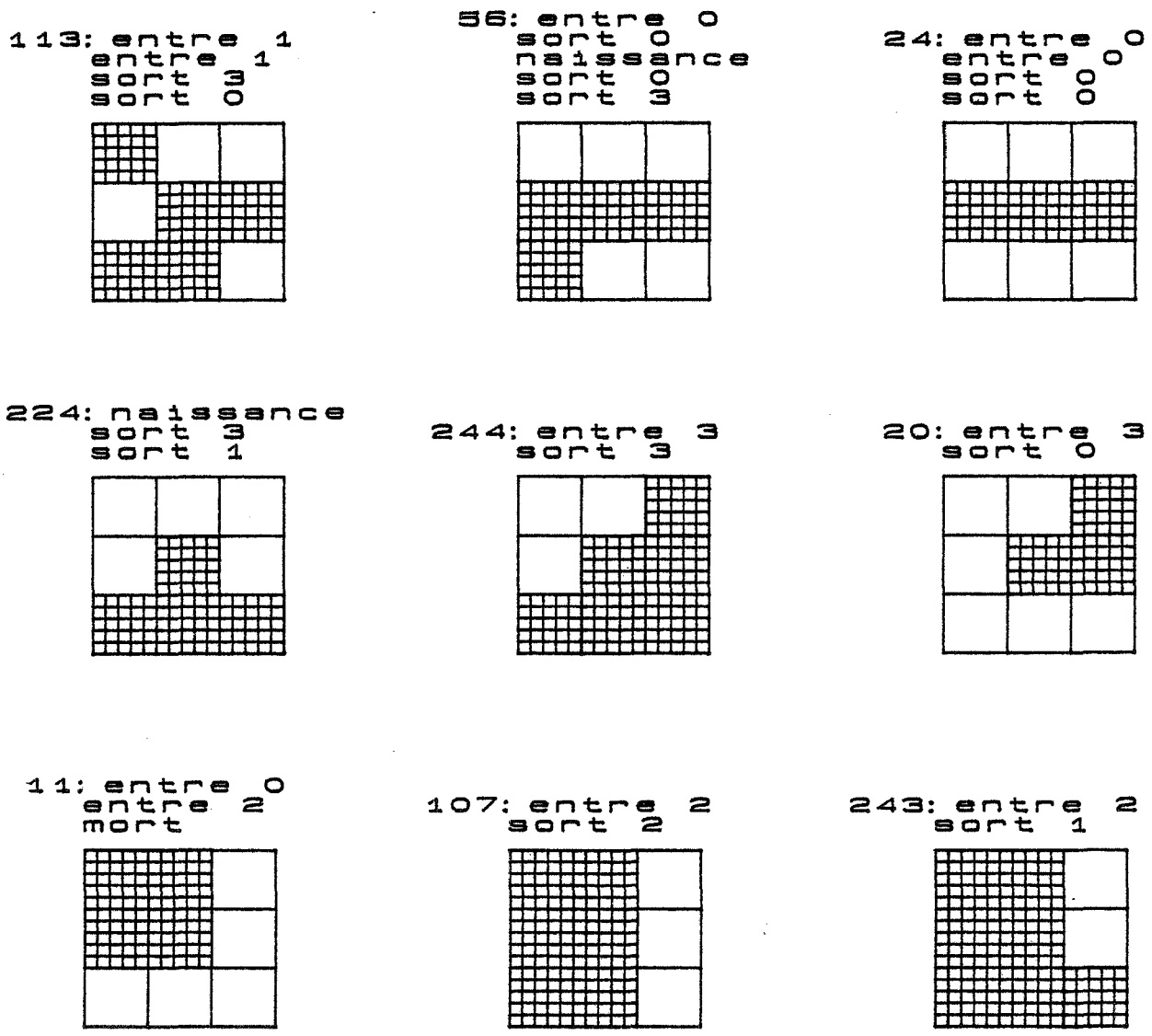


Figure 2.7.

Nous savons ainsi extraire l'information locale nécessaire. La cohérence entre ces données locales devra être maintenue lors du stockage de l'information au cours du balayage. Eventuellement d'autres informations disponibles au cours du balayage peuvent être stockées simultanément: périmètre, surface [AGR77]. L'ensemble du stockage peut être effectué de différentes manières,

suivant l'utilisation ultérieure du codage. Nous verrons plus loin des exemples d'implémentation.

2.5.2.2. Les travaux de Cederberg

Cederberg, dans un article de 79 [CED79] et la thèse publiée en 81 [CED81] expose une mise en oeuvre de ce principe en indiquant de nombreuses variantes possibles suivant l'utilisation ultérieure de l'image codée. Il suppose implicitement que l'analyse éventuelle de l'image est postérieure à son codage. L'intérêt de certaines implémentations détaillées par la suite est de permettre d'extraire de nombreuses caractéristiques de l'image au cours du codage lui-même.

Un effort de généralisation a été fait. Les mêmes algorithmes peuvent notamment être adaptés pour obtenir une description des contours par des suites de segments de droite: vectorisation.

Les mêmes algorithmes peuvent également s'adapter au codage d'images à plusieurs niveaux d'intensité ou de couleur; à chaque point de naissance, on attachera une "couleur" par paroi partant de ce point; chacune indique la couleur de la zone dont la paroi correspondante est une partie de la frontière gauche, pour un balayage de haut en bas et de gauche à droite; cette couleur est active jusqu'à la prochaine paroi rencontrée sur la ligne de balayage (fig.2.8.).

Point de naissance 3,2
couleur 1=gris foncé
couleur 2=blanc

Point de naissance 6,2
couleur 1=gris clair
couleur 2=blanc

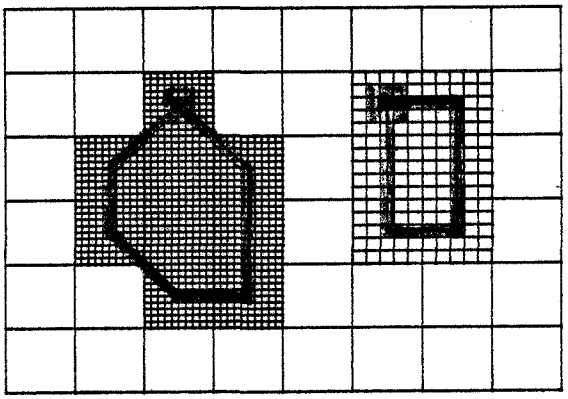
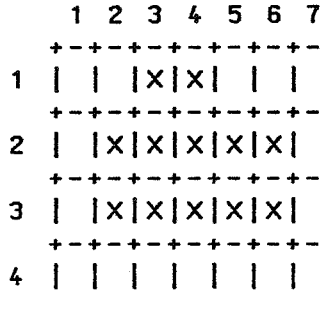


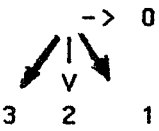
Figure 2.8.

2.5.2.3. Exemples

Elémentaire,

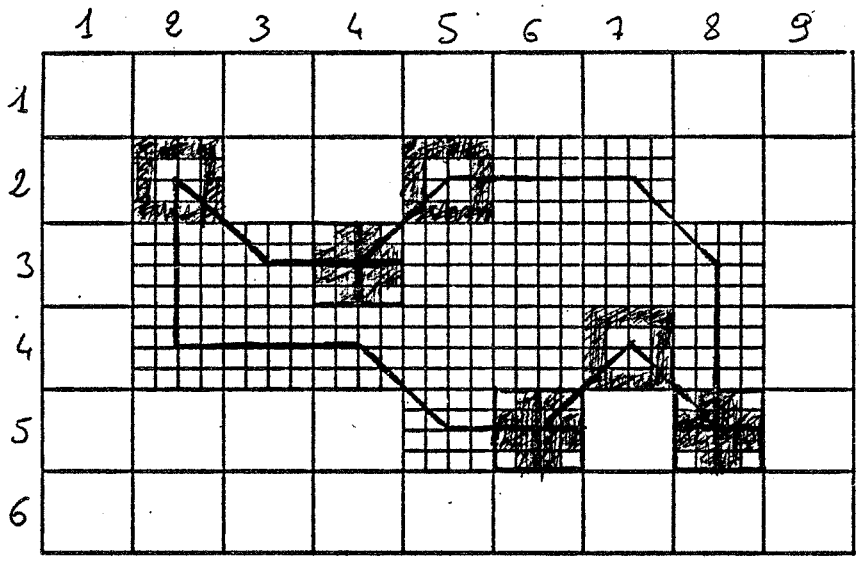


point de naissance: (1,3)
paroi 1 : 3,2,0,0,0,0
paroi 2 : 0,1,0,2



On peut mémoriser le fait que les parois 1 et 2 sont issues du point de naissance 1 et que la paroi 1 se ferme au même point que la paroi 2; ainsi on est capable de suivre le contour complet de l'objet détecté. En gérant une structure parenthésée, il est possible d'obtenir l'arbre descriptif des niveaux d'inclusions des groupes de parois formant des contours.

Nature morte



naissance (2, 2) (5, 2) (7, 4)

liens: 2130012002001312O

(2, 2) { 220010
10
(5, 2) { 3
00122
(7, 4) { 3
1

2.5.3. L'IMPLEMENTATION

L'effort de généralisation a conduit à décomposer l'algorithme en plusieurs étapes qui communiquent par des données intermédiaires. Le balayage de l'image fournit à l'algorithme d'analyse locale des paquets de neuf pixels voisins; l'analyse locale établit les caractéristiques de chaque paquet de pixels et les transmet à l'algorithme de stockage. Cette remarque constitue une voie de parallélisation du procédé (traitements pipe-line); en particulier le balayage et l'analyse se prêtent bien à une réalisation matérielle (cf.ch.2 ∫ 5.3.3).

2.5.3.1. Règles générales du codage

Le codeur peut être découpé en quatre modules:

->extraction d'une ligne de l'image
->extraction d'un point étudié et de se huit voisins
->analyse des caractéristiques du point
->stockage ou utilisation du résultat de cette analyse

Extraction d'une ligne de l'image:

Quelle que soit la méthode de lecture de l'image, il sera avantageux de disposer d'un tampon-mémoire de trois lignes d'image, organisé en file PEPS (première entrée, première sortie). En effet, pour les étapes suivantes il suffit de travailler sur ces trois lignes. Nous voyons qu'une mémoire d'image est inutile (fig. 2.9.).

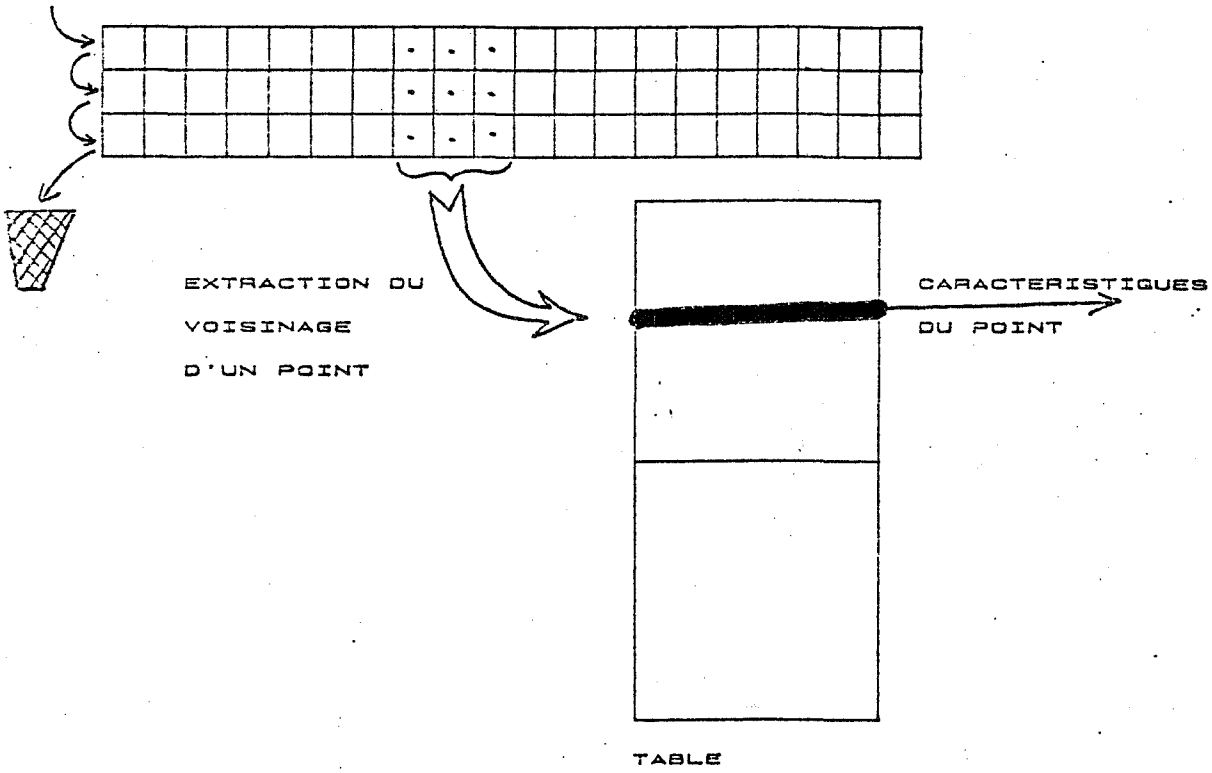


Figure 2.9.

Extraction d'un point étudié et de ses huit voisins:

La ligne analysée est la ligne centrale du tampon-mémoire. Pour analyser un point de cette ligne, on lit le point précédent et le suivant sur cette ligne ainsi que les points correspondants des deux autres lignes.

Analyse des caractéristiques d'un point:

Ce module prend neuf bits en entrée et restitue les caractéristiques énumérées au 2. Cederberg [CED79] a proposé une méthode d'analyse, mais la plus simple est sans doute la constitution d'une table qui fait correspondre à n'importe quelle zone 3X3 ses caractéristiques.

Stockage du résultat de l'analyse:

Ce module est très dépendant de l'utilisation ultérieure de l'image codée: décodage seul, traitement, reconnaissance, transmission, passage à un autre code...Nous traitons en 3.3 plusieurs exemples.

Allure de l'algorithme:

```
de la première à la dernière ligne faire

    du premier au dernier point faire

        observer le voisinage du point pour détecter ses
        caractéristiques ( point de naissance ou de mort,
        point d'un contour atteint par le haut à
        gauche,au-dessus ou à droite, ou par la gauche )
        si le point est un point de contour,stocker
        l'information le concernant

    fin

fin
```

2.5.3.2. Balayage et analyse

Nous détaillons ici la phase commune à toutes les implémentations : balayage de l'image et analyse des caractéristiques locales du point.

Les procédures **initialisations,détermine,stocke** seront détaillées par la suite pour différentes implémentations. La procédure détermine prend en entrée le point à analyser et son voisinage et retourne les caractéristiques locales que l'on souhaite utiliser; on peut la voir comme une table qui fait correspondre sa description "topologique" à chaque configuration de 9 pixels.

Par souci de clarté, nous donnons les versions élémentaires des algorithmes suivants. Les méthodes classiques d'optimisation pourront leur être appliquées [KNU81][MEY80]. Par exemple, pour l' **algorithme_maitre_du balayage** on pourra appliquer le principe du tampon circulant entre les tables A, B, C pour limiter les transferts de données.

algorithme_maitre_du_balayage;

```
type
  pixel=(allumé,éteint);
  une_ligne_d_image=array[0..max_pixel] of pixel;
var A,B,C:une_ligne_d_image;
begin

      initialisations;
      tampon_forcé_à_zero(A);
      index_de_ligne:=1;
      lire(B,index_de_ligne);
      for index_de_ligne:=2 to numero_de_ligne_maximum do
      begin

         lire(C,index_de_ligne);
         traiter_la_ligne_B ;
         A:=B;
         B:=C;

      end;
      tampon_forcé_à_zero(C);
      traiter_la_ligne_B;

  end;
```

traiter_la_ligne_B

```
begin

      for index_de_colonne:=1 to
            numero_de_colonne_maximum do

         if B[index_de_colonne]=allumé then

            traiter_le_pixel
                     (index_de_ligne-1,index_de_colonne);


  end;
```

```
traiter_le_pixel(index_ligne,index_colonne)

 var point_de_bord:boolean;
 begin

       (* considérer comme zone de test le carré de 9
      pixels centré sur le
       pixel (index_ligne,index_colonne) *)
       détermine(index_ligne,index_colonne,
                       point_de_bord,les_caractéristiques);
       if point_de_bord then stocke(les_caractéristiques);

   end;
```

2.5.3.3. Indications pour une implémentation matérielle

Les considérations précédentes font bien ressortir qu'une partie du processus consiste seulement à mémoriser trois lignes d'image, à en extraire un point et ses huit voisins pour adresser une table; il est clair que tout ceci est aisé à mettre en oeuvre avec des composants électroniques classiques. La table livrera par exemple à un processeur externe le numéro d'une action à effectuer.

Supposons que les données viennent d'un scanner du type utilisé en télécopie, il est simple de stocker séquentiellement les données arrivant dans des blocs mémoire; un bloc mémoire sera affecté à une ligne; il faudra quatre blocs pour travailler sur trois lignes pendant qu'on en lit une quatrième.

Le principal problème est d'adresser ces mémoires pour en tirer les bits décrivant le voisinage d'un point; l'astuce, couramment utilisée pour parvenir à ce style de double adressage écriture/lecture, consiste à provoquer à l'écriture une forte redondance dans les données. La méthode extrême consiste à stocker dans un octet les trois premiers bits ,puis dans l'octet suivant de stocker à nouveau deux des trois bits précédents auxquels on ajoutera un troisième, et ainsi de suite.

Une fois extraite l'information sur le voisinage d'un pixel, la lecture de la table ne pose aucun problème (voir parag. 5.2.1). Il ne s'agit que de l'adressage d'un bloc mémoire.

2.5.3.4. Variantes d'implémentation logicielle

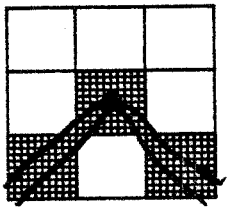
Nous allons détailler ici deux méthodes différentes d'implémentation des procédures **détermine et stocke** non explicitées au paragraphe 5.3.2.. La première est utile quand le propos n'est que de coder et de décoder. La deuxième permet de manipuler les contours présents dans l'image.

Dans la plupart des cas, le stockage se fait en deux étapes. Un stockage dynamique reçoit des données de l'analyse qu'il assemble. Lorsque certaines conditions sont remplies il passe une partie des données qu'il a accumulées à une procédure de stockage statique ou à une procédure de traitement. Dans l'exemple 5.3.4.2 , chaque fois qu'une composante connexe est constituée, son code est passé à la procédure statique.

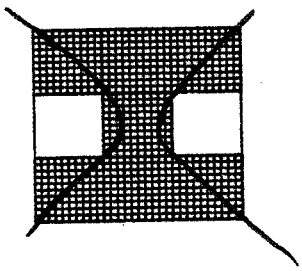
2.5.3.4.1. Implémentation sans structuration du stockage

détermine:

Cette procédure fournira les caractéristiques suivantes: le point est-il un point de naissance? simple ou double? quels liens partent de ce point? le point est_il un point de mort? simple ou double?.



Exemple de point de naissance double: 4 parois naissent



Exemple de point où passent deux parois

Figure 2.10.

stocke:

Met à jour deux listes séquentielles; l'une est la liste des points de naissance, chaque fois qu'un point de naissance est détecté, ses coordonnées sont ajoutées à la fin de cette liste; l'autre est la liste des liens, chaque fois qu'un ou des liens sont détectés en un point ils sont ajoutés à la fin de cette liste dans l'ordre 3,2,1,0; cet ordre correspond à l'ordre de rencontre des parois sur la ligne de balayage: de la plus à gauche à la plus à droite.

De plus la rencontre d'un point de mort doit conduire soit au marquage de ce point dans une liste, soit, ce qui est plus économique, à l'ajout dans la liste des points de deux liens conduisant à une configuration impossible de parois; nous avons choisi d'introduire dans la liste de liens les liens 1 et 3 dans cet ordre, ce qui correspond à un croisement entre la paroi de gauche et celle de droite.

Le décodage proposé est assez voisin de la technique utilisée par les codes par plage bidimensionnels. Pour générer une ligne, on utilise une ligne préparée lors de la génération de la ligne du dessus; cette ligne indique en chaque point combien de parois arrivent de la ligne du dessus. En chaque point:

- on regarde si le point est le premier non encore consommé dans la liste de points de naissance; si oui, on le retire de la liste, on lit les deux premiers liens non encore consommés dans la liste de liens; on commence la construction d'après ces liens de deux parois qui pendent vers la ligne du dessous.

- on consomme autant de liens qu'il y a de parois pendantes arrivant en ce point; si ces liens correspondent à une configuration impossible, on ne génére pas de pendantes vers la ligne suivante (point de mort).

2.5.3.4.2. Implémentation rendant possible la manipulation des contours:

Les principaux types utilisés:

```
point =record ligne,colonne: entier;end;

chainage=record
            valeur_chainon:0..3;
            suite_chainage:^chainage;
         end;

tête.de.paroi=record
                  tête_paroi_suivante :^tête_de_paroi;
                  tête_paroi_précédente:^tête_de_paroi;
                  pt_de_naissance :point;
                  début_chainage :^chainage;
                  fin_chainage :^chainage;
                  num_contour :^entier;
               end;
```

Ce type permet de définir une liste de parois ordonnée suivant la ligne de balayage; la première paroi est considérée comme étant la suivante de la dernière.(fig. 2.11.).

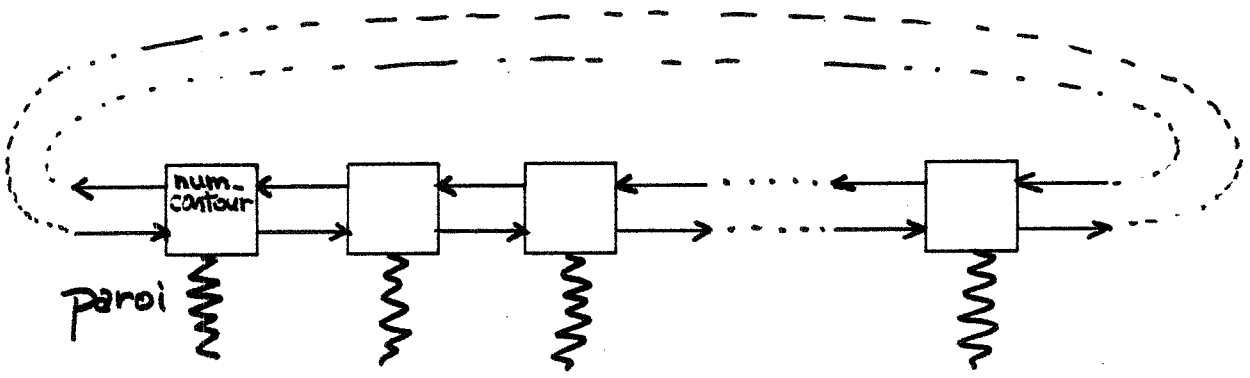
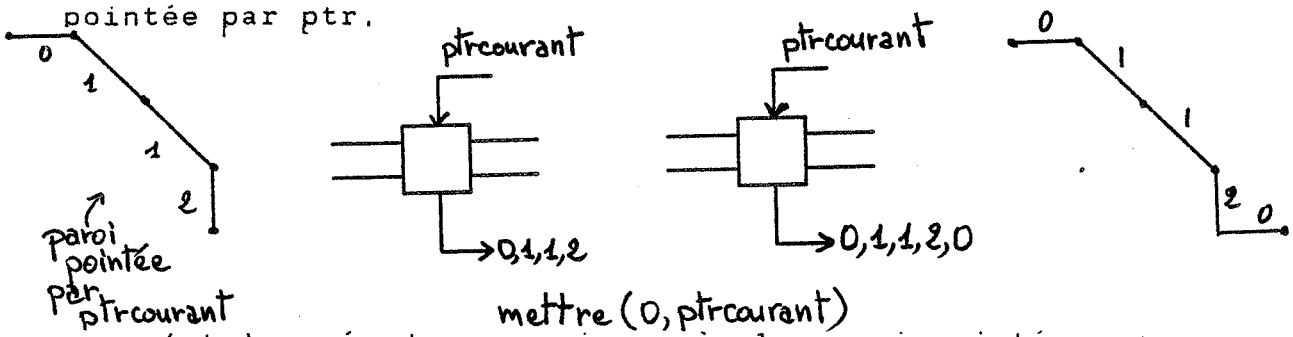


Figure 2.11.

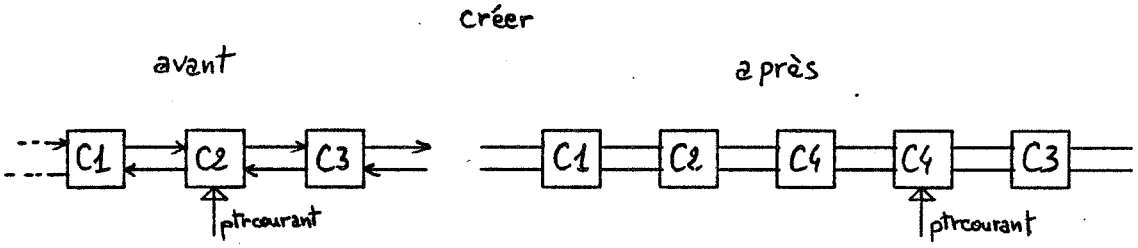
La procédure utilise la variable "courante" :
courante : ^tête.de.paroi;
Ce pointeur désigne la paroi à laquelle appartient le prochain lien trouvé.

Nous utilisons trois procédures de base sur cette structure de donnée; ces procédures ne seront pas détaillées ici, il suffit de savoir que:

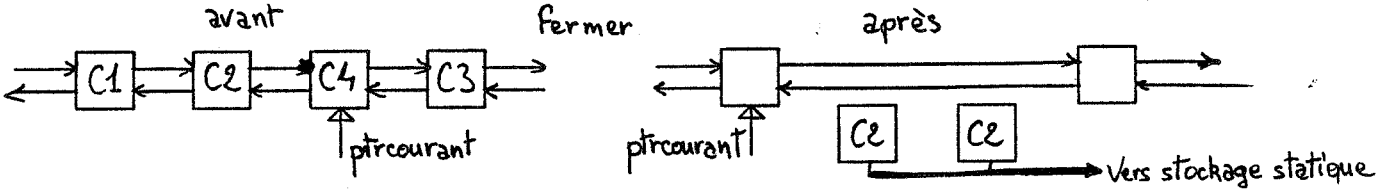
mettre(lien,ptr): met le lien au bout de la paroi pointée par ptr.



creer(ptr): crée deux parois après la paroi pointée par ptr et fait pointer ptr sur la derniere créée; de plus, attribue aux entiers ^num_contour des deux parois une valeur commune non encore attribuée à une autre paroi,



fermer(ptr): remonte ptr de deux parois,puis ferme les deux parois suivant la nouvelle paroi pointée par ptr; de plus, rend égaux les entiers ^num_contour des deux contours, afin de pouvoir les identifier par la suite comme appartenant à un même contour



Nous composerons les procédures d'un niveau supérieur suivantes:

```
p1:courante:=courante^.suivante;mettre(1,courante);
p2:courante:=courante^.suivante;mettre(2,courante);
p3:courante:=courante^.suivante;mettre(3,courante);
p4:mettre(0,courante);
p5:mettre(0,courante^.precedente);
p6:fermer(courante);
p7:creer(courante);
```

Au niveau encore supérieur on peut établir une table d'actions qui fait correspondre à chaque configuration une des 53 actions listées ci-dessous.

Il suffit d'observer les 256 configurations possibles du voisinage par 8-connexité d'un point appartenant à une tache pour établir la table. Il faut lister les caractéristiques de chaque configuration. Les caractéristiques utiles ici sont:

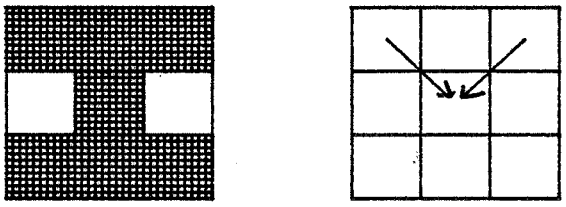
point de naissance? simple ou double?

lien(s) entrant(s) sur le point?

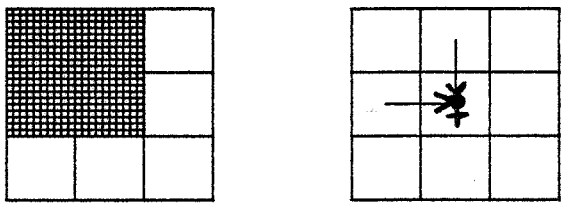
point de mort? simple ou double?

Pour générer cette table automatiquement, on pourra utilement se référer à [CED79]; Cederberg décrit des masques permettant de trouver les caractéristiques d'un point. Il utilise ces masques en chaque point et donc réalise à chaque fois un nombre non négligeable d'opérations booléennes pour trouver les caractéristiques d'un point. La table proposée ici, calculée une fois pour toute, limite notablement les calculs et se prête bien à une réalisation matérielle.

Par exemple la configuration 11 (cf.figure 2.12) correspond à l'action 35 (p4;p2;p6) puisqu'arrivent les liens 0 et 2, et que le point est un point de mort.



Configuration : 11
Action : 35



Configuration : 231
Action : 41

Figure 2.12: Exemples de configurations.

Les 53 actions tabulées sont:

```
  action 00:;
  action 01:p1;p1;p6;
  action
02:p1;p1;p3;p3;p6;p6;
  action 03:p4;p5;p6;
  action 04:p4;p1;p6;
  action
05:p4;p5;p3;p3;p6;p6;
  action
06:p4;p1;p3;p3;p6;p6;
  action 07:p7;
  action 08:p1;p1;
  action 09:p2;
  action 10:p4;p1;
  action 11:p4;
  action 12:p4;p5;
  action 13:p1;p1;p3;p6;
  action 14:p4;p5;p3;p6;
  action 15:p4;p1;p3;p6;
  action
16:p1;p1;p3;p6;p3;
  action 17:p4;p3;p6;p3;
  action 18:p1;p3;p6;p3;
  action 19:p7;p7;
  action 20:p1;p7;p1;
  action 21:p1;p7;
  action
22:p1;p1;p3;p6;p7;
  action 23:p4;p7;
  action 24:p4;p3;p6;p7;
  action 25:p1;p3;p6;p7;
  action
26:p4;p5;p3;p6;p3;
  action
27:p4;p1;p3;p6;p3;
  action
28:p1;p1;p3;p6;p7;p3;
  action
29:p1;p3;p6;p7;p3;
  action 30:p2;p2;p6;
  action 31:p1;p2;p6;
  action 32:p3;p3;p6;
  action 33:p2;p3;p6;
  action 34:p1;p3;p6;
  action 35:p4;p2;p6;
  action 36:p4;p3;p6;
  action 37:p2;p2;
  action 38:p1;p2;
  action 39:p3;p3;
  action 40:p2;p3;
  action 41:p1;p3;
  action 42:p1;
  action 43:p3;
  action 44:p4;p2;
  action 45:p4;p3;
  action 46:p2;p7;p2;
  action 47:p1;p7;p2;
  action 48:p3;p7;p3;
  action 49:p2;p7;p3;
  action 50:p1;p7;p3;
  action 51:p2;p7;
  action 52:p3;p7;
```

La logique de ces actions est dictée par l'ordre sur les parois imposé par la ligne de balayage. En un point on peut considérer que l'ordre sur les parois passant par ce point est imposé par la cohérence verticale entre les lignes.

Des extensions à ces algorithmes permettent d'établir l'arbre d'inclusion des contours de l'image. Leur utilisation pour le dessin animé est sans doute une voie intermédiaire entre le vectogramme et le pictogramme qu'il sera intéressant d'explorer. En effet, ce type de représentation devrait permettre de tirer parti des avantages des deux représentations. Nous ferons quelques remarques supplémentaires sur cette possibilité au cours du chapitre suivant. Dans cette thèse, nous avons développé les techniques associées à une représentation des dessins analogue à celle produite par une table à numériser. Le paragraphe suivant montre comment transformer les parois décrites sous forme de liens en des suites de segments de droite définis par leurs extrêmités dans le plan.

2.5.4. VECTORISATION

> Le problème est de passer de la représentation codée des contours à des lignes brisées représentant le dessin à traiter.

La solution retenue en première approche est la transformation brutale de chaque lien du code en un segment, puis le traitement de ces segments pour en réduire le nombre. Nous nous ramenons alors aux problèmes de filtrage de lignes brisées traités au paragraphe 6 de ce chapitre. Si on a adopté la méthode de codage fournissant les contours, chacun de ceux-ci est connu par une succession de liens qu'il est facile de transformer en une ligne brisée.

Pour une exploitation intensive de ce code, il serait judicieux de mettre en oeuvre un algorithme construisant des segments avec les liens successifs. Il est possible de construire des segments redonnant exactement les contours initiaux par un algorithme donné de tracé de segment [NEW79]. Cette construction pourrait être réalisée au fur et à mesure du codage.

Il est assez simple de générer un segment unique à partir d'une liste de segments colinéaires successifs. De plus, dans notre cas, les segments élémentaires n'ont que quatre directions possibles, ce qui simplifie les traitements.

Epaisseur des traits et vectorisation

Par construction du code, il apparaît que les traits d'un dessin sont représentés par le code de toutes les frontières entre le trait noir et le fond blanc. Les épaisseurs des traits sont donc représentées. Un trait correspond a une zone filiforme entre deux contours codés (fig. 2.13.).

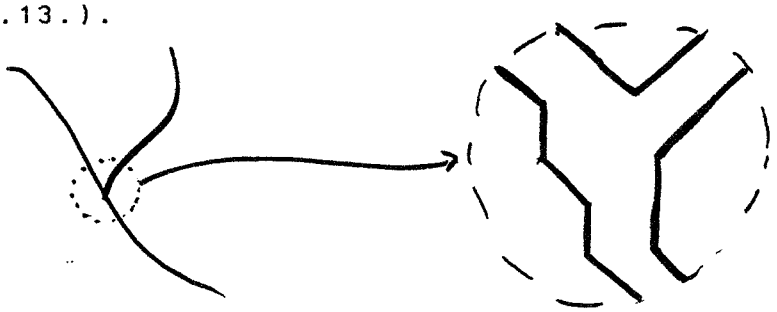


Figure 2.13: traits d'un dessin et contours codés

Une fois les contours vectorisés, tout filtrage est donc susceptible de modifier l'épaisseur des traits.

2.5.5. QUELQUES REMARQUES

La planche 2.1 donne quelques résultats relatifs au codage d'image obtenu par l'algorithme le plus simple (cf. 5.3.4.1).

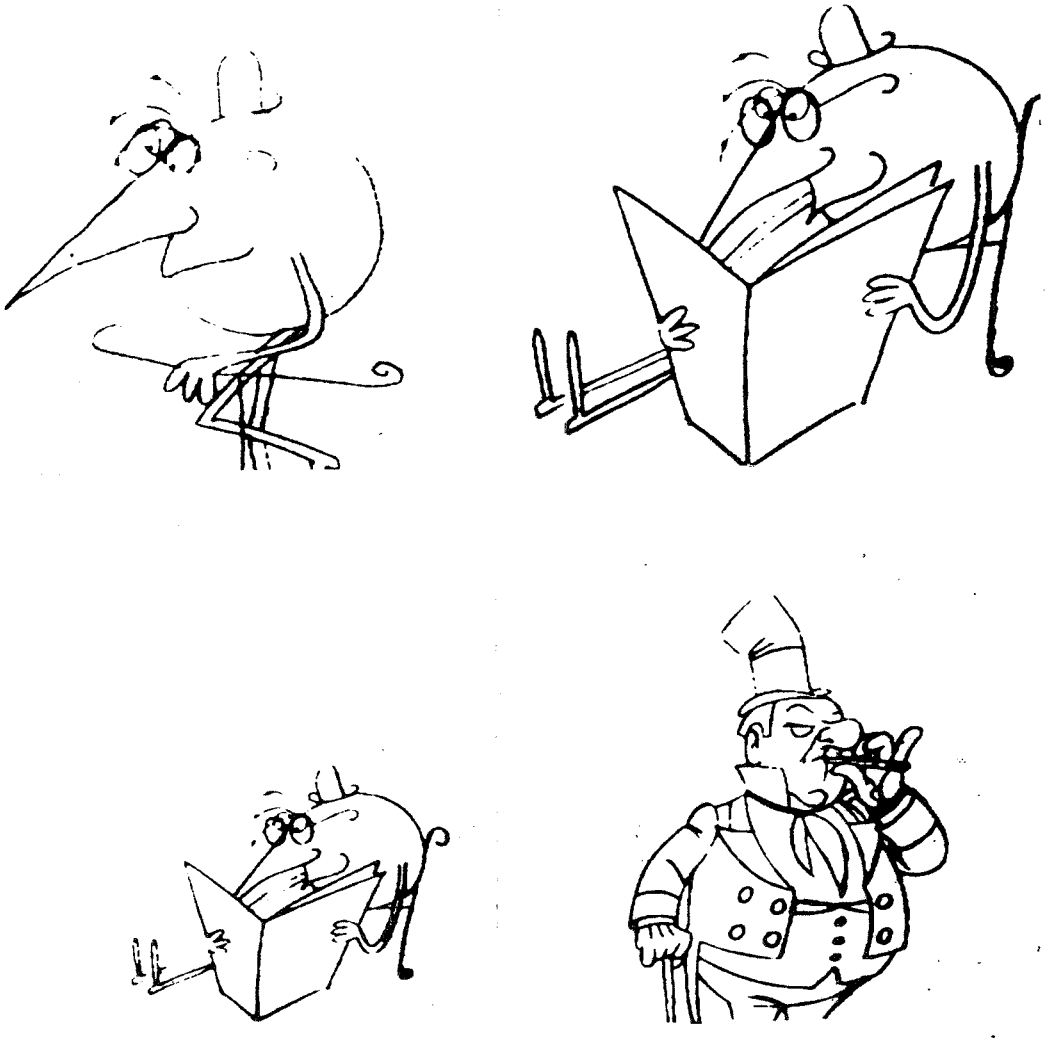


photo 1: marche 692 pts.de naissance,8100 liens, 4 Ko
photo 2: lit gros plan 1183 pts.de naissance,10641 liens, 6 Ko
photo 3: lit plan normal 639 pts.de naissance,6672 liens, 4 Ko
photo 4: magoo 538 pts.de naissance,10579 liens, 4 Ko

Planche 2.1.

Pour les images saisies par caméra, avec une résolution de 512x512, nous avons obtenu des codes occupant entre deux et six kilo-octets. Ils étaient composés d'un nombre de chaînons de l'ordre de 10000 et d'un nombre de points de naissances de l'ordre de 1000.

Nous avons obtenu des gains appréciables en ne stockant que les différences entre coordonnées de points de naissance et les différences entre chaînons successifs. Nous ne nous étendrons pas sur les techniques désormais classique d'optimisation des codages et renverrons à l'abondante littérature sur les codages: codage différentiel des points de naissance, codage différentiel des liens, code de Huffman ([IEE80], [CED81], ...).

Une caméra fournissant 512x512 pixels a été utilisée pour les tests. Ils ont montré l'intérêt de la technique de codage proposée, du fait de la compacité et de la facilité de manipulations ultérieures du code obtenu. Ils ont aussi révélé les faiblesses de cette technique de codage associée à une saisie caméra. A ce niveau de résolution, le crénelage des traits a une influence importante sur l'esthétique du résultat; seuls certains styles de dessin n'en sont pas trop affectés. De plus nous avons constaté que les épaisseurs de trait obtenues sont très sensibles aux conditions d'éclairement et aux caractéristiques de l'original (régularité du tracé...).

Nous proposons comme solution d'utiliser un numériseur à barre de diodes (scanner). Ainsi les problèmes d'éclairement s'estompent. La résolution obtenue (1768x2200) permet le filtrage pour un affichage vidéo anticrénelé (575x768). La qualité de l'image obtenue est alors comparable à celle obtenue par Stern [STE79] et utilisée par Wallace [WAL81] qui effectuent leur saisies par caméra à une résolution de 480x640 pixels en conservant 4 à 8 niveaux de gris pour les traits. L'avantage de la saisie par scanner est la possibilité, démontrée dans ce qui précède, de produire un vectogramme.

Le code proposé est en gros proportionnel à la longueur des frontières entre zones blanches et noires de l'image. Nous n'avons pas pu faire d'essai avec un numériseur à barre de diodes de résolution 2200x1768, mais on peut estimer à environ 4 fois plus la place prise par le codage d'une image à cette résolution. Il est possible de réduire cette place en ne codant les dessins qu'à l'intérieur de leur boîte englobante parallèle aux axes. Ce principe est utilisé par les systèmes de dessin animé représentant les

images par des pictogrammes et pour qui le problème de place est encore plus important [WAL81].

2.5.6. CONCLUSION

Nous avons décrit une méthode de codage d'images binaires en un balayage permettant la manipulation ultérieure des contours. En dehors de son utilisation évidente pour le stockage d'images sous forme compacte, ce codage pourra être utilisé comme pré-processeur d'une vectorisation ou d'une analyse.

Nous l'avons utilisé pour obtenir automatiquement une représentation des dessins sous forme de vectogrammes. Dans la suite, nous envisageons les traitements permis par cette représentation.

2.6. FILTRAGE D'UN VECTOGRAMME

> Les dessins étant connus sous forme de vectogramme, nous étudions ici comment réduire le nombre de points qui les composent. Les filtrages étudiés doivent aboutir à stocker la quantité de points minimum. Il faut cependant garder assez de points pour ne pas nuire à la qualité graphique du dessin finalement filmé.

La méthode de saisie peut conduire à la production d'un nombre exagéré de segments pour décrire le dessin. Par exemple, la saisie sur une table à numériser peut produire un grand nombre de points confondus aux endroits où l'utilisateur marque un temps d'hésitation. Une sauvegarde sans précaution de cette saisie conduira à un encombrement inutile des espaces de stockage.

Nous avons étudié et testé diverses méthodes de réduction de cette quantité de données. Les critères de choix entre les méthodes ont été:

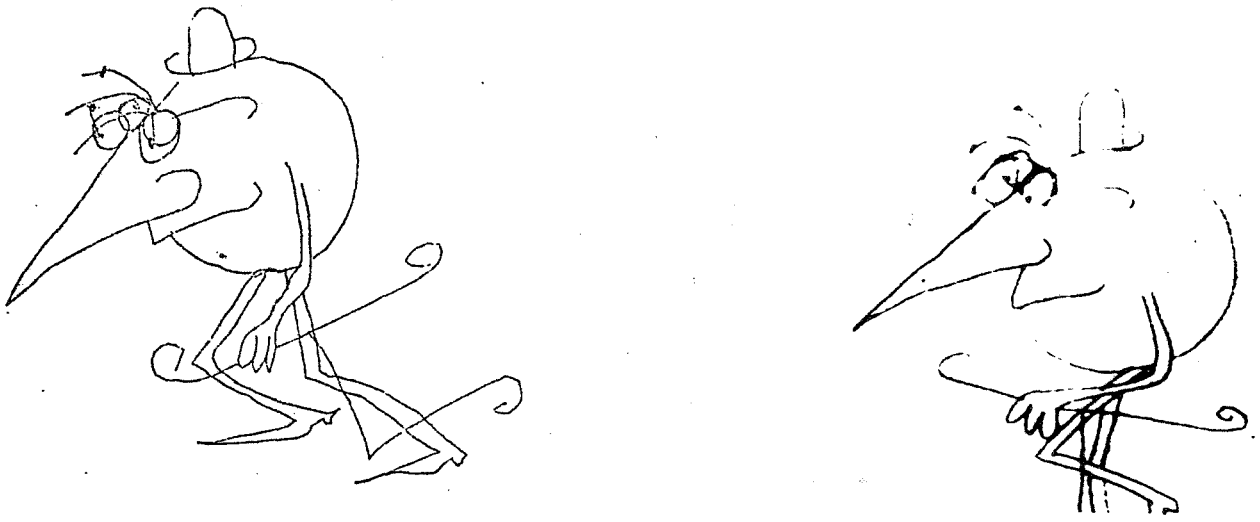
- d'une part de ne pas altérer la qualité graphique de l'image restituée,

- d'autre part de donner les résultats attendus rapidement sans réglage de paramètres quelle que soit l'image traitée . Ainsi, on peut prévoir d'utiliser ces méthodes sur une grande quantité de saisies sans intervention humaine.

Que le tracé soit courbe ou rectiligne, que le déplacement de la main de l'opérateur soit rapide ou lent, la table à numériser génère le même nombre de points par unité de temps. Un grand nombre de petits segments peuvent par exemple être générés pour représenter un trait droit en apparence à la résolution d'affichage. Il sera bon de remplacer tous ces segments par un seul.

La vectorisation du code décrit précédemment fournit une représentation des dessins par un grand nombre de segments courts. Le filtrage de ces derniers permet une réduction du stockage nécessaire et du volume des données manipulées lors des traitements.

Les exemples de la figure 2.14 sont représentatifs des essais que nous avons effectués. Le dessin saisi sur table à numériser comporte 1127 points et environ le même nombre d'arêtes; filtré, il ne comporte plus que 347 points (cf.ch2 parag.6). Le dessin codé à partir d'une saisie par caméra, en 80 secondes sur notre configuration, comporte 394 points de naissance et 6449 liens pour une résolution de 512x512 pixels.



Saisie: table à numériser Saisie: caméra

Figure 2.14

D'autre part il est possible de caler tous les points sur une grille, fine par rapport à la grille d'affichage, de telle sorte que le tracé apparent ne soit pas modifié, mais que les points très voisins soient réduits à un seul.

Les différents critères ci-dessus ont orienté nos recherches vers les nombreux travaux publiés sur la représentation de tracés numérisés sous forme de courbes évoluées (splines, arcs de cercles...) [PAV82], [LOZ83], [PLA83].

Ces travaux présentent des méthodes pour remplacer une ligne brisée composée de segments de droite par une ligne brisée composée d'arcs de courbe. On impose aux pictogrammes obtenus par affichage de chacune des deux représentations précédentes d'être identiques à une certaine résolution. La représentation par arc de courbes est en principe plus compacte et permet d'afficher le dessin à des échelles différentes en conservant un aspect lisse aux tracés.

Nous avons adapté la méthode décrite dans [LOZ83]; elle permet de transformer une ligne brisée composée de segments en une ligne brisée composée d'arcs de B-spline. Elle se décompose en quatre phases:

- réduire le nombre de points initiaux (élimination du "bruit" de saisie),
- trouver les points anguleux servant d'extrêmités aux arcs de cubiques qu'on va construire,
- trouver les points du tracé par où on imposera à la cubique de passer,
- calculer les points de contrôle de la cubique ainsi définie; ces points serviront au stockage du dessin sous forme d'arcs de cubique.

Les trois premières étapes nécessitent la donnée de paramètres qui conditionnent la précision du résultat final. Nous nous sommes heurté à la difficulté de trouver des paramètres remplissant les conditions suivantes:

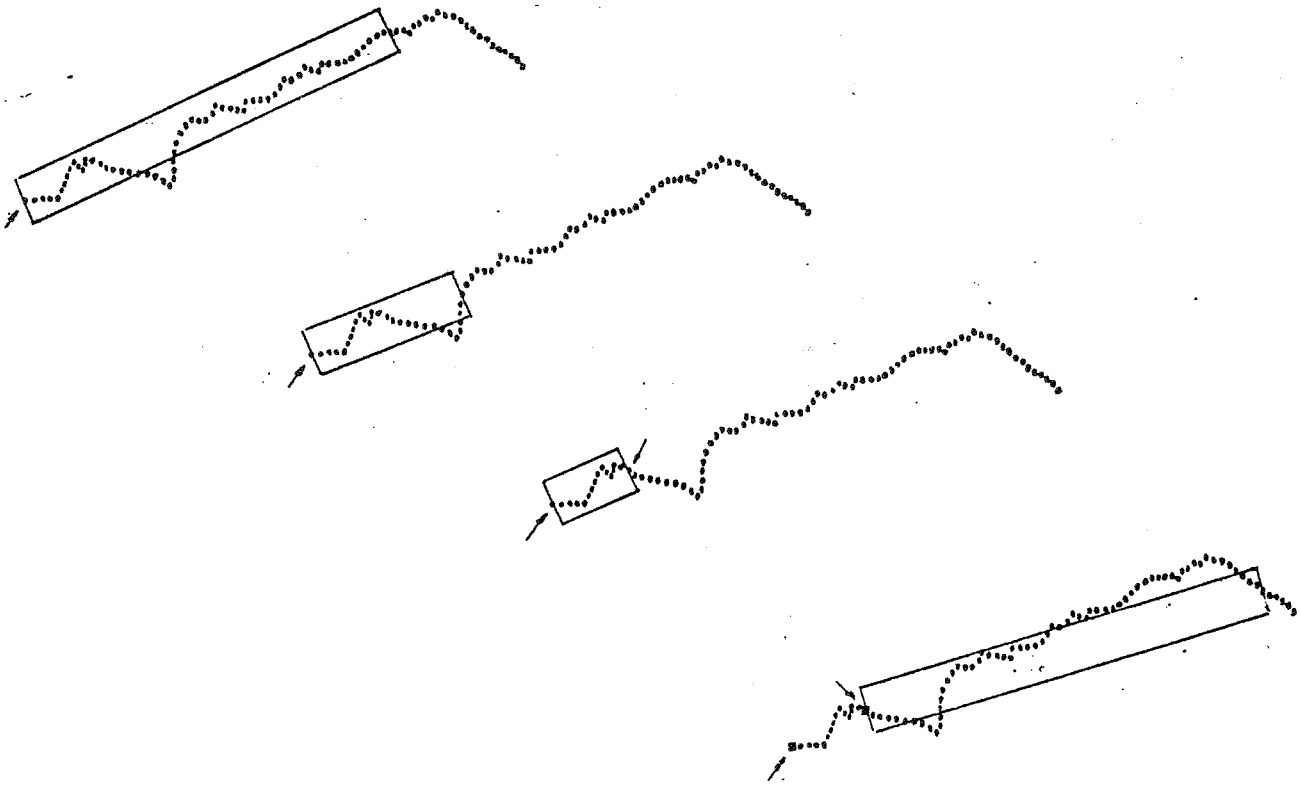
- donner moins de points de contrôle à stocker que de points dans le tracé initial,
- fournir une représentation du dessin à peu près identique au tracé initial à la résolution normale d'affichage,
- donner les résultats précédents quel que soit le tracé initial.

Le calcul des paramètres pour chaque dessin nous a paru trop coûteux. Ce problème est mentionné dans [PLA83] et [NEW79].

En conséquence, nous nous sommes contenté du filtrage obtenu grâce à la première étape. On filtre en fabriquant une liste de N points (où N est une puissance de deux), en construisant un rectangle centré sur la ligne joignant les deux points extrêmes et dont la largeur est le paramètre de réglage du filtrage; si un des points intermédiaires est hors de la boîte on divise la liste de points par deux et on réitère, sinon on mémorise le dernier point de la liste, puis on construit une nouvelle liste à partir de ce point et on réitère (fig. 2.15.).

Réduction du nombre de points d'une ligne brisée; les flèches indiquent les points conservés (fig.d'après [LOZ83]).

Figure 2.15.



Algorithme de filtrage

Nous proposons ici une représentation simplifiée de l'algorithme utilisé, la méthode ci-dessous ne permet de filtrer qu'une ligne brisée.

```
type
  point=record typ:char;x,y:real;end;
```

function nouveau_point:point;

Renvoie le premier point non encore utilisé de la ligne brisé; met à vrai la variable globale booléenne **fin_entrée** s'il n'y a plus de point après le point envoyé.

procedure émet(pt:point);

Envoie pt comme point suivant de ligne brisée filtrée.

procedure filtre;

```
const imax=6;
      deux_puis_imax=64;   var   largeursur2   :real;   (*
demi-largeur de la boite de test *)
    reponse :boolean;
    iptf,i:integer;
    maxlu :integer; (* nombres de points utiles dans la
table *)
    tab :array[1..1024] of pt_benson_bin;
```

function dans(pt1,ptf,pt:pt_benson_bin):boolean;

Cette fonction indique si le point pt est contenu dans le rectangle centré sur la droite pt1,ptf et dont la largeur divisee par deux est une variable globale du programme (largeurdiv2).

```
begin (* filtre *)
  writeln('largeur du filtre:');readln(largeursur2);
  (* lectures de points pour remplir la table depuis le
debut *)
  iptf:=deux_puis_imax;
  i:=0;
  while ((not fin_entree) and (i<iptf) ) do
  begin
    i:=i+1;
    tab[i]:=nouveau_point;
  end;
  iptf:=i; maxlu:=i;
  émet(tab[1]);
```

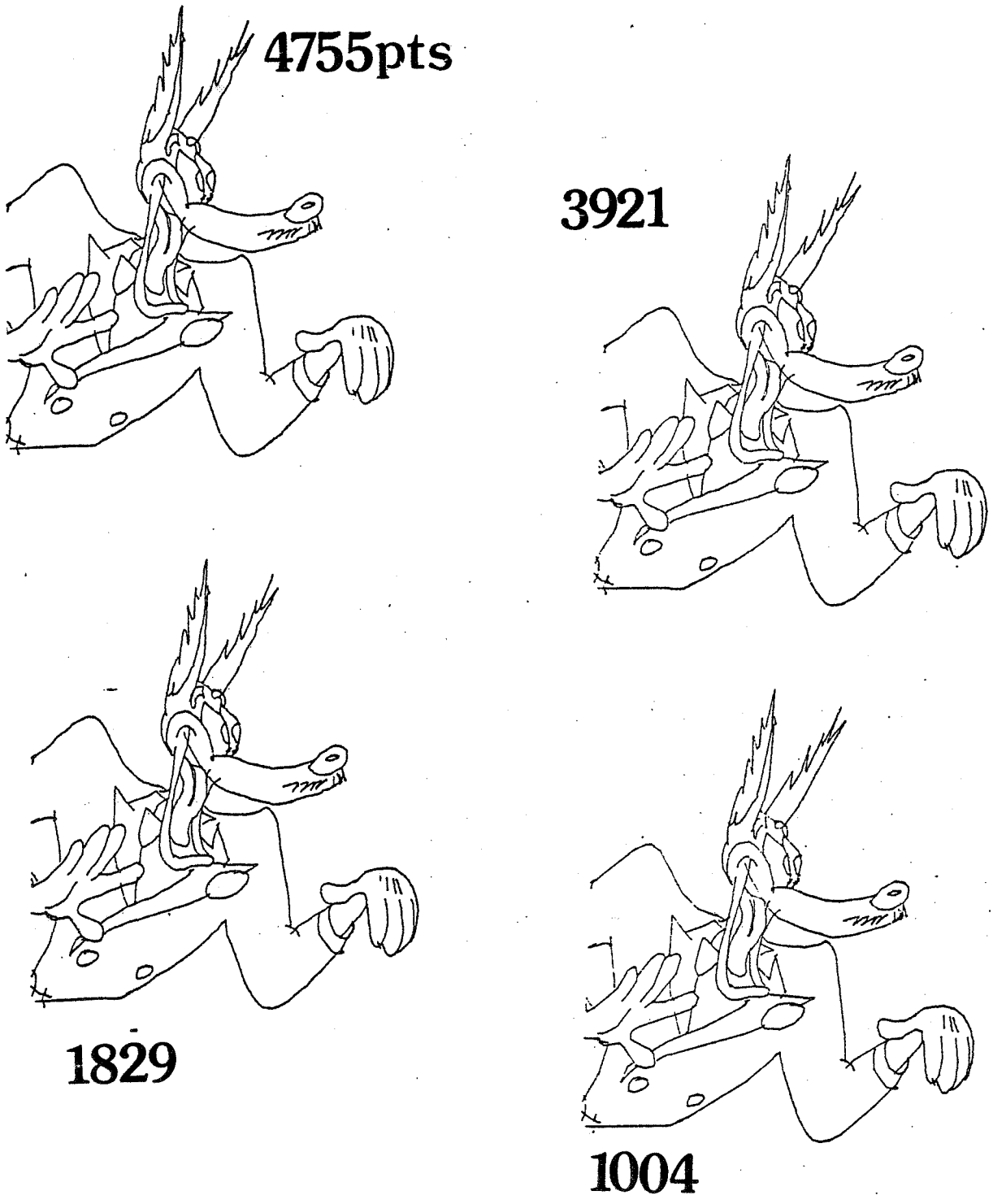
```
  repeat
    iptf:=iptf*2;
    repeat

(* pour tous les points entre le premier et le
iptf-ieme on regarde s'ils sont dans la boite centrée
sur les deux points extrêmes; si un point n'y est pas
on réduit la longueur de la boite de moitié et on
recommence, sinon le test de boite est terminé *)
        iptf:=iptf div 2;
        reponse:=true;
        for i:=2 to iptf-1 do
             reponse:=dans(tab[1],tab[iptf],tab[i])
                                          and reponse;
    until reponse;
    (* on sauve le dernier point de la boite *)
    with tab[iptf] do if (x<>tab[1].x) or (y<>tab[1].y)
then
                                     émet(tab[iptf]);

(* on prépare une nouvelle liste de points qui commence
par le dernier point conservé; pour cela on décale tous
les points à partir de lui vers le début de la liste *)
    if iptf<>0 then
       for i:=iptf to maxlu do tab[i-iptf+1]:=tab[i];
    (* on complete la liste dans la mesure du possible
    par des points lus dans le fichier source *)
    if maxlu>1 then i:=maxlu-iptf+1 else i:=1;
    maxlu:=i;
    iptf:=deux_puis_imax;
    while ((not fin_entree) and (i<iptf) ) do
    begin
      i:=i+1;
      nouveau_point;
    end;
    maxlu:=i; iptf:=maxlu;
  until maxlu<=1;
end; (* filtre *)
```

La figure 2.16. présente les résultats obtenus en filtrant un vectogramme saisi sur une table à numériser. Les dessins filtrés sont reproduits sur un traceur au format A4.

Figure 2.16.



Les saisies que nous avons effectuées sur une table à numériser nous ont toutes donné entre 500 et 5000 points. Un filtrage, sans dégradation apparente du dessin reproduit sur une mémoire d'image de résolution 512x640 pixels, ramène une saisie de 4000 points à environ 1000 points en 15 secondes de filtrage, lectures et écritures sur disque comprises. Le calage des points sur une grille de coordonnées entières comprises entre 0 et 5000 diminue encore légèrement le nombre de points sans diminuer la qualité. Les résolutions admises pour les images cinématographiques varient suivant les auteurs, mais sont toujours inférieures à 10000x10000.

Chaque point occupe actuellement 5 octets:

- 2 octets pour chacune de ses coordonnées

- 1 octet pour le marqueur.

Une image de 1000 points occupe donc 5 kilo-octets.

Le marqueur actuellement utilisé peut prendre deux valeurs : plume haute ou plume basse (cf parag.3). Il serait possible de lui faire prendre d'autres valeurs. Par exemple, en marquant des points comme étant "transparents": les segments arrivant sur ces points seront pris en compte lors des traitements (cf chap.3), mais ne seront pas tracés sur l'image finale. Le marqueur permet aussi d'indiquer une épaisseur de trait pour chaque segment. En fait, il pourra porter toute information supplémentaire à donner aux segments. Suivant l'application, on sera donc amené à le stocker sur un nombre variable d'octets.

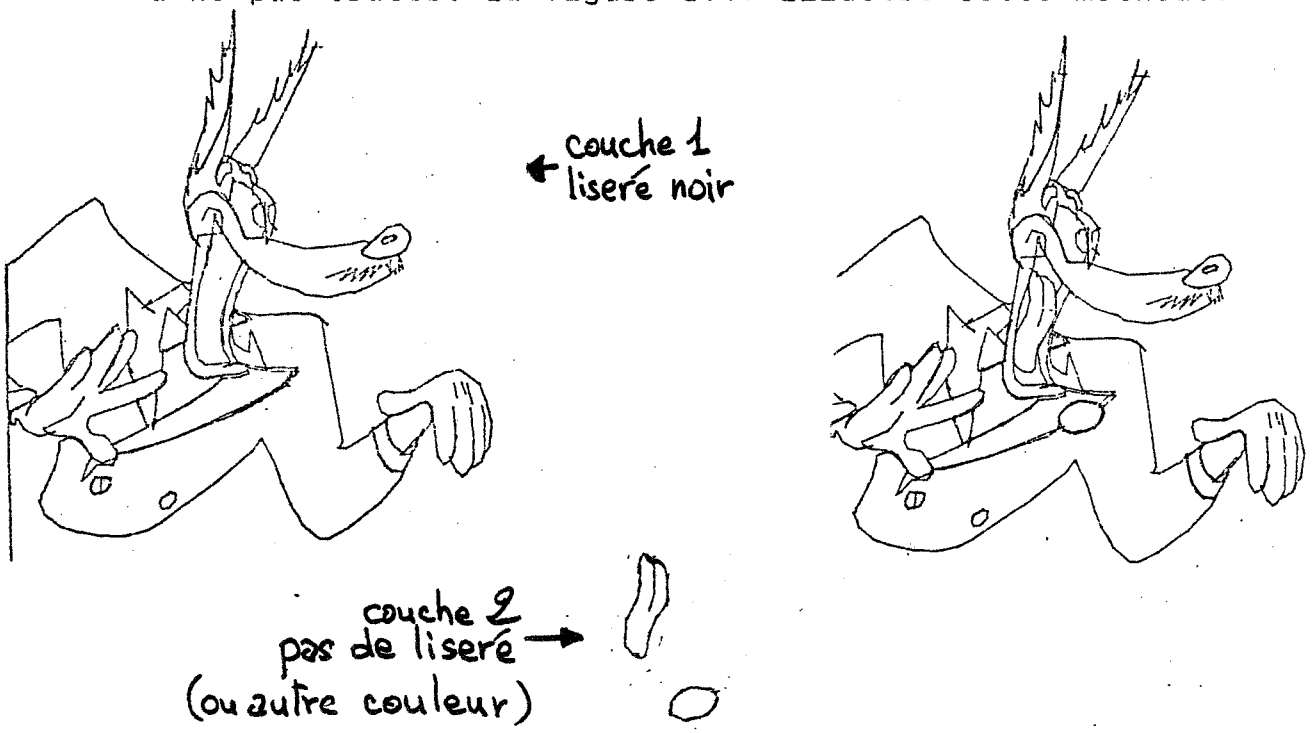
2.7. *PROBLEMES LATENTS*

Les problèmes que nous mentionnons ci-dessous sont des problèmes communs à tous les systèmes d'aide à la réalisation de dessins animés actuellement proposés et basés sur la préparation classique de l'animation.

Liseré de couleur

Le plus important de ceux-ci nous semble l'absence de prise en compte à la saisie d'une caractéristique commune à la plupart des images de dessin animé que nous avons observées : toutes ces images présentaient des zones de couleur dont la frontière était soulignée par un liseré de couleur généralement uniforme sur toute l'image et quelques zones sans liseré.

On voit qu'il est alors nécessaire de distinguer la saisie d'au moins deux types de traits: ceux qui seront tracés, distingués éventuellement par leur couleur de traçage et ceux qui ne le seront pas. Pour les saisies interactives sur table à numériser, il est simple de faire indiquer à l'opérateur de quel type sont les traits qu'il saisit. Pour les saisies automatiques: caméra, scanner, le problème est plus délicat. En attendant d'être en mesure de distinguer automatiquement les deux types de traits, par exemple par leur couleur sur l'original, nous proposons une solution influant sur la préparation du travail: on répartira les contours en deux couches; la première contiendra le dessin des contours à tracer; la deuxième se placera par dessus et contiendra les contours à ne pas tracer. La figure 2.17 illustre cette méthode.



Proposition de décomposition des dessins pour traiter le problème des contours non tracés ou de couleurs différentes.

Figure 2.17.

Dessins sans contour net

Certains types de dessins sont nécessaires pour obtenir des effets spéciaux. Leur usage est courant, bien que le nombre des dessins concernés soit petit relativement au nombre total des dessins d'un dessin animé. Il s'agit de dessins où ne figure pas de frontière nette entre des zones de couleur: fumée, véhicule en déplacement et tout mouvement rapide (fig. 2.18.).

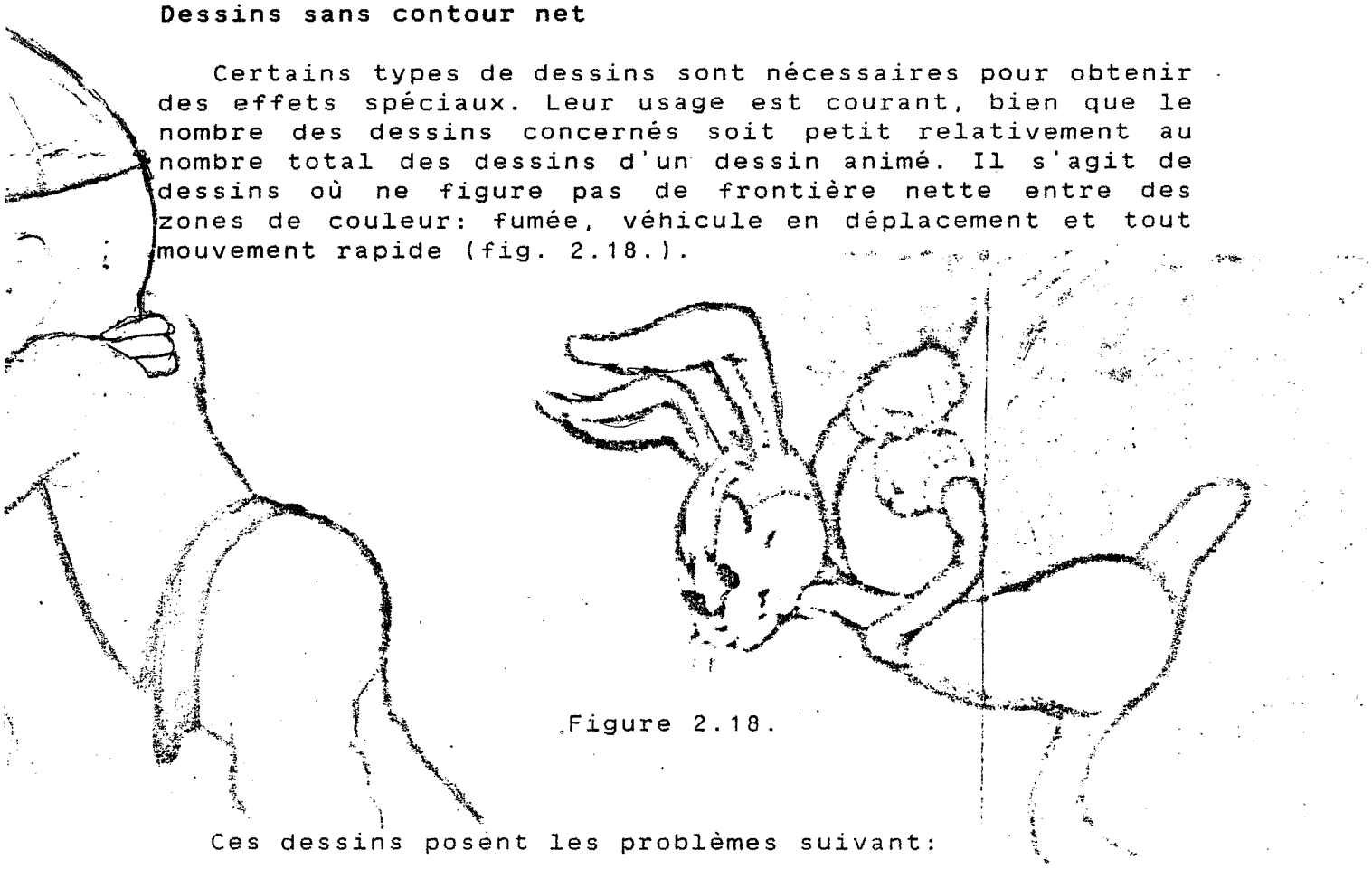


Figure 2.18.

Ces dessins posent les problèmes suivant:

film de test (linetest cf ch.3 2) le type d'effet recherché repose sur des nuances dans le dessin; si on utilise une méthode de test informatique reposant sur une représentation des images sur une mémoire d'image avec 1 bit par pixel, on ne pourra pas tester de tels effets. Voir au chapitre suivant la présentation des méthodes possibles pour le film de test.

gouachage: si le gouachage repose uniquement sur un remplissage de contour, on ne saura pas restituer ce type d'effet; noter que les systèmes actuellement commercialisés utilisent un gouachage par remplissage de contour.

2.8. COMPLEMENTS POUR LA COMPARAISON DES METHODES DE SAISIE

Une évaluation détaillée des méthodes de saisie et de représentation des données n'a de sens que relativement à un cahier des charges, précisant en particulier un budget, des contraintes de qualité ainsi que la productivité requise de la configuration mise en place.

Une telle évaluation sort du cadre de notre étude. Elle sera probablement complétée par ailleurs. Elle pourra faire appel aux techniques d'analyse de la valeur. Nous mentionnons maintenant quelques points clés en vue de sa réalisation.

Définition:

Les différentes méthodes de saisie fournissent des données décrivant numériquement les dessins avec plus ou moins de précision. On pourra choisir une méthode en fonction de la définition du résultat à obtenir, mais aussi en fonction de sa facilité de mise en oeuvre. Le vectogramme offre le maximum de précision, en acceptant de créer un vectogramme occupant la même place mémoire qu'un pictogramme, il est probable qu'on obtienne des images d'une qualité graphique équivalente; le vectogramme permet de tracer l'image à plusieurs définitions. Cependant il est plus complexe d'obtenir automatiquement la représentation d'un dessin sous forme de vectogramme.

Respect du tracé original:

Le respect du tracé original est un critère important de qualité mis en avant par les professionnels de l'animation. Il est clair que la saisie sur table à numériser ne permet pas ce respect. Celui-ci passe donc par une saisie optique de bonne définition. Les solutions par barre de diodes sont donc bien adaptées avec des définitions courantes de l'ordre de 2000 par 2000. Si on veut exploiter les dessins sous forme de vectogramme, il sera nécessaire d'améliorer les méthodes de transformation du pictogramme fourni par le scanner en vectogramme.

Possibilité d'antialiassage:

Parmi les saisies automatiques, seules les saisies par caméra ayant une trop basse résolution globale (nombre de pixels x nombre de bits par pixel) ne permettent pas un antialiassage correct. Les saisies par caméra offrant une saisie des dessins à une résolution de l'ordre de 512x512 sur plusieurs niveaux de gris permettent de conserver l'antialiassage naturel ainsi obtenu sur les traits par le mécanisme de saisie lui-même [STE79]. Les saisies courantes de type scanner fournissent une résolution de l'ordre de 2000x2000 et permettent donc d'antialiasser les traits pour un affichage vidéo de l'ordre de 500x500 par un simple filtrage. Certains scanner de haut de gamme fournissent jusqu'à 2000 par 2000 sur 256 niveaux de gris.

La résolution de saisie sur table à numériser est très supérieure à celle de tout affichage vidéo. De plus, elle fournit des segments définis géométriquement et non de façon discrète. L'antialissage dans ce cas ne pose donc pas de problème.

Automatisation envisageable:

Seules les saisies par numériseur à barre de diodes semblent pouvoir se prêter à un travail d'automatisation; elles tireront alors parti de leur analogie avec les appareils de reprographie. Les saisies par caméra pourraient éventuellement être automatisées; pour l'instant, elles sont utilisées en fonctionnement semi-automatique: le rôle de l'opérateur est seulement de placer les dessins sous la caméra. Les saisies par table à numériser semblent exclure toute automatisation.

Rapidité de la saisie:

Cet aspect dépend du degré d'automatisation qu'on aura atteint. Des saisies par caméra fournissent instantanément (1/25s) une représentation numérique de l'image à saisir. Les numériseurs par barre de diodes mettent environ 30 secondes par document à saisir. La saisie sur table à numériser prend plusieurs minutes.

Volume de stockage par dessin:

Le vectogramme semble prendre largement l'avantage de ce point de vue; sa facilité de manipulation est en parti contrebalancée par la difficulté qu'il y a à les saisir automatiquement ou semi-automatiquement. Il ne faut cependant pas perdre de vue que l'utilisation de la représentation par pictogramme conduit à utiliser des codages et des découpages des images qui sont au moins aussi coûteux en calculs et qu'on doit répéter à chaque affichage de l'image.

Evolutivité:

La représentation par vectogramme semble la plus à même de tirer partie de l'évolution des matériels d'affichage et de saisie. En effet, des dispositifs récents de saisie optique fournissent de façon intégrée une représentation de type vectogramme des contours de l'image [VIS83] Les mémoires d'image fournissent désormais couramment l'affichage très rapide de taches polygonales. Les matériels de report sur document diffusable autorisent des définitions de plus en plus élevées; ils offrent généralement aussi l'affichage de taches polygonales. Cette représentation est compatible avec la norme GKS pour l'informatique graphique [GKS84]; elle permettra donc dans un avenir proche de tirer parti des fonctionnalités des matériels intégrant cette norme.

Seule la saisie par barre de diodes semble pouvoir évoluer rapidement car elle fait appel à des technologies simples; de plus, la demande de matériels de télécopie devrait augmenter et entraîner une baisse des coûts des systèmes de saisie associés (scanner). La saisie par

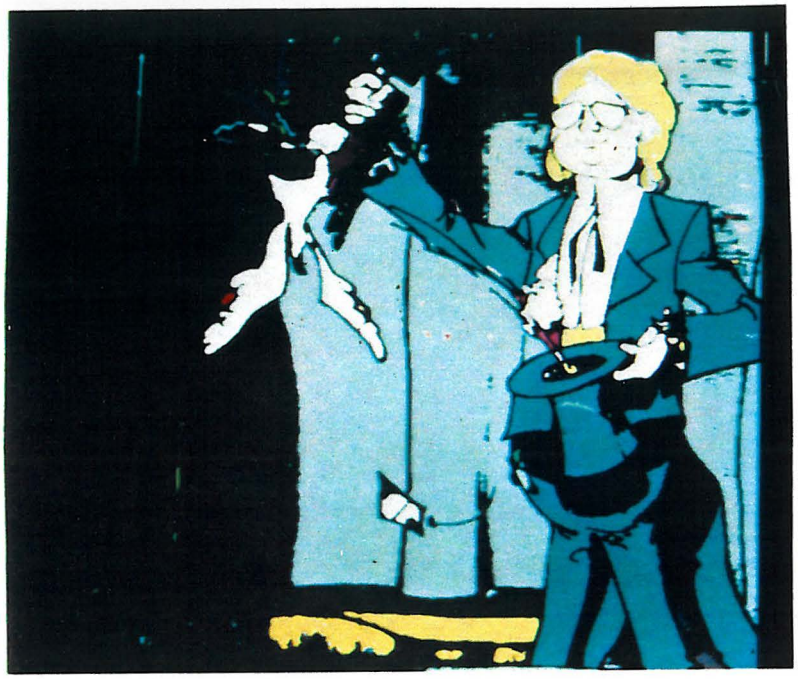
caméra fait appel à des technologies très différentes (numérisation, vidéo) et plus coûteuses.

Critère	Table à numer.	Caméra nir/blanc	Caméra 256 niv.de gris	Scanner
Définition spatiale	très bonne	basse	basse	bonne
Déf.globale spat.+coul.	bonne	basse	bonne	bonne
Manipulatio de saisie	forte	faible	faible	faible
Temps de saisie	élevé plusieurs mn	faible	faible	moyen 30 s.
Représent. obten.simpl	vecto. picto.	picto.	picto.	picto.
Volume de stockage	faible	moyen	élevé	élevé
Automatis.	difficile	possible	possible	possibl
Evolutivité e la saisie	faible	faible	faible	bonne
Evolutivité e l'utilisa tion du rés	très bonne	bonne	bonne	bonne

Une partie de cette discussion prendra tout son sens au vu des développements techniques du chapitre suivant.

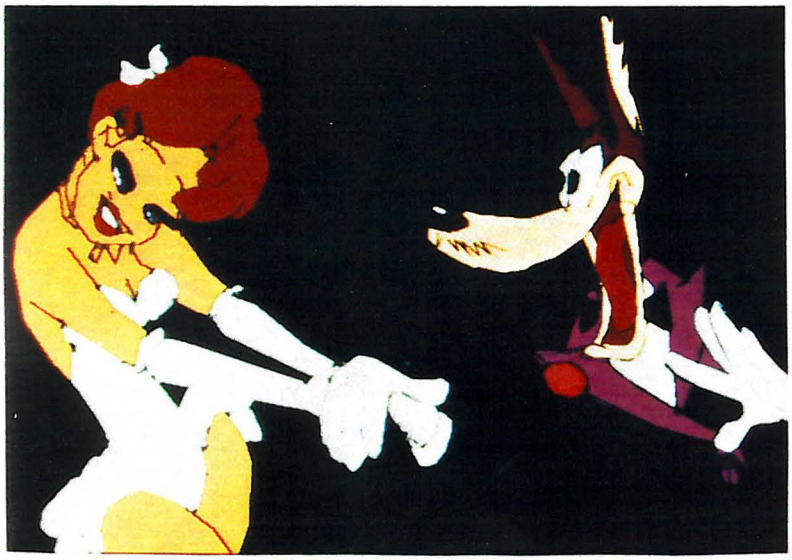
BIBLIOGRAPHIE DU CHAPITRE

[AGR77] [BAU84] [BLO82] [BUR76] [CED79] [CED81] [COU80]
[COU81] [DAN82] [FRE74] [GAN84] [GKS84] [IEE80] [KLI76]
[KNU81] [LOZ83] [LUC81] [MEY80] [MOR76] [NEW79] [ODG81]
[PAV82] [PLA83] [PRA80] [REE81] [SMI81] [STE79] [VIS83]
[WAL81] [WAL81] [YAS80]

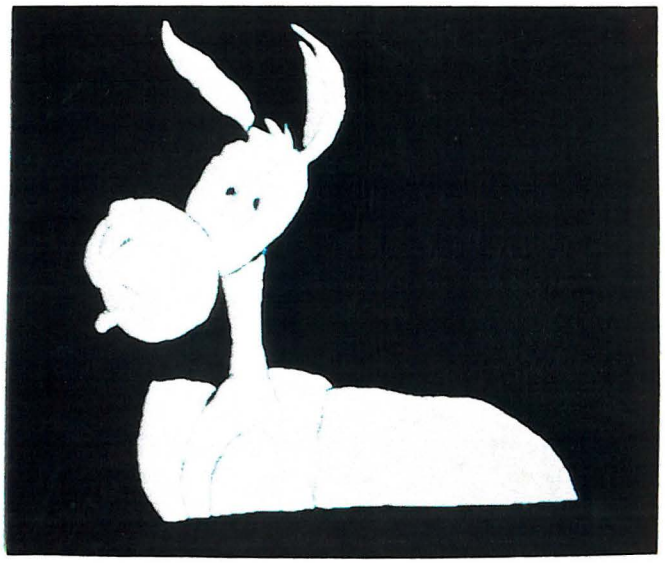


- A - gouachage et code par plage
CHAP. 2 et 3

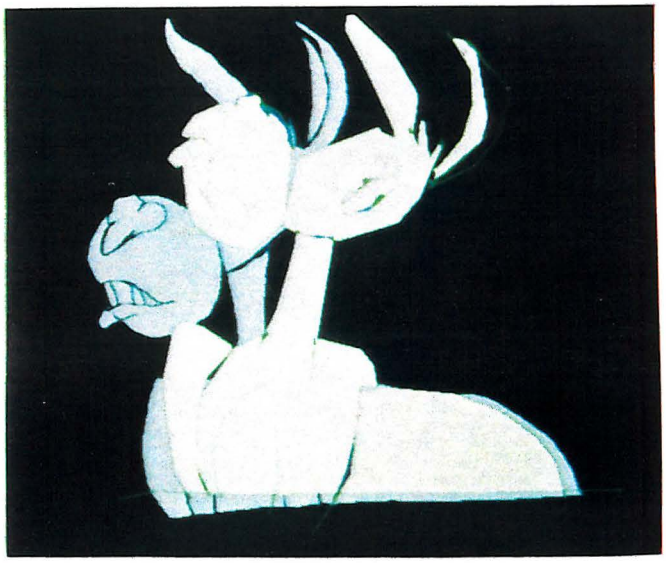
- B - gouachage - CHAP. 3

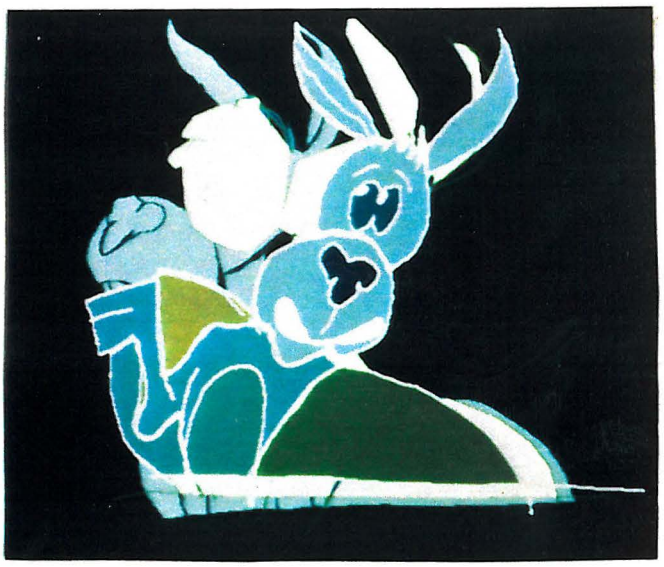


- C - assemblage



- D -





- D, E, F - propagation de la couleur dans une séquence de dessins.

CHAPITRE 3

DU DESSIN AU TRAIT
AU TOURNAGE DE L'IMAGE EN COULEUR

3.1. INTRODUCTION

Nous avons vu au chapitre précédent comment obtenir une représentation numérique des dessins au trait. Nous avons aussi vu pourquoi, parmi les différentes représentations possibles, nous avons choisi de travailler sur les vectogrammes.

Dans ce chapitre, nous étudions le passage des dessins au trait à des dessins en couleurs prêts pour le tournage. Ce qui inclut les opérations de contrôle, de mise en couleur des dessins et de composition des images à partir des dessins et des décors.

Chacune de ces opérations a des objectifs et des contraintes différentes. Pour chacune, il faut choisir une méthode appropriée d'affichage de dessin ou d'image. Nous justifierons la méthode retenue pour nos essais et envisagerons les principales alternatives.

Les méthodes de test de l'animation sur les dessins en noir et blanc sont présentées au paragraphe 2. Le paragraphe 3 montre pourquoi et comment nous avons structuré les données contenues dans les vectogrammes. Le paragraphe 5 détaille une méthode d'affichage de la structure construite. Des vérifications sur les dessins sont nécessaires avant leur mise en couleur; elles sont

présentées au paragraphe 4, tandis que les méthodes de gouachage sont présentées au paragraphe 6. Enfin, nous abordons rapidement au paragraphe 7 les problèmes informatiques liés au tournage.

Les quatre opérations essentiellement étudiées dans ce chapitre sont:

* Tests de l'animation.
définition: on utilise les dessins préparatoires pour tester l'animation qu'ils permettront d'obtenir

- méthode classique: feuilleter la pile de dessin ("flip-book) ou filmer les dessins image par image ("linetest"),
- autres méthodes utilisées: enregistrer les images sur des disques vidéo magnétiques ou utiliser des dispositifs informatiques spécialisés,
- méthodes étudiées: utilisation des représentations numériques des images sur des dispositifs informatiques standard ou spécialisés.

* Préparation des dessins pour rendre possible leur mise en couleur.
définition: passer des dessins préparatoires sur papier à leur copie utilisable par la méthode de mise en couleur choisie,

- méthode classique: traçage des dessins à l'encre de chine sur cellulo ou méthode reprographique ("Xeroxage"),
- méthode informatique actuelle: saisie numérique des dessins par caméra, obtention de leur représentation par pictogramme, affichage du dessin dans une mémoire d'image,
- méthode proposée: saisie numérique des dessins (cf.ch.2), obtention de leur représentation par vectogramme, structuration et vérification des données pour connaître les zones fermées qu'elles définissent, affichage dans une mémoire d'image.

* Mise en couleur des dessins.
définition: utilisation des dessins ainsi préparés pour attribuer une couleur à différentes zones; en général, il y a une couleur unie par zone entourée d'un trait du dessin,

- méthode classique: gouachage du cellulo avec un pinceau,
- méthode informatique actuelle: appliquer une méthode de remplissage de zone dans la mémoire d'image, donner la possibilité à un opérateur de choisir la couleur de chaque zone, sauver le pictogramme obtenu,
- méthode proposée: associer une référence à une couleur à la représentation de chaque zone fermée de l'image, dans la structure de donnée construite à l'étape précédente, sauver l'association entre la structure et les références de couleur.

* Assembler les dessins pour afficher l'image à tourner.
définition: regroupement de plusieurs dessins coloriés à l'étape précédente et du décor pour composer l'image finale,

- méthode classique: poser le décor, puis les cellulos les uns sur les autres en les calant les uns par rapport aux autres grâce à des perforations; ceci est fait sur un support de caméra nommé banc-titre,
- méthode informatique actuelle: afficher les pictogrammes dans une mémoire d'image avec une priorité d'affichage pour chacun d'entre eux,
- méthodes proposées: afficher les vectogrammes les uns sur les autres dans une mémoire d'image ou assembler les vectogrammes pour en faire un vectogramme unique ayant la même apparence puis afficher ce vectogramme; si le décor est connu sous forme de vectogramme, il est directement pris en compte dans l'opération précédente, sinon le ou les vectogrammes peuvent aisément être affichés sur un pictogramme; en production vidéo, il est même possible de mixer décor et parties animées par des méthodes vidéographiques: les parties animées assemblées sont émises par le système informatique, le décor est émis par une source vidéo quelconque, une régie vidéo permet l'assemblage des deux sources d'image.

Nous allons maintenant reprendre en détail chacun des points abordés ci-dessus.

3.2. TEST D'ANIMATION

Nous énumérons ici les méthodes actuelles et les méthodes envisageables pour tester l'animation en noir et blanc sur les dessins au trait, avant les étapes de mise en couleur et de tournage définitif.

Avant la production finale d'un dessin animé, il est courant de réaliser un test pour vérifier la qualité de l'animation. Dans le cas de l'animation traditionnelle, deux méthodes sont appliquées. L'une consiste à feuilleter rapidement la liasse composée des dessins successifs d'une séquence: c'est une méthode efficace et bon marché pour contrôler la géométrie du mouvement, mais pas son rythme; cette méthode est connue sous le nom anglais de flip-book. L'autre méthode consiste à filmer les dessins afin de contrôler l'efficacité des effets, l'exactitude des rythmes et l'esthétique des mouvements; le terme anglais "linetest" est employé pour désigner ce type de test.

Cette étape de la fabrication d'une dessin animé semble pouvoir bénéficier à très court terme des développements technologiques. Signalons que toutes les nouvelles méthodes proposées ici permettent une vision immédiate de l'animation à tester, alors que le linetest traditionnel nécessite des délais et des coûts d'exploitation importants, et que le flip-book ne permet qu'une vérification et une mise au point très sommaire. Ce dernier reste cependant la méthode la plus accessible à l'animateur en train de préparer ses dessins; nous ne reviendrons pas dessus par la suite.

Un plan de dessin animé dure de 5 à 30 secondes et comprend entre 60 et 720 images différentes. Tout dispositif de test évolué devra donc permettre de stocker et de visualiser de l'ordre de 700 images.

Les méthodes présentées ont été regroupées en trois classes: la méthode traditionnelle sur film, les méthodes basées sur un enregistrement vidéographique des dessins, et enfin celles supposant connue leur représentation numérique. Toutes ces méthodes permettent un contrôle approfondi de l'animation: effet, rythme, esthétique.

3.2.1. LE FILM DE TEST

La méthode utilisée traditionnellement consiste à filmer les dessins préparatoires exécutés en noir et blanc sur papier par les animateurs. On perçoit immédiatement la lourdeur et le coût d'une telle procédure. Le film doit être envoyé au développement, d'où il revient au plus tôt le lendemain. Il est ensuite visionné sur table de montage. Il est possible de gagner du temps par l'utilisation de films polaroid ou inversibles, mais difficilement de baisser ainsi le coût du test. Cette méthode ne paraît pas devoir évoluer rapidement, tant elle est liée à des processus chimiques longs et coûteux.

3.2.2. L'ENREGISTREMENT VIDEOGRAPHIQUE

3.2.2.1. Le disque vidéo magnétique

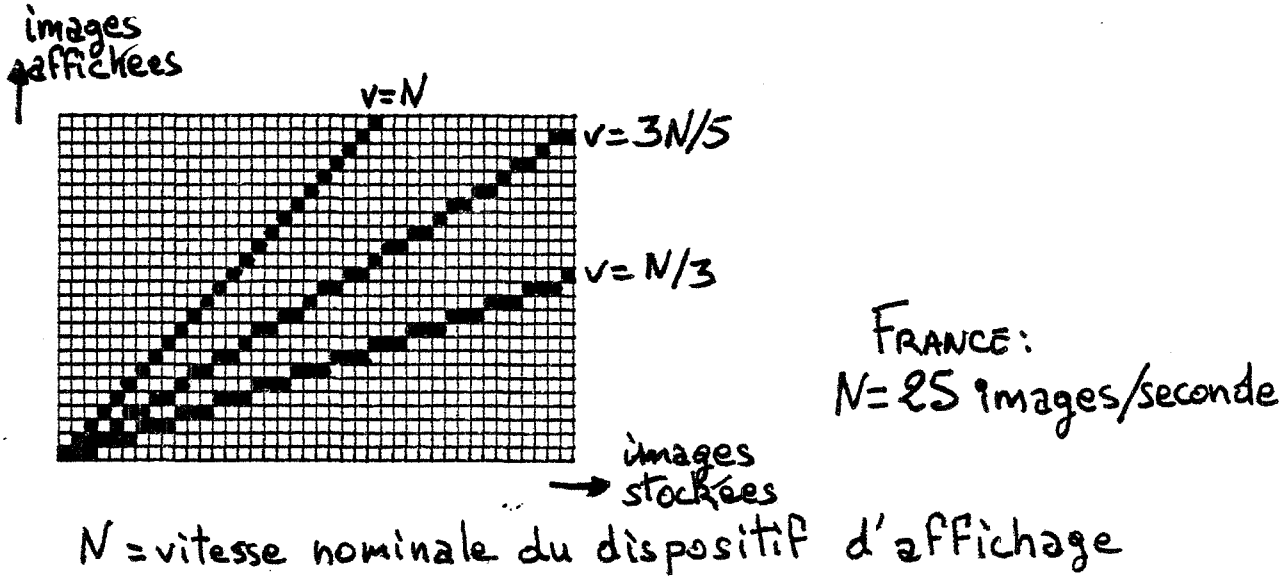
Des disques magnétiques permettent d'enregistrer environ 700 images, puis de les relire dans un ordre quelconque et à un rythme paramétrable. Ces disques sont depuis longtemps utilisés à la télévision pour les ralentis et les arrêts sur image lors des retransmissions en direct. Ils permettent d'enregistrer les images en couleur et image par image. Ils ont le défaut principal d'être assez coûteux, environ 300 Kf, et fragiles. Ils sont un bon médium pour réaliser l'enregistrement image par image en vidéo lors du tournage final. Ils ont été utilisés à cette fin par le système DAAO [COM84].

Nous avons participé à la réalisation d'un programme permettant de gérer un disque de ce type dont les principales fonctionnalités sont:

- enregistrement d'une image à une position choisie,
- choix de la séquence d'images affichées,
- choix de la vitesse d'affichage de la séquence; les vitesses possibles vont de l'image fixe à 125 images par seconde,
- options: cycle, affichage pas à pas.

Cette réalisation a permis de mettre en évidence le lien entre le problème du tracé d'un segment sur une grille de points et la simulation d'une vitesse d'affichage sur un dispositif permettant l'affichage de N images par seconde. En effet, si on associe l'image stockée i à la colonne i de la grille et la j-ième image affichée à la ligne j de la grille, la séquence qu'on affiche associe une image stockée à chaque image affichée, ces paires constituant les coordonnées de points de la

grille. La séquence affichée détermine donc un tracé sur la grille. Par exemple, l'affichage normal à V images par seconde est représenté par la diagonale principale de la grille. Inversement, étant donné un vitesse souhaitée k*V, le tracé sur la grille par une méthode classique [LUC81] d'un segment de pente k permet de déterminer quelle image afficher à chaque instant (figure 3.1.).



vitesse d'affichage et tracé sur une grille

Figure 3.1.

L'exploitation de ce principe est une voie d'illustration graphique des données de la feuille de prise de vue. Nous pensons étudier dans les prochains mois l'usage interactif de cette représentation.

3.2.2.2. Dispositifs de commande de magnétoscope

Des magnétoscopes peuvent être pilotés pour permettre l'enregistrement image par image. Ce mode de fonctionnement soumet bandes et mécanique du magnétoscope à rude épreuve. En effet, l'enregistrement sur une bande vidéo se fait sur une bande lancée à une certaine vitesse; les dispositifs d'enregistrement image par image contrôlent des séquences de rembobinage, prise de vitesse, arrêt, puis à nouveau rembobinage et ainsi de suite.

Les magnétoscopes spécialement conçus pour cette opération coûtent actuellement environ 1 million de francs. On trouve des dispositifs moins coûteux pour

piloter des magnétoscopes de série. On peut craindre l'influence de leur utilisation sur la durée de vie du magnétoscope et des bandes. Ces dispositifs donnent des résultats très inégaux; tous ne respectent pas les recommandations de la norme Broadcast, qui définit les caractéristiques de la vidéo professionnelle. Pour mémoire citons: le Vidéotoon [SPI83], l'Anivid [SAV84].

L'enregistrement est réalisé en plaçant un à un les dessins en face d'une caméra vidéo et en déclenchant le magnétoscope. La plupart de ces systèmes permettent l'enregistrement en couleur et peuvent donc être utilisés pour le tournage final; il faudrait cependant vérifier avec soin le respect de la norme Broadcast par chacun d'eux. Le visionnage peut succeder immédiatement à l'enregistrement.

3.2.2.3. Le disque vidéo optique

De nombreuses études sont menées dans le monde entier pour étudier les méthodes de stockage sur disque [FER84]. Nous avons déjà parlé des disques magnétiques dont la capacité de stockage est assez faible. Certaines techniques permettent de stocker plusieurs dizaines de milliers d'images sur une face de disque, mais nécessitent le pressage des disques sur des matériels coûteux. Ces techniques ne sont donc pas utilisables aujourd'hui pour le test d'animation.

Les disques optiques numériques permettent le stockage de dix milliards de bits par disques, soit environ 30000 images binaires de 512x512 pixels enregistrables une seule fois par l'utilisateur. Déjà ce type de disque est utilisé pour l'archivage d'images. Ces disques sont conçus au départ pour leur utilisation dans un environnement informatique. Il est probable que dans un avenir proche, cette technique sera utilisable pour le test d'animation, à moins que ses coûts d'exploitation ne baissent pas suffisamment.

3.2.3. LES IMAGES NUMERIQUES

3.2.3.1. Considérations générales

Au chapitre 2, les méthodes pour obtenir une représentation numérique des dessins ont été présentées. Nous étudions ici comment utiliser ces représentations pour visualiser l'animation. Les méthodes proposées ici présentent aujourd'hui des limites techniques contraignantes, qui seront autant que possible explicitées. L'évolution rapide de la technologie informatique devrait permettre de réduire ces limites et de rendre vite certaines de ces méthodes exploitables. Le test d'animation deviendra alors une fonctionnalité intégrée du système informatique de réalisation de dessins animés.

Quelle que soit la représentation numérique des dessins utilisée, le mécanisme du test peut se décomposer en quatre étapes:

-> transfert des données d'un stockage statique vers l'ordinateur hôte,
-> transfert des données de l'hôte vers le dispositif d'affichage avec traitement éventuel préalable pour fournir les données sous une forme compatible avec le dispositif d'affichage,
-> traitement des données par le dispositif d'affichage pour les rendre affichables,
-> affichage proprement dit.

A chaque étape, on peut associer le débit d'images qu'il autorise. La dernière étape doit permettre un débit supérieur ou égal à 25 images par seconde. La durée de la plus longue séquence qu'on pourra visualiser sur un matériel donné est conditionnée par chacun de ces débits. Cette durée est aussi conditionnée par le nombre d'images qui peuvent être représentées dans:

-> le dispositif de stockage statique,
-> l'ordinateur hôte,
-> le dispositif d'affichage, avant traitement,
-> le dispositif d'affichage, après traitement.

Dans la suite, nous comparons les performances possibles avec quatre représentations différentes des images:

-> le pictogramme binaire de 512x512 pixels, qui occupe 32 kilo-octets,
-> le codage par plage de ce pictogramme évalué à environ 6 kilo-octets,
-> le codage de ce pictogramme par la méthode proposée au chapitre 2 paragraphe 5, évalué à environ 3 kilo-octets; nous le nommerons code par chaînons,
-> le vectogramme, supposé comprendre 1000 segments, chacun représenté par 3 octets en moyenne, soit environ 3 kilo-octets; on suppose que le dessin est composé de lignes brisées et qu'il a environ le même nombre de segments que de points, chaque point étant représenté sur 3 octets.

Sur un disque dur de 10 méga-octet, on stockera environ:

-> 300 pictogrammes,
-> 1700 codes par plage,
-> 3400 code par chaînons,
-> 3400 vectogrammes.

Nous avons constaté sur plusieurs miniordinateurs du commerce, dans une configuration standard, des débits entre le disque et l'ordinateur de l'ordre de 25 kilo-octets par seconde, soit:

-> 1 pictogramme,
-> 4 codes par plage,
-> 8 codes par chaînons,
-> 8 vectogrammes.

Ces miniordinateurs peuvent utiliser 1 méga-octets pour le stockage des représentations d'image; ils peuvent donc stocker respectivement 30, 170, 340 et 340 dessins, suivant la représentation adoptée.

Les débits entre l'ordinateur hôte et le dispositif d'affichage peuvent être de plusieurs types. En effet, le dispositif d'affichage est:

-> soit un périphérique relié par une liaison série à l'hôte, le débit dépassant alors difficilement les 2,5 kilo-octets par seconde,
-> soit un périphérique relié par une liaison DMA-parallèle à l'hôte, le débit pouvant atteindre le méga-octet par seconde,
-> soit un coprocesseur de l'hôte, auquel cas aucun transfert n'est nécessaire, ou un transfert de mémoire à mémoire à un débit de l'ordre du méga-octet par seconde,
-> soit l'ordinateur lui-même, ce qui conduit aux mêmes débits que dans le cas précédent.

Le premier de ces cas doit être éliminé; il ne permet d'afficher qu'une séquence d'image contenue dans la mémoire d'image, sans mise à jour possible en cours de visualisation. Tous les autres cas sont compatibles avec la mise à jour de 25 images par seconde, puisqu'à un méga-octet par seconde on transmettra respectivement 30, 170, 340 et 340 dessins chaque seconde.

La vitesse de rafraichissement de l'écran d'affichage constitue un problème. Seuls les dispositifs fonctionnant avec des écrans à rémanence pas trop élevée sont compatibles de ce point de vue avec l'affichage de 25 images par seconde. Actuellement, il s'agit principalement des mémoires d'image avec écran vidéo et de certains générateurs spécialisés, capables de produire directement un signal vidéo à partir d'une représentation codée de l'image; nous reviendrons sur certains de ces dispositifs plus loin dans ce paragraphe. Une mémoire d'image est un ensemble de composants mémoire à double accès:

-> l'accès informatique permet d'adresser chaque élément de mémoire pour y lire ou y écrire des données numériques,
-> l'accès vidéographique permet à un dispositif électronique spécialisé de produire un signal vidéo par combinaison des bits contenus dans la mémoire; ce signal vidéo, fourni en entrée d'un moniteur vidéo, permet de visualiser une image.

L'image est représentée par N lignes de M points élémentaires ou pixels; chaque pixel est représenté sur B bits, ce qui permet de lui attribuer 2 puissance B valeurs. B conditionne donc le nombre de couleurs différentes que pourra prendre chaque point de l'image. Certaines mémoires se prêtent à un affichage modulaire de l'information qu'elles contiennent. Par exemple, le même dispositif peut contenir une image de 1024x1024 pixels de

8 bits ou 32 images de 512x512 pixels binaires.

Une mémoire d'image bit-map est une mémoire d'image qui fait partie de l'espace d'adressage du processeur d'un ordinateur. Dans certains cas une large part de l'espace d'adressage peut être utilisée comme mémoire d'image, il suffit alors d'indiquer au générateur vidéo quelle zone mémoire il doit lire. Il n'y a alors pas de temps de transfert de l'ordinateur au dispositif d'affichage, si ce n'est des transferts entre zones mémoire de l'ordinateur. Dans ces conditions on peut stocker des images en mémoire, puis les faire défiler rapidement en indiquant au générateur vidéo successivement le début de la zone mémoire de chacune des images. Le principe est utilisé par DAAO [COM84][BAU84] sur IBM-PC. Dans ce cas, la mémoire disponible ne permet de visualiser qu'une séquence de huit images. Sur une SM-90 équipée d'une bit-map 1024x2048 tout ou rien et d'un méga-octet de mémoire RAM, on stockera une trentaine d'images binaires 512x512; le renouvellement de ces images ne sera possible que sur une configuration où les échanges avec le disque sont gérés par un coprocesseur (cache-disque).

Pour visualiser une animation, on a besoin d'une mémoire d'image qui contienne au moins deux images binaires. Une image est affichée tandis que l'autre est mise à jour; tous les vingt-cinquième de seconde les rôles sont inversés; ce mécanisme est connu sous le nom de flip-flop. Quand elle est possible, la commutation est instantanée entre l'image lue par le générateur vidéo et celle accédée par la voie numérique.

Si Nm est le nombre d'images stockées en mémoire d'image, N1 le nombre d'images affichées dans le temps nécessaire pour générer une nouvelle image, alors le nombre total d'images affichables est donné par NT défini à l'aide de la suite S, où:

$S0 = Nm$

$S_n = S_{n-1} \text{ div } N1$

où div est la division entière

et $NT = \sum_n S_n$ (1)

Par exemple, supposons:

* une mémoire d'image permettant l'affichage de 96 images de 256x256 pixels binaires (mémoire de 512x512 pixels sur 24 bits chacun) et associée à un processeur graphique pouvant tracer 12500 segments par seconde, soit 500 segments chaque vingt-cinquième de seconde,
* une liaison parallèle permettant de transmettre 1 méga-octet par seconde, soit 340 vectogrammes de 1000 segments chaque seconde, entre l'ordinateur hôte et la

mémoire d'image; il est clair que ce débit ne limite pas la durée de la séquence affichée.

alors (1) s'écrit:
Nm = 96 images
N1 = 2
S0 = 96; S1 = 48; S2 = 24; S3 = 12;
S4 = 6; S5 = 3; S6 = 1;
Si = 0 pour tout i supérieur ou égal à 7
NT = 96+48+24+12+6+3+1 = 190 images

Pour le reste de la discussion, nous éliminerons le cas des pictogrammes, à cause du volume de stockage statique qu'ils nécessitent et la mémoire d'image reliée à l'ordinateur par une liaison série, à cause de sa lenteur excessive par rapport à nos objectifs.

Il apparaît alors que deux caractéristiques peuvent limiter la durée maximum de la séquence visualisable. La première est le partage des tâches par un même processeur: en effet, bien que les débits du disque vers l'ordinateur et de l'ordinateur vers le dispositif d'affichage puissent être compatibles avec une animation à 25 images par seconde, si le même processeur assure ces deux transferts, le débit global est divisé environ par deux. Il sera alors bon de disposer d'un coprocesseur qui prenne en charge un des deux transferts. A défaut, on pourra se contenter des 170 à 340 images stockables dans l'ordinateur avant de commencer la visualisation.

La deuxième caractéristique importante est la vitesse de traduction de la représentation de l'image: code par plage, code par chaînon, vectogramme en une représentation utilisable par le générateur vidéo. Plusieurs caractéristiques du dispositif d'affichage doivent être prises en compte:

-> un processeur spécialisé gère la communication avec la source extérieure de données,
-> un processeur spécialisé ou un dispositif électronique cablé assure la traduction pour l'affichage; ce genre de caractéristique existe sur des machines spécialisées, mais aussi dans certaines mémoires d'image, notamment pour la traduction du codage par plage et la génération de vecteurs; ainsi, on trouve des mémoires d'image permettant l'affichage de 1000 vecteurs de 10 pixels chaque vingt-cinquième de seconde,
-> un seul processeur gère la communication et la traduction.

Dans le dernier cas, il faut prendre garde aux vitesses de communication et de tracé de segments fournies par les fabricants de mémoire d'image; en effet, ces deux vitesses se limitent l'une l'autre quand un seul processeur gère les deux fonctions.

On peut concevoir à partir de procédés électroniques classiques un appareil fonctionnant
- soit de facon autonome,
- soit en périphérique d'un ordinateur hôte,
- soit comme coprocesseur d'un ordinateur.

Il s'agit d'un appareil ayant trois fonctionnalités essentielles :

* coder des images noir et blanc du type dessin de cartoon et les stocker (au moins 256),

* décoder ces images pour les afficher à un rythme pouvant aller jusqu'à 25 dessins différents par seconde,

* pouvoir communiquer ces dessins codés de l'appareil vers l'ordinateur hôte et inversement.

Le codage présenté au chapitre précédent se prête bien à une réalisation matérielle; sa réalisation a été rapidement esquissée au chapitre 2, de nombreuses améliorations algorithmiques ont été apportées par rapport à la présentation du chapitre 2 [HAB85]. Les taux de compression obtenus, la simplicité du codage comme du décodage observé permettent d'attendre d'un tel matériel des performances suffisantes pour permettre des tests de longue durée: si l'appareil dispose de 1 méga-octet en propre, on peut y stocker environ 300 images pour une visualisation temps réel. Un appareil répondant à certaines des caractéristiques mentionnées ici est actuellement commercialisé par une firme japonaise sous le nom de QAR.

Les configurations actuelles de mini-ordinateur et de mémoire d'image sont déjà proches des performances nécessaires pour permettre l'affichage d'une séquence de test d'animation. Les spécifications de produits annoncés commercialement semblent répondre pleinement aux besoins exprimés dans ce qui précède. On peut donc affirmer que le test d'animation est aujourd'hui compatible avec une configuration informatique de série.

3.3. STRUCTURATION DES DONNEES

CARTE PLANAIRE

Considérons maintenant les données telles que nous les fournissent les méthodes de saisie définies au chapitre précédent. Il s'agit de fichiers de segments de droite. Nous souhaitons connaître les zones fermées qui figurent dans l'image; pour cela, il devient nécessaire de structurer les données dont nous disposons: une gerbe de segments. La structure à définir doit nous permettre de suivre les frontières de toutes les zones fermées de l'image et ainsi d'attribuer une couleur à chaque zone.

L'affectation d'une couleur à chaque zone du cellulo peut se faire par pointage. Il faudra alors au moins disposer d'un outil logiciel indiquant dans quelle zone fermée se trouve un point donné.

Nous exposons ici une méthode permettant de construire une telle structure de données à partir d'un ensemble non structuré de segments de droite. Cette structure permet un accès rapide à de nombreuses informations de type topologique ou géométrique. Nous verrons dans la suite de ce chapitre et au chapitre suivant les avantages que la construction de cette structure donne au vectogramme sur le pictogramme. Nous verrons aussi comment cette structure permet de combattre les défauts liés à la représentation par vectogramme.

3.3.1. INTRODUCTION

Nous sommes partis de la définition d'un certain nombre de fonctionnalités nécessaires à notre application: saisie de dessins, retouches, gouachage, assemblage de plusieurs dessins, restitution.

Les interventions possibles sur les images non-structurées utilisées par les systèmes commercialisés d'aide au dessin animé nous ont paru limitées, en particulier par la résolution d'affichage; elles se prêtent moins que le vectogramme à l'exploitation des développements technologiques les plus récents: coprocesseurs graphiques, outils de report sur film, bibliothèque graphique intégrée (GKS [GKS84])... De plus la volonté d'exploiter la cohérence entre les dessins d'une même séquence rendait souhaitable une structuration

des dessins. Cette idée est développée au chapitre suivant (ch. 4).

Nous nous plaçons dans le cas des dessins saisis à la table à numériser ou au scanner haute résolution. Dans le cas de dessins saisis sur une table à numériser, ceux-ci sont connus comme une gerbe de segments de droite. Dans le cas de dessins numérisés. Le codage proposé au chapitre précédent pour stocker les images saisies par scanner fournit directement la description de tous les contours de l'image sous forme d'un grand nombre de petits segments de droite, les chaînons du code. Dans les deux cas, il est souhaitable de réduire la quantité de points et de segments par des filtrages. L'utilisation des méthodes présentées dans ce chapitre avec des images saisies par caméra est abordée au paragraphe 3.2.2..

Ce problème de structuration d'une gerbe de segments est décrit en détail par Gangnet et Michelucci dans [GAN84]. Plusieurs applications ont utilisé ce principe. Nous avons bénéficié pour notre application d'une implémentation des procédures de construction et d'interrogation d'une telle structure de données due à Dominique Michelucci, au sein de l'équipe Communications Visuelles de l'Ecole Nationnale Supérieure des Mines de Saint-Etienne.

Dans la suite, l'ensemble des structures adoptées sera globalement désigné par le nom de carte planaire. Ce terme est emprunté à la théorie des graphes [BER73] [COR80] et est lié aux définitions suivantes:

> Soit S un ensemble fini de sommets et A une partie de SxS, le graphe G=(S,A) est dit planaire s'il est possible d'en trouver une représentation dans le plan telle que les images de deux arêtes distinctes n'aient d'autres points communs que les images de sommets du graphe. Une carte planaire est ici une représentation plane d'un graphe planaire. Dans la suite, la carte planaire désigne une représentation particulière d'un graphe planaire; à chaque sommet du graphe sont attachés ses coordonnées dans le plan.

On trouvera aussi des résultats intéressants sur ce type d'objets mathématiques dans [JAC69], [SHO83].

3.3.2. DESCRIPTION D'UNE CARTE PLANAIRE

3.3.2.1. Le but poursuivi

Nous cherchons à construire une structure de données permettant au moins de trouver dans quelle zone fermée se trouve un point et l'arbre d'inclusion de ces zones fermées. De plus, nous aimerions savoir pour chaque arête quelles sont les deux zones fermées qu'elle sépare.

Chaque zone fermée est décrite par la liste de ses arêtes. Chaque arête est décrite par ses deux sommets extrêmités et les deux zones qu'elle sépare. L'arbre d'inclusion regroupe les zones fermées; une zone a pour descendantes les zones qu'elle contient; la racine de l'arbre représente une zone fermée virtuelle dont les arêtes se trouvent à l'infini dans le plan et contenant toutes les autres zones. Chaque sommet est décrit par ses deux coordonnées dans le plan et la liste des arêtes incidentes.

3.3.2.2. Les structures adoptées

Dans notre cas, nous désirons représenter à partir d'une suite de segments de droite la carte planaire ayant pour sommets les extrêmités des segments initiaux et leurs intersections éventuelles, et pour arêtes les segments initiaux ou leurs morceaux, en cas d'intersection.

Nous ne décrirons la construction d'une carte planaire que de façon informelle, renvoyant pour plus de précision à [GAN84]. Cette construction est inspirée des travaux de Bentley et Ottmann [BEN79]. On trouvera également une étude de ce type d'algorithmes dans [TAL85].

La construction:

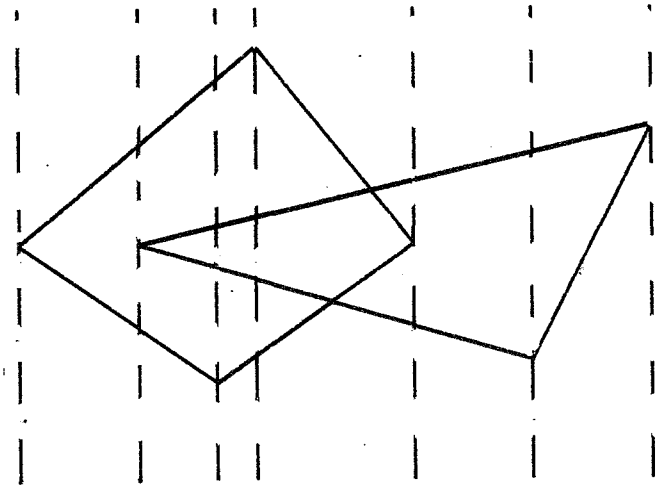
Nous partons d'un ensemble S de segments de droite connus par leurs extrêmités dans le plan. La premiére étape du traitement va être de passer de cet ensemble S à un ensemble S' dont les arêtes sont issues de celles de S en les brisant en leurs points d'intersection; les arêtes de S' n'ont donc au plus en commun que leurs extrêmités. Les arêtes de S' séparent des zones connexes du plan. Nous allons rechercher celles-ci et construire peu à peu une structure décrivant les inclusions de zones: l'arbre des bords.

Les étapes:

*On ordonne les sommets connus, d'abord suivant les abscisses, puis à abscisse égale suivant les ordonnées.

*On déplace une droite parallèle à l'axe des ordonnées perpendiculairement à cet axe, dans le sens des abscisses croissantes en lui donnant successivement les abscisses des sommets et des points d'intersection que l'on trouve au fur et à mesure. La méthode assure que l'on obtient toutes les intersections, et que les nouvelles intersections sont à chercher au delà de la droite de balayage courante.

*On gère l'ordre des arêtes d'après l'ordonnée de leur intersection avec la droite de balayage courante et les modifications locales de cet ordre en chaque sommet. L'adjacence suivant cet ordre est une condition nécessaire pour que deux arêtes puissent se couper. On trouvera donc toutes les intersections uniquement par des tests sur les arêtes adjacentes. Lorsqu'un nouveau point d'intersection est découvert il est inséré dans la liste des sommets.



Positions successives de la droite de balayage au cours de la construction.

Balayage pour la construction d'une carte planaire

Figure 3.2.

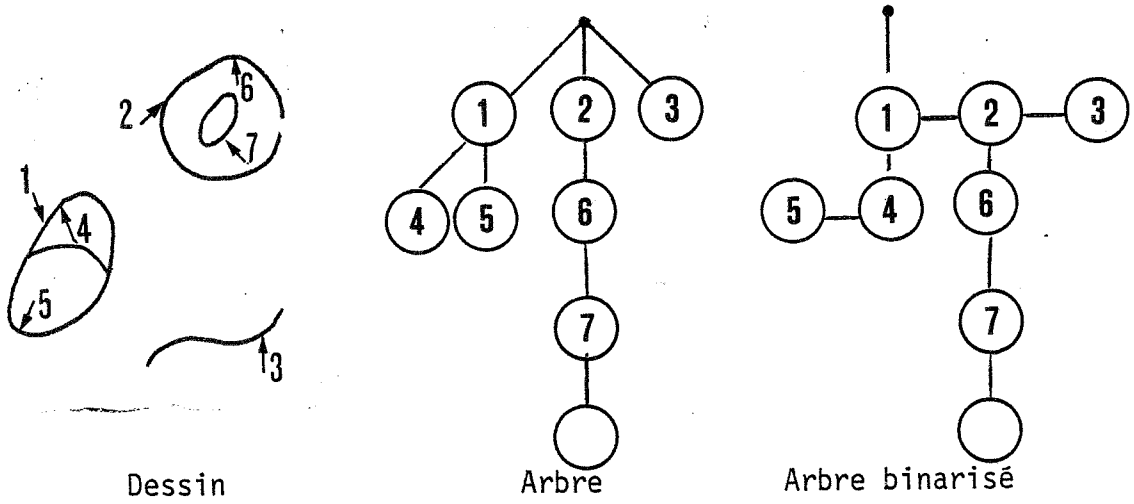
Dans le cas de saisies par caméra ou scanner, les chaînons du code proposé au chapitre 2 ne s'intersectent pas. Tout calcul d'intersection est donc inutile. La construction d'une structure analogue à celle présentée ici pendant le codage permettrait d'utiliser les méthodes

présentées dans ce chapitre. Des travaux sont en cours à l'Ecole des Mines pour la construction et l'utilisation d'une telle structure [BER85].

3.3.2.3. Définitions

* bord : nous avons désigné par bord la frontière de toute zone obtenue par le cheminement d'un bonhomme virtuel, qui avancerait en posant en permanence sa main gauche contre les arêtes considérées comme des murs. On peut distinguer deux types de bords : les bords externes parcourus dans le sens trigonométrique direct; les bords internes parcourus dans le sens inverse; la partie du plan comprise entre un bord interne et ses fils externes sera nommée zone, car elle correspond à une zone connexe de l'image (figure 3.3.); nous utiliserons dans la suite l'arbre d'inclusion des bords sous forme binarisée. A chaque bord, nous attacherons:

- une référence au bord qui le contient: père,
- une référence à un des bords qui sont au même niveau que lui: frère,
- une référence à un des bords qu'il contient: fils_aine.



Arbre des bords d'une carte planaire

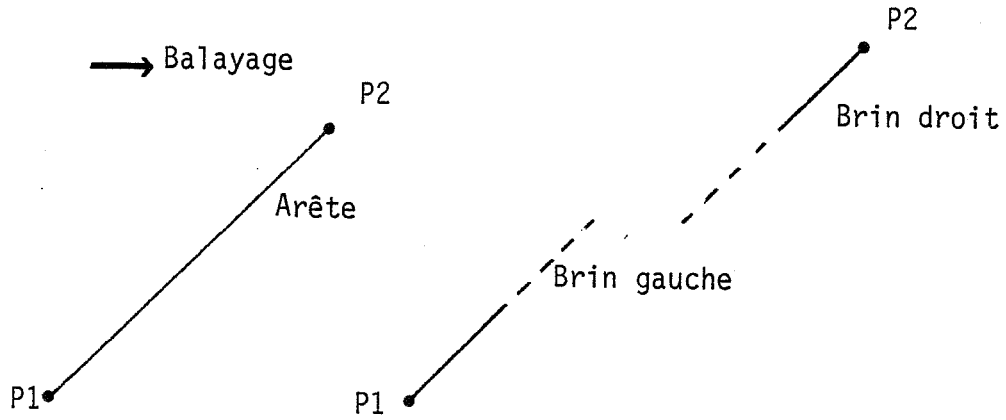
Figure 3.3.

* brin : un brin est une demi-arête. Un brin est décrit par:

  - une référence au bord passant par ce brin,

  - une référence à un brin d'une arête partageant la même extrêmité.

Ces références permettent des parcours de brin en brin, par exemple pour suivre un bord. Une arête sépare deux bords et est incidente à deux sommets, c'est pourquoi elle a pu être décomposée en deux brins, chacun incident à un des sommets et faisant référence à un des bords.



Décomposition d'une arête en brins

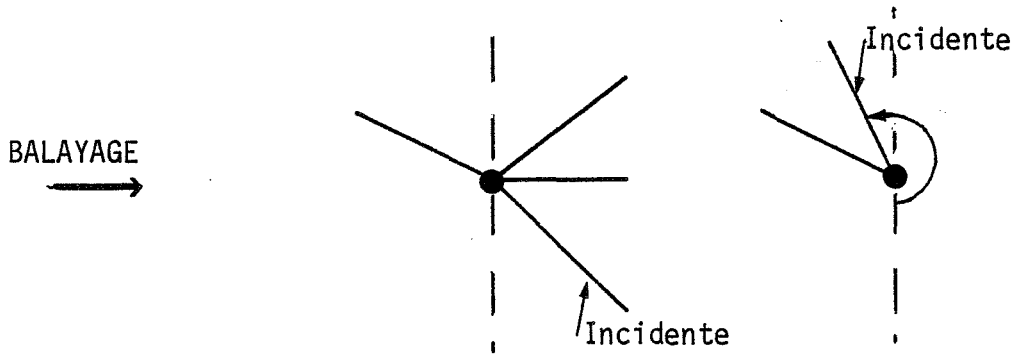
Figure 3.4.

* carte : désigne dans la suite un pointeur vers la structure complète, comprenant notamment l'arbre des bords et la liste des sommets.

Nous décrivons maintenant les structures de représentation d'une carte planaire dans l'implémentation Pascal utilisée.

```
Type
pt_sommet = ^t_sommet;
pt_arête = ^t_arête;
t_coord    = record x,y:real;end;
t_sommet   = record
               x_précédent,x_suivant: pt_sommet;
(* ces deux pointeurs donnent le précédent et le suivant
du sommet considéré, selon l'ordre défini ci-dessus  *)
               incidente : pt_arête;
               côté_incidente: 1..2;
(* incidente désigne une des arêtes ayant ce sommet pour
extrêmité; côté_incidente indique de quel côté  de la
```

```
droite de balayage se trouve l'incidente          *)
                coord : t_coord;
(* coordonnées du sommet considéré                *)
              end;
```



Arête incidente

Figure 3.5.

```
t_brin = record
            point : pt_sommet;
(* le sommet du brin considéré                    *)
            voisine : pt_arête;
            côté_voisine: 1..2;
(* indique le brin utile de l'arête voisine       *)
            bord    :pt_bord;
(* pointeur sur le bord unique passant par ce brin
         end;

t_arête =  record
              verticale : boolean;
(* vrai si l'arête est verticale, faux sinon      *)
              a,b       : real;
(* équation du support de l'arête y=ax+b ou x=a   *)
              demi      : array[1..2] of brin;
(* demi[1]=brin gauche de l'arête;demi[2]=brin droit
            end;

t_boite = record
             xmin,xmax,ymin,ymax: réal;
(* boîte englobante parallèle aux axes            *)
          end;

t_bord = record
            sens:(interne,externe);
            père,fils_aine,frère:pt_bord;
(* description de l'arbre d'inclusion de la carte planai
            départ: pt_arête;
(* pointeur sur une arête du bord
            boite: t_boite;
(* boîte englobante du bord parallèle aux axes
         end;
```

Les programmes utilisant cette structure de données y ont accés par un pointeur sur un enregistrement contenant un pointeur sur le premier sommet de la liste de sommets et un pointeur sur la racine de l'arbre des bords.

3.3.2.4. Les informations fournies par la structure de carte

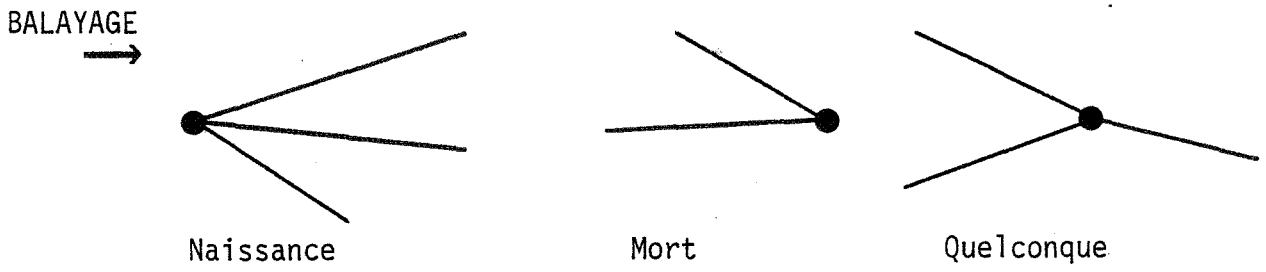
Le méthode choisie fournit les informations désirées sur les zones fermées de l'image; elle donne accès en plus à une liste ordonnée des sommets.

Les développements suivants supposent une compréhension de principe de ce qu'est une carte planaire. Nous utilisons par la suite des fonctionnalités offertes par l'implémentation de Dominique Michelucci.

Ces fonctionnalités doivent être perçues comme suit: à partir d'une liste d'arêtes et de sommets, on calcule une structure de données décrivant complètement la figure plane formée par les arêtes données. Cela fait, on dispose pour utiliser la carte planaire construite d'un ensemble de procédures permettant des traitements divers sur la structure de données produite. Il s'agit essentiellement de procédures de parcours de la structure, la liste qui suit n'est pas exhaustive.

Quelques procédures exploitant la structure de carte planaire:

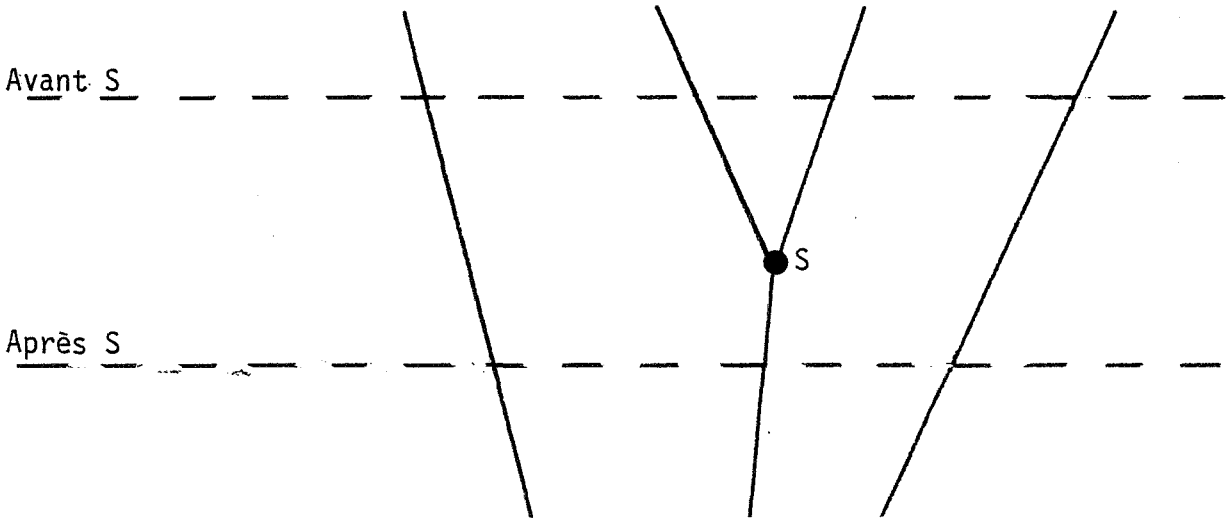
- Construction de la carte,
- Trouver dans quel bord de la carte se trouve un point donné,
- Trouver le bord le plus proche d'un point donné,
- Trouver l'arête suivant une arête donnée sur un bord donné,
- Trouver le sommet le plus proche d'un sommet donné,
- Trouver l'arête la plus proche d'un point donné,
- Suppression des arêtes pendantes: arêtes ayant un même bord marqué des deux côtés, et recalcul de la carte,
- Effectuer une procédure donnée pour tous les bords d'un sous-arbre donné de l'arbre des bords
- Effectuer une procédure donnée pour tous les sommets de la carte
- Effectuer une procédure donnée pour toutes les arêtes d'un bord donné
- Quel_type_de_point:
  cette fonction reçoit en paramètre un sommet de la carte et rend comme résultat le type de ce sommet; ce type est : point de naissance si toutes les arêtes partant de ce sommet sont au-delà d'une droite de balayage définie comme pour la construction de la carte et passant par ce sommet; point de mort si toutes les arêtes partant de ce sommet sont avant la droite de balayage; point quelconque sinon (figure 3.6.),



Types de sommets d'une carte planaire

Figure 3.6.

- Gestion_intervalles: cette procédure permet de gérer la liste ordonnée des arêtes coupant une droite de balayage. La liste des arêtes coupant la droite de balayage juste avant un sommet est fournie en entrée; la procédure la modifie pour fournir la liste d'arêtes coupant la droite de balayage entre le sommet et son suivant dans la liste de sommets. Cette procédure tient compte du fait qu'une pareille liste ne peut être modifiée qu'au franchissement d'un sommet de la carte puisque les arêtes de la carte ne se coupent qu'en des sommets de celle-ci. L'affichage par balayage d'une carte planaire, décrit au paragraphe 5, utilise cette procédure.



Liste d'intervalles avant et après un sommet

Figure 3.7.

3.3.3. QUELQUES ELEMENTS QUANTITATIFS AU SUJET DES CARTES PLANAIRES

Les cartes planaires de la plupart des dessins que nous avons utilisés, composés de gerbes de segments de droite, se calculent en des temps compatibles avec une interaction: quelques secondes. C'est pourquoi n'ont pas été développées de procédures de sauvegardes statiques de cette structure dynamique sophistiquée. Le stockage statique nécessaire est donc celui des sommets et des arêtes de la carte, augmenté éventuellement de caractéristiques attachées à certains objets composant la carte. Nous avons par exemple sauvegardé des couleurs attachées à chaque bord dans un arbre image de l'arbre des bords. Lorsqu'on reconstruit la carte planaire, la relecture simultanée de cet arbre permet de replacer la bonne couleur sur chaque bord.

Un arbre des bords et son arbre image

Figure 3.8.

L'ordre de grandeur de la place prise par la structure dynamique construite est fourni ici pour la conception des traitements ultérieurs et la définition des configurations matérielles susceptibles d'accueillir ces traitements. Les exemples que nous avons traités nous ont montré qu'on a toujours un nombre de points à peu près équivalent au nombre d'arêtes, et que ce nombre et très supérieur au nombre de zones fermées. Cette relation n'est pas fortuite; la relation d'Euler pour les graphes planaires donne:

$$s-a+f = p+1$$

où s est le nombre de sommets,
a est le nombre d'arêtes,
f est le nombre de bords internes, y compris le bord interne virtuel à l'infini,
p est le nombre de composantes connexes du graphe.

Dans l'implémentation actuelle:

-> 1 sommet utilise 25 octets,
-> 1 arête utilise 35 octets,
-> 1 bord utilise 33 octets,
en supposant les pointeurs codés sur 4 octets.

L'implémentation actuelle des cartes planaires a été utilisée principalement pour deux applications : dessins de plans de bâtiment et cellulos de dessin animé. Dans les deux cas, les données manipulées ont rarement dépassé s=a=1000 et f=100. Dans ce cas, la place mémoire nécessaire pour ranger une carte planaire en mémoire dynamique est d'environ 65 kilo-octets. Cette place est

tout à fait compatible avec les capacités des machines actuelles, y compris certains micro-ordinateurs. En pratique, la place occupée par une carte est environ proportionnelle à son nombre de sommets.

Remarque:

Le calcul de la carte correspondant à la figure 2.1.b prend environ 30 secondes. Le volume de données correspondant au stockage dynamique de la carte est d'environ 165 kilo-octets.

3.4. VERIFICATION DES DESSINS

3.4.1. LES CONTROLES NECESSAIRES

> Quelle que soit la méthode de saisie, il arrive que certaines zones qui devraient être fermées par un trait continu ne le soient pas. Ceci se traduit par l'impossibilité de différencier deux zones, notamment pour leur affecter des couleurs différentes.

Il a été nécessaire de proposer des outils pour se rendre compte de ces défauts et y pallier. Une méthode proposée plus loin résoud la plupart de ces cas. Si la saisie sur la table à numériser ou la préparation des dessins en vue de leur saisie par caméra a été faite avec soin, il reste alors très peu de cas à traiter; nous proposons des outils interactifs permettant de compléter les dessins au cours du gouachage.

Ce problème se pose aussi avec la méthode actuellement utilisée par les systèmes commerciaux [STE81]. Dans ces méthodes le coloriage se fait dans l'espace image, à la résolution de l'écran. Ainsi les seules ouvertures qui peuvent être gênantes sont en principe visibles. Cependant des ouvertures d'un pixel de large dans un tracé régulier sont très difficiles à percevoir. Nous verrons que les vectogrammes posent plus de problèmes que les pictogrammes, mais qu'il fournissent des méthodes puissantes pour corriger ou détecter les anomalies.

La représentation par vectogramme n'est pas contrainte par la résolution d'affichage. Ainsi les différentes étapes du travail peuvent se faire sur des équipements de visualisation aux performances différentes:

- affichage rapide et de haute qualité pour le tournage,
- affichage très rapide en couleur pour le gouachage,
- affichage très rapide pour les vérifications.

Mais ce découplage entre résolution des données et résolution d'affichage se retourne contre nous pour le traitement des défauts des dessins au trait.

En effet, certaines zones peuvent être ouvertes dans l'espace des données, mais sembler fermées à la résolution de l'image affichée. Ainsi nous aurons des ouvertures visibles que nous pourrons traiter interactivement et des ouvertures invisibles par l'opérateur de contrôle et pour lesquelles il faudra construire des outils spécifiques. Nous proposerons également des méthodes automatiques ou semi-automatiques pour la fermeture des bords visiblement ouverts. Nous verrons quelques idées pour améliorer le travail du vérificateur.

Les modifications du dessins entrainées par ces procédures nécessitent le recalcul de la carte pour son usage ultérieur. Dans certain cas, la possibilité de différer ce calcul sera signalée.

Notons que la représentation par vectogramme adoptée permet d'ajouter des arêtes invisibles pour assurer les fermetures de zones. Ainsi l'aspect extérieur du dessin n'est pas modifié, contrairement aux méthodes qui opèrent sur le pictogramme et nécessitent des zones séparées par des frontières visibles.

De plus, de nombreuses zones fermées, trop petites pour présenter de l'intérêt sur le plan visuel, peuvent être détectées afin d'éviter de leur appliquer les traitements ultérieurs. Une partie de ces petites zones sont générées par les opérations de fermeture automatique. Le détection proposée doit donc constituer la dernière étape des vérifications.

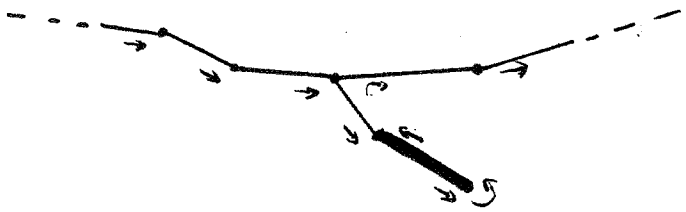
3.4.2. FERMETURES AUTOMATIQUES

3.4.2.1. Le prolongement des lignes brisées

Les dessins sont donnés sous forme d'une liste de lignes brisées. Des fermetures utiles sont assurées par l'adjonction d'une arête dans le prolongement de chaque extrêmité de chaque ligne brisée. Pour être efficace, cette procédure nécessite d'ajouter des segments relativement longs. Elle n'est donc graphiquement acceptable que si on marque les segments ajoutés pour éviter leur affichage sur l'image finale. Cette méthode est très simple, ne nécessite pas de calcul compliqué, ni la construction de structures de données sophistiquées. Elle est donc particulièrement attrayante.

3.4.2.2. Le prolongement des arêtes pendantes

La carte planaire nous permet de détecter un type particulier d'arête. Nous avons nommé arête pendante toute arête qui se succède à elle-même lors du parcours d'un bord de la carte planaire. La procédure donnant la suivante de toute arête sur un bord donné nous fournit une méthode simple pour détecter toutes les pendantes.



Arêtes pendantes et parcours d'un bord

Figure 3.9.

En dehors des endroits où une arête pendante figure volontairement, les pendantes apparaissent souvent aux endroits où une fermeture de bord devait être accomplie par la personne qui a assuré la saisie (figure 3.10).

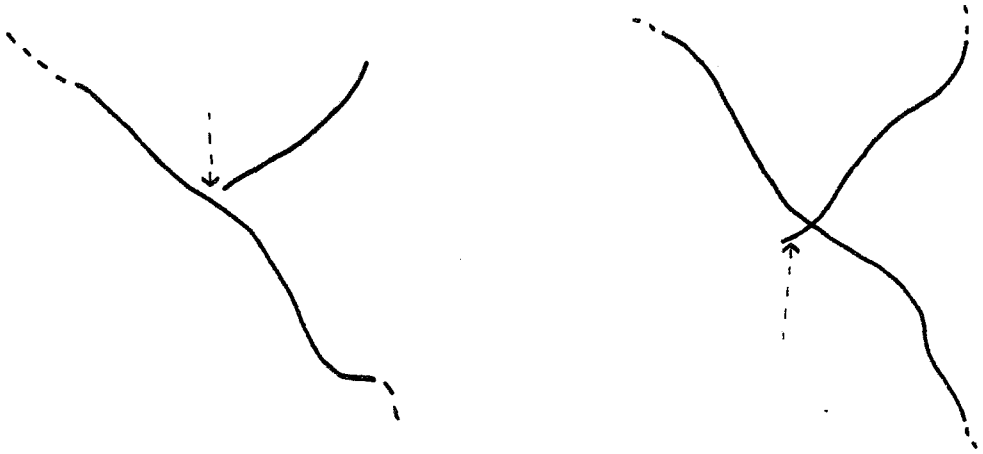


Figure 3.10.

La suppression des pendantes de très petite taille apporte une légère mise au net du dessin. C'est une procédure simple et rapide. Elle ne paraît pas devoir être systématisée pour les affichages vidéo, l'amélioration du dessin étant à peine perceptible sur une image fixe de 640x512 pixels. Cette procédure n'oblige pas à recalculer la carte pour une partie des traitements suivants. En effet, il s'agit seulement d'un marquage d'arêtes à supprimer, l'arbre des bords n'est pas modifié.

Pour fermer certains bords, il suffit de prolonger les arêtes pendantes. Si on a prévu de marquer certaines arêtes comme non vues, alors ce prolongement n'affecte pas l'aspect des données affichées.

3.4.2.3. Utilisation de la liste de sommets

Nous disposons d'une liste ordonnée des sommets de la carte. Il est alors simple de trouver les sommets proches d'un sommet donné en parcourant cette liste. Etant donné un point de cette liste, un point suivant est considéré comme proche de lui s'il est contenu dans un rectangle dont les côtés parallèles aux axes ont une taille donnée en paramètre. Le rectangle est construit au-delà du sommet de référence, au sens du balayage sur la carte et un de ses grands côtés est centré sur le point de référence.

Ainsi, pour un sommet donné, nous savons trouver le premier de ses successeurs proche de lui, s'il existe, en parcourant la liste de sommets. On teste alors si une arête existe entre ces deux sommets; si ce n'est pas le cas, on crée cette arête.

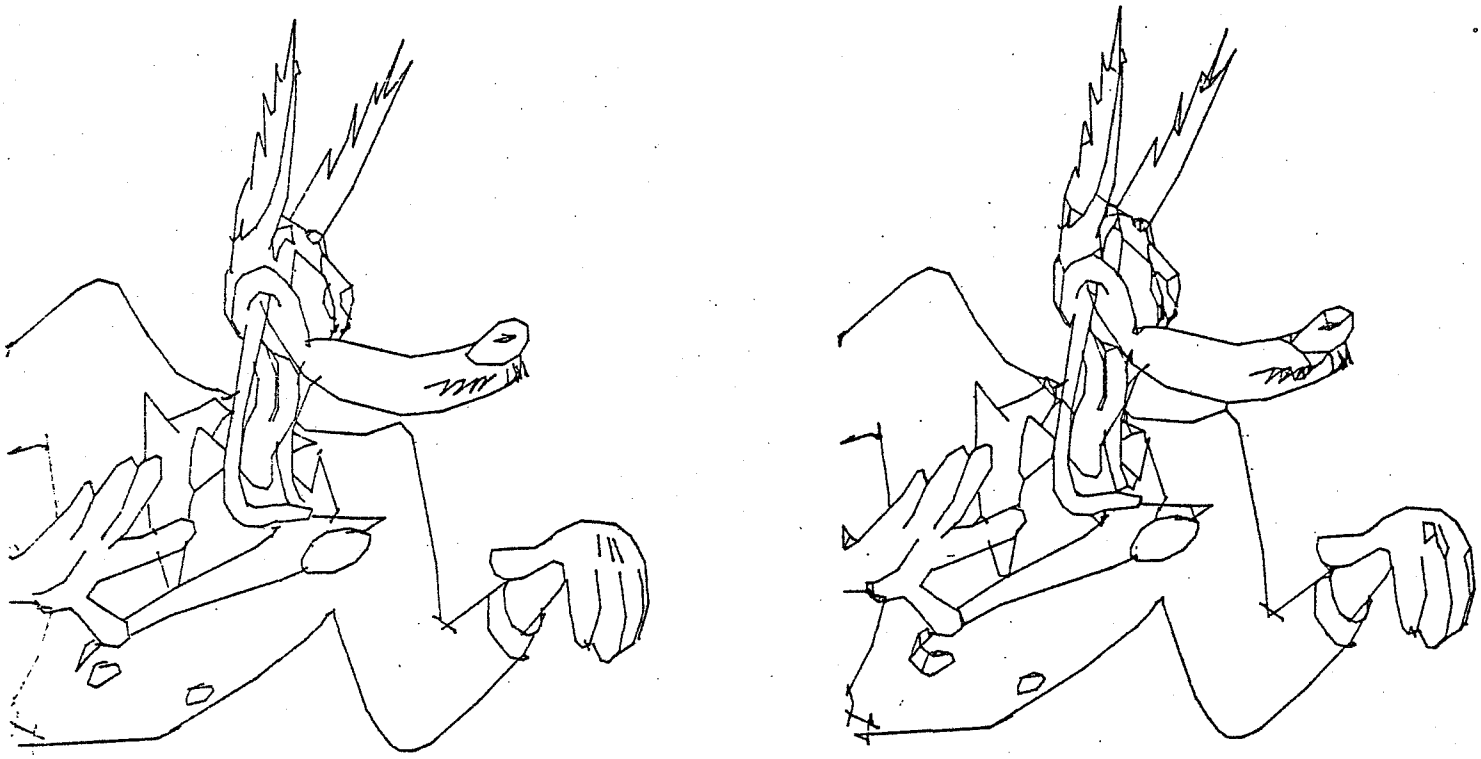
Algorithme de fermeture des bords à l'aide de la liste de sommets

```
début
  initialiser le sommet_courant au premier
  sommet de la carte;
  tant qu'on n'a pas atteint le dernier sommet
  début
    sommet_test<-sommet suivant
        sommet_courant dans la liste;
    tant qu'on n'a pas dépassé le dernier sommet
            et que la différence entre l'abscisse
            du sommet de test et celle du sommet
            courant est inférieure à un
            seuil donné alors
      début
        si la différence entre l'ordonnée du sommet
          de test et celle du sommet courant est
          inférieure à un seuil donné alors
        début
            on génère le segment du sommet courant
            au sommet test, en vérifiant éventuellement
            avant qu'ils ne sont pas déjà reliés,
            puis si le segment a été
            ajouté on prend le sommet suivant
            comme sommet courant;
        fin, sinon
        début
          sommet_test<-sommet suivant
                    sommet_test dans la liste;
        fin;
      fin;
    fin;
    sommet_courant<-sommet suivant
          sommet_courant dans la liste;
  fin;
fin;
```

Cette méthode n'est pas exempte de défauts. Elle assure presque cent pour cent des fermetures non visibles à la définition de l'affichage de contrôle, à condition de prendre pour taille de la boîte un côté correspondant à deux pixels de large. Les défauts apparaissent principalement sur les contours présentant un angle très aigu. Ces défauts ne conduisent qu'à une multiplication de petites zones. La figure 3.11 illustre la méthode et les défauts générés.

Figure 3.11.

Résultats d'une fermeture



Nous avons volontairement exagéré la taille de la boite utilisée (8 pixels) afin d'amplifier les défauts de la méthode.

Seule cette méthode a été utilisée intensivement dans les essais.

3.4.2.4. Proximité de deux portions de contours

Tous les contours sont définis par des lignes brisées composées de segments de droite. Nous dirons que deux segments de droite sont proches s'il existe un point du premier segment et un point du second tels que la distance de ces deux points soit inférieure à un seuil donné. Si deux segments de droite sont proches et non sécants alors une des extrêmités d'un des segments est proche de l'autre segment.

Proximité de deux portions de contours
soit A,B,C,D des points distincts du plan,
Définition 1:
le segment (A,B) est ε-proche du segment (C,D) si et seulement si il existe un point P appartenant au segment (A,B) et un point Q appartenant au segment (C,D) tel que la distance euclidienne de P à Q soit inférieure à ε.

Définition 2:
le point P est ε-proche du segment (C,D) si et seulement si il existe un point Q appartenant au segment (C,D) tel que la distance euclidienne de P à Q soit inférieure à ε.

Proposition:
si le segment (A,B) est ε-proche du segment (C,D) et que
l'intersection de (A,B) et (C,D) est vide
alors:
A est ε-proche de (C,D) ou
B est ε-proche de (C,D) ou
C est ε-proche de (A,B) ou
D est ε-proche de (A,B)

Démonstration:
La démonstration dont nous disposons est simple, mais longue; nous n'en indiquerons que les grandes lignes. Par un changement de repère, on ramène le segment (A,B) sur l'axe des x, le vecteur (A,B) étant un vecteur unitaire. On supposera qu'il existe un point P0 de (A,B) et un point Q0 de (C,D) à une distance inférieure à ε l'un de l'autre; si Q0 n'a pas son abscisse comprise entre 0 et 1, il est simple de montrer qu'alors A ou B est à une distance de Q0 inférieure à ε; sinon, une étude plus détaillée de la fonction "distance d'un point de (A,B) à un point de (C,D) est nécessaire.

Donc pour trouver tous les endroits où deux portions de contours sont proches l'une de l'autre, il suffit de chercher les endroits où un sommet de la carte est proche d'une arête de la carte. Pour cela nous utilisons un balayage de la carte analogue à celui décrit en détail au paragraphe 3.5 de ce chapitre pour l'affichage d'une carte planaire sur un écran à balayage. Par cette méthode, on est capable de construire la liste ordonnée des arêtes qui coupent la droite de balayage juste avant et juste après un sommet; quelques tests sur certaines de ces arêtes suffisent à trouver celles qui sont proches du sommet et à vérifier qu'elles ne lui sont pas déjà incidentes.

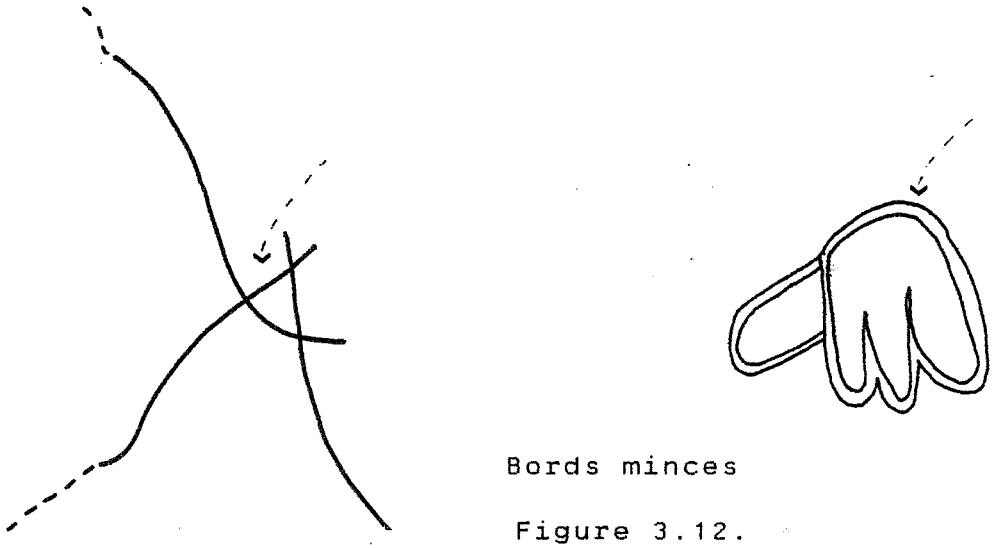
Ainsi, si il y a des arêtes proches d'un sommet, on peut relier celui-ci à l'arête la plus proche dont il ne fait pas partie. Cette procédure est relativement rapide et simple. Elle améliore les résultats de la méthode précédente. Elle n'est pas encore intégrée à l'ensemble de

notre application; seuls quelques tests isolés ont été effectués, car les méthodes déjà proposées résolvent automatiquement la plupart des cas, et la résolution des cas restants étant laissée aux procédures interactives.

3.4.3. MARQUAGE OU ELIMINATION DES ZONES MINCES

Il a paru nécessaire de détecter les zones fermées représentées dans la carte planaire et dont l'intérieur joue un rôle négligeable dans l'aspect visuel de l'image. Une fois détectées, ces zones peuvent être éliminées ou, plus simplement, recevoir la couleur des traits.

Toutes les méthodes de saisie utilisées produisent une grande quantité de bords dont l'intérieur n'est pas visible à la résolution d'affichage normale du dessin. Il nous a paru utile de les marquer pour éviter de leur appliquer certains traitements: gouachage, comparaisons entre bords. Il ne s'agit pas vraiment d'une vérification mais plutôt d'une phase de préparation aux étapes suivantes.



Bords minces

Figure 3.12.

Ainsi il est possible de vérifier automatiquement que l'opérateur de gouachage (cf.3.6.) a bien affecté une couleur à chaque zone fermée du dessin. Cette vérification constitue une aide automatique à son travail, mais n'a de sens que si elle signale à l'opérateur des zones visuellement significatives lors de l'affichage final. Notons qu'une telle zone peut ne pas être vue à la résolution d'affichage du poste de gouachage, sa détection entraînera par exemple un zoom automatique sur la partie de l'image concernée.

Sur la proposition de Marc Bertier [AHR85], nous avons utilisé un test de recherche d'objet mince, qui a donné des résultats satisfaisants. Pour tester si un contour polygonal est mince, à une définition donnée, c'est à dire avec un seuil choisi, nous utilisons le rapport de sa surface à son périmètre. Si ce rapport est inférieur au seuil, nous considérons que le polygone est mince. Ceci correspond à un polygone dont la surface est petite relativement à son périmètre.

Ainsi, pour un rectangle de côtés A et B, le rapport vaut AB/2(A+B). Un rectangle est mince lorsque la longueur d'un des côtés est très supérieure à celle de l'autre; par exemple si A est très supérieur à B, le rapport vaut approximativement B/2. Pour un seuil égal à un, les rectangles minces sont ceux qui ont un côté inférieur à deux unités de l'échelle choisie.

Elimination des petits bords

La détection précédente permet d'envisager d'éliminer les petits bords pour réduire les données et faciliter les traitements ultérieurs. Nous n'avons pas implémenté cette élimination. Le marquage nous ayant donné des résultats satisfaisants en temps de calcul et en facilité d'implémentation.

Ce problème est proche de celui reencontré dans la littérature sous le nom de squelettisation [ARC81][SHA81]. Les méthodes de squelettisation s'appliquent à n'importe quel type de contours. Le squelette d'un contour mince est en gros une ligne médiane allant d'un bout à l'autre du contour. En remplaçant un contour mince par son squelette, on élimine les contours minces de la carte, qu'il faut alors recalculer.

3.4.3.1. RECAPITULATIF

Les deux méthodes utilisant la liste des sommets de la carte planaire tendent à créer de nombreuses petites zones lorsque deux contours sont proches l'un de l'autre sur une longueur importante. Cela tend à se produire essentiellement dans deux cas:

-> un seul contour possédant un angle aigu (cf.fig.3.10.); ce cas est gênant car alors, soit on marque les petites zones ainsi créées, et elles prennent la couleur des traits, soit on ne les marque

pas et on multiplie le nombre de zones que l'opérateur de gouachage doit traiter. La première solution nous a paru acceptable sur nos tests.
-> des portions de deux contours sont à peu près parallèles; ce cas est courant pour les saisies par pictogramme codées et vectorisées; en effet, un trait épais du dessin original conduit à la fabrication de deux parois presque parallèles; ce cas n'est pas gênant, puisque les zones ainsi créées sont des morceaux de bords minces, eux-mêmes minces et détectés comme tels.

3.4.4. FERMETURES INTERACTIVES

Des ouvertures sont visibles à la résolution d'affichage du poste de contrôle. Il est nécessaire de fournir des outils logiciels à l'opérateur de contrôle pour détecter ces ouvertures et assurer les fermetures. Les outils construits utilisent une table à numériser. Certains d'entre eux peuvent servir également à détecter et corriger des ouvertures non visibles.

Ce problème se pose aussi pour les systèmes actuellement commercialisés pour l'aide au dessin animé. Une des solutions adoptées agit en amont du système informatique. Il s'agit d'imposer aux dessinateurs des normes de qualité lors de la préparation des dessins. Il semble malgré tout qu'une proportion importante des dessins doit être corrigée.

Il nous a paru utile de faciliter la tâche du vérificateur en tirant parti d'informations contenues dans la carte. Ces méthodes, pour être compatibles avec une interaction, nécessitent l'affichage très rapide des données utilisées. Les chiffres donnés au sujet du test d'animation montrent que l'affichage d'un dessin de 1000 segments sur une mémoire d'image en liaison rapide avec un ordinateur prend quelques secondes. Ce temps est bien compatible avec l'interaction. Sur une configuration disposant d'une mémoire d'image avec processeur spécialisé pour le tracé de vecteurs, nous avons observé un affichage en moins d'une seconde pour 1000 segments d'une longueur moyenne de 50 pixels.

La première méthode envisagée n'utilise que l'affichage rapide des arêtes de la carte. On affiche la carte bord par bord. Cela permet à l'opérateur de visualiser chaque zone fermée connue du logiciel. Nous n'avons pas approfondi l'ergonomie de cette méthode. Il sera ainsi peut-être préférable d'afficher la carte complète, puis d'effacer les bords et de les réafficher un par un. Avec un écran couleur, on pourra afficher la carte dans une couleur, puis chaque bord dans une autre couleur. Cette méthode semble efficace, mais fastidieuse.

Avec une mémoire d'image disposant d'un affichage rapide de polygones remplis en couleur, il est possible d'afficher la carte en donnant à chaque bord une couleur différente. D'un coup d'oeil, l'opérateur peut détecter les anomalies. Ce type d'affichage est tout à fait compatible avec la méthode de gouachage décrite plus loin, cette vérification pourra donc être menée simultanément au gouachage; elle nécessite un poste de travail plus coûteux.

Nous avons vu comment aider l'opérateur à détecter une anomalie. Nous pouvons l'aider ensuite à trouver la source exacte de l'anomalie: il lui faut quelquefois trouver une toute petite ouverture entre deux zones qui devraient être distinctes.

Nous avons défini plus haut les arêtes pendantes et vu leur corrélation avec les ouvertures intempestives. Leur affichage de façon distincte (clignotement, couleur différente des autres arêtes...) peut servir de guide visuel à l'opérateur.

Il est également possible d'afficher une fenêtre sur la carte qui englobe la zone anormale. L'agrandissement obtenu peut mettre en évidence la fuite. Enfin, on peut parcourir le bord suspect en promenant une "loupe" sur son pourtour. Il s'agit d'afficher des fenêtres successives sur la carte, centrées sur les arêtes du bord à vérifier.

3.4.5. RESULTATS DES TESTS

La plupart des images obtenues au cours des essais n'ont fait appel qu'à trois des techniques énoncées ci-dessus, les résultats de ces dernières étant satisfaisants. Il s'agit de la fermeture automatique par voisinage entre points, suivie d'un affichage en couleur avec ajout interactif de segments.

Quelques images ont été plus difficiles à fermer. Le recours aux différentes méthodes interactives proposées a suffit. Il sera nécessaire de soigner la conception du poste d'interaction pour que le recours à ces outils logiciels soit simple. Dans l'état actuel, les méthodes proposées sont presque toutes isolées les unes des autres. Si la fermeture n'est pas correcte avec les outils de base, le processus devient long et fastidieux.

Les résultats obtenus montrent qu'il n'y a pas de problème insurmontable pour l'utilisation de la représentation par vectogramme et que celle-ci multiplie les problèmes de définition de zones de couleur, mais fournit des solutions impossibles à envisager dans une représentation par pictogramme..

3.5. AFFICHAGE D'UNE CARTE PLANAIRE SUR SURFACE A BALAYAGE

3.5.1. INTRODUCTION

Une couleur est supposée avoir été attribuée à chaque zone d'une carte planaire. Ce paragraphe présente plusieurs méthodes pour afficher une carte en couleur sur un écran à balayage géré par une mémoire d'image.

Une littérature abondante traite du problème de l'affichage d'un ensemble de taches contigues sur une surface à balayage [NEW79] [SEC83] [HEG83]. Les méthodes réalisant un balayage sur l'ensemble des taches, en mettant à jour une liste des arêtes présentes sur la ligne de balayage, sont signalées par ces auteurs comme fournissant les meilleurs résultats sur le plan théorique, mais aussi pratique.

L'algorithme décrit au paragraph 5.2 entre dans cette catégorie. La structure de carte permet d'obtenir par de simples affectations des informations qui, dans les méthodes citées, nécessitaient des opérations de tris sur les arêtes et les sommets.

La méthode tire donc parti du précalcul de la carte. La comparaison avec un autre algorithme dépend beaucoup du contexte d'utilisation. En effet, soit la carte a déjà été calculée à d'autres fins et les opérations en résultant sont partagées par plusieurs applications, soit la carte

doit être calculée pour la mise en couleur et il faudrait intégrer l'analyse de l'efficacité du calcul de la carte dans l'analyse de l'efficacité du remplissage proposé.

Pour notre part, la carte servant tout au long du processus, seule l'efficacité du remplissage proprement dit sera étudiée. Nous renvoyons le lecteur aux publications présentes [GAN84] et à venir sur les cartes planaires pour l'efficacité du calcul d'une carte planaire.

La méthode permet l'antialiassage, par le calcul d'une image suréchantillonée que l'on filtre pour l'afficher. Le calcul fournit l'image ligne par ligne et il est possible de stocker des informations indiquant si un groupe de pixel à filtrer est uni ou non. Le filtrage n'est donc effectué qu'aux endroits où il est indispensable: transition entre zones de couleurs différentes.

Autres méthodes d'affichage envisageable

La méthode d'affichage proposée ne tire parti d'aucune particularité du périphérique d'affichage. Il peut être souhaitable lors des diverses interactions sur les images de prendre en compte de telles particularités pour obtenir un affichage très rapide.

On trouve principalement deux types d'affichages rapides sur de nombreuses mémoires d'image:

- l'affichage d'images codées par plage,
- l'affichage de polygones.

Nous montrons plus loin que l'algorithme proposé permet de générer des images codées par plages. Le cas des affichages de polygones est plus complexe. En effet, l'intégration de ce processus à la capacité de traitement de la mémoire d'image peut avoir été faite à divers niveaux:

- affichage de polygones convexes seulement; il faut découper les polygones non convexes en polygones plus petits avant de les transmettre [WEI80][HEG83],
- affichage de polygones à frontière connexe seulement; dans notre cas, le respect de cette contrainte est simple, il suffit d'afficher les contours en descendant dans l'arbre des bords,
- affichage de polygones fournis dans les coordonnées de l'écran sans dépassement hors des limites de l'écran; il faut alors assurer un conversion des

coordonnées utilisateurs en coordonnées de l'écran et couper les polygones ainsi définis par les limites de l'écran [FOL82][NEW79],
- affichage de polygones quelconques fournis dans un repère quelconque, moyennant une matrice de changement de repère transmise au préalable [LER84].

Sans avoir cité tous les cas, on peut se rendre compte que le prétraitement préalable à la transmission du polygone est très variable d'une configuration à l'autre. Les terminaux à venir respectant la norme GKS [GKS] seront de la dernière catégorie mentionnée.

En dehors d'une interaction, où quelques défauts de l'image peuvent ne pas être gênants, l'affichage polygone par polygone pose plusieurs problèmes. Il travaille au niveau de l'écran et rend difficile l'antialiassage. Comme nous connaissons les polygones de façon précise, il est cependant possible d'afficher d'abord brutalement les polygones, puis de traiter de façon plus précise les pixels traversés par des arêtes. Cette méthode n'est avantageuse sur celle que nous proposons au paragraphe 3.5.2 que si l'affichage des polygones par le matériel est très rapide. En effet, dans le cas de suréchantillonage l'algorithme décrit ci-après permet également de ne filtrer que les pixels traversés par une arête. Le deuxième problème posé par l'affichage par polygone est le fait que les mémoires d'images n'assurent pas toute un remplissage correct: deux tracé de deux polygones voisins laisse parfois des trous; ce problème est lié à des questions de choix de méthodes de tracé de segment sur lesquelles nous ne reviendrons pas ici [FOL82].

Pour finir, signalons une méthode n'entrant dans aucune des catégories ci-dessus et présentant des avantages pour le gouachage (cf.parag.3.6). Cette méthode consiste à afficher les arêtes des polygones puis balayer l'image en affectant une même valeur à tous les pixels appartenant à une même zone connexe; la quatre connexité sur une grille de points est utilisée, les points d'arêtes servent de limite et gardent leur valeur initiale. Une analyse locale de l'image analogue à celle définie au chapitre 2 pour le codage permet de faire cette opération en deux passes sur l'image. Ce type d'affichage pose des problèmes évidents pour l'antialissage des traits; des études récentes fournissent des ébauches de solution [FIS84].

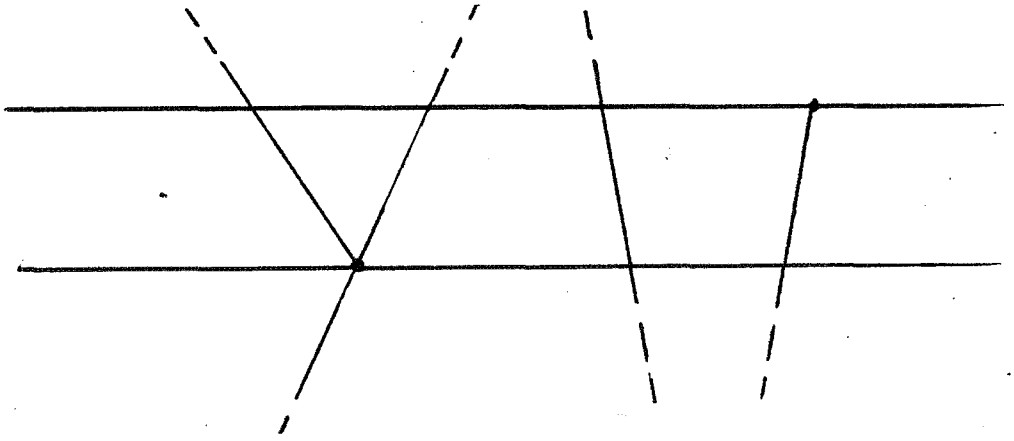
3.5.2. L'ALGORITHME D'AFFICHAGE ET SES PARAMETRES

3.5.2.1. Principe

Nous allons construire ligne par ligne l'image à afficher. Pour cela, nous allons balayer la carte planaire comme elle l'a été pour sa construction: balayage par une droite parallèle à un axe déplacée parallèlement à l'autre axe.

Nous disposons de la liste des sommets de la carte ordonnée suivant ce balayage. Elle va servir pour définir les positions successives données à la droite de balayage.

Observons la tranche de carte comprise entre deux droites de balayages calées sur deux sommets successifs de la carte. Par définition de la carte, les arêtes traversant cette tranche ne peuvent se couper que sur une des droites de balayage. Les arêtes traversant la tranche précédente traversent aussi celle-ci, sauf si elles se terminent sur le premier des deux points. Les arêtes n'existant pas dans la tranche précédente et traversant la tranche courante, partent nécessairement du premier des deux sommets. La figure 3.13 illustre l'allure d'une telle tranche.



Tranche entre deux sommets d'une carte planaire

Figure 3.13.

Dans une telle tranche, il existe un ordre invariant entre les arêtes qui la traverse (arêtes actives). On peut donc y définir une liste d'intervalles; chaque arête défini un intervalle entre l'arête précédente et elle. La structure de carte nous permet de trouver dans quel bord est situé l'intervalle associé à une arête, puisqu'il existe des pointeurs de chaque arête vers les bords

auxquels elle appartient.

La mise à jour de cette liste d'intervalles doit être faite au franchissement de chaque sommet de la carte, en supprimant les intervalles correspondant aux arêtes disparaissant sur ce sommet et en ajoutant les intervalles liés à l'introduction des arêtes naissant en ce sommet.

Nous venons de définir un découpage de la carte en tranches indépendantes de la résolution d'affichage choisie et sur lesquelles sont définies des listes d'intervalles. Le calcul de l'image pour un échantillonage donné induit un découpage supplémentaire de la carte; en plus des droites de balayage passant par un sommet, la carte va être découpée par des droites correspondant aux transitions entre les lignes de pixels.

Une ligne de pixels sera ainsi composée d'une ou plusieurs tranches sur lesquelles seront définies des listes d'intervalles. Le calcul de la ligne de pixel sera fait en traitant chacune des tranches la composant.

Dans la méthode choisie, un traitement préparatoire est fait sur chaque tranche composant la ligne. Celle-ci est ensuite calculée en combinant les résultats préparatoires.

Le traitement préparatoire d'une tranche consiste à marquer les pixels traversés par les arêtes actives de la tranche et à mémoriser le bord dans lequel est contenu le premier pixel non marqué suivant chaque groupe de pixels marqués. Tous les pixels non marqués suivants un tel pixel appartiennent au même bord que lui.

Suivant la qualité des résultats souhaités, la combinaison des données ainsi obtenues sur chaque tranche se fera de façon plus ou moins sophistiquée; en effet, la même structure de base de l'algorithme permet notamment de calculer la contribution exacte de chaque bord à chaque pixel, seule les procédures **traite_tranche** qui stocke des données en vue de l'affichage, et **affiche_ligne** qui utilise ces données doivent être modifiées. La même structure générale peut être concervée pour balayer et afficher plusieurs cartes à la fois, avec une priorité d'affichage et une liste de zones transparentes pour chaque carte. Dans la version de base de l'algorithme, présentée en détail plus loin, nous nous sommes contentés d'afficher en noir tous les pixels traversés par une arête et d'attribuer à chaque pixel la couleur du dernier bord qui lui a été attaché.

3.5.2.2. Les paramètres

Ils règlent l'affichage.

Une fenêtre permet de définir quelle partie des données doit être affichée. Cela permet de stocker des scènes plus grandes que la zone d'affichage et de n'en afficher qu'une partie à chaque instant. Nous avons contraint cette fenêtre à être parallèle aux axes; elle est donc définie par un quadruplet de réels (xmin,xmax,ymin,ymax). Pour afficher une fenêtre non parallèle aux axes, il faudra appliquer la transformation appropriée sur les données avant le calcul de la carte. Ceci ne constitue même pas un obstacle à une interaction, puisque les cartes se calculent en quelques secondes.

Une cloture sur la zone d'affichage permet de définir sur quelle partie de l'écran on veut projeter la fenêtre sur les données. Nous fournissons également un paramètre règlant de quelle façon la proportion hauteur/largeur de la fenêtre sera ajustée sur la proportion correspondante de la cloture :

- agrandir la fenêtre pour atteindre la proportion de la cloture,
- diminuer la fenêtre pour atteindre la proportion de la cloture,
- pratiquer une homothétie entre la fenêtre et la cloture.

Un échantillonage en x et un en y permettent de définir le nombre de sous-pixels calculés par pixel affiché pour une image suréchantillonée, ou le nombre de fois qu'on répète chaque pixel calculé pour une image souséchantillonée. Nous avons imposé à ces valeurs d'être des multiples ou des diviseurs du nombre de pixel affichés dans les directions correspondantes.

Une liste de numéros de couleur indique que les zones affectées de ces numéros sont transparentes. Ceci permet de conserver une image déjà affichée pour certaines parties de la nouvelle image calculée.

3.5.2.3. Les variables locales

La conjonction entre les différents paramètres permet d'ajuster les variables xpas et ypas internes à l'algorithme d'affichage. Ces variables définissent les dimensions d'un rectangle sur les données correspondant à un pixel de l'image calculée.

La variable sommet_courant est un pointeur sur un sommet de la liste de sommets de la carte. Elle indique que ce sommet est le prochain à atteindre au cours du balayage de la carte.

La variable x_courant contient l'abscisse de sommet_courant.

La variable x_ancien sert à mémoriser la position précédente de la ligne de balayage sur la carte. Le couple x_ancien, x_courant défini une tranche sur la carte qu'il est facile de traiter pour l'affichage.

La variable x_ligne indique l'abscisse de la fin de la ligne de pixel en cours de calcul.

La variable premier_intervalle est un pointeur sur le premier d'une liste d'intervalles. Un intervalle est un enregistrement défini en Pascal par: pt_interv = intervalle;

```
intervalle = record
                    arete:pt_arete;
(* pointe sur l'arête qui termine l'intervalle *)
                    precedent,suivant :pt_interv;
(* pointent sur l'intervalle précédent et l'intervalle
suivant *) end;
```

Deux variables marq_bord et marq_ligne permettent de stocker des informations préparatoires à l'affichage. Dans l'implémentation actuelle ces variables sont de types tableau. Les tableaux ont été dimensionnés par le nombre maximum de pixel qu'on a supposé souhaitable de calculer (4096 dans notre implémentation). Ces tableaux seront utilement remplacés par une structure dynamique, car ils sont creux; ainsi l'échantillonage ne sera pas limité par l'implémentation.

La variable marq_bord est une table d'entiers, on y stockera des numéros de bords de la carte. L'élément i de cette table, associe au pixel i de la ligne en cours de calcul le numéro du bord dans lequel il est contenu. Un

numéro négatif indique qu'aucun bord n'a encore été assoscié à ce pixel.

La variable marq_ligne est une table de booléens. L'élément i de cette table indique si le pixel i de la ligne en cours de calcul est traversé par une arête de la carte ou pas. La variable marq_nul est une variable annexe permettant de réinitialiser rapidement marq_ligne. Pour optimiser le filtrage dans le cas d'un suréchantillonage perpendiculaire au sens du balayage, il faut gérer plusieurs tables analogues à marq_ligne (une par ligne de pixels calculée).

La variable no_ligne indique le numéro de la ligne de pixel en cours de calcul.

Les algorithmes suivants sont donnés dans leur implémentation en langage Pascal. Ils correspondent à la version de base du programme ni suréchantillonée, ni souséchantillonée, sans gestion de zones transparentes, les pixels traversés par une arête sont affichés en noir. La gestion de ces variantes s'obtient par des extensions simples des algorithmes fournis mais qu'il aurait été long et fastidieux de décrire en totalité. Des optimisations sont possibles; elles nécessitent l'adjonctions de variables annexes qui auraient obscurci la présentation du principe utilisé. Elles ne font pas appel à des techniques de programmation sophistiquées, mais correspondent juste au stockage temporaire de valeurs calculées plusieurs fois dans l'implémentation présentée. Par exemple, on pourra gérer incrémentalement l'évolution des intersections des arêtes avec la ligne de balayage.

3.5.2.4. L'algorithme

PROCEDURE PRINCIPALE

```
begin
  with fenetre do
  begin
    (*   initialisation balayage      *)
    x_ligne:=xmin;
    premier_intervalle:=nil;
    sommet_courant:=cartavoir^sommet_un;
    if sommet_courant<>nil then x_courant:=sommet_courant^
    else x_courant:=xmax+1;
    (* on change de sommet courant jusqu'à en trouver un
    fenetre *)
    while x_courant<xmin do
      if sommet<>nil then
      begin
        gestion_intervalles(premier_intervalle);
        sommet_courant:=sommet_courant^.x_suivant;
        if sommet_courant<>nil then x_courant:=sommet_cour
        else x_courant:=xmax+1;
      end;
    (* on initialise les tables de marquage *)
    for k:=0 to nx_pix-1 do
    begin
      marq_nul[k]:=false;
      marq_bord[k]:=-1; (* numero impossible pour un bord
    end;
    no_ligne:=0;
    (*  traitement ligne par ligne     *)
    while x_ligne < xmax do
    begin
      marq_ligne:=marq_nul;
      x_ancien:=x_ligne;
      x_ligne:=x_ligne+x_pas;
      while x_courant < x_ligne do
      begin
        traite_tranche(x_ancien,x_courant);
        x_ancien:=x_courant;
        if sommet_courant<>nil then
        begin
          gestion_intervalles(premier_intervalle);
          sommet_courant:=sommet_courant^.x_suivant;
          if sommet_courant<>nil then x_courant:=sommet_co
          else x_courant:=xmax+1;
        end;
      end;
      traite_tranche(x_ancien,x_ligne);
      affiche_ligne(no_ligne);
      no_ligne:=no_ligne+1;
    end;
  end;
end;
```

TRAITEMENT D'UNE TRANCHE

procedure traite_tranche(xi,xf:real);

Traitement préparatoire à l'affichage d'une ligne de pixels; il s'agit d'etudier ce qui se passe dans une tranche définie plus haut,

Cette procédure est faite n+1 fois par ligne de pixel où n est le nombre de sommets contenus dans la ligne; sur l'ensemble de l'image cette procédure est donc effectuée Nt fois avec:
Nt = Ns + Nl
où Ns = nombre de sommets de la carte planaire
Nl = nombre de lignes sur l'image calculée

L'expression de cette procédure en Pascal est:

```
begin
  premier_bord_courant;
  met_a_jour_marq; end; (* traite_tranche *)
```

avec les procédures **premier_bord_courant** et **met_a_jour_marq** suivantes:

procedure met_a_jour_marq:

Pour chaque arête coupant la tranche, on va marquer les pixels qu'elle traverse dans marq_ligne et marquer dans marq_bord le numéro du bord suivant l'arête dans le sens du balayage; à chaque franchissement d'une arête, la variable intervalle_courant est mise à jour.

procedure premier_bord_courant:
Cherche le bord englobant le premier pixel de la ligne calculée; le résultat n'a de sens que lorsque cette définition a un sens, c'est à dire lorsque le premier pixel de la ligne est entièrement contenu dans un bord de la carte et non pas partagé entre plusieurs bords.
Cette procédure s'exécute en deux étapes:
1)trouver l'intervalle contenant le premier pixel
2)trouver le bord entourant cet intervalle
Le résultat est une mise à jour de bord_courant et de intervalle_courant

CALCUL D'UNE LIGNE DE PIXELS

Nous présentons ici une méthode pour fabriquer une ligne de pixels à partir des informations préparées dans marq_ligne et marq_bord. Une ligne de pixels est représentée ici par une table d'autant d'entiers qu'il y a de pixels dans la ligne. Ce choix correspond au périphérique d'affichage utilisé pour nos essais.

Les informations receuillies dans marq_ligne et marq_bord pourraient servir à générer directement un codage par plage de chaque ligne. L'efficacité de l'algorithme d'affichage devient donc à partir d'ici liée au périphérique utilisé.

```
procedure affiche_ligne(no_ligne:int);
var i, couleur_courante, ligne:integer;

begin
  couleur_courante:=bord_courant^no_coul_bord;
  for i:=0 to ny_pix do
  begin
    if marq_ligne[i] then couleur_courante=arete_coul
    else if (couleur_courante=arete_coul) then
    begin
      if marq_bord[i-1]>=0 then couleur_courante:=marq_bor
      else couleur_courante:=arete_coul;
    end;
    tab[i]:=couleur_courante
  end;
  envoi_au_perif_d_affichage(tab);
end;          (* affiche_ligne   *)
```

Complexité

Les évaluations qui suivent correspondent à l'affichage d'une carte entièrement contenue dans l'écran. Des optimisations des algorithmes précédents sont possibles, notamment en affichant rapidement les lignes entièrement contenues dans le bord le plus externe de la carte; il s'agit en particulier des lignes avant le premier sommet et après le dernier sommet.

C = Ns * (G + T) + Nl * (T + A) où
Ns est le nombre de sommets de la carte affichée
Nl est le nombre de lignes de l'image calculée
G est la complexité de la procédure gestion_intervalles
T est la complexité de la procédure traite_tranche
A est la complexité de la procédure affiche_ligne

Dans l'implémentation actuelle, A est proportionnel au nombre de pixels par ligne de l'image calculée, T est proportionnel au nombre moyen d'arêtes par ligne, G est proportionnel au nombre moyen d'arêtes se terminant par sommet (Na/Ns).

Résultats

La structure de l'algorithme proposé prend en compte les résultats publiés sur l'affichage d'un ensemble de taches polygonales. Elle se prête bien à de nombreuses variantes d'implémentation.

Sur les exemples traités, à résolution fixée, le temps d'affichage varie à peu près linéairement en fonction du nombre de sommets de la carte. Sur les mêmes données, augmenter la résolution, conduit à augmenter linéairement le temps consacré à G et T. Le temps consacré à A est intimement lié à la configuration utilisée et à la procédure calcule_ligne qui en découle; dans notre cas, ce temps augmente proportionnelement au nombre de pixels.

3.6. GOUACHAGE

3.6.1. RESULTATS A ATTEINDRE-METHODES UTILISEES

Un dessin est supposé affiché sur un écran. Nous étudions ici les méthodes pour associer une couleur à chaque zone fermée de ce dessin.

La gouache est étalée au dos des cellulos traditionnels. Les gouacheurs, munis de pinceaux, traitent ainsi chaque cellulo l'un après l'autre.

L'opération informatique correspondante prend une forme analogue quoique moins laborieuse. Chaque dessin devra être affiché sur un écran en couleur. Le dessin apparaîtra d'abord en noir sur fond blanc. Un opérateur sélectionnera une couleur et désignera sur l'écran un point de la zone devant prendre cette couleur.

Pour cela, on distingue trois opérations principales :

- affichage initial des dessins,
- remplissage des zones par des couleurs,
- correction des zones mal fermées.

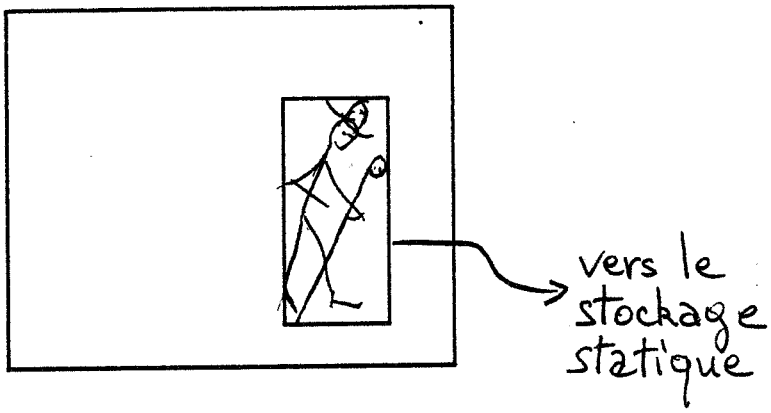
Actuellement les systèmes d'aide à la réalisation de dessins animés basés sur la préparation traditionnelle des dessins utilisent une représentation numérique des images par pictogramme [STE81][FIS84]. Dans ces systèmes, l'affichage des dessins n'est en fait que la lecture du pictogramme initialement saisi par caméra.

La mise en couleur des dessins se fait:

- soit avec des méthodes de remplissage de zone par propagation de couleur; ces méthodes travaillent directement sur le pictogramme. A partir d'un point désigné par un opérateur, on étend la couleur dans toutes les directions jusqu'à rencontrer des points définis comme frontière de la zone. Dans notre cas, les points frontières sont les sombres traits du dessin. Cette méthode permet de traiter assez simplement les zones mal fermées en rajoutant dans le pictogramme les traits nécssaires. Des méthodes, travaillant directement sur des images antialiassées, permettent de changer la couleur d'une région en maintenant

l'antialiassage [FIS84].
- soit par des méthodes de coloriage à priori. Dans toutes les méthodes, chaque zone n'est en général pas remplie directement par la représentation numérique de la couleur, mais par un indice sur une table où figure une représentation numérique de la couleur: triplet (rouge, vert, bleu), triplet (luminance, teinte, saturation). La méthode d'affichage décrite à la fin du paragraphe 3.4 peut servir à cet effet. La plupart des mémoires d'image actuelles gérent matériellement cette indirection (table de couleur). Ainsi, il suffit de changer la représentation numérique associée à un indice de la table pour changer la couleur affichée de tous les points qui ont reçus cet indice. Ceci permet d'afficher l'image en ayant affecté un numéro arbitraire à chaque zone connexe de l'image. La table de couleur contenant au départ du noir pour les numéros correspondants aux traits et du blanc pour les autres numéros, l'image apparaît au trait noir sur fond blanc. L'affectation d'une couleur à une zone n'est en fait qu'une modification de la description de couleur associée au numéro de cette zone, elle s'opère donc instantannément. Cette méthode pose un problème pour la correction des zones mal fermées; en effet, toute adjonction de traits de fermeture nécessite le recalcul du remplissage à priori, ce qui rend difficile la conservation de l'information sur les couleurs déjà attribuées à des zones. Une alternative consiste à mixer cette méthode et la précédente: les remplissages avant correction sont assurés à priori, les remplissages après correction sont assurés par propagation.

Ensuite, le pictogramme en couleur est stocké. Pour limiter la place qu'il occupe, diverses solutions sont adoptées. La première consiste à ne stocker qu'une fenêtre sur l'image; cette fenêtre englobe toute la partie dessinée de l'image (figure 3.13.). De plus, on stocke en fait un codage par plage de ce pictogramme [WAL81].



Stockage de la portion utile d'un pictogramme

Figure 3.13.

Les remplissages matériels étant désormais très rapides, il peut être avantageux de ne conserver que la version noir et blanc du dessin à laquelle on ajoutera les coordonnées des points désignés par l'opérateur et la référence à une couleur associée à chacun de ces points.

Les couleurs affectées à un personnage figurant sur un cellulo donné sont connues à l'avance. Elles ont été choisies lors de la définition des modèles (cf.chap.2). Elles pourront donc être affichées avec une indication de leur destination dans une colonne de menu (figure 3.15.). Ceci n'est possible qu'en combinant des informations contenues dans une représentation informatique des modèles et de la feuille de prise de vue. En effet, il faut trouver quel personnage apparaît sur quels dessins (feuille de prise de vue), puis chercher les couleurs associées à ce personnage (modèles).

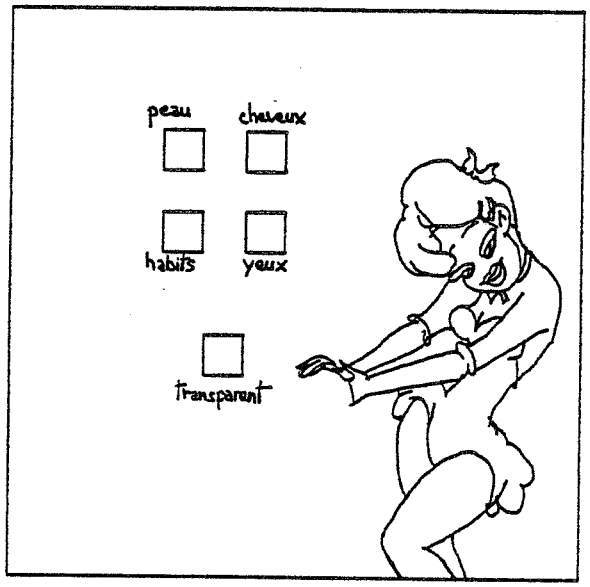


Figure 3.15.

Il est nécessaire de pouvoir attribuer la "couleur" transparente à des zones, puisque l'affichage pour le tournage se fait en assemblant plusieurs dessins sur un fond.

Il n'est pas nécessaire d'antialiasser l'image pendant son gouachage. Moins de couleurs différentes sont donc nécessaires sur le poste de gouachage que sur le poste de tournage. De plus, le gouacheur ne manipule que des références à des couleurs: la couleur affichée ne sert que de controle visuel à son travail. On peut se contenter de moins de nuances pour ce controle que pour l'affichage final. Une table matérielle de 256 couleurs parait largement suffisante.

Alors, chaque pixel de l'image en cours de gouachage est représenté par huit bits; ils servent d'adresse dans une table de couleur associée à chaque dessin.

3.6.2. GOUACHAGE D'UN VECTOGRAMME

Il est possible d'afficher le vectogramme sur une mémoire d'image et d'appliquer sur le pictogramme obtenu les méthodes mentionnées ci-dessus. Stocker l'image en couleur sous forme de pictogramme nous ferait perdre au dernier moment les avantages de la compacité du

vectogramme pour le stockage et les traitements.

Les principales caractéristiques des méthodes décrites ci-dessus peuvent être concervées pour le gouachage d'un vectogramme: affichage avec gouachage à priori, désignations de zones pour l'affectation des couleurs. Le fonctinnement externe du poste de gouachage reste identique; la représentation interne des images est beaucoup plus compacte: liste de lignes brisées et, par exemple, couples point de désignation-couleur.

Nous proposons de réaliser l'affichage de la carte planaire du dessin par une des méthodes décrite aux paragraphes 4 et 5, et en utilisant le principe de coloriage à priori décrit au paragraphe précédent. Une façon de le faire est de parcourir l'arbre des bords de la carte en affectant un numéro à chaque bord interne (zone). On ajoutera au stockage du vectogramme les coordonnées des points désignés par l'opérateur de gouachage et la référence à une couleur associée à chacun de ces points. Les coordonnées de ces points sont connues dans le repère de l'écran; au prix d'un changement de repère, il est préférable de les stocker dans le repère des données. On obtient ainsi un gouachage plus indépendant des conditions d'affichage.

Notons une particularité du dernier affichage proposé au paragraphe 4. Celui-ci réalise un coloriage à priori indépendant de la carte et dépendant de la résolution d'affichage. Il en résulte que deux zones connexes différentes sur l'écran au sens de la quatre connexité, peuvent ne correspondre qu'à une zone de la carte planaire. Ceci est une voie pour la détection interactive des zones mal fermées. En effet, l'opérateur va attribuer une couleur à chacune des deux zones de l'image affichée et lorsqu'on va reporter dans la carte un couple (numéro de zone, couleur), il va y avoir une contradiction: une même zone de la carte reçoit se voit affecter deux numéros de zone différents. La détection de ce fait peut être l'occasion de déclencher un processus de troncature automatique du bord là où il passe près de lui-même.

L'opérateur choisit une couleur et désigne une zone où il doit la mettre. Les coordonnées du point, obtenues en coordonnées d'écran, sont transformées en coordonnées du point correspondant de l'espace des données. Une fonction sur la carte permet de savoir à l'intérieur de quel bord se trouve ce point. Il suffit alors d'associer dans une table le numéro de ce bord et la couleur à y mettre. Pendant l'interaction, une image de cette table est contenue dans la table de couleur du périphérique d'affichage. En chargeant la couleur à l'emplacement

convenable de la table de couleur, la zone désignée apparaît instantanément dans la couleur choisie.

La table associant une couleur à chaque numéro de bord d'un dessin suffit à sauver le gouachage de ce dessin.

Notons que, par rapport aux méthodes précédentes, cette méthode facilite le gouachage des petits bords puisqu'elle permet le zoom par programme sur des parties du dessin. L'information manipulée n'est à aucun moment liée à la définition du périphérique d'affichage ou à son mode de fonctionnement.

Il est facile de détecter que l'opérateur n'a pas attribué de couleur à un bord. Pour cela, on tiendra à jour une table de booléens indiquant pour chaque bord si il a reçu une couleur ou non. Il est évident que cette information n'a d'intérêt que pour les bords suffisament grands pour être affichés. Nous avons vu une méthode pour les reconnaître (cf.parag.3).

Ainsi, le vectogramme se prête aussi bien que le vectogramme à des opérations interactives de gouachage; de plus, il permet l'utilisation d'outils supplémentaires pour faciliter le travail, du fait notamment de la connaissance explicite du contour de chaque zone. La qualité de l'image finale n'est pas restreinte par la qualité de l'image au moment du gouachage: en particulier, il est simple à partir des données obtenues au gouachage de calculer une image en couleur antialiassée par suréchantillonage et filtrage. Enfin, même associé à des couleurs, il permet le stockage des dessins sous une forme très compacte.

Les opérations de gouachage sont simples. Elles peuvent être gérées sur un micro-ordinateur personnel muni d'une mémoire d'image couleur, et d'un disque dur pour le stockage. Il est ainsi possible de réaliser des postes de gouachage relativement peu coûteux.

3.7. TOURNAGE

Toutes les méthodes de préparation de l'animation décrites précédemment permettent de produire un ensemble d'images fixes en couleur. Au cours du tournage, on obtient la reproduction de ces images sur un support diffusable à 25 images par seconde ou plus: film, bande ou disque vidéo, générateur informatique spécialisé.

La tâche d'assembler des dessins et des décors pour chaque image d'une séquence d'animation a toujours été un sérieux problème par les méthodes conventionnelles de banc-titrage et a contribué au prix élevé de la production d'animation. De plus, deux contraintes matérielles limites les images qu'il est possible de créer ainsi:

- au-delà de 5 cellulos, la couche d'acétate ainsi constituée manque de transparence et conduit à une dégradation excessive de l'image;

- les mouvements relatifs des images et de la caméra sont limités par les caractéristiques mécaniques des dispositifs de tournage utilisés (banc-titre: cf.ch.1)..

Une heure de dessin animé nécessite la composition d'environ 40000 images à partir de 120000 dessins et de décors. Si cinq minutes sont nécessaires pour composer et filmer chaque image, il faut 150 jours, 24 heures sur 24, pour tourner le film.

Les solutions retenues pour la composition informatique des images doivent donc conduire à minimiser ce temps. Elles peuvent répondre à des critères de qualité différent : images antialiassées ou non, tournage en vidéo ou sur film... En pratique, actuellement, le problème du temps d'affichage et d'assemblage des images est une sérieuse limitation au choix du support final. Parmi les méthodes de report citées au premier chapitre, seules celles utilisant la vidéo permettent un rythme de production compatible avec les conditions concurentielles courantes de production. Par exemple, le traçage sur film d'une image couleur à l'aide d'un COM demande plusieurs minutes pour une image 4000x4000.

Dans la fin de ce chapitre, les cartes planaires sont supposées être la représentation numérique des cellulos. Les deux termes sont donc considérés comme synonymes pour

simplifier l'exposé.

Jusque là, la méthode informatique proposée est proche du processus classique de production de dessin animé: traçage, puis gouachage cellulo par cellulo. Il faut maintenant assurer l'assemblage des cellulos. Nous supposerons que la composition des images est définie sur une feuille de prise de vue classique qu'on envisagera d'informatiser par la suite.

En suivant cette feuille, l'opérateur peut rappeler le décor et les cellulos de l'arrière vers l'avant. La méthode la plus simple d'affichage consiste à afficher brutalement les dessins colorés les uns sur les autres; le masquage est assuré automatiquement par l'ordre d'affichage et l'utilisation de couleurs définies comme transparentes. Cette méthode ne permet pas d'assurer un antialiassage correct entre les cellulos. Si ce dernier est nécessaire il faudra donc procéder différemment.

Une variante de cette méthode consiste à l'appliquer à une image sur-échantillonée puis filtrée. On part du décor. Puis on affiche sur lui le premier dessin par la méthode de balayage de la carte planaire décrite plus haut. Pour les sous-pixels transparents ainsi générés, on utilise la couleur du décor à l'emplacement correspondant. L'image obtenue est considéré comme nouveau décor pour les dessins suivants. Les études sur l'antialisaage ont montré que dès que plus de deux polygones chevauchent un pixel, l'antialiassage obtenu par une telle méthode n'est pas mathématiquement exact et qu'il a des résultats fâcheux dans certains cas.

La méthode la plus correcte consisterait à calculer une carte planaire obtenue par concaténation des différents dessins dans un ordre défini. L'opération élémentaire est la concaténation de deux cartes, une liste de bords de l'une étant réputé cacher l'autre. L'affichage de cette carte résultante permettrait un antialiassage correct, puisqu'elle donne une représentation dans l'espace des données du résultat de la composition.

Cette concaténation a été implémentée pour une autre application, mais n'a pas encore été intégrée à la nôtre. L'algorithme utilise un balayage des dessins à assembler simultané au balayage d'une carte calculée à partir de tous les segments de ces dessins; ce balayage est analogue à celui présenté ci-dessus pour l'affichage d'une carte planaire sur un écran raster. L'opération peut être vue comme le découpage d'un groupe de polygone par un autre; Weiler [WEI80] et Hegron [HEG83] ont présentés des

résultats sur ce sujet.

Dans l'implémentation et avec la configuration matérielle actuelle, il faut environ 40 secondes d'affichage par dessin. Si on admet qu'en moyenne chaque image est composée de trois dessins, il faut environ 2 minutes pour composer chaque image, auquel il faut ajouter le temps d'enregistrement, que nous suposerons de 30 secondes. Dans une optique industrielle, il est possible d'optimiser le programme d'affichage écrit en Pascal standard. Le tournage ainsi défini peut être totalement automatisé; il permettra alors de tourner 48 secondes de dessin animé par jour. Avec un dispositif d'affichage de technologie plus récente que celui de notre configuration, on peut estimer à plus d'une minute par jour et donc 30 minutes par mois la possibilité de production au niveau du poste de tournage. Nous compléterons cette argumentation au chapitre de conclusion.

L'existence de l'opération de concaténation permettrait d'adapter à nos données les propositions de Wallace [WAL81] pour l'optimisation du tournage. Il s'agit d'éviter de répéter des traitements identiques sur plusieurs images successives. Une telle méthode n'est applicable qu'en relation avec une informatisation de la feuille de prise de vue. Le principe est de faire un super-cellulo d'un groupe de cellulos voisins se répétant sur plusieurs images successives; ce super-cellulo temporaire a l'apparence du groupe qui lui a donné naissance et la représentation en machine d'un seul cellulo.

Remarque:
Wallace propose dans ce même article une méthode de transparence au bord des zones pour traiter les problèmes d'antialiassage sur l'image composée. Cette méthode pourrait être adaptée à nos données. Elle peut se généraliser pour obtenir des effets spéciaux (fumée, vitre, flou de vitesse...). Une couleur est alors (R,V,B,transparence). Lorsqu'on fait une concaténation de cellulos, il faut alors tenir compte de la transparence. Cette méthode est reprise te complétée dans [FIS84].

Les optimisations de la composition des images pour le tournage passent par l'exploitation intelligente de la feuille de prise de vue. Nous constatons une fois de plus que la simulation informatique de ce document constitue une des priorités à venir.

3.8. CONCLUSION

Nous avons vu dans ce chapitre comment obtenir les images prêtes pour le tournage à partir des dessins en noir et blanc.

Nous avons vu que le vectogramme se prête bien au tournage sur des périphériques de type très différents (vidéo, film) . Cette représentation des images fourni des résultats de qualité au moins équivalente à ceux obtenus avec les pictogrammes et pour des durées de traitement probablement inférieures. Une évaluation précise reste à faire, mais suppose une meilleure information sur les systèmes opérants sur des pictogrammes, que celle que j'ai pu réunir.

REFERENCES BIBLIOGRAPHIQUES DU CHAPITRE

[ARC81] [BAU84] [BEN79] [BER73] [COM84] [COR80] [COU79] [FER84] [FIS84] [GAN84] [GKS84] [HEG83] [JAC69] [LUC81] [MIT82] [NAC83] [NEW79] [PAV81] [SAV84] [SEC83] [SHA81] [SHO83] [SPI83] [STE79] [TAL85] [WAL81] [WEI80]

CHAPITRE 4

COMPARAISON ENTRE LES IMAGES D'UNE SEQUENCE

Un des axes de recherche en analyse de scènes bi ou tri-dimensionnelles est l'analyse de séquences d'images et plus particulièrement la détection des invariants d'une image à l'autre. De nombreuses applications sont concernées: robotique, génie biomédical, surveillance... Beaucoup de ces applications nécessitent l'analyse en temps réel des séquences reçues par un dispositif vidéographique.

Dans le cadre de notre application (dessin animé), l'objectif est plus limité. Nous disposons pour chaque image de la description sous forme de liste de segments de droite des contours la composant. Nous allons tenter de reconnaître d'une image à la suivante les zones qui peuvent être appariées. Nous pourrons ainsi, par exemple, propager un attribut de chaque zone de la première image vers la ou les zones correspondantes des images suivantes. L'application principale de ces résultats concernera le transport de l'attribut d'une zone d'une image à l'autre en vue de l'automatisation de l'opération de gouachage. Ces mêmes résultats devraient aussi améliorer les méthodes d'intervallage automatique, en les rendant moins contraignantes lors de la saisie des dessins.

Nous ne prétendons pas apporter de résultats nouveaux en analyse de séquence d'images, mais plutôt montrer comment des résultats connus peuvent être utilisés dans notre application.

4.1. INTRODUCTION

Le problème que nous envisageons ici complète naturellement l'exposé du gouachage des dessins animés tel qu'il a déjà été abordé (cf.ch.3 parag.6) Nous pensons que la forte cohérence qui existe d'une image à l'autre dans une séquence animée peut être exploitée pour propager les couleurs de la première image vers les suivantes de la séquence. Nous tentons d'aborder ce problème de facon plus générale.

> "Les tentatives pour décrire les phénomènes dynamiques enregistrés sur des séquences d'images tendent à modéliser les scènes représentées sur ces images comme une configuration d'objets qui ont des (séquences d') états bien définis" [NAG81].

Les travaux sur des séquences d'image peuvent être regroupés suivant deux techniques. L'une considère l'image globalement (table de points) et étudie l'évolution temporelle de cet ensemble (différence d'une table à la suivante...). L'autre technique tend à détecter des contours sur chaque image, puis à étudier l'évolution temporelle de ces seuls contours [HUA84].

Pour notre application, nous avons supposé que nous disposions pour chaque image de la description des contours présents. Nous allons tenter de reconnaître d'une image à la suivante les zones qui peuvent être appariées. Nous pourrons ainsi propager la couleur, ou tout autre attribut: texture, fonction d'évolution de chaque zone de la première image vers les zones correspondantes des images suivantes, mais aussi calculer éventuellement des interpolants entre zones appariées.

Nous avons recherché une heuristique qui donne des résultats satisfaisants dans un certains nombres de cas prédéfinis et fréquents: mouvements relativement lent, style graphique proche du dessin animé classique. Cette heuristique est prévue pour indiquer à un opérateur les correspondances sur lesquelles pèsent la plus forte incertitude.

2. SITUATION DU PROBLEME

2.1. HYPOTHESE DE REPRESENTATION DES DONNEES

On fait l'hypothèse que les images sont connues sous forme d'un ensemble de segments de droite. Les algorithmes basés sur l'analyse de séquences d'images connues sous forme de table de pixels n'ont pas été approfondis.

Ce choix a été fait en corrélation avec d'autres travaux sur l'informatisation de la production de dessins animés. Cette hypothèse est vérifiée par les méthodes de saisies de dessins de base de certains systèmes ([COM84], [COU80]), nous avons montré dans les chapitres précédents les avantages de cette représentation et proposé une méthode pour passer d'une représentation par table de pixels à une représentation par segments de droite.

Les travaux sur le gouachage des images de dessins animés ont jusqu'ici pris deux directions:

*un opérateur travaille image par image sur des images connues sous forme de table de pixels; les temps observés sont de deux à douze minutes par image pour du cartoon classique. Ces temps sont mesurés sur notre implémentation. Ils semblent liés à l'opération elle-même: un dessin du type traité comprend de 15 à 30 zones; l'opérateur doit indiquer une couleur, puis désigner la zone où affecter cette couleur; si cette opération élémentaire prend 8 secondes, le gouachage d'un dessin de 15 zones prend 2 minutes.

*interpolation automatique entre formes décrivant des dessins clés; ces formes ont une couleur d'où sont déduites les couleurs des formes intermédiaires.

Le gouachage image par image reste un procédé lourd. L'interpolation automatique répond bien à certains types d'animation, mais ne donne pas satisfaction à un grand nombre d'animateurs; leur travail, leur talent est justement de donner de la vie à toutes les images intermédiaires.

Nous proposons donc que, de façon assez traditionnelle, tous les dessins soient réalisés par les animateurs, mais que le processus de gouachage soit traité semi-automatiquement par une heuristique travaillant sur des séquences entières, sous le contrôle d'un opérateur.

Pour mémoire, voici les résultats énoncés par Markle [MAR84] au sujet d'un système de mise en couleur semi-automatique de films tournés en noir et blanc; ces résultats ont été obtenu sur un film de Laurel et Hardy et correspondent donc à des séquences avec peu ou pas de mouvements de caméra:

* 96 % des pixels d'une image prennent la même valeur sur l'image suivante.
* si un pixel non compris dans les précédents est entièrement entouré de pixels de la même valeur, on obtient de bons résultats en lui attribuant automatiquement cette valeur
* la coloration d'une séquence de film prend 10 secondes par image en moyenne, compte tenu de la coloration interactive de la première image de la séquence.

4.2.2. DIFFICULTES

La forte cohérence d'une image à la suivante dans une séquence d'images est vérifiée pour la plupart des séquences de dessin animé. C'est en particulier vrai pour le dessin animé commercial (long métrage ou série) pour lequel le volume de dessins produit impose une simplification et une homogénéisation des styles graphiques entre les intervenants (ch.1 fig. 6).

La cohérence visuelle que nous cherchons à exploiter est difficile à traduire par des critères numériques. Une manifestation courante des problèmes posés est la suivante: un ensemble de traits, définissant une zone visuelle, peuvent ne pas définir un contour fermé (ch.2 fig. 2). Nous avons vu au chapitre précédent des méthodes de fermeture de zones.

Par la suite nous supposerons que les zones devant être fermées le sont. Par contre, nous devons prendre en compte les fusions ou scissions possibles entre deux zones de même couleur intervenant lors d'un changement d'image (fig. 4.1).

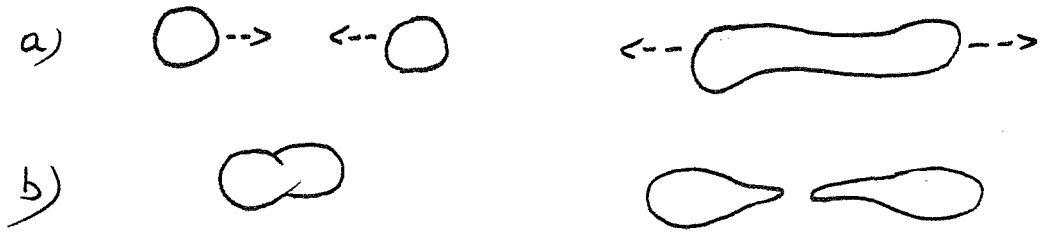


Figure 4.1.

La propagation ne peut pas être totalement automatique. D'une image à la suivante, des zones entrent dans l'écran par les côtés, d'autres apparaissent au milieu de l'écran à la suite d'un mouvement qui s'y produit. Il s'agira par exemple d'un personnage pénétrant dans l'écran ou d'un oeil apparaissant lorsqu'un visage de profil se tourne pour se présenter de face. Il sera nécessaire d'attribuer interactivement des couleurs à ces zones.

De plus, la masse de documents à traiter oblige à un traitement rapide de chacun. Il est donc hors de question de faire des analyses très sophistiquées sur chaque séquence d'image. En conséquence, les résultats obtenus ne sont pas entièrement satisfaisants. L'amélioration des techniques d'analyses connues et celle des performances des machines devraient permettre d'améliorer nettement le rendement de la propagation semi-automatique des couleurs. Pour l'instant, nous nous sommes plutôt intéressé à des méthodes rapides, permettant une propagation sous contrôle interactif d'un opérateur.

Nous pensons qu'il est plus facile de gouacher progressivement une image vierge, quitte à disposer d'aides puissantes pour le faire, que de vérifier après coup les couleurs attribuées automatiquement aux zones d'une image par un processus sophistiqué, même s'il donne des résultats satisfaisants dans un nombre très important de cas. Il est possible de mélanger les deux méthodes, en aidant le gouachage interactif par des informations calculées auparavant par un programme automatique.

Supposant les zones de chaque image connues, le traitement à effectuer peut être décomposé en trois processus:

-> la sélection des paires de zones à comparer,
-> la comparaison proprement dite,
-> la décision en fonction des résultats de la comparaison.

Ces trois opérations peuvent être très imbriquées. Par exemple, le résultat de la comparaison de deux zones peut être une indication précieuse pour la sélection des prochaines zones à comparer. Il est probable qu'à l'avenir, l'utilisation des techniques développées en intelligence artificielle facilitera la construction de tels processus.

La comparaison pourra se faire sur des critères de plusieurs types:

-> métrique: surface, périmètre, dimension et position de la boîte englobante,
-> topologique: zones voisines, zones incluses, zone englobante,
-> dynamique: fonction d'évolution des différentes zones, jusqu'à l'image de référence.

Les résultats qui suivent constituent une validation préliminaire de ces concepts.

4.3. STRUCTURATION

4.3.1. CARTES PLANAIRES

La structuration d'un ensemble de traits sous forme d'une carte planaire a paru un outil puissant pour la comparaison d'images successives. En effet la carte nous fournit de nombreuses informations topologiques et métriques sur les images étudiées.

Les cartes planaires obtenues sur des images de dessin animé ne dépassent pas cinq niveaux pour l'arbre des bords. Une bonne finesse de représentation est obtenue avec environ 1000 points, 1000 arêtes et 100 bords par dessin. Ces chiffres correspondent à une représentation statique d'environ 8Ko et une représentation dynamique d'environ 80Ko. Plus de la moitié des bords sont des bords minces (cf.ch.3 parag.4) qu'on peut négliger lors de la phase d'analyse et de reconnaissance de formes. On leur attribuera la couleur des traits de séparation entre zone ou la couleur de bords voisins (ch.3 fig.2).

L'hypothèse est que la description d'une image sous forme de carte planaire fournit une information assez stable d'une image à la suivante dans une séquence. Les principales exceptions correspondent aux modifications très rapides de l'image et aux styles graphiques touffus.

La notion de zone a été présentée au chapitre 3 paragraphe 3. Disposant de N cartes planaires, représentant N images, l'exploration des arbres des bords de ces cartes doit permettre d'associer chaque zone de la première carte avec une ou plusieurs zones de chacune des cartes suivantes. Nous nous sommes limités à N=2; ce choix a simplifié les tests, mais nous a privé des critères dynamiques de comparaisons.

4.3.2. PRINCIPE DES METHODES MISES EN OEUVRE POUR LA COMPARAISON ENTRE CARTES PLANAIRES

Les solutions élémentaires consistent à comparer les cartes planaires correspondant à deux dessins successifs d'une séquence. Nous n'avons pas abordé les solutions qui travailleraient sur toute la séquence, en cherchant à tirer parti de données dynamiques globale. Pour ces solutions, on construit une représentation de l'évolution entre les deux premières images pour étudier la troisième. A chaque nouvelle image, on affine la représentation de l'évolution; cette représentation aide à la compréhension des images suivantes [NAG84].

Les deux implémentations utilisées commencent par une même sélection des zones à comparer: on élimine de toute comparaison les zones minces (cf.ch3.parag.4). Dans une implémentation, aucune autre sélection n'est effectuée. Dans l'autre, les comparaisons ne sont effectuées qu'entre des zones d'un même niveau de l'arbre des zones (cf.ch.3 parag.3); nous détaillons cette méthode au paragraphe 4.2. de ce chapitre.

En deuxième lieu, la comparaison entre deux zones est réalisée en deux étapes:

-> pour un ensemble de critères, on compare les deux zones; le résultat de chaque comparaison est une valeur entre 0 et 1; le résultat 1 correspond à l'identité entre les deux zones du point de vue du critère; le résultat 0 correspond à la différence maximum entre les deux zones.,
-> la combinaison des résultats précédents permet d'établir une évaluation, comprise entre 0 et 1, de la correspondance éventuelle entre les deux zones considérées.

En troisième lieu, la décision de propager la couleur d'une zone de la première carte vers une zone de la deuxième est prise au cours d'un processus interactif. Les deux cartes sont affichées sur un écran couleur; en

utilisant la table de couleur du matériel (cf.ch.3 parag.6), on affiche les deux cartes avec les traits noirs sur fond blanc. Pour chaque zone de la première carte, on va faire les opérations suivantes:

-> envoyer la couleur de la zone à son adresse dans la table de couleur,
-> envoyer la même couleur à l'adresse de la zone de la deuxième carte supposé lui correspondre; ainsi, les deux zones apparaissent à l'écran avec la même couleur,
-> si l'opérateur valide la correspondance, on enregistre la couleur pour la zone de la deuxième carte, puis on passe à la zone suivante de la première carte,
-> si l'opérateur invalide la correspondance, il doit alors fixer lui-même la couleur du zone de la deuxième carte.

Les critères retenus pour établir les comparaisons vont être présentés, puis les structures des deux implémentations utilisées pour les tests seront explicitées.

4.3.2.1. Les comparaisons élémentaires

Critères métriques

La surface et le périmètre de deux zones peuvent être comparés. De plus, ils permettent de calculer un critère de forme: la circularité est le rapport de la surface d'un polygone au carré de son périmètre. C'est un nombre sans dimension, donc indépendant de l'échelle de définition de la forme. Le cercle possède la circularité maximale ($1/4\pi$). Ces trois valeurs peuvent être calculées une fois pour toutes et attachées à la carte qu'elles concernent; elles ne sont pas recalculées à chaque comparaison entre zones.

Levine, Lenoble et Youssef [HUA83] mettent en évidence l'efficacité et les problèmes des critères ci-dessus. Ils montrent notamment qu'il est aisé de construire des polygones ayant même surface et même périmètre, donc même circularité, mais de formes très différentes. Ils suggèrent de faire une discrimination supplèmentaire à l'aide de la moyenne des angles entre arêtes successives du contour (courbure moyenne). Ce critère n'a pas été utilisé, d'autres critères demandant moins de calculs ayant donné des résultats satisfaisant pour l'utilisation interactive qui en est faite.

Pour chacune de ces valeurs, nous avons utilisé un critère de comparaison basé sur la différence relative entre la valeur du critère sur la zone de la carte 1 et celle du critère pour la zone de la carte 2:

1 - $f(v1,v2) = |v1-v2| / \max(v1,v2)$

Plus les valeurs sont grandes, plus cette fonction est significative. En effet, une faible variation de surface, par exemple, est d'autant moins sensible qu'elle se rapporte à une grande surface. Il en résulte que la fonction f proposée de meilleurs résultats pour les zones de plus grand périmètre ou surface.

Recouvrement

Une façon de vérifier que deux zones ont des positions proches consiste à évaluer leur surface commune. Pour simplifier, la surface commune de leurs boîtes englobantes parallèles aux axes a été calculée. Cette simplification diminue le temps de calcul et ne nous a pas paru pénalisante dans la majorité des cas traités.

Ce critère a d'autant plus de sens que les zones observées sont grandes. En effet, si on observe un objet dont la surface et le diamètre sont grands relativement à l'écran, et un objet petit se déplaçant à la même vitesse, le petit objet peut se déplacer de plusieurs fois sa plus grande dimension tandis que dans le même temps que d'une portion de sa plus petite dimension (fig.4.2)..

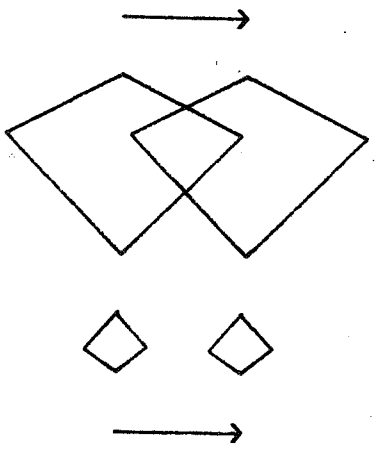


Figure 4.2.

Germe de couleur

Le paragraphe 6 du chapitre 3 montre comment a été effectué le gouachage de la première carte: un opérateur a désigné sur un écran un point de chaque zone pour y

affecter une couleur. Si les coordonnées des points ainsi désignés (germes) ont été conservés, ils peuvent être utilisés pour le gouachage de la carte suivante. On suppose ici qu'un point qui était dans une zone de la première image est probablement dans la zone correspondante de l'image suivante. Encore une fois, pour des raisons analogues à celles invoquées pour les critères précédents, cette supposition n'est acceptable que sur des grandes zones.

Cette utilisation pose un problème: que se passe-t-il lorsqu'on compare la deuxième carte avec sa suivante? Utiliser les germes de la première carte pour gouacher la troisième ou les suivantes conduit à des erreurs excessives. Il est nécessaire de faire évoluer les germes de carte en carte avec la zone qu'ils désignent. Nous proposons comme invariant de chaque germe ses coordonnées relatives à la boîte englobante de la zone qu'il désigne. Cette méthode peut être adaptée aux méthodes de gouachage actuellement en usage (cf.ch.3 parag.6), à condition de connaître pour chaque image la boîte englobante de chaque zone connexe de couleur uniforme.

Le nombre insuffisant de tests effectués sur cette méthode ne permet pas d'affirmer qu'elle donne de bons résultats. Combinée à d'autres critères, en particulier sur la taille des zones considérées, elle est sans doute une des voies à explorer.

Critères topologiques

Une carte planaire fournit des informations de type topologique. Ainsi, l'arbre des bords représente les inclusions entre zones; plus on s'éloigne de la racine de cet arbre, plus on rencontre des zones de petite taille..

Nous avons signalé que la plupart des critères précédents donnent de meilleurs résultats pour les zones de grande taille. Ceci a suggéré le principe de la deuxième implémentation de la propagation semi-automatique: commencer la comparaison à la racine de l'arbre et la poursuivre par niveaux de descendants (cf.ch.3 parag.3).

Le dual d'une carte planaire, exprimant les relations de voisinages entre zones a été construit afin d'exploiter la cohérence d'une image à la suivante. La construction de cette structure s'est avérée trop coûteuse en temps de calcul pour s'intégrer utilement dans nos implémentations

interactives.

Au sujet d'autres critères

Les publications sur l'analyse de scènes et la reconnaissance de formes mentionnent quantité de critères permettant d'évaluer la ressemblance ou la correspondance entre deux formes. Nous avons volontairement utilisés les critères les plus simples; ce choix s'est avéré positif, puisque les calculs mis en oeuvre sont compatibles avec l'interaction et que cette dernière permet de rattraper les erreurs dues à la grossièreté des moyens de comparaison employés.

4.4. IMPLEMENTATIONS

4.4.1. CONSTRUCTION ELEMENTAIRE

Nous avons vu qu'un dessin de dessin animé comprend de 15 à 30 zones visibles de couleur uniforme. Une solution élémentaire pour comparer deux dessins consiste à comparer deux à deux toutes les zones des deux images en établissant ainsi une table de comparaison. L'utilisation de la structure de carte planaire donne un accès facile à la description géométrique de chaque zone. Les lignes de la table correspondent aux zones de la première carte. Les colonnes de cette table correspondent à celles de la seconde carte. On range dans l'élément de la ligne i et de la colonne j de la table la correspondance entre le i-ième élément de la première carte rencontré lors d'un parcours ordonné de l'arbre des bords et le j-ième élément de la deuxième carte.

Nous utilisons une table 50x50. Les critères de comparaison sont: la variation relative de surface, la variation relative de périmètre, la variation relative de circularité, le recouvrement des boîtes englobantes relatif à la surface de la plus grande des deux zones.

Le temps de ce calcul s'est avéré compatible avec l'interaction. Pour chaque zone de la première carte, seule la zone de la deuxième ayant la meilleure correspondance est proposé à l'opérateur. Les erreurs d'une comparaison aussi grossière sont aisément corrigées interactivement. Le nombre de zones correctement appariées nous a paru permettre un gain de temps de gouachage, bien qu'une estimation sur un grand nombre de dessin n'ait pu

être faite.

Les photos C à F -page 55 chapitre 2- présentent une séquence d'image qui nous a permis de valider la méthode. Toutes les grandes zones du corps et de la tête y ont été correctement appariées d'une image à la suivante.

4.4.2. IMPLEMENTATION STRUCTUREE

Considérons deux cartes visuellement très semblables. On va tenter d'identifier les portions d'image en correspondance d'une carte à l'autre. Pour cela, on va parcourir en parallèle l'arbre des bords des deux cartes en construisant un arbre image de l'arbre de la deuxième carte et contenant des informations sur les correspondances possibles.

Les types de base servant à construire cet arbre en Pascal sont:

```
pt_cousinage = ^cousinage;
pt_liste_de_cousins = ^liste_de_cousins;
liste_de_cousins = record
                     bord_de_première_carte:pt_bord;
                     evalue_correspondance:real;
                     liste_suite:pt_liste_de_cousins;
                   end;
cousinage = record
              fils,frère,père:pt_cousinage;
              liste_de_cousins_possibles:pt_liste_de_cousi
            end;
```

On fait deux opérations sur cet arbre:

-> création dirigée par un critère: ayant un critère de comparaison entre deux zones, on construit l'arbre de comparaison entre les deux cartes; cette construction n'est pas réciproque: arbre(c1,c2)<>arbre(c2,c1).

La comparaison est établie depuis la racine. On compare une fratrie de la première carte C1 avec une fratrie de la deuxième C2. On compare la descendance d'une zone Z2 de C2 à la descendance d'une zone Z1 de C1 pour laquelle la correspondance de (Z2,Z1) est maximum. A chaque noeud N de l'arbre A de comparaison est attachée une liste des zones de C1 pouvant correspondre à la zone de C2 obtenue par un parcours de l'arbre des bords de C2 analogue à celui qui permet d'aller de la racine de A au noeud N.

-> Modification dirigée par un critère: ayant construit un arbre de comparaison de C1 avec C2, on veut modifier cet arbre en fonction d'un nouveau critère: ajout, modification ou suppression de correspondances possibles.

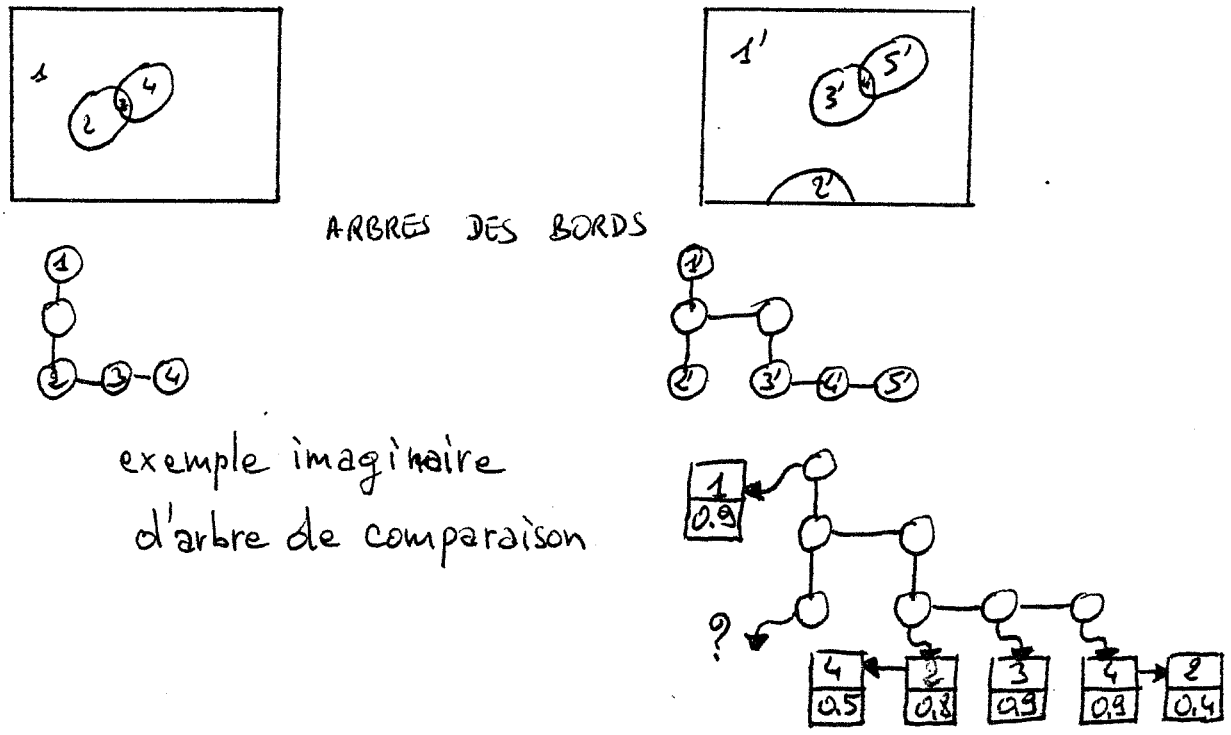


Figure 4.3.

Des essais ont été effectués avec une construction de l'arbre de comparaison dirigée par un critère composé: variation relative de surface, de périmètre et de circularité, recouvrement. Cette méthode permet de limiter le nombre de comparaisons effectuées, mais nécessite la gestion d'une structure plus sophistiquée que celle de la première implémentation. Les résultats obtenus sont analogues dans les deux méthodes.

La méthode pourra évoluer par la définition d'un ensemble de règles agissant sur un noeud de l'arbre ou induisant un parcours dans l'arbre, ceci afin de diminuer le nombre de comparaisons effeçtuées en augmentant l'exploitation de la structure de carte planaire.

Une autre évolution possible consiste à calculer les comparaisons en temps différé, à stocker ces comparaisons, puis à les exploiter de façon interactive. Par exemple, supposons un gouacheur travaillant 8 heures par jour et 2 minutes sur chaque dessin; les 16 heures restantes, le système peut calculer les comparaisons et bénéficier de 4 minutes de calcul par paire de dessins successifs. Dans ces conditions, des méthodes sophistiquées de comparaison pourraient être exploitées. En particulier, on pourra alors exploiter des résultats connus sur les cartes planaires [JAC69] [SHO83].

4.5. AUTRE UTILISATION DE LA COMPARAISON DE DESSINS

Plusieurs méthodes d'intervallage automatique entre dessins ont été proposées [BUR82] [BAE69] [COM84]. Elles permettent à l'animateur de ne donner que quelques dessins pour définir une séquence. Ces dessins correspondent grossièrement à ce que sont les dessins clés en animation traditionnelle (cf.ch1). Les autres dessins sont calculés.

Toutes ces méthodes reposent sur une saisie contraignante: d'un dessin à l'autre, les traits qui se correspondent doivent être décrits dans le même ordre; le système opère ensuite des interpolations plus ou moins sophistiquées sur ces traits.

Nous proposons une saisie quelquonque des dessins conduisant à leur représentation par des vectogrammes (cf.ch.2). La construction de la carte planaire (cf.ch.3) permet alors de désigner facilement des zones à apparier pour l'intervallage. Le temps passé par un opérateur pour réaliser cette mise en correspondance peut être réduit par une comparaison préalable automatique des dessins.

6. CONCLUSION

Les méthodes simples de comparaison entre dessins proposées dans ce chapitre permettent de réduire le temps nécessaire pour le gouachage d'une séquence de dessin animé. Une évaluation sur un grand nombre de séquences est nécessaire pour valider réellement la méthode proposée, dans un cadre opérationnel.

Il est cependant intéressant de constater ce qui se passe fonctionnellement: une proposition de couleur est faite à l'opérateur de gouachage, sa validation éventuelle est très rapide; si la proposition est fausse, l'opérateur l'invalide, puis sélectionne la bonne couleur. On peut donc comparer la méthode classique (cf.ch.3) à la méthode avec proposition automatique:

-> méthode classique: choix d'une couleur, désignation d'une zone,
-> méthode semi-automatique: validation ou invalidation et choix d'une couleur.

Cette technique n'aura une réelle efficacité que dans un environnement soigneusement étudié pour permettre à l'opérateur de valider ou d'invalider une propagation de couleur après un simple coup d'oeil sur un écran. Ceci est nécessaire pour compenser le fait qu'en principe un contrôle sur une image demande un effort d'attention supérieur à celui nécessaire pour une désignation.

BIBLIOGRAPHIE DU CHAPITRE

[BAE69] [BUR82] [COM84] [COU80] [HUA83] [JAC69] [NAG81] [SHO83]

CHAPITRE 5

CONCLUSION

Nous avons étudié aux chapitres précédents diverses techniques informatiques permettant d'utiliser le vectogramme comme représentation des images de dessin animé.

Nous avons montré au chapitre 2 comment une représentation des dessins par des vectogrammes peut être obtenue. Il est clair que pour une exploitation intensive de cette représentation, il faudra approfondir les voies suggérées pour l'automatisation de la saisie. Nous avons vu que la représentation obtenue est très compacte, peut être très précise, mais qu'elle pose un problème encore mal résolu pour la numérisation des traits à épaisseur variable.

Le chapitre 3 a permis de mettre en évidence que la représentation par vectogramme se prête bien au test d'animation, en exploitant les possibilités d'affichage rapide de segments sur les mémoires d'image de technologie récente. Dans ce même chapitre, des méthodes aboutissant à l'obtention de l'image en couleur prête pour le tournage ont été présentées. Il apparait que les méthodes proposées pour l'affichage et l'assemblage final sont encore un peu lentes; une évaluation précise d'une version optimisée du programme d'affichage sur une mémoire d'image de type bit-map va être effectuée. Les résultats obtenus sur notre configuration sont prometteurs.

Nous avons introduit au chapitre 4 une méthode permettant d'accélérer le gouachage de séquences de dessins, en exploitant la structuration de ceux-ci permise par leur représentation sous forme de vectogramme. Nous

avons ainsi montré que la cohérence temporelle entre les dessins peut être exploitée de façon simple.

A notre avis, l'ensemble des résultats obtenus est un encouragement à poursuivre l'étude de représentations de type vectogramme pour les applications travaillant sur un grand nombre d'images composées de plages de couleur unie. Un effort important doit être fait pour l'amélioration des méthodes de saisie, en particulier par leur automatisation.

La saisie sur table à numériser n'est envisageable que si elle permet un gain suffisant de temps sur l'ensemble de la réalisation. En particulier, les vectogrammes peuvent être utilisés par des méthodes d'interpolation automatique. L'évaluation des temps nécessaires pour passer du dessin sur papier à l'image en couleur n'a pas été faite. Les essais effectués permettent cependant de donner quelques indications sur ce que serait une configuration informatique utilisant les outils présentés dans les chapitres précédents.

Nous allons décrire différentes configurations matérielles adaptées aux différentes étapes de la production du dessin animé: saisie, gouachage, tournage. Le poste de tournage doit fonctionner automatiquement; il doit disposer d'un dispositif de stockage statique important; sur la base des chiffres proposés au chapitre 3, nous envisagerons un poste permettant le tournage de 45 secondes d'animation par jour. A 3 dessins par image et 12 images par seconde, cela représente environ 1500 dessins à préparer sur les autres postes: saisie, vérification, gouachage.

Le gouachage des dessins, effectué lors des tests, demandait de l'ordre de 4 minutes par dessin si les couleurs du dessin étaient pré-sélectionnées. Le gouachage étant interactif, nous suposerons une journée de 8 heures de travail par opérateur; chaque opérateur devrait ainsi pouvoir gouacher environ 120 dessins par jour.

La production de 1500 dessins demande donc environ 12 postes de gouachage. Nous avons vu que cette opération demande peu de calcul et une mémoire d'image de qualité réduite. On pourra, par exemple, établir chaque poste sur un micro-ordinateur muni d'une mémoire d'image permettant d'afficher rapidement des polygones quelconques. La préférence sera accordée à des micro-ordinateurs munis d'un système d'exploitation multi-tâche. En effet, on pourra ainsi tirer parti de la faible utilisation de l'unité de traitement par le processus interactif de

gouachage pour opérer en tâches de fond la plupart des traitements automatiques préalables au gouachage. La désignation des zones par l'opérateur pourra être faite avec un joystick, une souris, ou une table à numériser.

La saisie des dessins peut être réalisée soit avec un scanner, soit avec une table à numériser. Dans le cas de la table à numériser, il faut de 2 à 10 minutes de saisie par dessin; nous prendrons une moyenne de 5 minutes par dessin; ceci amène à environ 100 dessins saisis par jour et par personne. Il faut donc 15 postes de saisie. Un poste de saisie pourra être composé d'un micro-ordinateur personnel muni d'un écran bit-map noir et blanc d'assez bonne définition (environ 500x500) et d'une tablette à numériser. Un tel poste ne subit pas la gêne d'un système multi-utilisateur (cf.ch.2). Il sera avantageux d'utiliser une interface avec la tablette à numériser possédant sa propre capacité de traitement: filtrage, callage, changement de repère.

La saisie par scanner est semi-automatique, mais ne fourni pas directement de vectogramme. Chaque saisie prend environ 1 minute, soit environ 500 dessins saisis par jour. Les traitements ultérieurs sur le résultat de telles saisies demandent un complément d'étude: il est envisageable d'adapter tous les traitements du chapitre 3 au résultat d'un codage analogue à celui du chapitre 2. Ce travail restant à faire, nous ne définirons pas plus une configuration basée sur ce principe.

Nous venons d'énoncer ce qui nous parait la principale voie d'étude à explorer en prolongement des travaux effectués. Par ailleurs, un important travail d'ingénirie est à fournir pour évaluer précisément les coûts et les performances d'installations informatiques analogues à celle que nous venons de définir. Une telle installation repose sur la possibilité de communication entre des appareils aux performances diverses, adaptées à la tâche auxquels on les destine. Il est en effet nécessaire que des données transitent entre les postes de saisie, de gouachage et de tournage: la notion de réseau local d'ordinateurs prend donc une place importante dans la définition d'un studio informatique de dessin animé classique.

Nous pensons que l'informatisation d'un studio de dessin animé pose d'importants problèmes de choix des équipements et d'évaluation des investissements. Les exposés techniques des chapitres précédents masquent quelque peu des notions qu'il est indispensable de prendre en compte pour une exploitation en grandeur réelle: formation du personnel, fatigue des opérateurs, fiabilité

du matériel, adaptation du produit à faire à l'outil mis en oeuvre, support final du film. Vu les progrès dans les communications entre ordinateur, il nous paraît souhaitable d'axer les choix vers une structure modulaire, prête à de nombreuses évolutions et pouvant fonctionner dans une configuration minimale très restreinte. Ainsi, l'informatisation pourrait se faire progressivement. Il s'agirait d'acquérir l'équipement minimum permettant de disposer d'un poste de saisie et d'un poste de gouachage -ordinateur, mémoire d'image bit-map couleur, table à numériser- afin d'assurer de petites productions ponctuelles. L'acquisition du poste de tournage se justifierait par la production d'autres types d'images: animation par table de couleurs et par liste d'affichage. La maîtrise des équipements ainsi acquise peu à peu permettrait d'orienter les investissements ultérieurs.

ANNEXE 1

ANNEXE

Ecole des Mines
Dept. Informatique

RESUME

CONFIGURATION MATERIELLE

Les travaux présentés dans ce rapport sont réalisés sur le matériel informatique de l'Ecole des Mines.

l'ordinateur hôte est un Hewlett-Packard 9000 multiutilisateur avec:

. 1 processeur 32 bits

. 1,5 méga octects de mémoire centrale

. 1 disque de 120 méga octets

Tous les logiciels sont développés en Pascal sous HP-UX, système d'exploitation de la famille UNIX (1).

* la mémoire d'image est un LEXIDATA solidview (2). Ces caractéristiques principales sont:

. une résolution de 640 x 512 points,

. une table de couleur de 12 bits en entrée et 3x8 en sortie soit 4096 couleurs représentables,

. 12 plans mémoire accédants la table de couleur,

. 12 plans mémoire de profondeur,

. 4 plans réservés à la superposition (overlay),

. diverses opérations microprogrammées.

* une table à numériser Benson.

---

(1) UNIX est une marque commerciale déposée par A.T. & T.

(2) SOLIDVIEW est une marque commerciale déposée par LEXIDATA.

ANNEXE 2

BIBLIOGRAPHIE

[AGR77] AGRAWALA, ASHOL & alii,
"A sequential approach to the extraction of shape features",
Computer graphics and image processing Vol 6 1977.

[ARC81] ARCELLI C.,
"Pattern thinning by contour tracing",
Computer graphics and image processing Vol 17 1981.

[BCT84] BANC-TITRE,
Mensuel sur le cinéma d'animation,
ed. TARCUS.

[BAU84] BAUDELAIRE P.,
"Etude comparative des systèmes informatiques d'assistance au dessin animé",
Agence OCTET,Ministère de la culture, Paris 1984

[BEN79] BENTLEY J.L., OTTMAN T.A.,
"Algorithms for reporting and counting geometric intersections",
IEEE Trans.on Comput. Vol.C-28,Num.9,1979.

[BER73] BERGE C.,
Graphes,
Gauthier-Villars, Paris 1973.

[BLO82] BLOCH M.,
"Génération de taches bicolores",
Thèse de Docteur-ingénieur, Ecole Nat.des Mines, Saint-Etienne 1982.

[BRI84] BRION P.,
TEX AVERY,
ed. du Chêne.

[BUR76] BURTNYK N. and M.WEIN,
"Interactive Skeleton Techniques for Enhancing Motion Dynamics in Key Frame Information" ,
Communications of the ACM,Vol19,no10,October 1976.

[CED79] CEDERBERG Roger,
"Chain-link coding and segmentation for raster scan devices",
Computer graphics and image processing Vol 10 1979.

[CED81] CEDERBERG Roger,
"On the coding, processing, and display of binary images",
Linkoping Studies in Science and Technology. Dissert. no57 Linkoping University, Sweden.

[COM84] COMPARETTI G.,
"Outil de Production Industrielle de Dessin Animé par Ordinateur"
Premier Colloque Image, CESTA-GRETSI, Biarritz, Mai 1984.

[COR73] CORI R.,
"Un code pour les graphes planaires et applications"
Thèse de doctorat d'Etat, Paris VI, 1973.

[COU81] COUEIGNOUX Philippe,
"Character generation by computer",
Computer graphics and image processing Vol 16,1981.

[COU80] COUPIGNY Francis,
"Animation par ordinateur",
Proc. AFCET, NANCY. 1980

[DAN82] DANIELSON Per-Erik ,
"An improved segmentation and coding algorithm for binary and nonbinary images" ,
IBM J.Res.Develop. vol.26 No 6 Novembre 82.

[FIS84] FISHKIN K.P., BARSKY B.A.,
"A Family of New Algorithms for Soft Filling"
Computer Graphics, Vol.18, Num.3, 1984.

[FOL82] J.D. FOLEY, A. VAN DAM,
Fundamentals of Interactive Computer Graphics,
Addison Wesley, 1982.

[FRE74] FREEMAN Herbert ,
"Computer processing of line-drawing images" ,
Computer surveys Vol 6 No 1 Mars 74.

[GAN84] GANGNET M.,Michelucci D.,
"Un outil graphique interactif ",
Rapport interne,Ecole Nationale des Mines de St Etienne,no 84-12,Octobre 1984.

[GKS84] Computer Graphics, Special GKS Issue, 1984.

[HUA83] HUANG T.S.,
Image Sequence Processing and Dynamic Scene Analysis,
Proc.of NATO study institute 1982, RFA, Berlin Springer 1983.

[IEE80] Proceedings of the IEEE ,
"Spécial issue on image coding",
Vol.68 No 7 1980.

[JAC69] JACQUES A.,
"Constellation et propriété algébriques des graphes topologiques"
Thèse de troisième cycle, Paris VI.

[KLI76] KLINGER A.,Dyer C.R. ,
"Experiments in picture representation using regular decomposition" ,
Computer graphics and image processing Vol 5 1976.

[KNU68] KNUTH D.E. ,
The art of computer programming,
Addison-Wesley, 1968.

[LOZ83] LOZOVER O. and K.PREISS ,
"Automatic Construction of a cubic B-Spline representation for a general Curve" ,
Computers and Graphics,Vol7,no2,pp149-153,1983.

[LUC81] LUCAS M. ,
"La réalisation des logiciels graphiques interactifs" ,
Ed. Eyrolles,Paris,1981.

[LUC83] Conférencier représentant LUCAS FILM,Forum International des Nouvelles Images Informatiques, Mt Carlo,Fevrier 1983.

[LUC84] LUCIANI A., CADOZ C.,
"Modélisation et Animation Gestuelle d'Objets - Le Système ANIMA"
Premier Colloque Image, CESTA-GRETSI, Biarritz, Mai 1984.

[MAR70] MARCHI S.de, AMIOT R.,
Le dessin animé d'amateur,
ed. Paul MONTEL.

[MEY80] MEYER B. et C.BAUDOIN ,
"Méthodes de programmation" ,
Ed.Eyrolles,Paris,1980.

[MIC84] D. MICHELUCCI, M. GANGNET,
"Saisie de Plans à Partir de . Tracés à Main-Levée",
Proc. MICAD'84, Mar. 1984.

[MOI84] MOISSINAC J.C.,
"Codage et analyse d'images par balayage",
Proc.conférence COMPUTIM, EPA, Strasbourg 1984.

[MOR76] MORRIN T.H. ,
"Chain-link compression of arbitrary black-white images" ,
Computer graphics and image processing Vol 5 1976.

[NEW79] W.M. NEWMAN, R.F. SPROUL,
Principles of Interactive Computer Graphics,
McGraw-Hill, 1979.

[ODG81] ODGERS C.R.,
"Criteria for Choosing a Camera for Use in a Video Digitizing System",
Tutorial:Two-Dimensional Computer Animation,Siggraph 1981,pp 117-128.

[PAV82] PAVLIDIS T.,
"Algorithms for Graphics and Image Processing",
Springer-Verlag, 1982.

[PAV83] PAVLIDIS T.,
"Curve fitting with Conic Splines",
ACM TOG, Vol.2, Num.1, 1983.

[PRA80] PRATT W.K.,
"Combined symbol matching facsimile data compression system" ,
Proceedings of the IEEE, Vol 68, no 7, Juillet 1980.

[REE81] REEVES W.T ,
"Intbetweening for Computer Animation Utilizing Moving Point Constraints" ,
Computer Graphics ,Vol15,no3,August 1981,pp263-269.

[SEC83] SECHER E.,
SECHER E., "Conception et réalisation d'un logiciel de saisie et de restitution de cartes élémentaires",
Rapport technique no 28 1983, IRISA, Rennes .

[SHA80] SHAPIRO B., PISA J., SKLANSKY J.
Computer Graphics and Image Processing no 15 1981

[SHO83] SHOUKRY A.A.F.,
"Représentation des problèmes de reconnaissance à l'aide de la théorie des graphes: application à l'analyse topologique et statistique de tracés manuscrits",
Thèse de docteur-ingénieur, Nancy I, 1983.

[SHO79] SHOUP R.G.,
"Color Table Animation",
Comput. Graph. 13-2, Aug. 1979.

[SIG79] Proc.Siggraph 1979-1983.

[SMI81] SMITH A.R ,
"Paint" ,
Tutorial:Two Dimensional Computer Animation Siggraph 1981,pp 36-70.

[STE79] STERN Garland ,
"Softcel- An application of raster scan graphics to conventionnal cel animation" ,
Proc. SIGGRAPH 1979.

[TAL85] TALLOT D.,
"Quelques propositions de mise en oeuvre d'algorithmes combinatoires"
Ecole des Mines de Saint-Etienne, à paraître.

[WAL81] WALLACE B.A. ,
"Merging and transformation of raster images for cartoon animation" ,
Proc. SIGGRAPH 81.

[WAL81] Wallace B.A. ,
"The control of merging and transformation of raster scan images for cartoon animation",
Tutorial Animation SIGGRAPH 81.

[WEI80] WEILER K.,
"Polygon Comparison using a graph representaion",
Computer Graphics Vol14 no3 1980, Proc.Siggraph.

[YAS80] YASUDA Y. ,
"Overview of digital facsimile coding techniques in Japan" ,
Proceedings of the IEEE, Vol 68 ,no 7, Juillet 1980.

dernière page de la thèse

AUTORISATION DE SOUTENANCE

VU les dispositions de l'article 3 de l'arrêté du 16 avril 1974,

VU le rapport de présentation de Monsieur BAUDELAIRE

Monsieur Jean-Claude MOISSINAC

est autorisé à présenter une thèse en soutenance pour l'obtention du diplôme de DOCTEUR-INGENIEUR, spécialité Informatique

Fait à Saint-Etienne, le 20 décembre 1984

Le Directeur de l'EMSE,

M. MERMET
INGÉNIEUR GÉNÉRAL DES MINES
DIRECTEUR
ÉCOLE NATIONALE SUPÉRIEURE DES
MINES DE SAINT-ÉTIENNE

AIDES INFORMATIQUES A LA REALISATION DE DESSINS ANIMES

Jean-Claude MOISSINAC

n° d'ordre: 44II

Mots-clés:
Synthèse d'Images
Dessin animé assisté par ordinateur
Codage d'image
Gouachage d'image
Saisie de dessins
Analyse de séquences d'images

RESUME

Partant d'une description du processus de réalisation classique de dessins animés, ce travail développe des techniques informatiques pour la représentation et la manipulation numérique des dessins: saisie, gouachage, assemblage.

Les techniques étudiées reposent sur la représentation des dessins par des suites de lignes brisées; chaque fois que cela a été possible ces techniques sont comparées aux techniques habituellement utilisées travaillant sur des tableaux de points.