



Génération de polices d'imprimerie pour photocomposeuse digitale

Marc Hourdequin

► To cite this version:

Marc Hourdequin. Génération de polices d'imprimerie pour photocomposeuse digitale. Réseaux et télécommunications [cs.NI]. Ecole Nationale Supérieure des Mines de Saint-Etienne; Institut National Polytechnique de Grenoble - INPG, 1978. Français. NNT : 1911II . tel-00827676

HAL Id: tel-00827676

<https://theses.hal.science/tel-00827676>

Submitted on 29 May 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**ECOLE NATIONALE
SUPERIEURE DES MINES
DE SAINT-ETIENNE**

**INSTITUT NATIONAL
POLYTECHNIQUE
DE GRENOBLE**

N° d'ordre : 11 11

THESE

présentée par

Marc HOURDEQUIN

pour obtenir le grade de

**DOCTEUR-INGENIEUR
"SYSTEMES ET RESEAUX INFORMATIQUES"**

**GENERATION DE POLICES
D'IMPRIMERIE POUR
PHOTOCOMPOSEUSE DIGITALE**

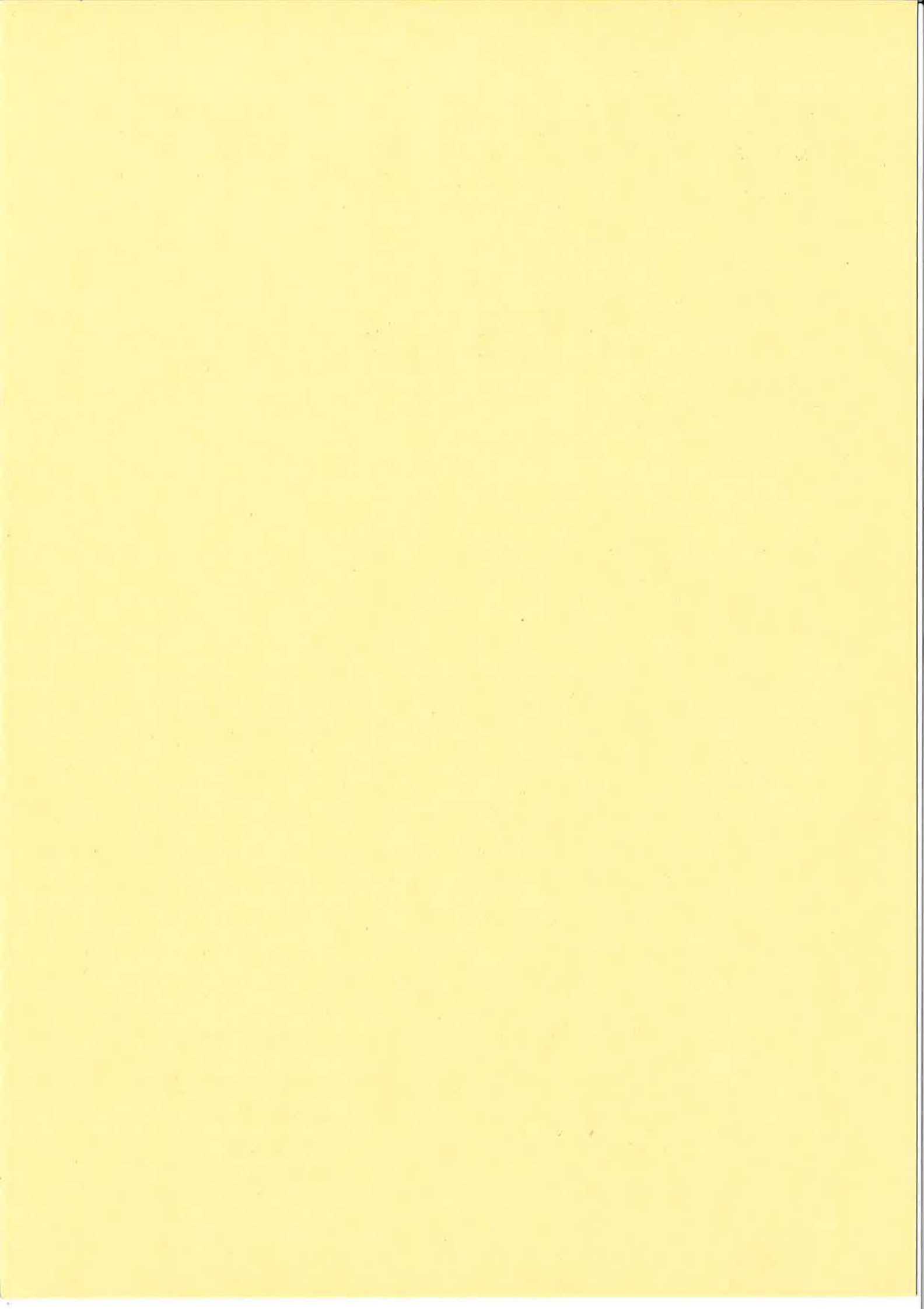
Soutenue à Saint-Etienne le 6 Novembre 1978 devant la commission d'examen :

Président

M. VEILLON

Examineurs

**MM. COUEIGNOU X
DENJEAN
LUCAS
MAHL**



**ECOLE NATIONALE
SUPERIEURE DES MINES
DE SAINT-ETIENNE**

**INSTITUT NATIONAL
POLYTECHNIQUE
DE GRENOBLE**

N° d'ordre : 11 11

THESE

présentée par

Marc HOURDEQUIN

pour obtenir le grade de

**DOCTEUR-INGENIEUR
"SYSTEMES ET RESEAUX INFORMATIQUES"**

**GENERATION DE POLICES
D'IMPRIMERIE POUR
PHOTOCOMPOSEUSE DIGITALE**

Soutenue à Saint-Etienne le 6 Novembre 1978 devant la commission d'examen :

Président

M. VEILLON

Examineurs

**MM. COUEIGNOUX
DENJEAN
LUCAS
MAHL**

ECOLE NATIONALE SUPERIEURE DES MINES DE SAINT-ETIENNE

Directeur: M. G. ARNOUIL
Direction des Recherches: M. P. COUEIGNOUX
Direction des Etudes: M. A. COINDE

PROFESSEURS DE 1^{ère} CATEGORIE

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

PROFESSEURS DE 2^{ème} CATEGORIE

MM. GUITOUD-RIBAUD Serge	Informatique
LOWYS Jean-pierre	Physique
TOUCHARD Bernard	Physique Industrielle

PROFESSEURS ASSOCIES DE 2^{ème} CATEGORIE

MM. DAVOINE Philippe	Géologie
PERRIN Michel	Géologie

DIRECTEUR DE RECHERCHE

M. LESBATS Pierre	Métallurgie
-------------------	-------------

MAITRES DE RECHERCHE

MM. BISCONDI Michel	Métallurgie
BOOS Jean-Yves	Métallurgie
COUEIGNOUX Philippe	Informatique
Mlle FOURDEUX Angéline	Métallurgie
MM. LALAUZE René	Chimie
LANCELOT Francis	Chimie
LE COZE Jean	Métallurgie
THEVENOT François	Chimie
TRAN MINH Canh	Chimie

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Président: M. P. TRAYNARD
Vice-Présidents: M. R. PAUTHENET
M. G. LESPINARD

PROFESSEURS TITULAIRES

MM. BENOIT Jean	Electronique - Automatique
BESSON Jean	Chimie Minérale
BLOCH Daniel	Physique du Solide - Cristallographie
BONNETAIN Lucien	Génie Chimique
BONNIER Etienne	Métallurgie
BOUDOURIS Georges	Electronique - Automatique
BRISSONNEAU Pierre	Physique du Solide - Cristallographie
BUYLE-BODIN Maurice	Electronique - Automatique
COUMES André	Electronique - Automatique
DURAND Francis	Métallurgie
FELICI Noël	Electronique - Automatique
FOULARD Claude	Electronique - Automatique
LANCIA Roland	Electronique - Automatique
LONGUEUE Jean Pierre	Physique Nucléaire Corpusculaire
LESPINARD Georges	Mécanique
MOREAU René	Mécanique
PARIAUD Jean Charles	Chimie - Physique
PAUTHENET René	Electronique - Automatique
PERRET René	Electronique - Automatique
POLOUJADOFF Michel	Electronique - Automatique
TRAYNARD Philippe	Chimie - Physique
VEILLON Gérard	Informatique Fondamentale et Appliquée

PROFESSEURS SANS CHAIRE

MM. BLIMAN Samuël	Electronique - Automatique
BOUVARD Maurice	Génie Mécanique
COHEN Joseph	Electronique - Automatique
GUYOT Pierre	Métallurgie Physique
LACOUME Jean Louis	Electronique - Automatique
JOUBERT Jean Claude	Physique du Solide - Cristallographie
ROBERT André	Chimie Appliquée - Chimie des Matériaux
ROBERT François	Analyse Numérique
ZADWORNÝ François	Electronique - Automatique

MAITRES DE CONFERENCES

MM. ANCEAU François	Informatique Fondamentale et Appliquée
CHARTIER Germain	Electronique - Automatique
CHIAVERINA Jean	Biologie - Biochimie - Agronomie
IVANES Marcel	Electronique - Automatique
LESIEUR Marcel	Mécanique
MORET Roger	Physique Nucléaire Corpusculaire
PIAU Jean Michel	Mécanique
PIERRARD Jean Marie	Mécanique
SABONNADIÈRE Jean Claude	Informatique Fondamentale et Appliquée
MME SAUCIER Gabrielle	Informatique Fondamentale et Appliquée
M. SOHM Jean Claude	Chimie Physique

CHERCHEURS DU C.N.R.S. (Directeur et Maîtres de Recherche)

M. FRUCHART Robert

Directeur de Recherche

MM. ANSARA Ibrahim

Maître de Recherche

BRONOEL Guy

Maître de Recherche

CARRE René

Maître de Recherche

DAVID René

Maître de Recherche

DRIOLE Jean

Maître de Recherche

KLEITZ Michel

Maître de Recherche

LANDAU Ioan Doré

Maître de Recherche

MATHIEU Jean Claude

Maître de Recherche

MERMET Jean

Maître de Recherche

MUNIER Jacques

Maître de Recherche

Je remercie :

Monsieur Gérard VEILLON, Professeur à l'ENSIMAG et à l'Institut National Polytechnique de GRENOBLE, qui m'a fait l'honneur de présider le jury de cette thèse.

Monsieur Robert MAHL, Chef du Service Economique et Financier à la DIELI, qui a bien voulu accepter de faire partie du jury.

Monsieur François DENJEAN, de la société LOGABAX, qui m'a permis de mieux apprécier les problèmes industriels liés à l'étude faite.

Monsieur Michel LUCAS, Maître Assistant à l'Université de GRENOBLE, dont les conseils m'ont permis d'améliorer la communication homme-machine.

Monsieur Philippe COUEIGNOUX, Maître de Recherches à l'Ecole des MINES de SAINT-ETIENNE, qui est l'initiateur de cette étude, et qui n'a jamais ménagé son aide ni ses conseils.

Monsieur Serge GUIBOUD-RIBAUD, Directeur du Département Informatique, ainsi que les membres de ce département qui, par leurs conseils, m'ont permis de mener les travaux à leur aboutissement.

Madame BONNEFOY, Messieurs LOUBET et DARLES, qui ont assuré, avec beaucoup de soins et de compréhension, la frappe et le tirage de cet ouvrage.

Je tiens à remercier tout particulièrement Madame AVONDO qui a réalisé l'ensemble des photographies publiées.

AVERTISSEMENT

Des contacts ayant été pris avec des industriels, les algorithmes utilisés n'ont pu être décrits. Certains de ceux-ci n'étant que des améliorations d'algorithmes existants (génération point par point d'arcs de cercle) ou résultant d'une nouvelle approche non spécifique à une exploitation industrielle (tel que le découpage automatique d'un contour en une structure codée permettant d'engendrer en une seule passe la tâche correspondante) feront l'objet, moyennant l'accord des industriels, de publications ultérieures.

Afin de réduire le travail de documentation du lecteur, la bibliographie n'a pas été établie de manière exhaustive. Elle comprend les articles ou ouvrages de base. Les références bibliographiques se trouvent regroupées, suivant leur pôle principal d'intérêt, à la fin des sous-chapitres. L'ensemble de la bibliographie est reprise en fin de cet ouvrage.

RECAPITULATIF

INTRODUCTION	1
CHAPITRE 1	
Principes de la Typographie	9
La Photocomposition Programmée	17
Informatique	29
Micro-Informatique	35
CHAPITRE 2	
Historique.	47
Le Générateur de Caractères	51
Edition de Caractères	77
CHAPITRE 3	
Mesures Effectuées sur la Simulation	87
Vers une Réalisation	127
CONCLUSION	133
BIBLIOGRAPHIE	137

I N T R O D U C T I O N

Depuis quelques années, l'introduction de l'informatique dans l'imprimerie a renouvelé les techniques et les concepts de cette dernière.

L'évolution s'est opérée suivant deux axes principaux :

- développement de logiciels afin de faciliter la transmission, l'édition et l'archivage des textes destinés à être photocomposés.
- développement de matériels pour améliorer les performances et les caractéristiques des machines de photocomposition.

En ce qui concerne ce dernier axe, l'amélioration des performances, en particulier la vitesse de composition, a été acquise au prix d'une diminution de la qualité typographique des caractères.

Cette thèse propose une méthode permettant de remédier à cet inconvénient sans perdre sur les performances déjà obtenues.

La difficulté du problème est liée aux paramètres suivants :

- la finesse de définition d'un caractère dépend de la quantité d'information sur celui-ci.
- la rapidité de composition dépend du nombre de données à traiter. De ce fait, elle varie en sens inverse de la finesse de définition.

La solution adoptée est issue d'un compromis entre une compression d'informations (jeux de caractères digitaux pour la photocomposeuse) obtenue par des considérations linguistiques et numériques, et une rapidité de décodage obtenue par une architecture multiprocesseurs appropriée.

Le problème est traité dans sa totalité, à savoir la définition des formes (encodage) et leur génération (décodage). Il en résulte de nombreuses applications intermédiaires ou dérivées :

- création de jeux de caractères pour les photocomposeuses existantes.
- création automatique de jeux de caractères dégradés pour l'impression des épreuves
- stockage de courbes régulières sous forme condensée.
- utilisation en dessin animé (coloriage par ordinateur de l'intérieur de contours complexes).

La compréhension de ce travail nécessite des connaissances dans les domaines suivants :

- principe de la typographie
- photocomposition programmée
- informatique classique
- micro informatique

Le premier chapitre présente un bref exposé sur chacun de ces domaines afin de permettre au lecteur, quelle que soit sa spécialité, de tirer bénéfice de ce travail.

En particulier, les deux dernières parties sont destinées à donner, aux lecteurs non-informaticiens, quelques notions en informatique et micro-informatique.

Le second comporte trois parties :

- historique du travail.
- méthode de génération de caractères à haute définition.
- édition de jeux de caractères à haute définition.

Le dernier chapitre est réservé aux mesures faites sur le système, ses possibilités et les extensions possibles.

CHAPITRE I

PRINCIPES DE LA TYPOGRAPHIE

Nous nous limiterons à l'étude des caractères romains, bien que les notions dégagées soient d'ordre plus général.

Le but de cet exposé est de familiariser le lecteur avec les termes techniques couramment utilisés en typographie et de justifier, par avance, le traitement linguistique de l'information.

UNITE DE LA REPRESENTATION VISUELLE D'UN CARACTERE :

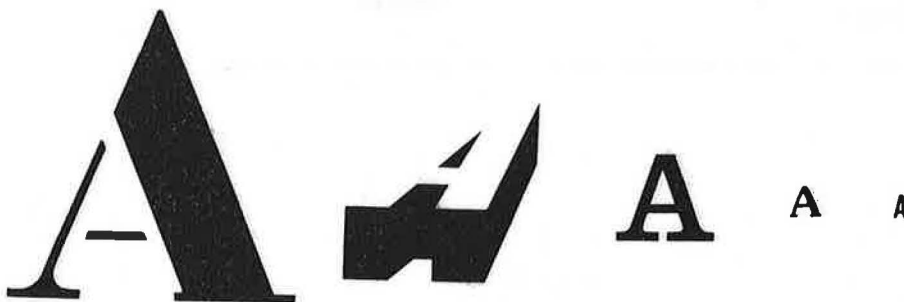
La forme des caractères alphabétiques a subi des transformations tout au long de l'histoire. Ces dernières ont abouti à avoir pour un même caractère un large choix de représentations.

Ces représentations peuvent être regroupées en plusieurs grandes familles :

- majuscules A B
- minuscules a b d
- italiques a b d

Dans une même famille la forme peut varier à l'infini.

Exemple :



Cependant l'acceptation de différentes familles, implique l'existence de critères de sélection. Plusieurs études ont été réalisées afin de formaliser ces critères intuitifs ; ces études sont citées en bibliographie. Elles ont permis de sélectionner les familles de paramètres communs à chaque type de lettres.

UNITE D'UN JEU DE CARACTERES :

Un jeu de caractères comprend les majuscules, les minuscules, les chiffres de 0 à 9, les signes spéciaux (on y adjoint parfois les italiques, et les petites majuscules). Un tel ensemble s'appelle une police.

Remarque Les majuscules sont dites hauts de casse.

Les minuscules sont dites bas de casse.

Tous les caractères d'une police, pour des besoins esthétiques évidents, possèdent une certaine unité dans leur représentation.

Ainsi : ceci n'est pas esthetique

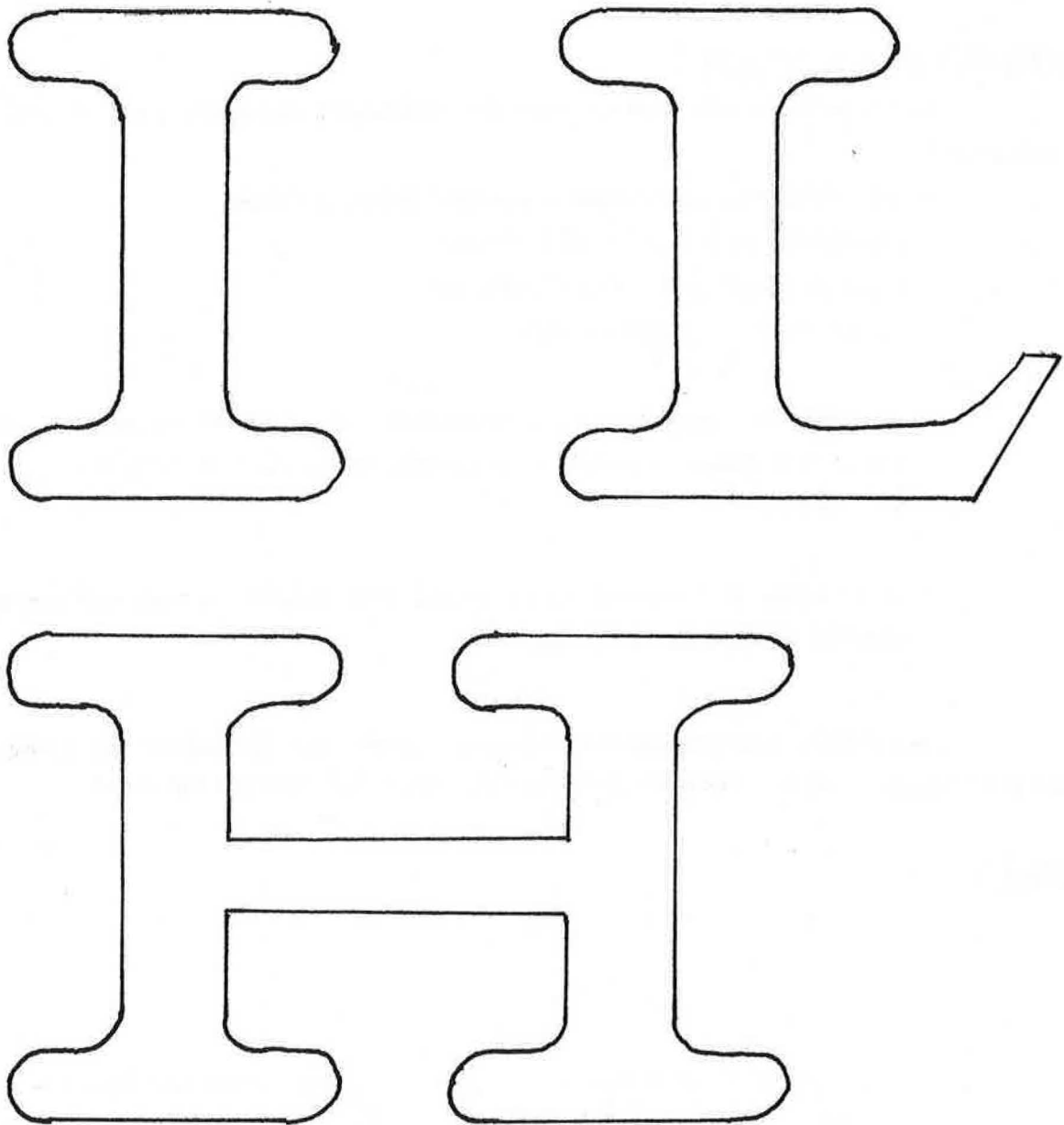
Ici aussi, les critères sont intuitifs. Les artistes ont cependant établi quelques règles :

- Tous les caractères hauts de casse ont la même hauteur apparente (à l'exception du J et du Q).
- Les propriétés des bas de casse sont illustrées par le dessin suivant:



- La graisse (épaisseur des traits gras) est, à peu de chose près, la même quel que soit le caractère haut de casse. Il en est de même pour les bas de casse.
- Les embellissements (parties non nécessaire à la reconnaissance de la lettre) sont, à de faibles variations près, identiques.

EXEMPLE :



On retrouve le motif "empattements"

(à une transformation géométrique près)



15 fois

Cette remarque est utilisée pour réduire la redondance des informations sur une police donnée.

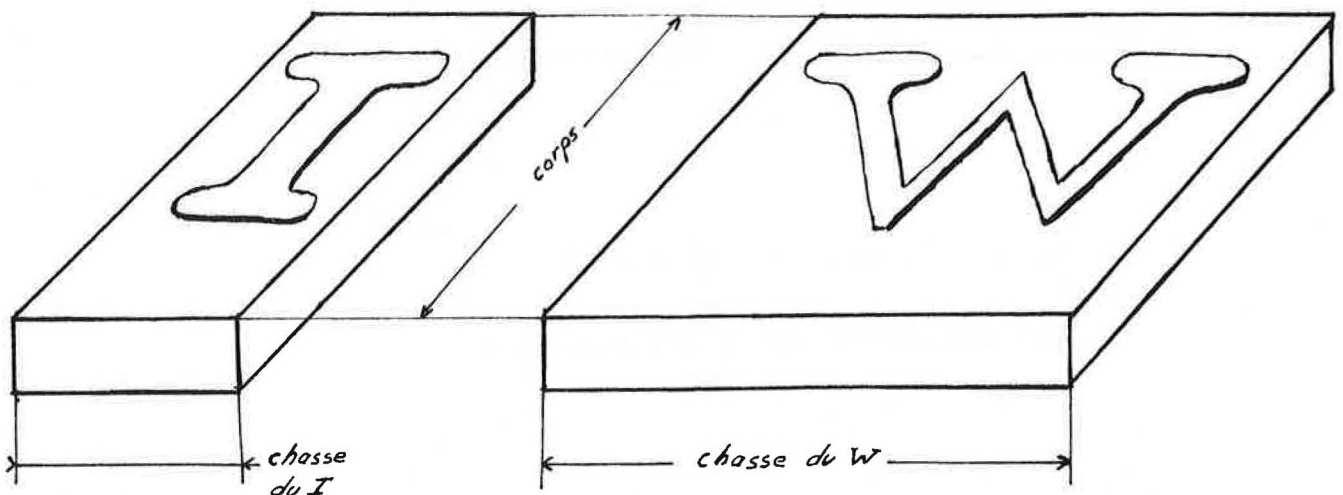
PARAMETRES LIES AUX LETTRES :

Le typographe utilise un certain nombre de paramètres qu'il est utile de connaître.

- une dimension se mesure en point pica ou didot
 - 1 point pica \neq 0,37 millimètres
 - 1 point Didot \neq 0,35 millimètres
 - 1 cicero = 12 points didot
- la somme des zones 1, 2, 3 constitue le corps de la police. La gamme des corps couramment utilisée varie de 5 à 72 points (pica ou didot).
- la largeur de l'espace dans lequel est défini un caractère est appelée chasse de ce dernier.

Autrefois, les caractères étaient portés sur des blocs de plomb. Ces deux paramètres (corps, chasse) correspondaient à la section du bloc.

EXEMPLE :



- Un autre paramètre important est la longueur de justification. Lorsque l'on écrit une ligne de texte, les lettres n'ayant pas une chasse constante, il peut rester un espace non utilisé en bout de ligne. Cet espace est réparti également entre les mots, éventuellement les lettres d'un même mot. Cette opération s'appelle la justification. La longueur de justification est la longueur maximale que l'on peut répartir entre les mots d'une ligne de texte.

REMARQUE : La justification n'est pas une propriété intrinsèque d'un caractère. De ce fait, elle est un terme de composition et non de typographie.

BIBLIOGRAPHIE

PRINCIPE DE LA TYPOGRAPHIE

LIVRE DE VULGARISATION (on trouvera en particulier une bibliographie plus étendue).

- [1] PEIGNOT Jérôme
"De l'écriture à la typographie"
Paris, Gallimard, Collection Idées - 1967.

PETIT DICTIONNAIRE DES TERMES COURANTS

- [2] AVIS F.C.
"Type face terminology"
London, Glenwien Press - 1965

CATALOGUE DE POLICES

- [3] Specimen book of Monotype printing types (vol I et II)

ARTICLES DE RECHERCHE SUR LA STRUCTURE DES CARACTERES

- [4] COUEIGNOUX Philippe
"Generation of Roman Printed Fonts"
Massachussets Institute of Technology, Department of Electrical Engineering,
Ph.D Thesis 1975
- [5] COX C.H., III, COUEIGNOUX, B.A. BLESSER, M. EDEN
"Skeletons : a link between theoretical and physical letters descriptions"
Department of Electrical Engineering, M.I.T.
- [6] MERGLER H.W., VARGO P.M.
"An approach to computer assisted letter design"
The journal of Typographic Research - Vol 3 - Number 4 - 1968
- [7] SHILLMAN P.J.
"Character recognition based on phenomological attributes : theory and Methods"
P.h. D. Thesis, Department of electrical engineering, M.I.T. - 1974

LA PHOTOCOMPOSITION PROGRAMMEE

HISTORIQUE :

L'imprimerie a débuté avec Gutenberg (vers 1420), celui-ci a le premier utilisé des caractères typographiques mobiles. Leur assemblage temporaire 'compose' le texte sur une forme. Cette forme sert à tirer les exemplaires suivant un procédé d'encrage et de transfert s'appliquant globalement à tous les caractères de la forme. La vitesse de composition est d'environ mille cinq cents signes par heure. Cette limitation est liée à la vitesse de sélection manuelle des caractères.

Vers la fin du dix neuvième siècle, sont apparues des machines à composer. Elles mécanisent les opérations de sélection et d'assemblage des caractères. Elles se divisent en deux grandes familles :

- celles qui fondent les caractères par lignes entières. Leur principe a été trouvé en 1884 par Mergenthaler et utilisé par les fondeuses 'Linotype' et "Intertype".

- celles qui assemblent et alignent les caractères fondus un à un. Elles ont été mises au point en 1887 par Lanston et diffusées par "Monotype".

Grâce à ces machines, la productivité passe de 1500 à 9000 signes par heure.

L'étape suivante en matière de composition a été l'utilisation d'un film photographique en remplacement du chassis de plomb. Pour comprendre le phénomène, nous devons brosser un historique rapide des progrès de l'imprimerie dans le domaine du tirage.

A l'époque de Gutenberg, on utilise la presse à bras. En 1812 Koenig crée une presse mécanique à vapeur. Elle imprime cylindre contre plan. Grâce à elle, la productivité passe de 300 à 1000 feuilles par heure. Les machines actuelles employant ce principe permettent d'imprimer 3000 feuilles par heure.

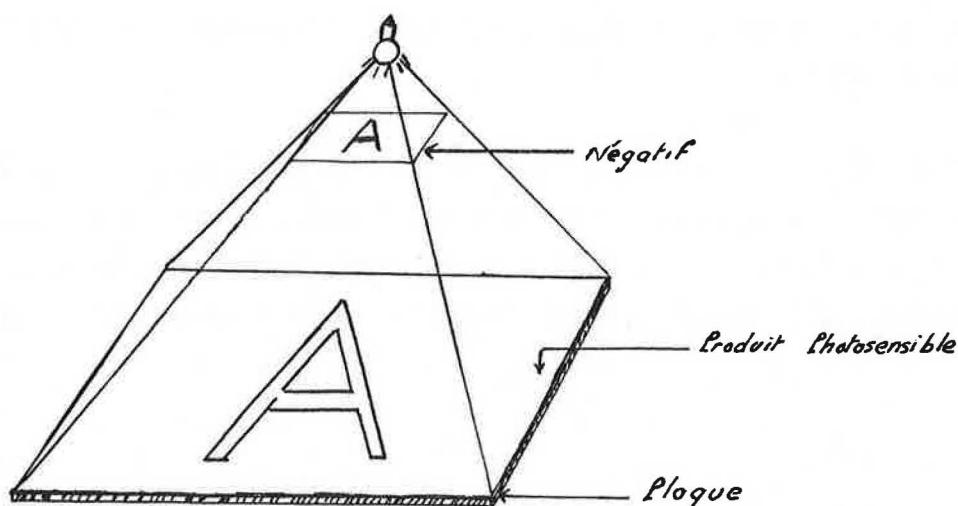
Vers la fin du dix neuvième siècle, le principe d'impression cylindre contre cylindre permet d'atteindre 10000 feuilles par heure. Les machines utilisant ce concept nécessitent une étape intermédiaire ; en effet, la forme plane issue des fondeuses ne peut être rendue directement cylindrique. On prend une empreinte de celle-ci grâce à un carton humide, et à partir de ce "négatif" on confectionne la matrice cylindrique qui sera utilisée pour le tirage.

Actuellement, deux autres procédés sont couramment utilisés : il s'agit de l'impression offset et de l'héliogravure.

- impression offset : on utilise une plaque de zinc ou d'aluminium enduite d'un produit photosensible. Seul la surface insolée est réceptive à l'encre.
- héliogravure : on utilise une plaque de métal gravée par une attaque chimique, en creux. Ces creux sont de minuscules alvéoles qui vont retenir l'encre.

Ces plaques relativement minces, sont aisément incurvables.

Ces deux procédés font appel à un système de photographie : c'est-à-dire à un film transparent qui porte l'image des feuilles finales. C'est ce dernier qui permet l'insolation sélective des plaques.



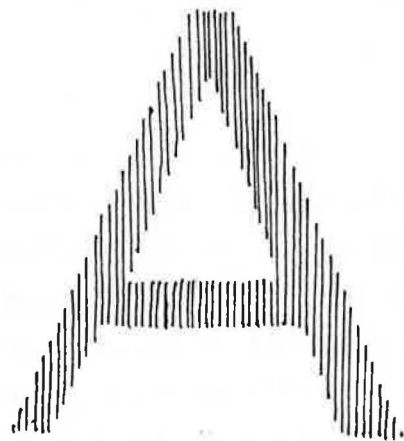
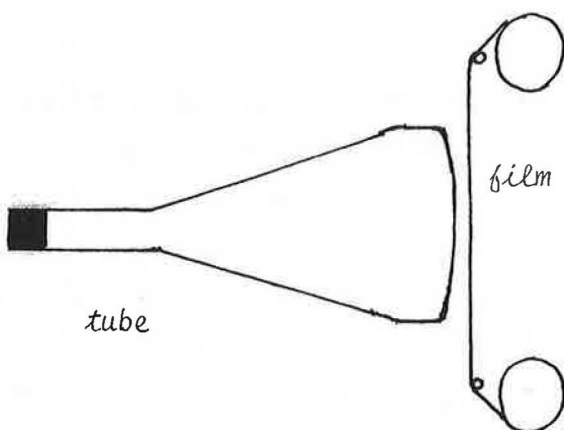
Le film peut être obtenu par deux voies :

- l'une consiste à photographier une épreuve obtenue par les procédés typographiques classiques.
- l'autre, essentiellement électronique, consiste à composer directement le film.

Notre centre d'intérêt est cette seconde voie. Exposons les différentes étapes qui ont permis son aboutissement.

Le film a servi de porte d'entrée à l'électronique. En effet, on sait réaliser des films photographiques par des moyens électroniques (écriture du texte sur un écran cathodique, puis photographie de ce dernier). Or une image sur un tel écran n'est rien d'autre qu'une suite de points. L'image présente à ce stade un caractère essentiellement digital. Elle est par suite traitable par des moyens informatiques.

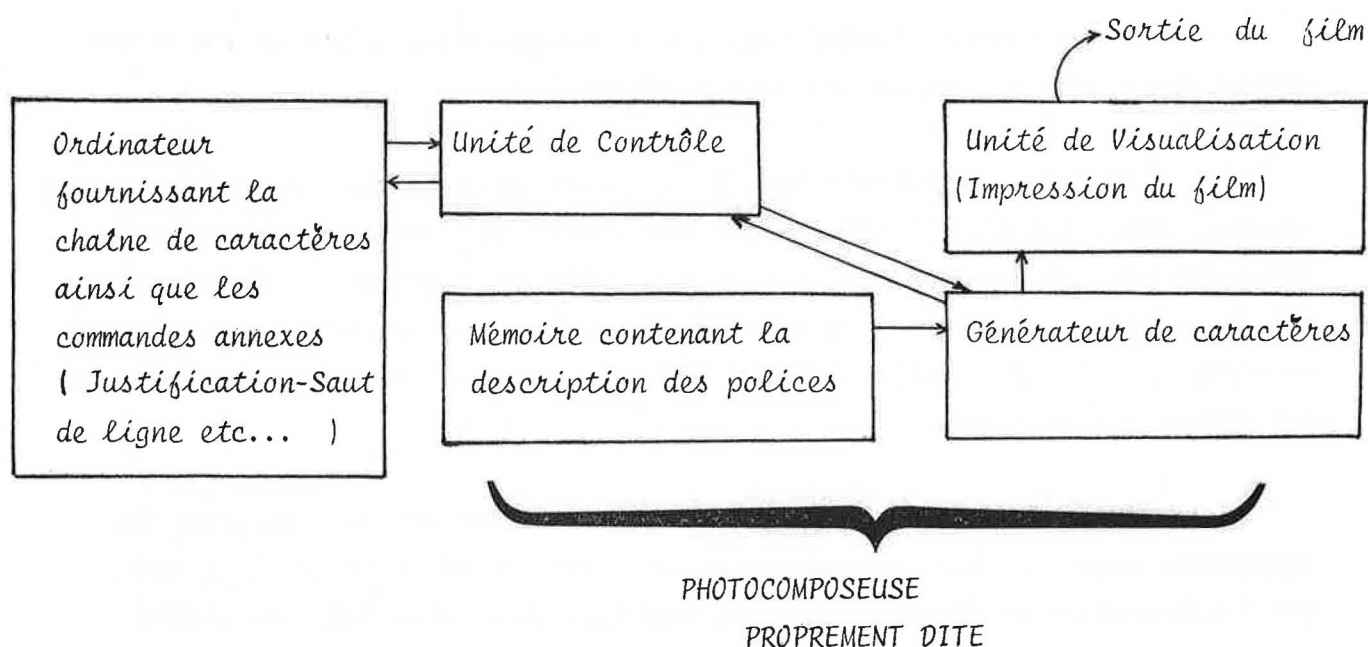
Cette évolution est favorisée par la baisse continue du prix des composants digitaux, leur accroissement de puissance et de densité. Le fait que l'information se présente dans ces machines sous forme digitale permet de lui appliquer aisément toutes les théories relatives au traitement de cette dernière.



Détail d'une lettre formée
sur l'écran

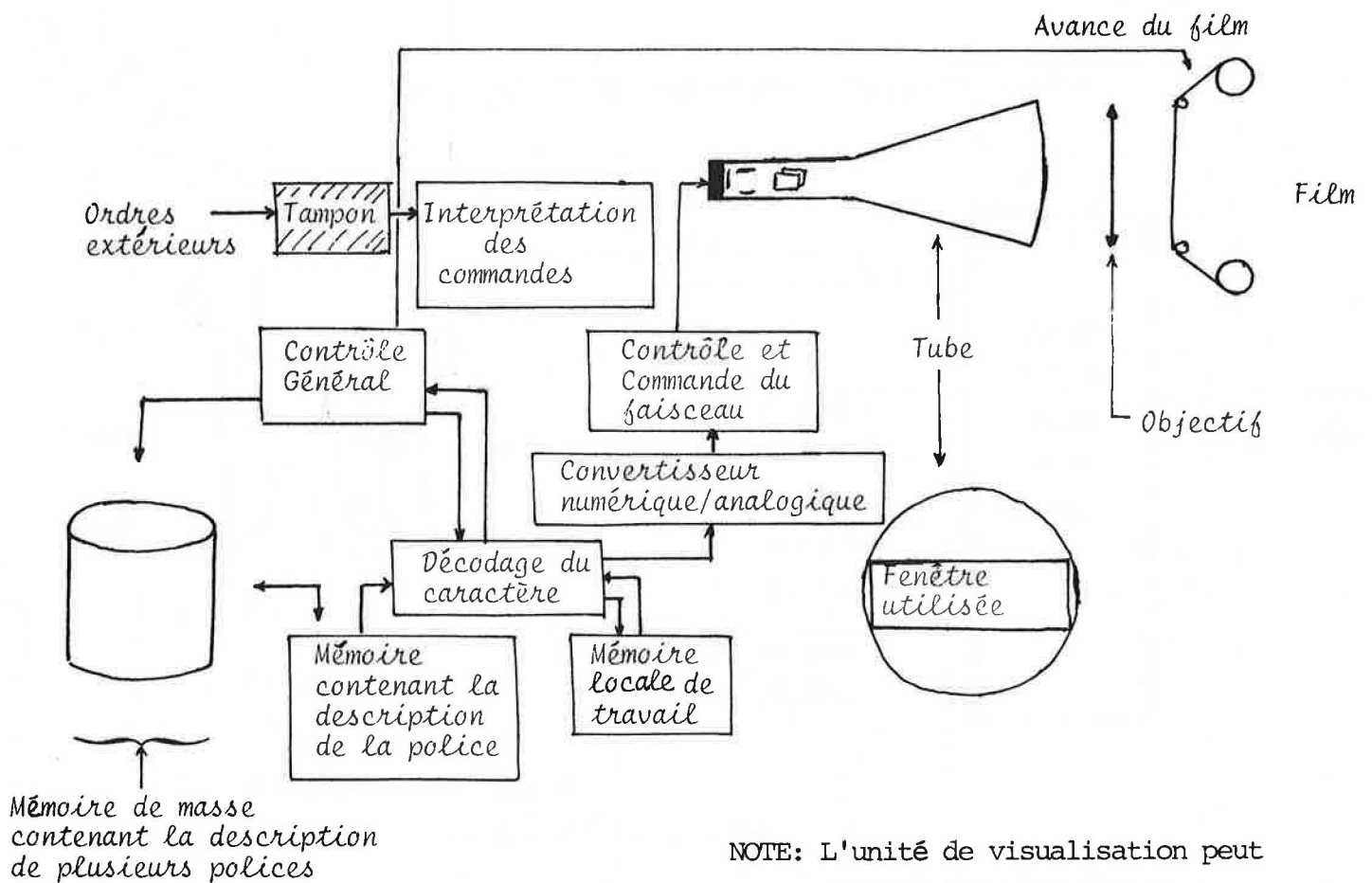
LES PHOTOCOMPOSEUSES

La photocomposeuse est l'organe qui permet, à partir du texte justifié, de sortir le film. Le schéma de base d'une photocomposeuse est le suivant :



Cette technique en supprimant certains problèmes en a créé de nouveaux. Ainsi, pour une même police, les caractères de corps différents ne résultent pas d'une simple homothétie. Par exemple, l'oeil (c'est-à-dire le contour intérieur) du 0 est plus important dans le corps cinq que dans le corps soixante douze. La solution adoptée en photocomposition est la multiplication des jeux de caractères pour une même police ; par exemple, un jeu est utilisé du corps 72 au corps 20, un second jeu du corps 30 au corps 10, un dernier du corps 12 au corps 5. Les corps intermédiaires sont obtenus par "réduction photographique".

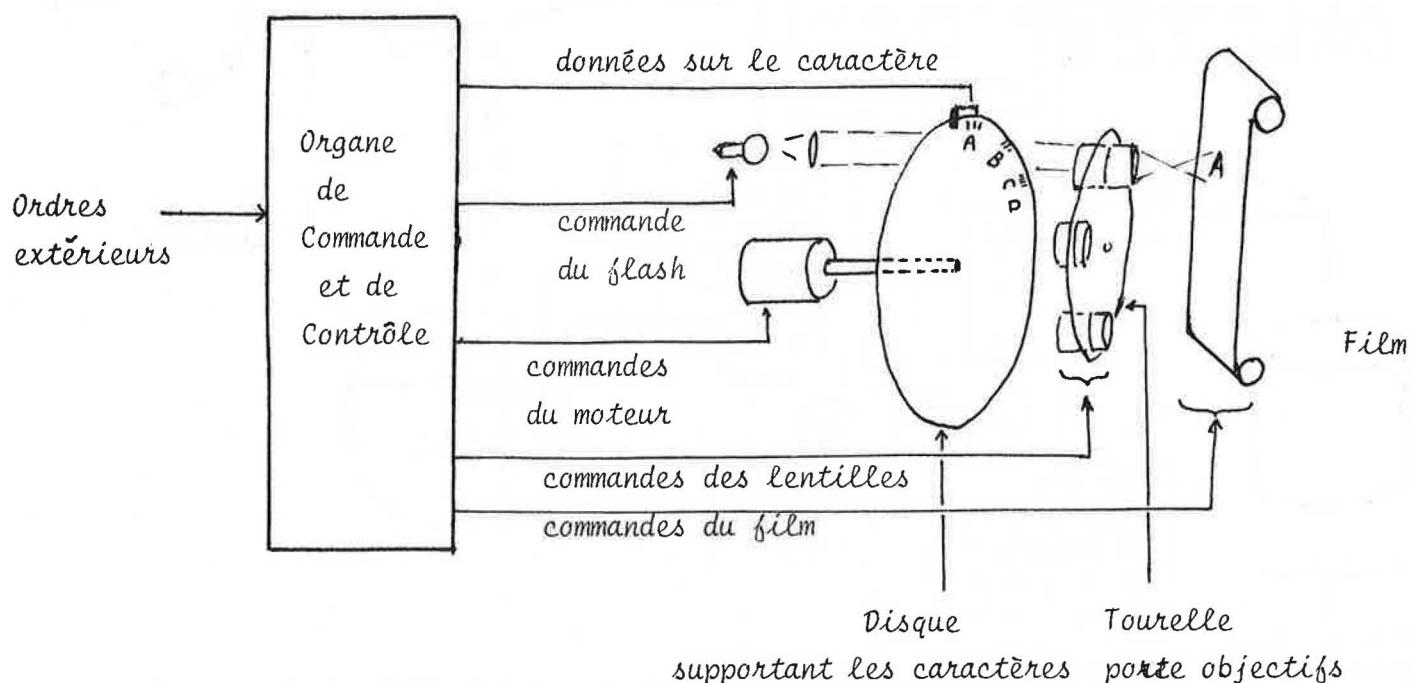
Détail d'une photocomposeuse programmée :



Les contraintes propres à cette technique ont amené les artistes à créer de nouveaux styles de lettres tels que les caractères Univers.

Le schéma fonctionnel que nous avons donné est identique pour les photo-composeuses purement "photographiques" et, pour celles dites programmées (ou digitales).

Détail d'une photo-composeuse "photographique" :

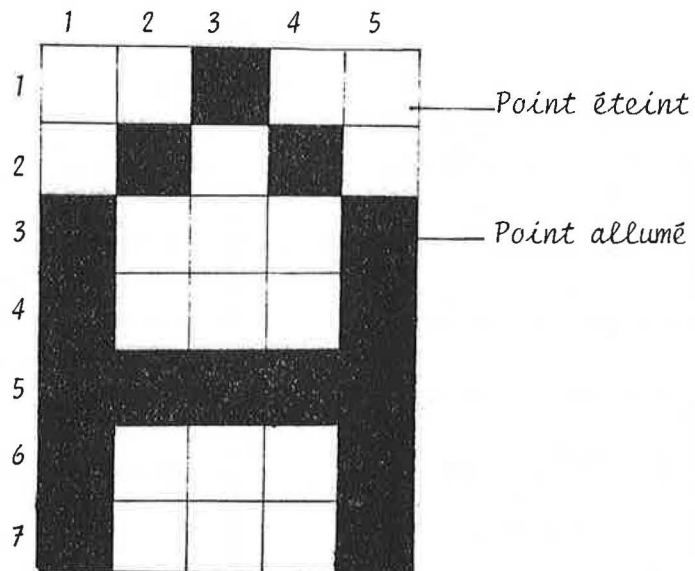


Nota : Par souci de simplification, le mécanisme permettant de caler le caractère sur la ligne n'a pas été représenté.

Ce type de machine permet de traiter 50 caractères par seconde.

Le générateur de caractère lit dans une matrice les points à allumer pour "dessiner" ce dernier. La qualité du dessin dépend de la dimension de la matrice de départ.

Par exemple, si la matrice de départ est de dimension cinq par sept, le A sera défini de la manière suivante :

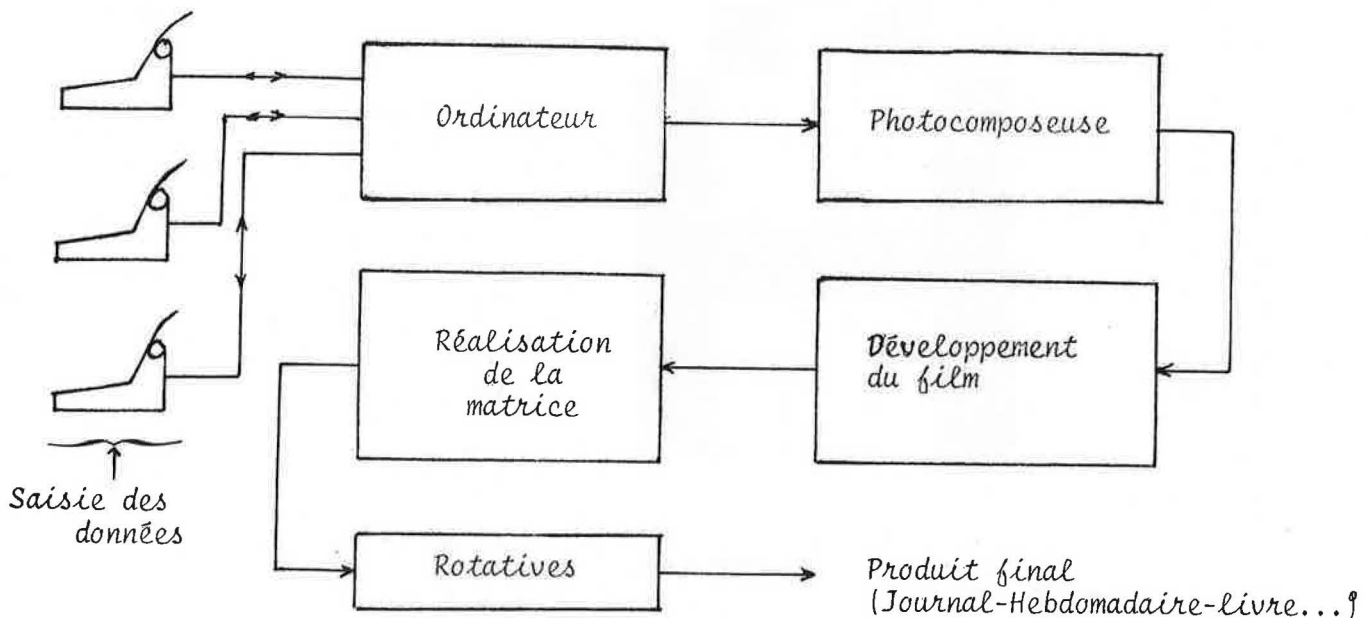


Pour les petits corps, la taille usuelle des matrices est 30 par 30. Elle excède rarement 100 par 100. Ce type de machine permet l'impression de 1000 signes par seconde. La qualité obtenue est médiocre, surtout pour les corps importants. Ceci est lié à la faible capacité des matrices utilisées (capacité limitée pour des raisons économiques).

SCHEMA FONCTIONNEL GLOBAL :

Une photocomposeuse n'est pas utilisable en elle même ; elle doit s'insérer dans une chaîne allant de la saisie des informations aux feuilles imprimées.

Le schéma général d'un tel ensemble est le suivant :



REMARQUE : Les étapes de corrections n'ont pas été représentées, cependant, leur importance ne doit pas être sous-estimée.

Le calculateur automatise ou facilite les opérations suivantes :

- saisie et correction des informations,
- justification des lignes,
- coupure syllabique des mots,
- mise en page,
- pilotage de la photocomposeuse.

La photocomposeuse permet de sortir le film (et, en général, si on le désire des microfiches). Le tirage du film permet de créer les matrices cylindriques qui seront utilisées par les rotatives.

La rotative permet d'obtenir les feuilles qui une fois découpées et reliées constitueront le document final (journal, livre, quotidien...).

CONCLUSION.

Toute l'évolution se résume en un seul phénomène : accroître la productivité (mesurée à l'heure actuelle en millions de signes heure). Cette évolution s'est traduite par une baisse constante de la qualité typographique des caractères. Elle a amené de ce fait une diversification du marché :

- l'imprimerie de luxe reste acquise à la composition classique.
- celle n'ayant pas une grande diffusion se tourne vers l'utilisation de photocomposeuses "photographiques".
- l'imprimerie de grande diffusion (journaux, annuaires...) tend à utiliser les photocomposeuses programmées.

L'amélioration de la qualité graphique des caractères générés par ces dernières, et la baisse de leur prix, amorce la tendance inverse (c'est-à-dire une réunification du marché potentiel).

BIBLIOGRAPHIE

LA PHOTOCOMPOSITION PROGRAMMEE

LE POINT DE VUE D'UN ARTISTE

- [1] FRUTIGER A.
"The IBM selectric composer-The evolution of composition technology"
I.B.M. Journal - Janvier 1968

ARTICLE DE SYNTHESE

- [2] LARUE José
"Composition automatique des textes"
Technique de l'ingénieur - Informatique N° H21 - Septembre 1975

COURS EXPOSANT L'ETAT ACTUEL DES TECHNIQUES

- [3] BACCHUS P. - ERABAND P. - FRETIN M. - LHEUREUSE B. - PROFIZI J.C.
"La composition programmée des textes"
Séminaire I.R.I.A. - Avril 1977

I N F O R M A T I Q U E

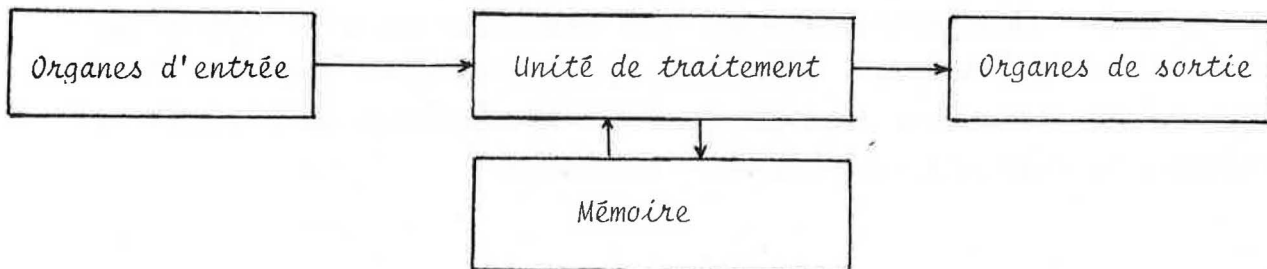
L'ordinateur est l'outil qui traite et mémorise l'information digitale. Il constitue notre outil de base. Il est par la suite nécessaire de connaître son principe et les principales techniques auxquelles il a donné naissance.

INTRODUCTION

Les fonctions principales d'un ordinateur sont la saisie de données du monde extérieur, leur traitement, la restitution des résultats (au monde extérieur).

Le traitement de données numériques n'est qu'un cas particulier. La machine est capable d'effectuer n'importe quel type de traitement sur n'importe quel type de données, à condition que ces dernières aient reçu une codification digitale (ou soient par nature des informations digitales).

Il est formé d'une unité de traitement effectuant les opérations sur les données, d'organes d'entrée pour saisir l'information, d'organes de sortie restituant l'information traitée, et d'une mémoire où sont stockées les données initiales, finales, les résultats intermédiaires ainsi que les programmes (ces derniers indiquent à l'unité de traitement les différentes opérations élémentaires à effectuer).



LA FONCTION MEMOIRE :

La mémoire est constituée de cellules élémentaires capables de mémoriser un 1 ou un 0. Une telle cellule s'appelle un bit. A chaque bit est attachée une valeur sémantique (un bit n'a pas de signification par lui-même) ; ainsi pour une matrice de points définissant un caractère, chaque bit de cette dernière indiquera si le point physique correspondant sur l'écran doit être allumé (la valeur du bit est alors par exemple 0) ou éteint (la valeur du bit est alors pour notre exemple, 1).

En général, on ne traite que rarement un bit isolé, de ce fait on regroupe ces informations élémentaires par paquets appelés mots. Les longueurs courantes de mots sont 8 bits (on parle alors d'octets), 16 et 32 bits.

Ceci permet de créer facilement des valeurs sémantiques plus élevées ; par exemple si on utilise une machine à mots de 8 bits et que l'on veut stocker des caractères, la valeur sémantique attachée à 00000000 sera A, celle attachée à 00000001 sera B, 00000010 représentera C, 00000011 D, 00000100 E, et ainsi de suite.

Remarque : 1 kilo octets (ou 1 k octets) = 1024 octets

1 kilo mots (ou 1 k mots) = 1024 mots

Chaque mot de la mémoire comporte une adresse qui lui est propre. Cette adresse est le moyen de retrouver l'information contenu dans le mot.

LANGAGE MACHINE

Chaque machine comprend un certain langage.

Le langage compris par l'ordinateur est le langage machine. Ce dernier décrit les opérations élémentaires que peut effectuer l'unité de traitement. Il est formé d'un ensemble fini d'instructions qui sont codées sur des mots (l'unité de traitement attache donc une valeur sémantique à ces derniers). L'exécution d'une instruction demande un certain temps qui est lié à la complexité de celle-ci. Il est en moyenne, de l'ordre de quelques micro-secondes.

Une opération d'addition est en moyenne 5 fois plus rapide qu'une opération de multiplication (et encore, toutes les machines ne possèdent pas cette instruction). Par exemple sur PDP 11/40 l'addition (16 bits à 16 bits) s'effectue en 1,5 micro-secondes, la multiplication (16 bits par 16 bits) en 9 micro-secondes.

Lorsque l'on désire effectuer un traitement rapidement, il faut limiter à la fois le nombre et la complexité des instructions exécutées.

Exemple : Sur les ordinateurs à mots de 8 bits pour ajouter 8 bits à 8 bits, il faut 3 mots mémoires (1 pour le 1er opérande, 1 pour le second, 1 pour le résultat), et au moins une instruction (celle d'addition).

Pour ajouter 16 bits à 16 bits, il faut 6 mots et au moins 2 instructions : additionner le 1er mot du 1er opérande au 1er mot du second et mémoriser la retenue, additionner le 2ème mot du 1er opérande au 2ème du second et additionner la retenue.

Ceci nous amène à quelques conclusions :

- plus on effectue des calculs précis, plus il faut de bits pour mémoriser l'information.
- plus on utilise de bits, plus il faut effectuer d'opérations élémentaires.
- plus on utilise d'opérations élémentaires, plus on allonge le temps de calcul.

Donc une augmentation de la précision de calcul se solde par un encombrement mémoire plus grand et par une diminution de la vitesse. En matière de calculs, lorsque l'on désire être rapide et utiliser le moins de mémoire possible, il faut établir un compromis entre les deux facteurs.

CONCLUSION

Ce chapitre n'a pas l'ambition d'initier le néophyte aux techniques informatiques. Ce dernier, pour plus ample information, devra consulter les ouvrages cités en bibliographie. Il rappelle les grandes lignes en insistant sur les points fondamentaux qui seront utilisés par la suite.

BIBLIOGRAPHIE

INFORMATIQUE

OUVRAGES PERMETTANT DE COMPRENDRE LE FONCTIONNEMENT DES ORDINATEURS

- [1] DONDOUX J. - MARANO Ph. - MERLIN J.C.
"Introduction à l'informatique - structure et programmation des ordinateurs".
Librairie Armand Colin - 1971

- [2] MEINADIER Jean-Pierre
"Structure et fonctionnement des ordinateurs"
Editions Larousse - 1975

EXPOSE DES TECHNIQUES DE TRAITEMENT DE L'INFORMATION

- [3] MAHL Robert, BOUSSARD J.C.
"Algorithmique et structure de données" (171 pages)
Laboratoire d'Informatique - Université de Nice 1977

M I C R O - I N F O R M A T I Q U E

Lorsque que l'on parle d'ordinateurs, le néophyte imagine une machine monstrueuse, de prix élevé.

Cette image était vrai voilà 20 ans. Les progrès ont été tels, qu'elle est maintenant totalement erronée.

Depuis le début de cette décennie, on sait construire des ordinateurs plus performants que les monstres des années cinquantes, sur une plaque de silicium de quelques dizaines de millimètres carrés. La chute des prix a été tout aussi extraordinaire : depuis 1977, on vend de telle machine aux amateurs.

Afin de comprendre le phénomène, faisons un bref rappel de l'évolution. Puis, nous exposerons les conséquences principales des possibilités des machines actuelles.

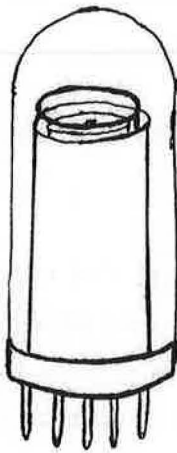
EVOLUTION DE LA TECHNOLOGIE :

Jusqu'en 1958, les ordinateurs étaient construits à base de lampes (analogues à celles que l'on trouve dans les anciens postes de télévision).

A partir de 1958, on dispose du transistor. Fonctionnellement, il effectue le même travail qu'une lampe, mais il consomme beaucoup moins d'énergie, il est infiniment plus fiable, il est plus facile à utiliser, et son encombrement est 50 fois plus faible. Un transistor est réalisé sur un morceau de silicium, sa surface est de l'ordre de la dizaine de millimètres carrés.

Dès 1965, apparaissent les circuits intégrés : on regroupe en un seul boîtier plusieurs transistors (souvent interconnectés afin de réaliser une certaine fonction). La taille des boîtiers augmente peu par rapport à un boîtier contenant un seul transistor. La fiabilité par boîtier reste la même.

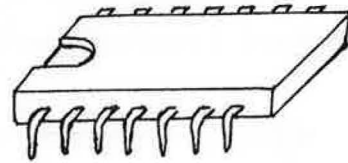
A partir de 1972, apparaît la technique LSI (LARGE SCALE INTEGRATION, en français : INTEGRATION A GRANDE ECHELLE) qui permet de regrouper en un seul boîtier plusieurs milliers de transistors. Ces transistors diffèrent légèrement des précédents : ils ne demandent pratiquement pas de puissance de commande, ils consomment de ce fait que peu d'énergie. Cette technique permet d'obtenir en un seul boîtier un ordinateur complet.



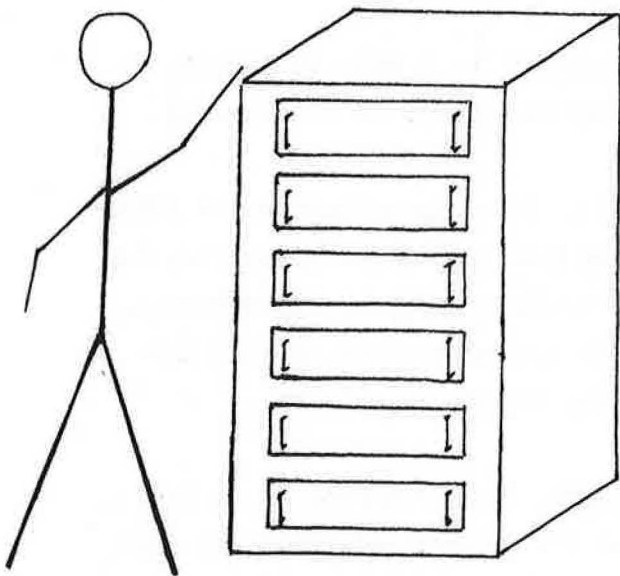
Tube



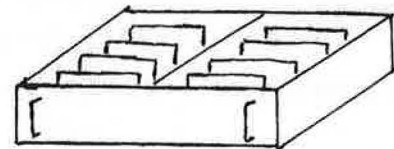
Transistor



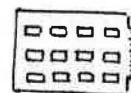
Circuit Intégré



Première génération : Tubes



Deuxième génération : Transistors



Troisième génération : Circuits Intégrés



Quatrième génération : Circuits L.S.I

Encombrement des différentes technologies pour une même fonction

Un facteur fondamental est le prix pratiquement constant d'un boîtier (lampes incluses). Un circuit LSI de grande diffusion coûte environ 30 francs. Le fait de pouvoir disposer de supports physiques aussi puissants à un prix aussi faible, a modifié profondément les concepts classiques de l'informatique, et a donné naissance à une nouvelle branche de cette science : la micro-informatique.

LA MICRO-INFORMATIQUE :

Le but de la micro-informatique est de trouver des applications, à ces nouveaux composants électroniques.

Un de ces terrains de choix est le remplacement des automatismes à relais ou électroniques conventionnels. Un autre terrain est le domaine grand public :

- jeux pour télévision ;
- automate de machine à laver ;
- centrale de surveillance pour les voitures, etc...

Les réalisations dans le domaine informatique sont :

- réalisation de terminaux intelligents ;
- réalisation à faible coût, de systèmes multiprocesseurs ;
- réalisation de micro-ordinateurs (c'est-à-dire dont le prix est inférieur à 100 000 francs).

Le point important pour nous, est le second ; on peut répartir une tâche entre plusieurs processeurs (travaillant en parallèle) et ainsi obtenir rapidement le résultat recherché.

Exemple 1: On a deux suites de nombres : A_1, A_2, \dots, A_n et B_1, B_2, \dots, B_n que l'on veut additionner par paire : $A_1 + B_1, A_2 + B_2, \dots, A_n + B_n$. Si l'on dispose que d'une seule machine, il faut effectuer séquentiellement n additions. Si l'on dispose de n machines, il suffit de faire exécuter à chaque machine une seule opération (la machine numéro j , effectue $A_j + B_j$). Ces machines étant indépendantes, peuvent travailler en parallèle ; on a ainsi notre résultat final n fois plus rapidement.

Exemple 2 : Soit un système d'acquisition de données; toutes les milli-secondes, il doit obtenir A et B, et effectuer A+B. On suppose que les temps d'obtention des données et de calcul sont les mêmes et légèrement inférieurs à la milli-seconde.

Si l'on ne dispose que d'une seule unité de traitement, le problème est insoluble, le temps total de traitement étant sensiblement de trois milli-secondes.

L'exemple 1 nous suggère d'effectuer l'acquisition de A et B en parallèle, puis d'effectuer la sommation. On doit disposer de deux machines élémentaires. Le temps total est de l'ordre de deux milli-secondes.

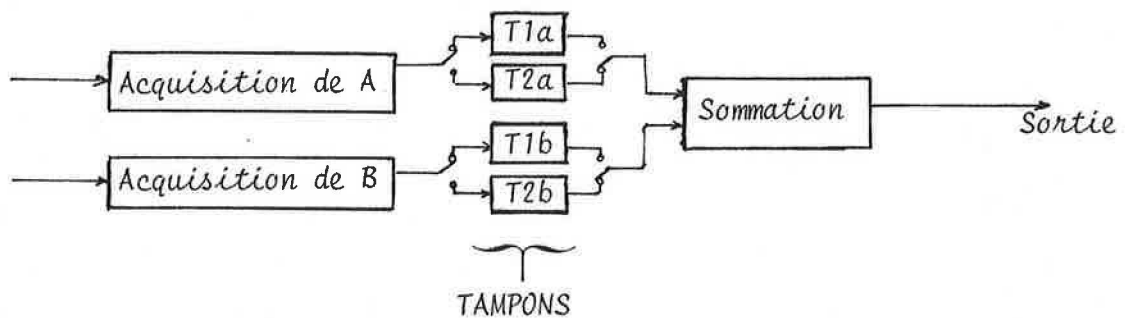
Pour améliorer le temps d'acquisition, on peut envisager d'effectuer durant l'acquisition de A_n et B_n , la sommation de A_{n-1} et B_{n-1} . Il faut trois machines élémentaires. Le temps d'acquisition tombe à une milli-seconde (le temps de traitement reste de deux milli-secondes). Le problème est résolu.

Evolution temporelle des opérations :

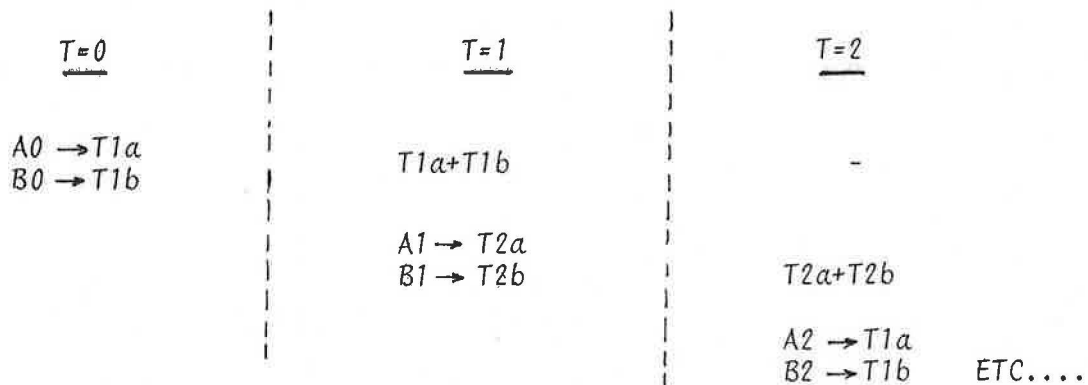
<u>T=0</u>	<u>T=1</u>	<u>T=2</u>	
Acquisition de A_0 Acquisition de B_0	Calcul de A_0+B_0		
	Acquisition de A_1 Acquisition de B_1	Calcul de A_1+B_1	
		Acquisition de A_2 Acquisition de B_2	ETC...

Nous observons qu'il est nécessaire de pouvoir mémoriser simultanément deux couples de données (A_n, B_n et A_{n-1} et B_{n-1}); on emploie une méthode de double mémorisation permettant, en plus de la fonction de stockage, le fonctionnement en "parallèle".

Représentation :



Fonctionnement :



$A0 \rightarrow T1a$ signifie : placer la donnée A0 dans tampon T1a.

Ce qu'il faut retenir : étant donné un algorithme que l'on sait découper en sous algorithmes pouvant être traités en parallèle, on peut construire une machine spécialisée très rapide et bon marché le traitant. Plus on augmente le degré de parallélisme, plus on va vite.

BIBLIOGRAPHIE

MICRO- INFORMATIQUE

- [1] DANIEL R. - GLYNN M.
"Microprocessors : Technology, architecture and applications"
(Disponible aux éditions Lavoisier) 1976
WILEY - Interscience publication

- [2] SAINT.PATRICE Alain
"Etude d'un système de développements pour applications multiprocesseurs"
Rapport de D.E.A. ; Ecole Nationale Supérieure des Mines de Saint-Etienne
1978

CHAPITRE II

HISTORIQUE

Le sujet m'a été proposé en octobre 76 sous la forme suivante :

Peut-on à l'heure actuelle créer un générateur de caractères engendrant des lettres ayant une définition égale à celle obtenue grâce au plomb, en conservant les avantages des photocomposeuses programmées ?

Les trois premiers mois ont été consacrés à la bibliographie. C'est lors de cette phase que la contrainte la plus importante est apparue :

- les caractères doivent être définis dans des matrices d'au moins 512 par 512 (la digitalisation est alors à peine visible pour des caractères de corps 72). En effet, un corps 72 représente un caractère de 24 mm de haut ; ce caractère devant se trouver représenté dans une matrice 512 par 512, la dimension d'un point est d'environ 50 microns. Le pouvoir de résolution de l'oeil est d'une minute d'angle. La distance à partir de laquelle il commence à confondre les points est donc de : 20 cm ($0,05/20 = \text{tangente } 1 \text{ minute}$).
- les exigences des artistes sont de beaucoup supérieures : 2048 points pour du 72 points. Ceci est dû au fait que l'oeil est qualitativement plus sensible que ne laisse supposer son pouvoir de résolution quantitatif. Il est particulièrement sensible aux écarts suivants :
 - . variation en palier par rapport à une ligne régulière ;
 - . écart au parallélisme ;
 - . écart par rapport à l'horizontale ou à la verticale.

De plus l'artiste regarde un caractère à la loupe ou sur agrandissement.

Le stockage d'une matrice 2048 par 2048 exige 4 194 004 bits, soit pour une police complète environ 537 millions de bits. Le bit revenant à 2 centimes, le prix de la mémoire est donc d'environ 10 millions de francs. Le lecteur comprendra que même en ayant recours à des méthodes spéciales de codage des matrices, une telle solution n'est pas économiquement envisageable.

Le problème à l'issue de ces trois mois était devenu :

comment créer rapidement une matrice 2048 par 2048, ceci à partir de données plus compactes ?

Durant les mois suivants le problème a évolué. Il a été découpé en sous-ensembles. Chaque sous-ensemble a été résolu. Cette étude a abouti à la réalisation d'une simulation de la machine finale. Cette simulation a été terminée fin juin 77.

Il est apparu que l'introduction des données est une opération longue et délicate, étant donné leur compacité ; or cette opération est d'un coût économique non négligeable. Aussi, a-t-il été décidé de construire un système permettant de faciliter cette opération.

Le système adopté ne comporte pas de nouveauté fondamentale : c'est un système graphique spécialisé doublé d'un bibliothécaire, auquel est ajoutée une interface permettant de traduire les données mémorisées sous une forme exploitable par le générateur de caractères.

La définition de ce système et sa réalisation ont demandé respectivement deux et trois mois.

Depuis le mois de janvier 78, nous testons les différentes tâches et mesurons leurs performances.

CONCLUSION

L'ensemble du système se révèle être à l'usage, non seulement un système d'édition et de génération de caractères à haute définition, mais aussi un système d'édition et de génération de tâches en temps réel (les caractères étant un cas particulier de tâches).

LE G E N E R A T E U R D E C A R A C T E R E S

SYSTEMES EXISTANT ACTUELLEMENT :

Nous avons vu que le prix de la mémoire pour stocker une police définie dans des matrices 2048 par 2048 est de 10 millions de francs. Les photocomposeuses digitales actuelles utilisent parfois des matrices 100 par 100, ce qui implique un coût mémoire pour un codage tel quel de l'information de 25 600 francs. Ce prix est encore trop élevé (il faut ajouter le prix de l'électronique permettant d'accéder à cette mémoire, le prix des cartes, etc..., et n'oublions pas qu'en général on a plus d'une police en ligne).

On emploie de ce fait des techniques plus efficaces qu'un codage brut. Les deux techniques les plus élémentaires sont :

- le codage par matrices creuses ;
- le codage par plages.

Avant de décrire ces techniques, il nous faut faire une remarque préliminaire. Au lieu de coder dans la matrice tous les points qui doivent être allumés on ne code que le contour de la lettre ; on sait que les points à l'intérieur d'un contour doivent être allumés.

Exemple : I codé dans une matrice 32 par 32.

[illegible]

CODAGE BRUT

[illegible]

CODAGE DU CONTOUR

Pour un codage 16 par 16, on a :

```

000000000000000000
001111111111111100
011111111111111110
011111111111111110
001111111111111100
0000011111100000
0000011111100000
0000011111100000
0000011111100000
0000011111100000
001111111111111100
011111111111111110
011111111111111110
001111111111111100
000000000000000000
000000000000000000

```

CODAGE BRUT

```

000000000000000000
001111111111111100
010000000000000010
010000000000000010
001110000000111100
0000010000100000
0000010000100000
0000010000100000
0000010000100000
0000010000100000
001110000000111100
010000000000000010
010000000000000010
001111111111111100
000000000000000000
000000000000000000

```

CODAGE DU CONTOUR

A - CODAGE PAR MATRICES CREUSES

La matrice issue du codage par contour comporte relativement peu de bits à 1, par suite, au lieu de mémoriser la matrice nous mémoriserons la position des bits à 1.

Exemple sur le I :

Numérotons J le numéro de la ligne, K celui de la colonne.

Pour le cas 16 par 16 :

Le codage de J nécessite 4 bits, de même celui de K, on repère les bits à 1 par la suite : (J1, K1), (J2, K2)... soit ici : (2, 2), (2, 3), (2, 11)... Il nous faut 54 octets, or le codage brut nécessite 32 octets.

Pour le cas 32 par 32 :

Le codage de J nécessite 5 bits, de même celui de K. Il y a 100 bits à 1, il faut donc 125 octets pour repérer leur position, or le codage de la matrice brut s'effectue sur 128 octets, on gagne donc 3 octets.

Le codage par matrice creuse est donc d'autant plus rentable dans notre cas, que la matrice de départ est grande. Il n'est pas rentable pour les petites matrices.

B - CODAGE PAR PLAGES

De même que le codage par matrices creuses, le codage par plages va utiliser le fait que la matrice issue du codage du contour comporte relativement peu de bits à 1. Il va considérer le fichier de manière séquentielle (colonne 1, colonne 2...) et coder la distance entre 2 bits consécutifs à 1.

Exemple sur le I :

La première indication repère le numéro de la colonne de départ.

Pour le cas 16 par 16 :

Le codage de la colonne de départ nécessite 4 bits ; la plus grande variation est de 12 positions binaires, nous prendrons également 4 bits pour la coder. Il nous faut environ 25 octets. On gagne par suite 4 octets par rapport au codage brut.

Pour le cas 32 par 32 :

Un raisonnement similaire montre qu'il faut environ 64 octets. On gagne par suite 64 octets.

Remarque : Le codage par matrices creuses permet une modification du fichier en temps réel, contrairement au codage par plages.

Ce qu'il faut retenir :

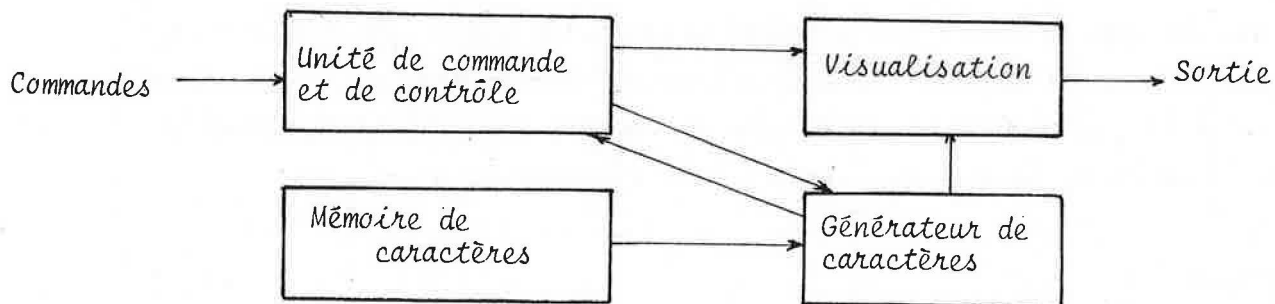
- il existe de nombreuses formes possibles de codage de la matrice de départ.
- ces formes, quelle que soit leur performance, ne permettent pas de stocker (du point de vue économique) de grosses matrices.
- elles n'utilisent pas les propriétés linguistiques et numériques des lettres, hormis le fait que le codage de la matrice par contour fournit une matrice comportant peu d'éléments à 1.

Un code est d'autant plus efficace qu'il est adapté aux caractéristiques de la structure à représenter.

PRINCIPES DE LA SOLUTION ADOPTÉE :

A) Principe général

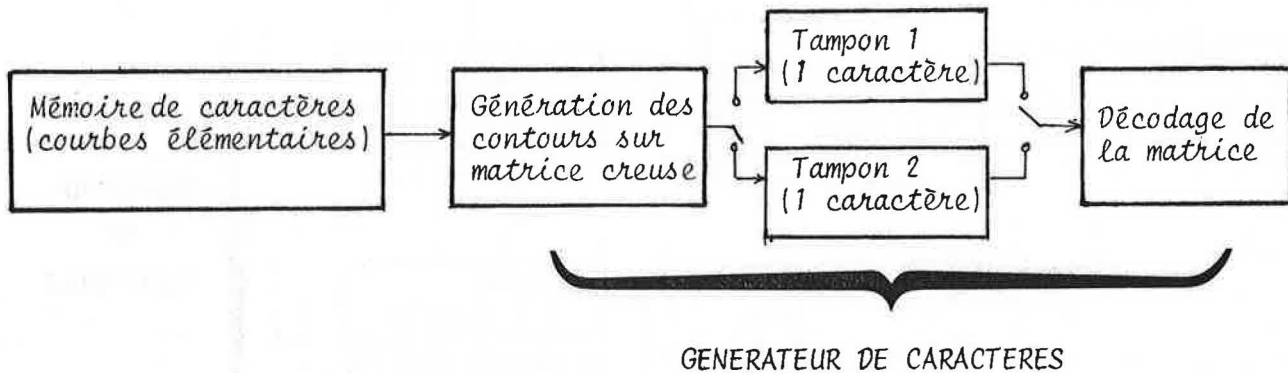
Rappelons le principe d'un générateur de caractères :



Ce type de machine substitue en fait à un codage brut des lettres, un codage "brut" du contour.

Nous pouvons affiner ce dernier en disant qu'il est obtenu par la concaténation d'une suite de courbes élémentaires (utilisation de la propriété de régularité des contours). A partir de ce codage on peut, en calculant point par point chaque courbe élémentaire, réobtenir le codage par matrice creuse.

Nous avons alors le schéma suivant :



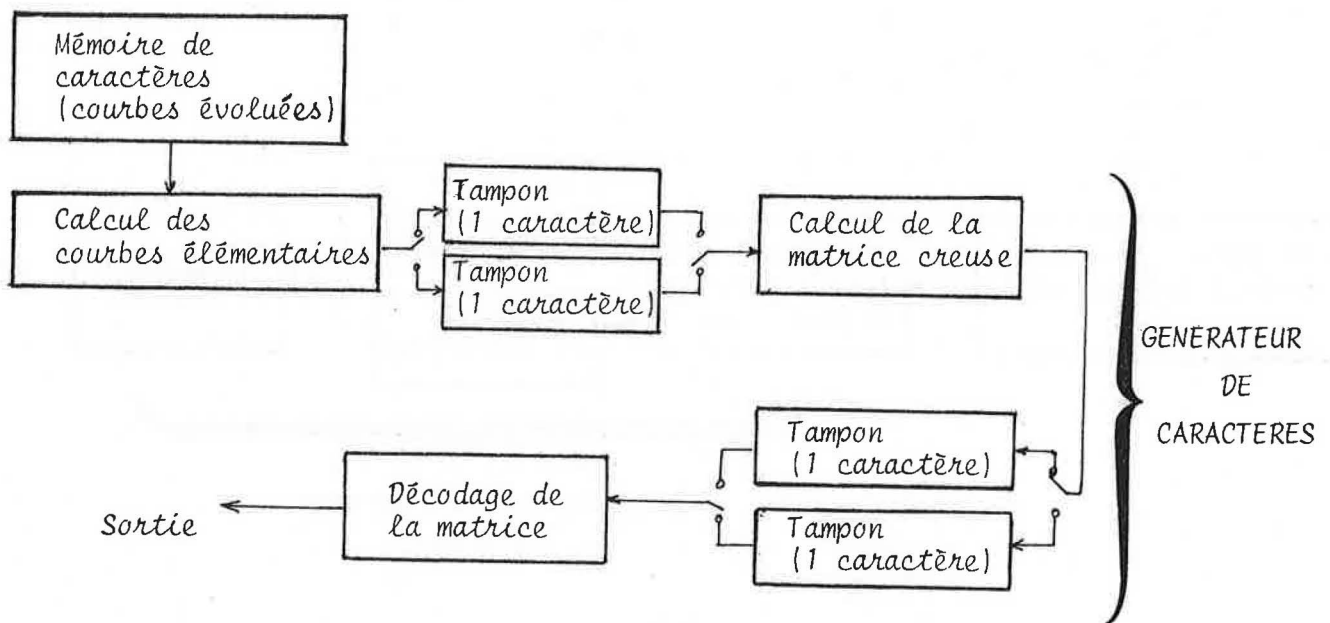
Le lecteur attentif aura remarqué que les deux étapes de décodage sont indépendantes :

- le premier étage calcule la matrice du caractère suivant à générer.
- le second étage décompacte la matrice qui vient d'être calculée.

Afin de conserver, la vitesse globale du système, il est nécessaire que les calculs effectués soient simples, par suite, la famille de courbes utilisées doit être réduite à sa plus simple expression.

L'étape suivante est évidemment d'utiliser une famille de courbes plus évoluées ; et obtenir par décodage du code des caractères réalisés à l'aide de cette dernière, le code dans la famille minimale (ici aussi, on utilise la propriété de régularités des contours). L'expérience montre que cela est réalisable sans perte des performances.

Le schéma est alors le suivant :

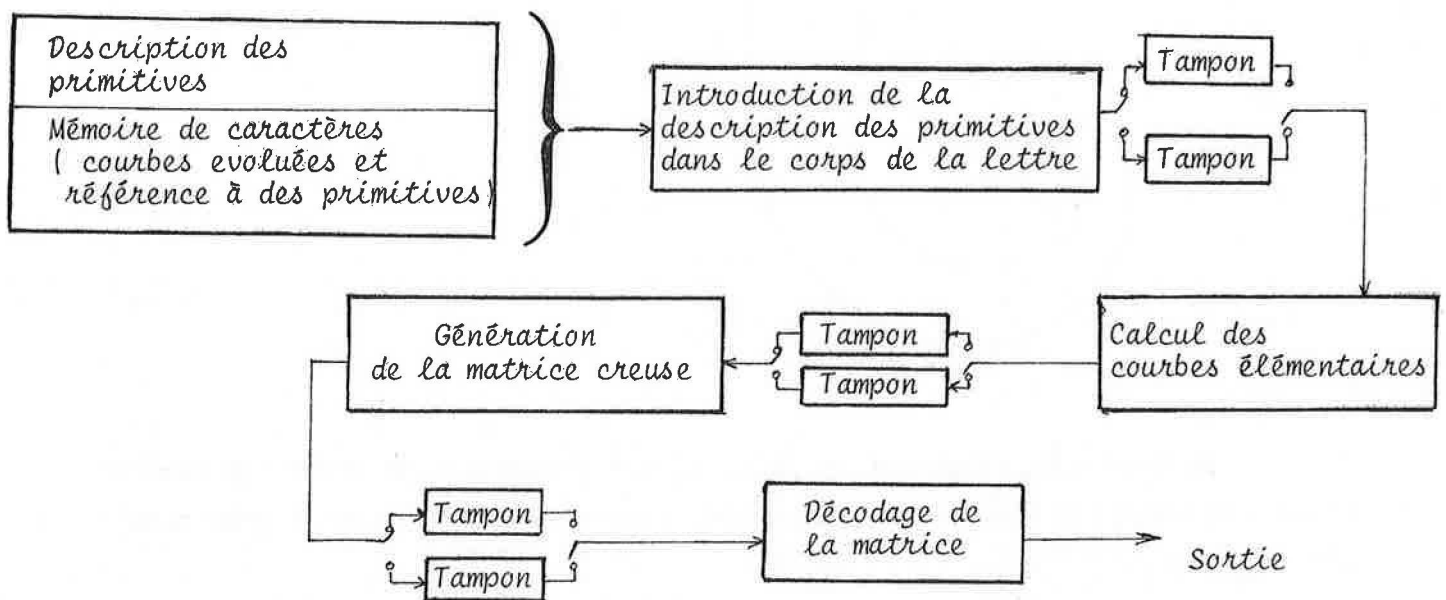


A ce stade, nous avons épuisé les possibilités de compaction du code par des considérations numériques. Pour aller plus avant, nous devons nous intéresser aux propriétés linguistiques des caractères romains.

La propriété la plus facile à utiliser est celle de la similitude des embellissements, voire de parties caractéristiques de certains caractères, à travers une même police. Nous introduirons donc une étape supplémentaire qui à partir des caractères décrits comme une suite de concaténations de courbes et de primitives (une primitive est une entité graphique utilisée de façon répétitive à travers une même police*) fournira les caractères décrits en tant que contour (s) "pur (s)".

* Cette notion ne correspond pas à la notion de primitive actuellement utilisée en infographie.

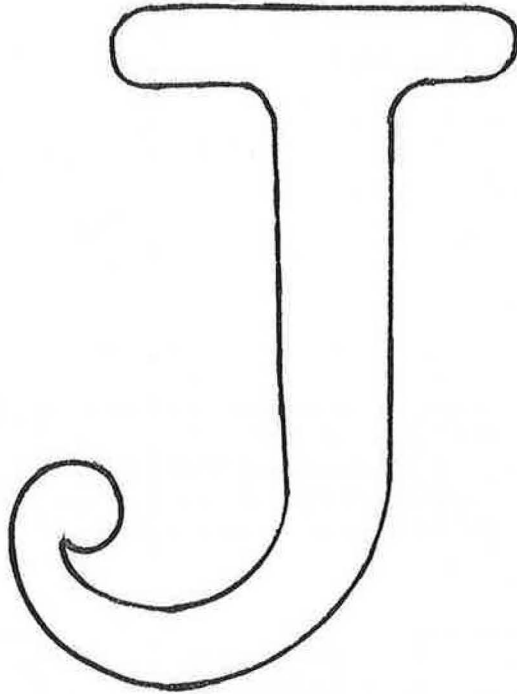
Le schéma est alors le suivant :



Remarque : Les embellissements présentent généralement de faibles variations entre eux. Ainsi pour le J, l'empattement n° 1, est légèrement plus long que le numéro 2, ceci afin d'équilibrer la lettre.

Empattement N° 1

Empattement N° 2



Si l'on veut respecter ce fait, il est nécessaire de coder les variations d'un embellissement d'une lettre par rapport au modèle choisi pour constituer une primitive.

Le premier étage de décodage est chargé d'inclure les variations apportées à une primitive lors de son introduction dans le corps de la lettre.

Le raisonnement que nous suivons présente un caractère récursif, et nous pouvons le poursuivre, mais n'oublions pas que chaque fois qu'une étape est ajoutée cela signifie pour la machine réelle plus d'électronique. Jusqu'au niveau présent ce fait est compensé par la baisse du coût mémoire.

Cette approche permet de conserver les performances globales sans augmenter le coût final (celui mémoire inclu). Le prix du changement d'un jeu de signes est moins élevé.

Deux problèmes n'ont pas été abordés :

- . comment conserver la vitesse de 1000 caractères par seconde lorsque l'on génère des matrices 2048 par 2048 ?
- . comment résoudre le problème de la mise à l'échelle des matrices creuses ?

B) Considérations sur la vitesse :

Notons que la vitesse des deux premiers étages dépend peu de la dimension de la matrice à générer ; ceci est dû au fait que le nombre de données à considérer à ces niveaux ne dépend que de la forme des caractères.

Par suite le problème ne se pose que pour les deux derniers étages de décodage.

Il est facile d'obtenir par décodage du caractère décrit dans la famille minimale, la matrice creuse décrivant le contour. Le seul travail de la dernière étape de décodage est alors de parcourir cette dernière, colonne par colonne (on suppose que l'on génère la lettre en une seule passe en la balayant de la gauche vers la droite).

1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	x
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	
4	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
5	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
6	0	0	1	1	1	0	0	0	0	0	0	1	1	1	0	0	
7	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	
8	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	
9	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	
10	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	
11	0	0	1	1	1	0	0	0	0	0	0	1	1	1	0	0	
12	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
13	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
14	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
y																	

Matrice décrivant le contour

La dernière étape extrait, par exemple, la colonne N° 13
(colonne N° 13 : 0 1 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0)
et l'examine dans le sens de la numérotation croissante des lignes. Au départ,
le spot est éteint. Lorsqu'elle rencontre le premier 1 elle allume le spot,
au second elle l'éteint, au 3ème elle le rallume et ainsi de suite, après quoi
elle passe à la colonne suivante.

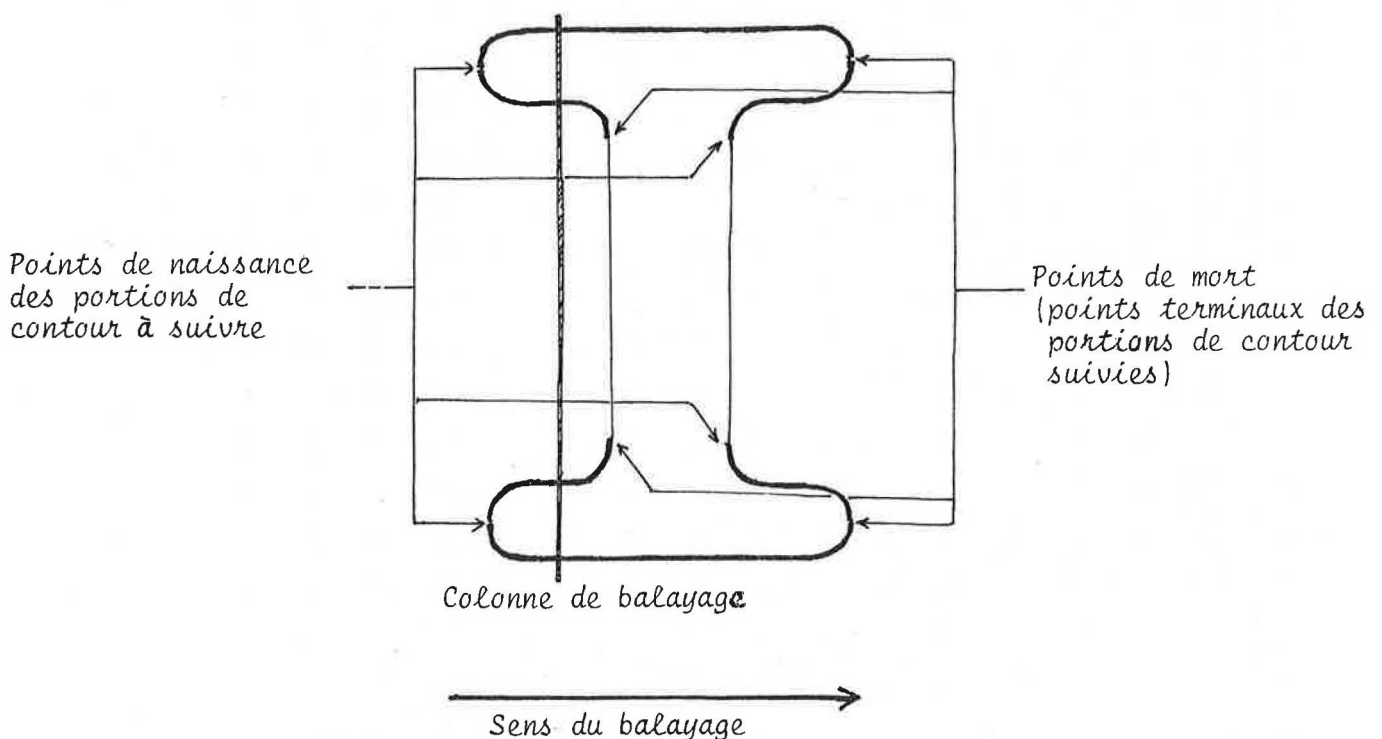
Remarque : Ceci revient à tracer les segments $\Delta 1$ et $\Delta 2$ définis par :

$\Delta 1$: M1 (x=13, y=2), M2 (x=13, y=5)

$\Delta 2$: M3 (x=13, y=11), M4 (x=13, y=14)

Ce travail paraît trivial. Cependant, son coût financier est non négligeable ; en effet, cette étape de décodage opère sur deux tampons de (2048 x 2048) bits, soit près de 8,4 millions de bits. Moyennant la connaissance, a priori, des portions de courbes à suivre simultanément lorsque l'on balaye la lettre de la gauche vers la droite, on peut engendrer successivement les colonnes de la matrice creuse.

EXEMPLE

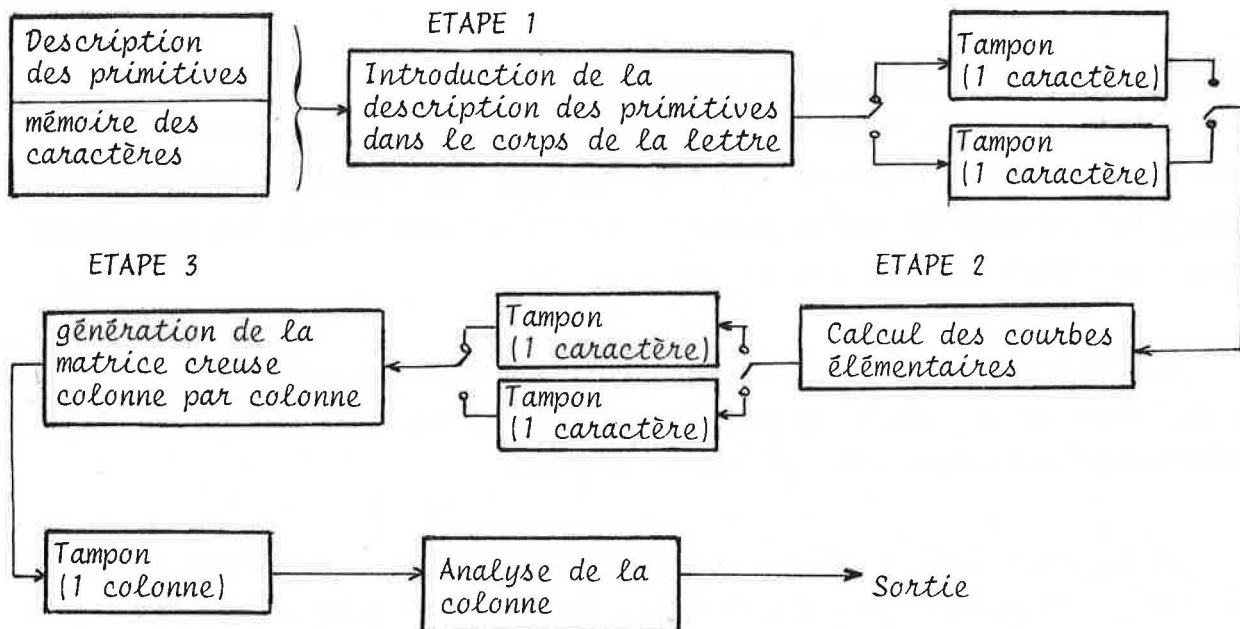


Par suite, pour chaque lettre, en plus de sa description en courbes évoluées et primitives, on doit posséder une structure décrivant les portions de courbes à suivre simultanément. (Par abus de langage, cette structure sera appelée "structure d'arbre").

Outre le fait qu'elle remplace les "énormes" tampons de caractères précédemment nécessaires par des tampons de colonnes *, cette méthode permet d'accélérer le processus de décodage par deux voies :

- la connaissance a priori des colonnes contenant effectivement de l'information.
- la possibilité de décoder plusieurs courbes élémentaires en parallèle.

Le schéma est alors le suivant



* En fait, le décodage des courbes élémentaires, fournit directement les coordonnées des points de contour. La suite de ces dernières suffit à rendre compte des tampons de colonnes.

C) Considérations sur la mise à l'échelle

A priori, la mise à l'échelle peut s'effectuer à tous les niveaux du décodage. Examinons les différentes possibilités :

- Avant la 1ère étape de décodage :

Il faut opérer des réductions sur des données numériques. Ceci est facilement faisable. Cependant, étant donné la non-connaissance, a priori, des primitives que va utiliser une lettre, il est nécessaire de traiter l'ensemble du fichier des caractères.

- Avant l'étape de décodage des courbes évoluées.

L'inconvénient précédent disparaît. Cependant, les courbes évoluées à traiter étant plus petites, la précision relative des calculs sera moins bonne.

- Avant l'étape de génération colonne par colonne de la matrice creuse.

Le traitement s'effectue comme précédemment au niveau des caractères. Comme il se situe après l'étape de décodage des courbes évoluées, il n'y aura pas d'effets de bord sur la précision du calcul des paramètres des courbes élémentaires.

- Après l'étape de génération des colonnes.

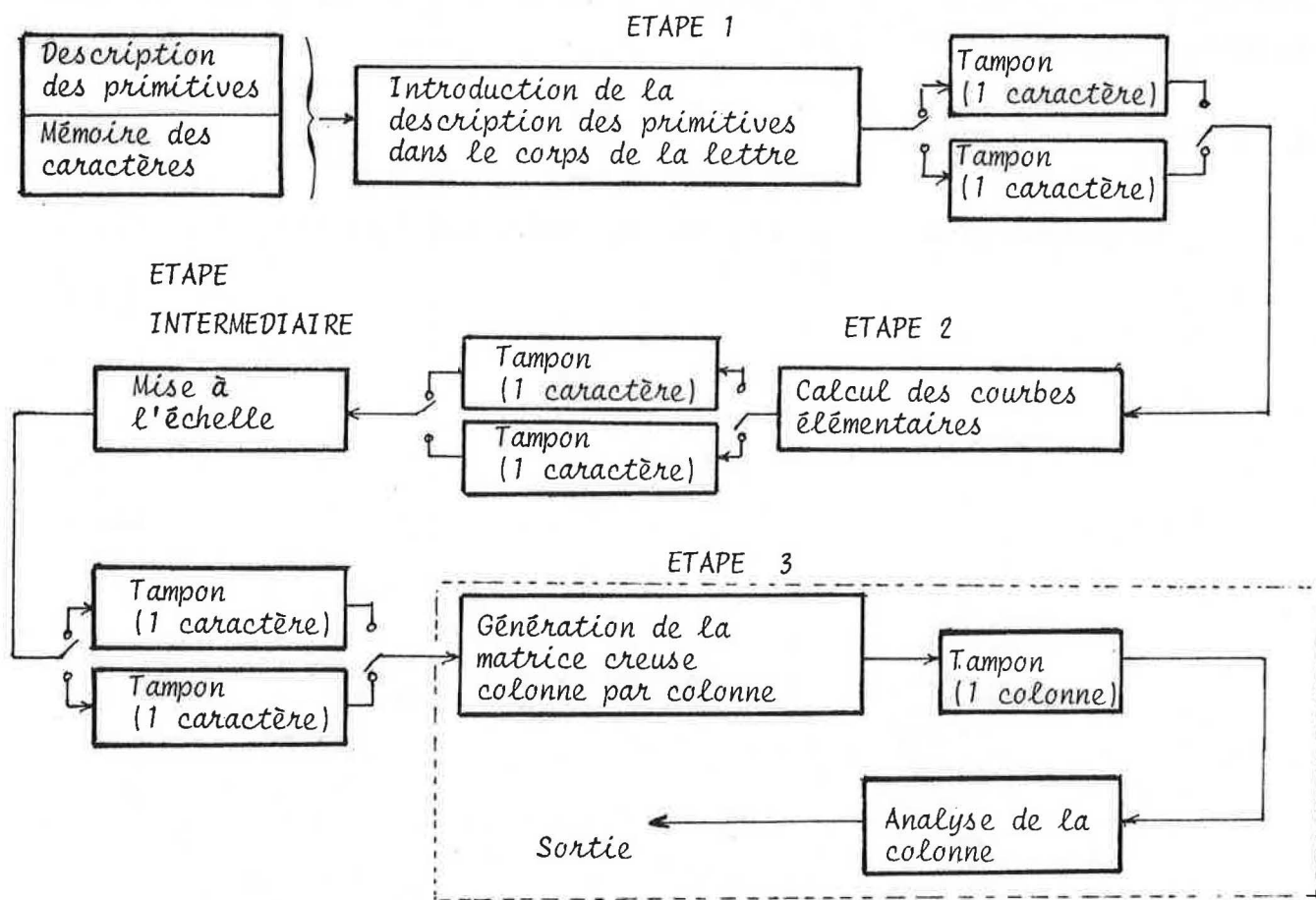
Le traitement ne s'effectue plus ici sur des données purement numériques ; Il est de ce fait plus complexe. Par exemple, pour une réduction 1/2 on sélectionnera les colonnes de numéro pair, et sur les données de ces colonnes on effectuera une homothétie de rapport 1/2*.

Contrairement au cas précédent, les dimensions de la matrice creuse ne sont pas réduites suivant le rapport de réduction ; par suite, le temps global de génération du fichier terminal reste constant.

Il apparaît donc que la meilleure solution est celle consistant à intercaler l'étape de mise à l'échelle entre la 2ème et 3ème étape de décodage.

* - Noter que le cas général devra traiter toutes les fractions, telles que 51/83.

Le schéma est alors le suivant :



OPTIONS CHOISIES :

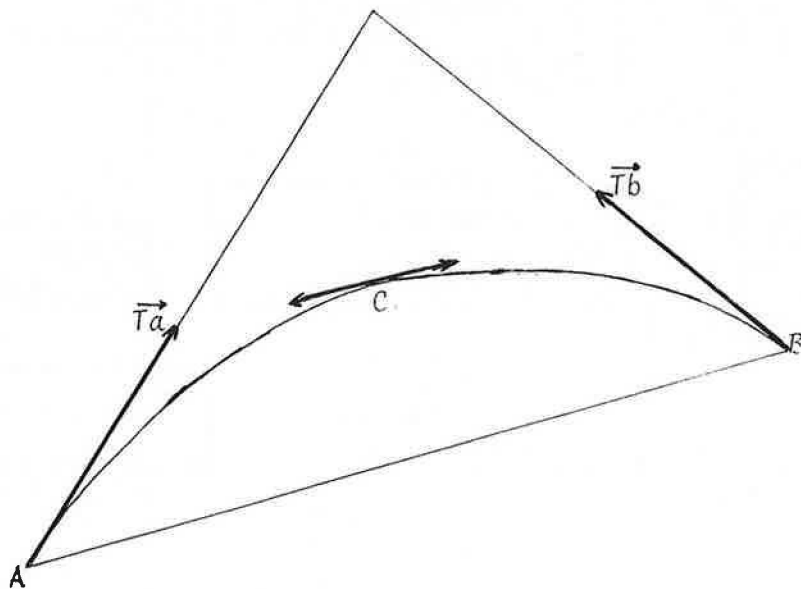
A - FAMILLE MINIMALE RETENUE :

La famille minimale est constituée par les arcs de cercles et les droites. En effet, on connaît des algorithmes très performants les générant[1]. Les algorithmes adoptés sont tels qu'ils fournissent pour un pas donné dans une direction, l'accroissement dans l'autre.

Ils sont issus d'une modification du principe de ceux calculant les accroissements en testant le signe de la dérivée totale de la fonction (droite ou cercle) au point à calculer [6, 7, 8, 9]

B - FAMILLE EVOLUEE RETENUE :

La famille retenue est basée sur les paramètres suivants [11 , 5]



- A Point de départ
- B point d'arrivée
- $\vec{T_a}$ demie-tangente au point de départ
- $\vec{T_b}$ demie-tangente au point d'arrivée
- C point de carrure (point de la courbe dont la tangente est parallèle à AB)

Jusqu'à présent on essayait directement de calculer les points de la courbe à partir de cette description (codée sous forme d'une équation). L'expérience a montré qu'il est pratiquement impossible alors de respecter la contrainte de temps.

Citons les principales équations utilisées :

$$- X = (\cos T) * * N \quad [10]$$

$$Y = (\sin T) * * N$$

$$- X = (A1 * T * T + B1 * T + C1) / (A2 * T * T + B2 * T + C2) [11]$$

$$Y = (D1 * T * T + E1 * T + F1) / (D2 * T * T + E2 * T + F2)$$

$$- X = A1 * T * T * T + B1 * T * T + C1 * T + D1 [12]$$

$$Y = A2 * T * T * T + B2 * T * T + C2 * T + D2$$

La génération des points étant obtenue en considérant les équations aux différences finies.

Le lecteur remarquera que les solutions précédemment citées, consistent en fait à passer directement de la représentation d'une courbe évoluée, à la représentation point par point.

Notre solution, au contraire, décompose une courbe évoluée en arcs de cercle, puis, à partir de ces derniers, génère la représentation point par point : Ceci sépare bien la complexité due aux calculs sur les courbes évoluées de celle qui résulte du nombre de points à engendrer, d'où le gain de vitesse.

Pour notre part, le problème est de décomposer la courbe en arcs de cercles et droites.

L'étude de la position des cercles osculateurs montre que la famille est engendrabable à partir d'arcs de cercle (la courbe obtenue n'est évidemment pas de classe c infini). Ces arcs correspondent à des faisceaux de cercles. Nous avons retenu la réalisation qui donne la meilleure impression visuelle. Le calcul des paramètres liés aux arcs s'effectue en 64 opérations élémentaires (addition, soustraction, multiplication, division, extraction de racines carrées).

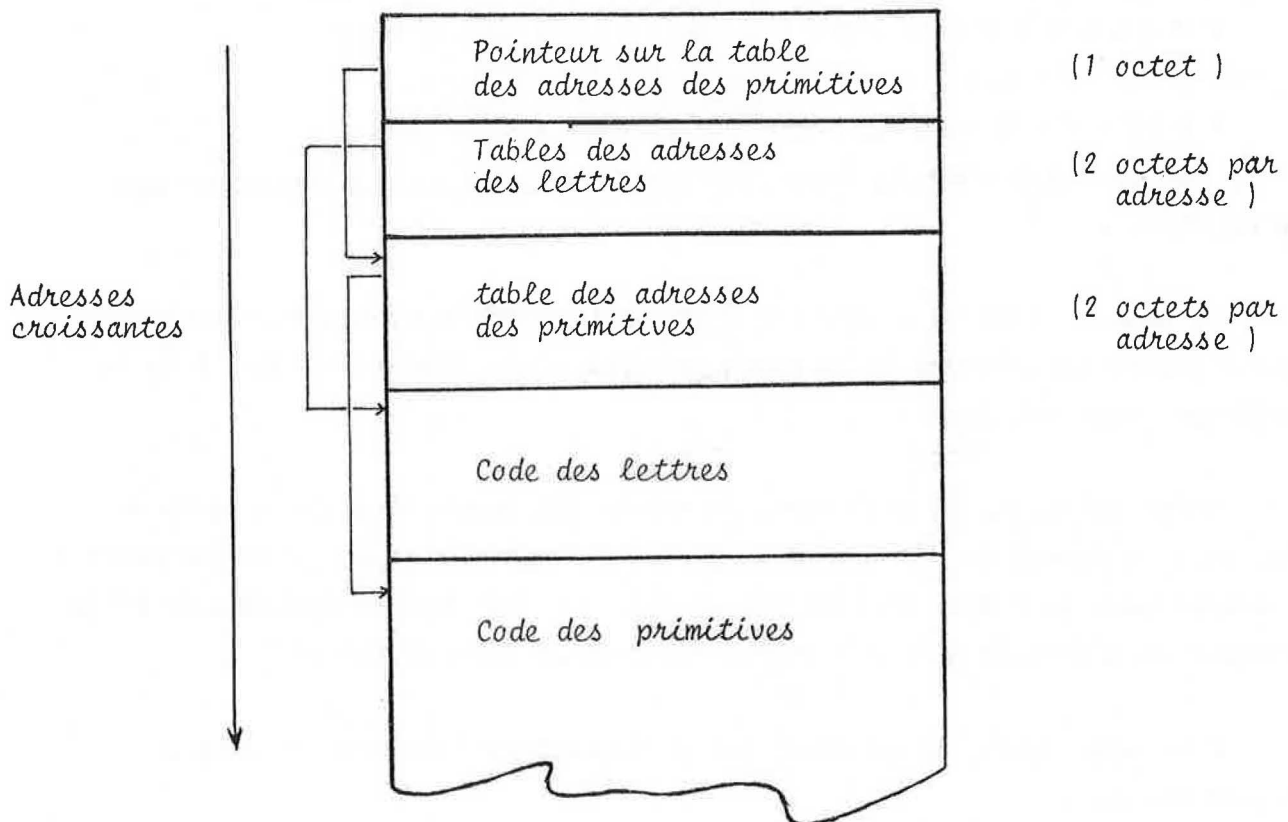
C - DESCRIPTION D'UNE POLICE

Comme nous l'avons vu précédemment, nous avons à décrire les caractères en tant que concaténation de portions de courbes et de primitives. De ce fait, le code originel comprend deux parties :

- un fichier décrivant les caractères,
- un autre décrivant les primitives.

Chaque caractère ou primitive porte un numéro d'ordre qui permet de les retrouver à partir d'une table de pointeurs.

Nous obtenons la structure suivante :



La longueur des mots machines est en général un nombre entier d'octets. Or, pour coder des grandeurs allant de 0 à 2048, il nous faut 11 bits, par suite, nous utiliserons 2 octets pour stocker une grandeur. Les 5 bits restants seront utilisés pour fournir des indications particulières.

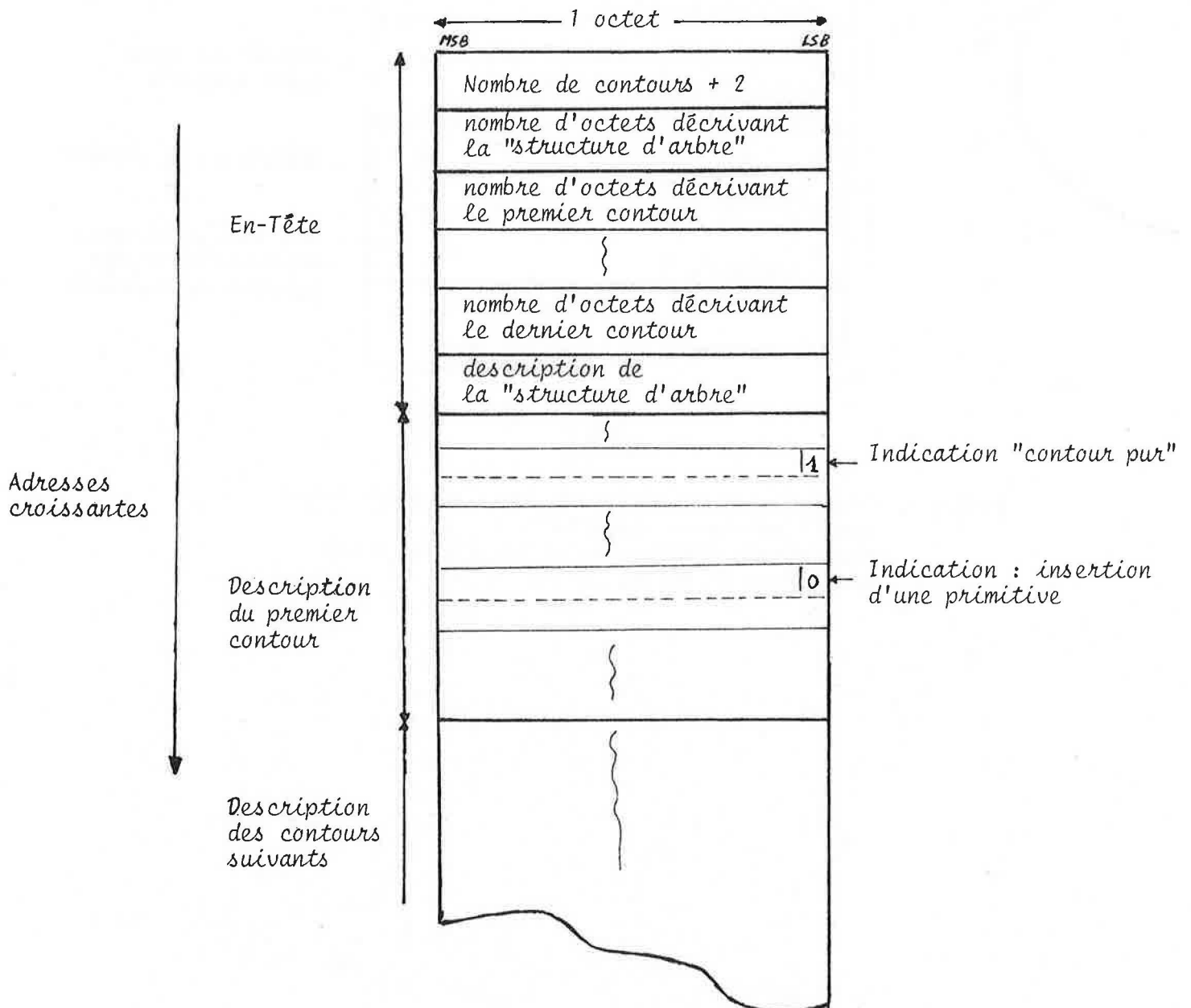
Exemple : La majorité des tangentes que nous aurons à coder sont soit des tangentes verticales, soit des tangentes horizontales. Lorsque l'on fournit les coordonnées d'un point, on codera directement sur les bits libres cette indication (ou le fait que les données suivantes soient les tangentes).

Grâce à un tel codage, le nombre de bits nécessaires par caractère pour coder une police est d'environ 750.

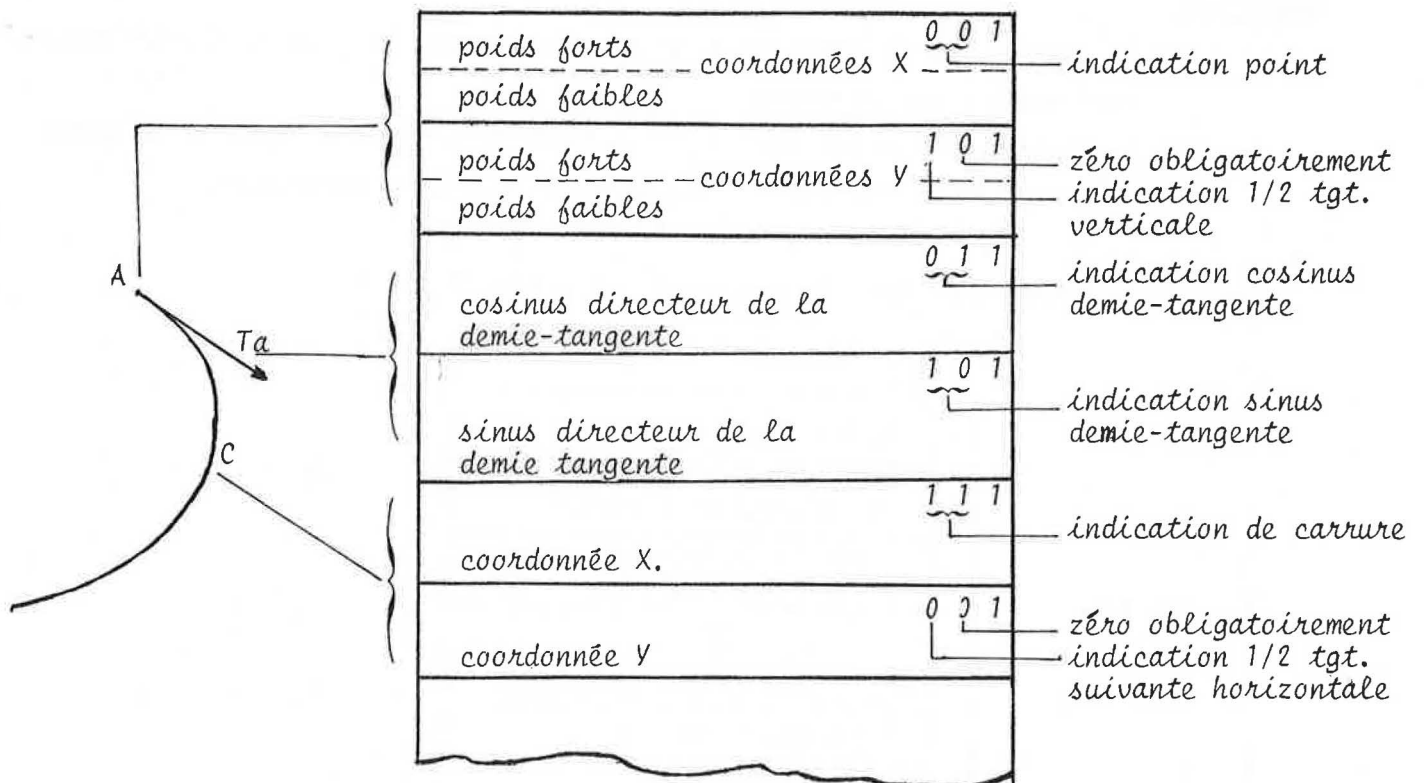
Remarques :

- les variations à apporter à un embellissement lors de sa concaténation sont codées sur un octet.
- la structure d'arbre indiquant comment le troisième étage doit opérer, est stockée en tête de la description de chaque caractère.

Nous avons donc pour un caractère le schéma suivant :



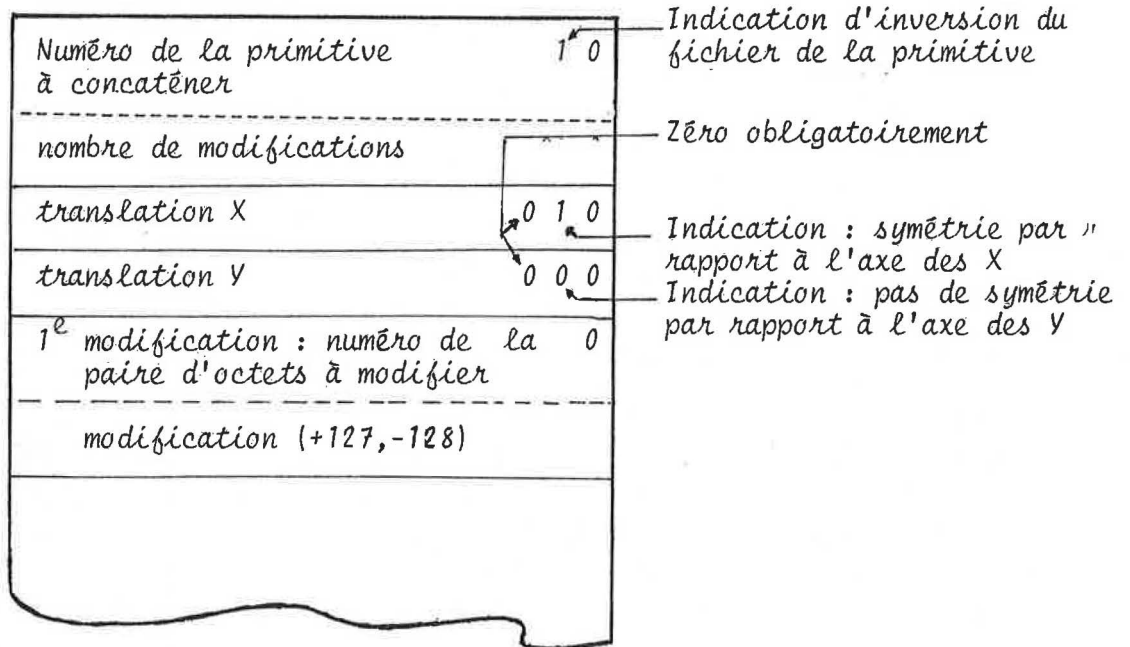
Détail "contour pur"



1 octet

- Règles :
- l'indication explicite de demie-tangente prime sur l'indication implicite .
 - deux points consécutifs correspondent à une indication de droite .

Détail "indication de primitive"



CONCLUSION :

Ainsi ont été résolus les différents problèmes. Le lecteur trouvera dans le chapitre suivant la description des performances du simulateur et l'extrapolation de ces dernières à une éventuelle réalisation utilisant les possibilités actuelles de la technique.

BIBLIOGRAPHIE

LE GENERATEUR DE CARACTERES

OUVRAGE DE REFERENCE

- [1] MORVAN Pierre, LUCAS Michel
"Image et ordinateur - Introduction à l'infographie interactive"
Librairie Larousse 1976

TECHNIQUES DE CODAGE

- [2] MURAT KUNT
"Comparaison de techniques d'encodage pour la réduction de redondance d'images"
Thèse 183, Département d'électricité
Ecole Polytechnique de Lausanne - 1974
- [3] COUEIGNOUX Ph.
"Comparaison of type face by contour encoding"
S.M. Thesis Department of electrical engineering, M.I.T. - 1973
- [4] COUEIGNOUX Ph.

Ph.D Thesis, Department of electrical engineering, M.I.T. - 1975
Chapitre 4
- [5] WURDLE Caroline
"Artspeak : A graphics language for Artists"
ACM Sigplan notices - Vol II, N° 6, Juin 1976
Pages 32 à 38

ARTICLES TRAITANT DES METHODES DE GENERATIONS D'ARCS DE CERCLE

- [6] WON L. CHUNG
"Notes on circle generation algorithms"
Computer Graphics and Image Processing - N° 6 - 1977
Pages 196 à 198
- [7] DENERT Ernst
"A method for computing point of a cercle using only integers"
Computer Graphics and Image Processing - N° 2 1973 - Pages 83 à 91
- [8] JORDAN Bernard W., LENON William J., HOLM Barry D.
"An improved algorithm for generation of non parametric curves"
I.E.E.E. - Transactions on Computers - Vol C22 - Numéro 12 - 1973
- [9] HORN Berthol K.P.
"Circles generators for display devices"
Computer Graphics and Image Processing N° 5 - 1976 - Pages 280 à 288

SUR LE DECODAGE DE COURBES EVOLUEES

- [10] MERGLER H.W. and VARGO P.M.
"On approach to computer assisted letter design"
The journal of Typographic Research, Vol II, N° 4, Octobre 1968
Pages 299 à 322

- [11] COUEIGNOUX Ph.
Ph. D Thesis - M.I.T.
Chapitre 3

- [12] COUEIGNOUX Ph.
Ph.D. Thesis - M.I.T.
Appendice II

EDITION DES CARACTERES

Afin d'automatiser la production des caractères exploitables par le générateur, il a été créé un éditeur de caractères. Ce dernier est composé de trois parties :

- un bibliothécaire mémorisant les données,
- un éditeur de lettres,
- un interface permettant de produire le fichier final à partir des données mémorisées.

LE BIBLIOTHECAIRE

Il gère un espace de quatre méga octets. Cela permet de mémoriser une vingtaine de polices.

Chaque police est repérée par un nom ; ce nom est limité à 12 caractères alphanumériques. Le fichier alloué à une police est divisé en deux sous fichiers :

- l'un est réservé à l'édition des primitives (chaque primitive est repérée par un nom d'au plus 12 caractères),
- l'autre est réservé à l'édition des lettres.

Le nombre de primitives est limité à 1024, de même le nombre de lettres. Un fichier de primitives comprend au plus 256 courbes élémentaires. Un fichier lettre comprend au plus 256 courbes élémentaires ou références à des primitives.

L'espace alloué aux différents fichiers est géré dynamiquement (c'est-à-dire alloué au fur et à mesure des besoins, et retiré s'il n'est plus utilisé).

La forme générale des commandes est : NOM1/CLE : NOM2

Les différents niveaux se signalent par l'impression d'un mnémonique :

- PO* niveau opération sur polices
- PR* niveau opération sur primitives
- CH* niveau opération sur caractères

A chaque niveau les opérations possibles sont :

- déclaration de fichier
- opération sur un fichier déclaré
- destruction d'un fichier déclaré

EDITEUR DE LETTRES

Comme nous l'avons vu précédemment, l'éditeur comprend deux parties : l'une permettant l'édition des primitives, l'autre permettant l'édition des lettres.

Chaque partie se subdivise en deux modules :

- le premier réalise l'édition
- le second facilite la correction

Ces modules permettent l'entrée des données de deux manières distinctes :

- une entrée par une tablette graphique
- une entrée directe à partir d'un clavier.

De même, il y a deux modes de visualisation des données :

- les données entrées sont directement affichées sous forme numérique
- les courbes correspondant aux données sont tracées sur un terminal graphique.

Nous avons dit qu'un fichier de primitive ou de lettre comprend au plus 256 courbes. Une courbe est définie par :

- son point d'arrivée,
- la demi-tangente de la courbe en ce dernier
- le point de carrure
- la demi-tangente au point de départ.

Le point de départ est fourni lors de la définition de la courbe précédente. Le premier point (il n'appartient à aucune ligne éditée) est dit point origine, et est entré comme première donnée.

Pour les lettres, nous aurons en plus la possibilité d'indiquer l'insertion de primitives et des variables attenantes à ces dernières (symétrie, inversion, variations).

L'INTERFACE

L'interface permet de traduire les données mémorisées sur une police (ou sur un certain nombre de caractères de cette dernière) en des données exploitables par le générateur de caractères.

Son plus important travail est de calculer la structure qui permettra au troisième étage de décodage de générer le caractère en temps réel. Ce travail étant long et complexe, la traduction des données est relativement lente : 30 secondes par caractère.

L'interface constitue à lui seul trente pour cent de la tâche. Le fichier produit n'est pas inclus dans le bibliothécaire.

DETAILS SUR L'ENTREE DE DONNEES

Ainsi que nous l'avons vu précédemment, une ligne de "texte" est constituée par quatre paramètres. En mode d'entrée tablette graphique, une demi-tangente est fournie en pointant un point quelconque situé sur cette dernière.

L'utilisateur dispose sur la tablette d'un certain nombre de touches de fonction qui ont pour objet de fournir des indications particulières (indication des droites, tangentes verticales ou horizontales etc...) et de limiter les interactions avec un clavier (touche de changement de mode, indications du paramètre à modifier etc...)

Chaque donnée entrée est vérifiée ; la validité est signalée par la sortie du caractère "bell" ; ceci permet à l'utilisateur de se consacrer uniquement à la tablette.

REMARQUE QUANT A L'IMPLEMENTATION.

Des précautions particulières ont été prises pour garantir les données tant au point de vue erreurs du système (la validité des données est vérifiée chaque fois que le système les utilisent) qu'au point de vue erreurs dues au matériel (tel qu'un défaut du secteur) ; dans ce dernier cas, le rattrapage s'effectue par deux passages consécutifs dans une routine quelconque manipulant les données suspectes (en aucun cas, la perte ne peut être supérieure à une ligne de "texte").

CONCLUSION

Ce système permet donc de générer des ensembles de contours. Les applications ne se limitent pas aux caractères. A ce sujet, le lecteur trouvera dans cette thèse plusieurs graphismes qui ont peu de choses communes avec ces derniers.

BIBLIOGRAPHIE

EDITION DE CARACTERES

RUDNIK Antoine

"Réalisation d'une base de données sur mini-ordinateur"

Rapport de 3ème année - Ecole Nationale Supérieure des Mines de
Saint-Etienne - 1978

LUCAS Michel

"La programmation des consoles de visualisation"

Université Scientifique et Médicale de Grenoble
Laboratoire d'Informatique

CHAPITRE III

MESURES EFFECTUEES SUR LA SIMULATION

Le simulateur a été réalisé sur un PDP 11/40 possédant la multiplication (16 bits par 16 bits) et la division (32 bits par 16 bits) cablées.

Les organes périphériques utilisées sont :

- un traceur électrostatique GOULD ayant une définition de 200 points par pouce.
- un écran graphique Tektronix 4015
- une tablette graphique Tektronix 1024 x 1024 points.

Les mesures effectuées sont de quatre types :

- encombrement de la réalisation
- encombrement des fichiers traités
- qualité de reproduction obtenue
- temps d'exécution des différentes phases de décodage.

Mais avant de donner le résultat de ces mesures, il est utile d'indiquer les contraintes liées à la réalisation actuelle.

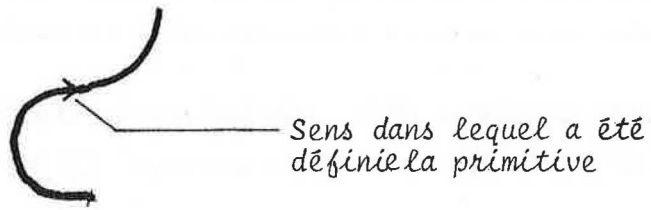
A) CONTRAINTES DE LA REALISATION ACTUELLE

NOTION DE FIGURE

Par rapport à un système graphique conventionnel, la notion de figure est remplacée par celle de primitive. Cette dernière fournit une méthode aisée de concaténation de courbe.

Le système actuel ne permet pas d'effectuer une rotation quelconque sur les primitives ; seules les symétries sont permises. De plus, une primitive peut être concaténée dans le sens de lecture ascendante ou descendante du fichier (c'est-à-dire dans le sens de sa description ou dans le sens inverse. Il y a donc huit façons de concaténer ces derniers).

Exemple :



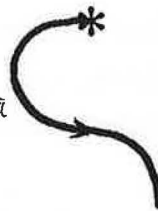
Primitive : "Empattement"



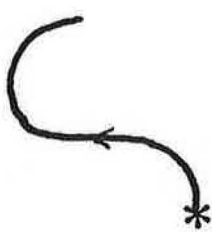
Concaténation directe



Concaténation avec symétrie par rapport à l'axe des Y



Symétrie "X" Symétries "X" et "Y"



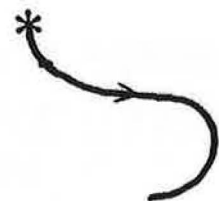
concaténation avec inversion du fichier de la primitive



Inversion du fichier et symétrie "y"



Inversion du fichier et symétrie "X"



Inversion du fichier et symétries "X" et "Y"

- * Point de concaténation
- ➔ Sens de parcours

NOTION DE ZONE DE COLORATION (TACHES)

Les zones à colorer sont délimitées par les contours. Lorsque la figure présente plusieurs contours, on applique la règle suivante :

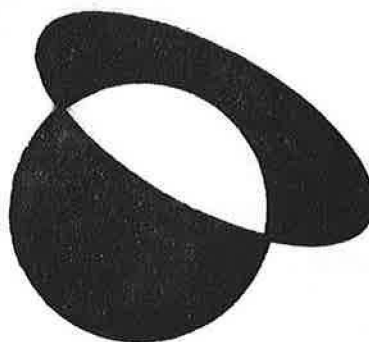
- Tout point situé à l'intérieur d'un nombre impair de contour est à colorer.

Note : Par convention zéro est considéré comme un nombre pair.

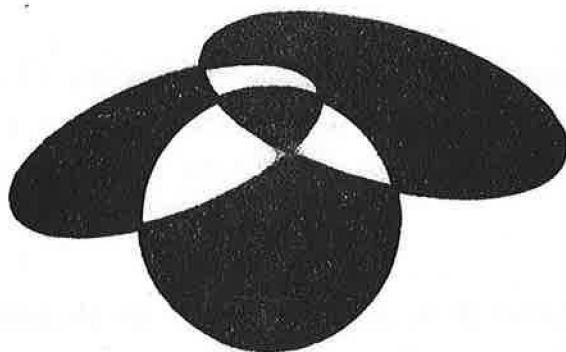
- Tout point situé sur un contour est traité de la même manière que les points intérieurs à ce dernier.



Cas de 1 contour



*Cas de 2 contours
se coupant*

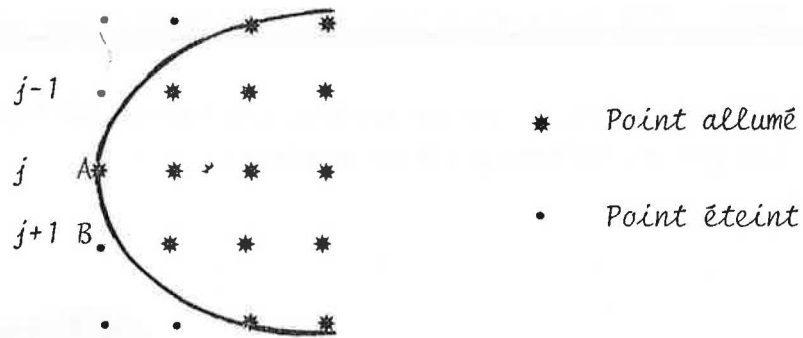


*Cas de 3 contours
se coupant*



*cas de 2 contours
distincts*

Le lecteur remarquera que la méthode d'analyse colonne par colonne, respectueuse par nature la règle précédemment énoncée ; le nombre de points de contour situé avant le point courant sur la colonne donne la parité cherchée, sauf dans le cas suivant :



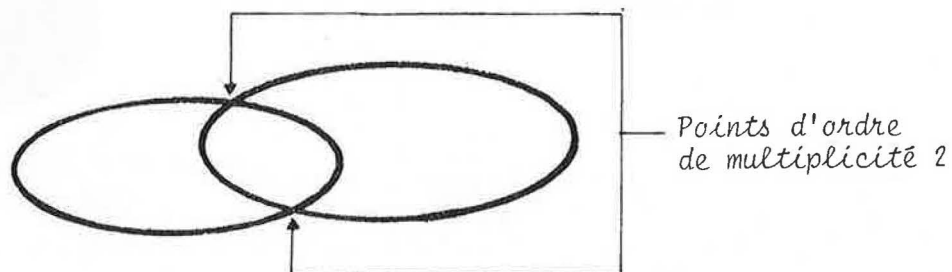
Le point $A(i, j)$ est un point allumé, le point suivant $B(i, j+1)$ doit être éteint.

Si on applique brutalement la règle d'analyse de colonne B sera allumé.

On résout le problème en remarquant que A est un point de naissance et en faisant la convention que les processus chargés de suivre les portions de courbes en parallèle génèrent leur point de départ.

Ainsi A étant un point de naissance de 2 portions de courbe, il sera généré en parallèle par deux processus. L'analyse de la colonne donnera : tracer le "segment" défini par les points A, A. Le problème des points terminaux est résolu de façon identique.

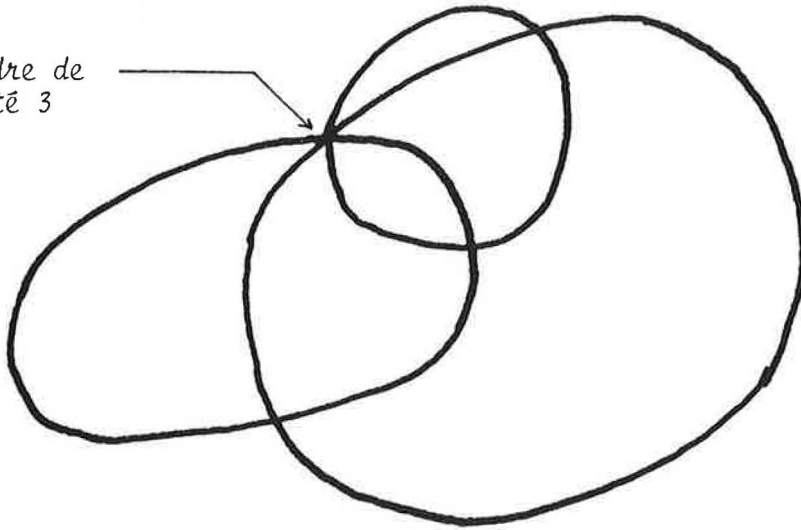
Remarque : On dit que l'ordre de multiplicité du point A est 2. Un phénomène identique se passe lors de l'intersection de contours.



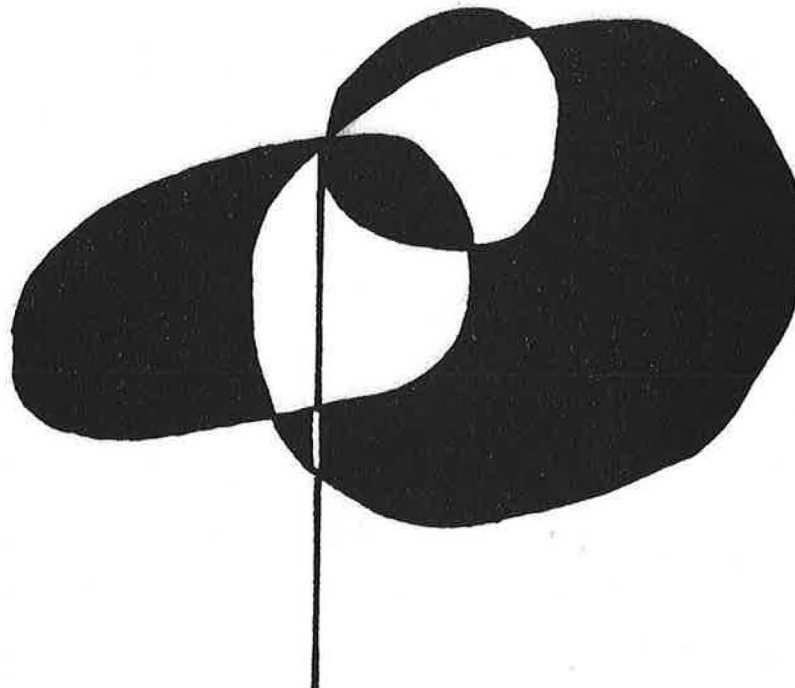
Le système actuel ne permet pas de résoudre les cas où l'ordre de multiplicité des points est supérieur à 2.

Ainsi, pour le cas suivant :

Point d'ordre de
multiplicité 3



Il fournira :



POINTS DE NAISSANCE

1°) Rappel

Les points de naissance sont les points de départ du suivi de parties de contours lorsque l'on examine la lettre de la gauche vers la droite

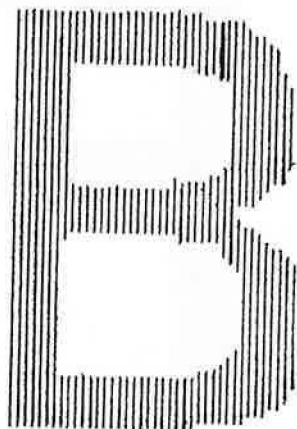
2°) Contraintes

Les points de naissance doivent être explicitement définis dans la description du dessin. Un point de carrure ne peut être un point de naissance.

PRECISIONS DES CALCULS.

Afin d'effectuer les traitements numériques le plus rapidement possible, la précision des calculs a été réduite au minimum. Les opérations sont effectuées soit sur des entiers soit sur des nombres codés en virgule fixe. L'erreur de positionnement des points intermédiaires est au plus de deux unités. Cette erreur est d'autant plus gênante que les arcs de cercles sont petits.

Dans la pratique, elles deviennent visibles lorsque l'on réduit la matrice à des dimensions de l'ordre de 70 x 70.



Matrice de
64 par 64 points

IMPRESSION DES RESULTATS

Les figures ci-jointes illustrent la qualité graphique que le système permet d'obtenir

RAPPEL : Ces figures ont été réalisées sur un traceur électrostatique ayant une définition de 200 points par pouce.

UNIVERS MEDIUM 689

matrices de 512 par 512 points

A B C

D E F G

H I J K

L M N

O P Q

R S T

U V W

X Y Z

UNIVERS MEDIUM 689

matrices de 256 par 256 points

A B C D E F
G H I J K L M
N O P Q R S
T U V W X
Y Z

UNIVERS MEDIUM 689

matrices de 128 par 128 points

A B C D E F G H I

J K L M N

O P Q R S T U V

W X Y Z

UNIVERS MEDIUM 689

matrices de 64 par 64 points

A B C D E F G H I J K L M N O P Q

R S T U V W X Y Z

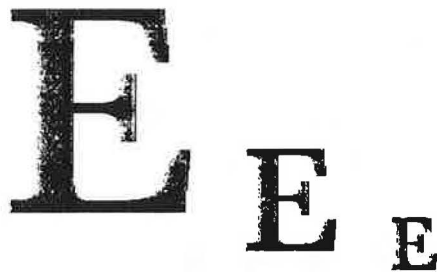
a b c d e f g h i j k l m n o p

q r s t u v w x y z



E majuscule, Baskerville 169

Réduction imprimante 1/1, 2/5, 1/5, 1/10, 1/20
Réduction photographique 1/1



K k

k k k

K minuscule, Baskerville 169

Réduction imprimante 1/1, 2/5, 1/5, 1/10, 1/20
Réduction photographique 1/1

Q

Q

Q

Q

Q

Q majuscule, Baskerville 169

Réduction imprimante 1/1, 2/5, 1/5, 1/10, 1/20

Réduction photographique 1/1

DIFFERENCES ENTRE POSSIBILITES DE L'EDITEUR ET DU GENERATEUR

L'éditeur comporte des restrictions par rapport au générateur. Ces dernières sont :

- Obligation de fermer un contour par une droite
- Obligation de concaténer une primitive par une droite
- Impossibilité de définir des variations sur les primitives.

CONCLUSION

- La réalisation et l'étude pratique du système ont montré que
- Les limitations dues à l'éditeur ne sont pas fondamentales.
 - Il serait utile de pouvoir définir, au niveau de l'éditeur, des primitives issues de concaténations de primitives, ainsi que de caractères issus de concaténations d'autres caractères.
 - Il serait utile d'avoir un système simple d'empilage des différentes versions d'un caractère ou d'une primitive.

B) ENCOMBREMENT DE LA REALISATION

L'encombrement global du simulateur (écrit en assembleur PDP 11) est de 28 K octets. Cet ensemble se subdivise en :

- 6 K octets pour la réalisation de la première étape
- 6 K octets pour la réalisation de la deuxième étape
- 1 K octet pour l'étape intermédiaire
- 15 K octets pour la dernière étape.

Ces chiffres sont purement indicatifs ; ainsi pour la dernière étape de décodage, afin de respecter les propriétés qu'aurait une machine traitant les différentes parties d'un contour en parallèle, l'étape 3 comporte 5 "sous-programmes" identiques simulant chacun un processeur capable de suivre une portion de courbe.

C) ENCOMBREMENT DES FICHIERS

Le système comporte plusieurs fichiers. Lors de la description du générateur de caractère, nous avons indiqué assez précisément la structure des fichiers d'entrée de l'étape 1. Il est aisé, à partir de cette dernière de se faire une idée de l'encombrement d'une police.

Le simulateur et le générateur de caractères ont permis de vérifier que le but recherché (\approx 700 bits par caractères) était atteint. Ainsi, pour les majuscules de la police Univers (voir tableau page 116), nous avons obtenu une moyenne de 605 bits par caractère (et ceci sans utiliser la notion de primitive). L'encombrement d'une police du type Baskerville est d'environ 800 bits par caractère (en utilisant la notion de primitive).

Ceci montre qu'un fichier de 16 K octets est suffisant pour coder la totalité d'une police de 128 caractères.

L'écart possible de l'encombrement d'une lettre par rapport à la moyenne peut être très important ainsi (voir tableau page 116), le I majuscule Univers occupe 27 octets, alors que S occupe 149 octets.

La taille des fichiers intermédiaires (tampons) doit être fixée en fonction des maximaux. Parmi toutes les lettres réalisées, la lettre présentant l'encombrement maximal est le g minuscule type Baskerville. C'est par suite en fonction de ce dernier que l'on doit choisir la taille des tampons. Les tailles préconisées sont :

- 512 octets, pour les tampons situés entre les étapes 1 et 2
- 1024 octets pour les tampons situés entre les étapes 2 et 3.

Le facteur de décompaction du code entre les deux dernières étapes est d'environ 3. Le facteur dépend peu de la lettre. Le facteur d'extension entre les deux premières étapes, dépend de l'utilisateur ; si ce dernier n'a pas utilisé les primitives, le facteur est 1. Pour la police Baskerville que nous avons codée, le facteur d'extension est d'environ 3.

D) QUALITE DE REPRODUCTION :

Les caractères doivent être définis dans des matrices 2048 x 2048 ; Or nous ne disposons comme entrée que de la tablette graphique 1024 x 1024. De plus, la précision de désignation sur la tablette est d'environ ± 5 points.

La tablette étant cependant de dimension respectable (environ 20 x 20 cm), les lettres à reproduire doivent être de grandeur similaire. Nous n'avons pas pu obtenir d'originaux de cette taille. Par suite, nous avons procédé à un agrandissement photographique de polices tirées du catalogue de Monotype . Les polices sont de corps 9 pour Univers Médium 689 et 10 pour Baskerville 169 ; l'agrandissement photographique choisi est 50. Le bruit résultant de cette opération est important.



Z

X

W

N

ORIGNAUX

Echelle 1/1

SYNOPSIS IN 9D ON 10 POINT

ABCDEFGHIJKLMNOPQRSTUVWXYZ&Æ

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyzæ

ABCDEFGHIJKLMNOPQRSTUVWXYZ&Æ

abcdefghijklmnopqrstuvwxyzæ

1234567890 1234567890

.,:;!?"-([†\$£\$*—\$£\$†])-"?/;,:.

UNIVERS MEDIUM 689

SYNOPSIS IN 10 POINT

ABCDEFGHIJKLMNOPQRSTUVWXYZ&

ABCDEFGHIJKLMNOPQRSTUVWXYZÆ

abcdefghijklmnopqrstuvwxyzfifflææÆ

ABCDEFGHIJKLMNOPQRSTUVWXYZ&

abcdefghijklmnopqrstuvwxyzfifflææÆ

.,:;!?"-([†\$£\$*—\$£\$†])-"?/;,:.

1234567890 1234567890

BASKERVILLE 169

La police Univers étant relativement simple, a pu être introduite par l'intermédiaire de la tablette. Pour la police Baskerville, il a fallu se résoudre à relever tous les paramètres et les introduire numériquement.

Les épreuves obtenues étant sorties sur un traceur électrostatique d'une définition de 200 points par pouce, ne sont pas à l'échelle. Pour obtenir différentes forces de corps, nous avons opéré des réductions photographiques de 1/18^{ème} et 1/9^{ème} respectivement sur des matrices de taille 1/1 et 1/2 par rapport à la matrice de codage.

Le lecteur peut juger par les planches qui suivent du résultat de ces opérations.

A B C D E F G H I J K L M N
O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p
q r s t u v w x y z

Epreuve imprimante, rapport de réduction 1/1
Réduction photographique 1/9

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

Epreuve imprimante, rapport de réduction 1/1
Réduction photographique 1/18

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

Epreuve imprimante, rapport de réduction 1/2
Réduction photographique 1/9

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

Epreuve imprimante, rapport de réduction 1/2
Réduction photographique 1/18

& Æ Œ œ 0123456789

no p q r s t u v " . , : ! ?

G H I J K L M N O

Epreuve imprimante, rapport de réduction 1/1
Réduction photographique 1/9

& Æ Œ œ " . , : ! ? 0123456789

no p q r s t u v G H I J K L M N O

Epreuve imprimante, rapport de réduction 1/1
Réduction photographique 1/18

& Æ Œ œ " . , : ! ? 0123456789

no p q r s t u v G H I J K L M N O

Epreuve imprimante, rapport de réduction 1/2
Réduction photographique 1/9

& Æ Œ œ " . , : ! ? 0123456789 *no p q r s t u v G H I J K L M N O*

Epreuve imprimante, rapport de réduction 1/2
Réduction photographique 1/18

A B C D E F G H I J K L
M N O P Q R S T U V
W X Y Z

a b c d e f g h i j k l m n o p
q r s t u v w x y z

Epreuve imprimante, rapport de réduction 1/1
Réduction photographique 1/9

A B C D E F G H I J K L M N O P Q R S T
U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

Epreuve imprimante, rapport de réduction 1/1
Réduction photographique 1/18

A B C D E F G H I J K L M N O P Q R S T U

V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

Epreuve imprimante, rapport de réduction 1/2
Réduction photographique 1/9

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

Epreuve imprimante, rapport de réduction 1/2
Réduction photographique 1/18

BASKERVILLE 169

E) MESURES DE TEMPS

Afin d'alléger l'exposé, seules les principales mesures seront exposées dans le détail. Les caractères retenus pour cet exposé sont les majuscules de la police "Univers Medium 689" de Monotype. Ils permettent d'obtenir aisément les principaux résultats.

PRELIMINAIRES

Les mesures de temps ont été effectuées grâce à une horloge synchronisée sur le secteur ; de ce fait, la résolution est de 0.02 secondes.

Les temps à mesurer étant de l'ordre de quelques milli-secondes, nous avons procédé de la manière suivante :

- chaque module a été découpé en autant de tronçons que nécessaire.
- chaque tronçon a été inclu dans une boucle.

Ainsi, pour mesurer le temps d'exécution du tronçon X, on l'exécute N fois et on divise le temps obtenu par N.

Le temps d'exécution de la boucle est donné par l'exécution d'un tronçon de longueur nulle.

Les mesures ci-jointes ont été effectuées de cette manière.

TEST UNIVERS MEDIUM - Majuscules

LETTRES	Nombre de droïtes	Nombre de courbes	Largeur de la lettre	Encombrement Fichier en octets	Temps d'exécution (en millisecondes)								
					Etape 1	Etape 2	Etape 3				Etape intermédiaire		
							Réduction				Réduction		
							1/1	1/2	1/4	1/8	1/2	1/4	1/8
					±0,005	± 0,01	± 0,2	± 0,2	± 0,2	± 0,2	± 0,02	± 0,04	±0,04
A	11	0	892	62	1.09	1.34	292.2	147.2	74.6	38.4	2.22	2.20	2.20
B	9	8	656	138	2.29	40.78	344.8	173.6	88.6	46.6	8.92	8.95	8.95
C	2	8	756	92	1.55	41.98	363.5	182.6	92.2	47.2	7.50	7.50	7.55
D	6	4	746	76	1.31	22.40	292.0	147.0	74.0	38.2	4.82	4.85	4.85
E	12	0	557	69	1.12	1.32	245.1	124.0	62.8	33.0	2.34	2.35	2.40
F	10	0	558	56	0.95	1.13	185.2	92.8	47.2	24.8	1.98	1.95	2.00
G	8	7	786	109	1.78	37.11	399.8	201.4	102.4	52.6	7.66	7.70	7.70
H	12	0	703	59	1.06	1.27	173.7	87.4	44.8	22.8	2.22	2.20	2.25
I	4	0	135	27	0.52	0.61	33.5	16.7	9.2	4.6	0.96	0.95	0.95
J	4	4	583	59	1.06	21.02	181.9	91.6	46.2	24.0	4.34	4.35	4.35
K	11	0	671	60	1.02	1.22	294.4	117.4	59.4	31.0	2.12	2.15	2.10
L	6	0	577	35	0.66	0.77	142.1	71.6	36.2	18.6	1.26	1.30	1.30
M	13	0	978	63	1.12	1.35	295.5	148.2	74.8	38.2	2.36	2.35	2.40
N	10	0	717	49	0.92	1.11	196.5	98.8	50.2	25.6	1.90	1.85	1.90
O	4	8	720	90	11.56	42.71	388.8	193.8	97.6	49.6	7.58	7.60	7.65
P	8	4	625	86	1.49	21.52	225.8	113.8	58.0	30.4	5.28	5.30	5.30
Q	7	8	880	119	1.99	43.15	423.8	212.8	107.6	54.8	8.60	8.62	8.60
R	11	6	653	123	2.06	30.28	266.4	134.4	68.4	35.8	7.38	7.40	7.40
S	4	12	686	149	2.45	61.95	430.8	216.6	109.6	57.2	11.28	11.30	11.30
T	8	0	721	43	0.79	0.94	177.8	84.4	45.2	23.2	1.58	1.55	1.60
U	6	4	734	67	1.19	21.52	231.6	116.2	58.6	30.2	4.66	4.65	4.70
V	2	0	784	43	0.79	0.94	226.8	113.8	57.6	29.4	1.58	1.60	1.60
W	13	0	1224	63	1.12	1.35	361.6	131.4	91.8	46.6	2.36	2.35	2.35
X	12	0	819	69	1.12	1.32	320.2	161.4	81.6	42.4	2.36	2.35	2.35
Y	9	0	818	47	0.86	1.02	226.6	113.8	57.8	29.4	1.72	1.75	1.70
Z	10	0	716	59	0.99	1.17	330.4	166.0	84.2	43.4	2.04	2.05	2.05
Tot.	218	73	18695	1912	42.86	401.28	6992.8	3513.7	1780.6	918.0	107.02	107.17	107.40
Moy.	8.4	2.8	719	73.54	1.65	15.43	269.9	135.1	68.5	35.3	4.12	4.12	4.13
Remar que	Fichier : 1966 octets (1912+tables d'entrées)						Le temps d'exécution est indépendant du facteur de réduction						

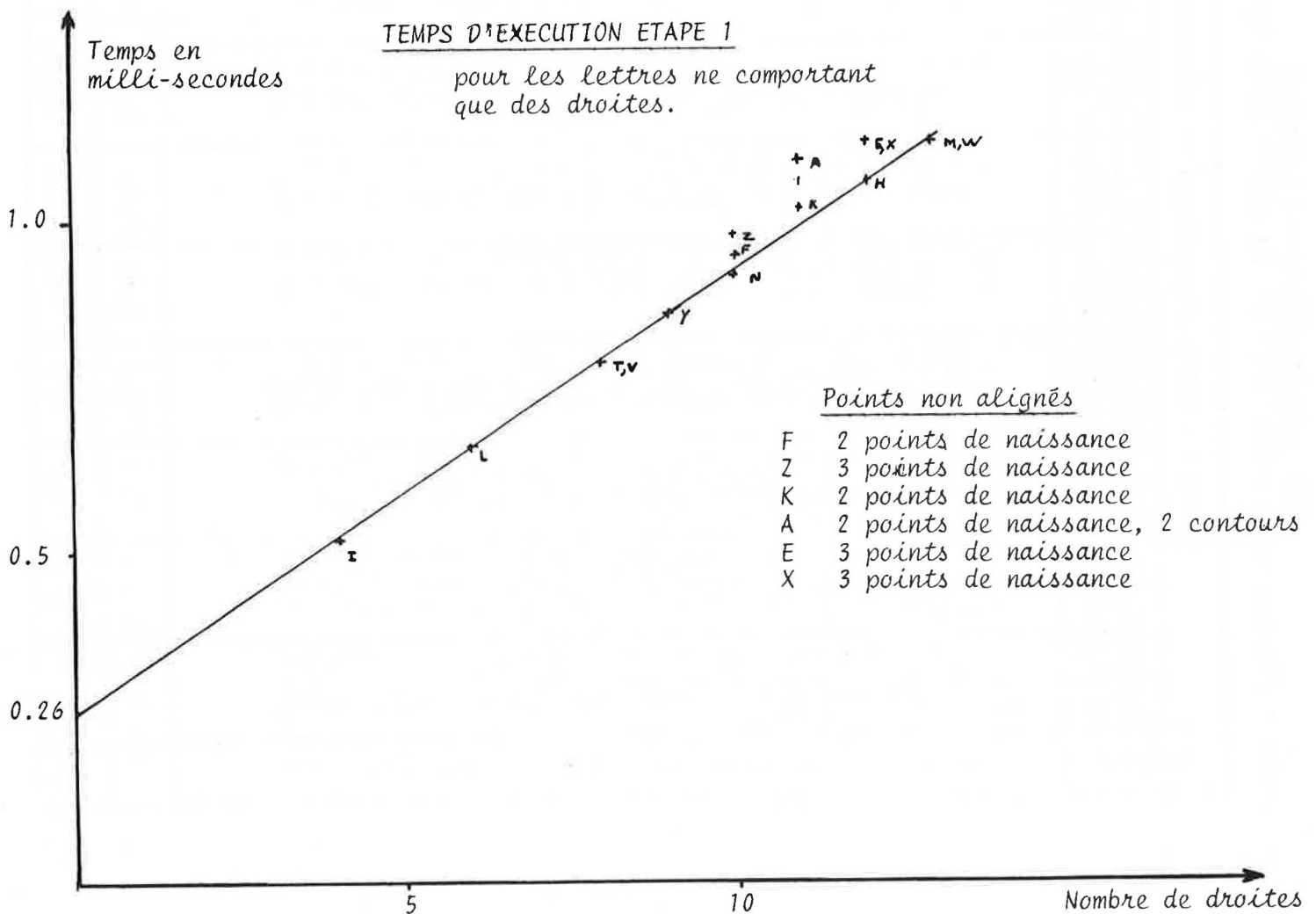
TEST UNIVERS MEDIUM - MAJUSCULES
Répartition des temps entre le calcul de points et le tri lors de l'étape 3

LÉTTRES	ETAPE 3 - TEMPS D'EXECUTION EN MILLI-SECONDES									
	Réduction 1/1		Réduction 1/2		Réduction 1/4		Réduction 1/8		Réduction 1/16	
	Gen.	Tri	Gen.	Tri	Gen.	Tri	Gen.	Tri	Gen.	Tri
	± 0,2	± 0,4	± 0,2	± 0,4	± 0,2	± 0,4	± 0,2	± 0,4	± 0,2	± 0,4
A	190,0	102,2	96,0	51,2	49,0	25,6	25,6	12,8	13,8	6,4
B	225,2	119,6	113,8	59,8	58,6	30,0	31,6	15,0	18,6	7,6
C	260,8	102,7	131,2	51,4	66,6	25,6	34,4	12,8	18,4	6,4
D	200,6	91,4	100,6	46,4	50,8	23,2	26,6	11,6	14,4	5,8
E	138,4	106,7	70,4	53,6	36,2	26,6	19,4	13,6	10,8	6,8
F	111,8	73,4	56,2	36,6	29,0	18,2	15,4	9,4	8,6	2,4
G	274,4	125,4	139,2	62,2	71,4	31,0	37,0	15,6	20,2	7,8
H	102,4	71,3	51,8	35,6	27,0	17,8	13,8	9,0	7,8	4,4
I	20,0	13,5	10,2	6,5	5,6	3,6	3,0	1,6	2,0	6,6
J	122,8	59,1	62,0	29,6	31,4	14,8	16,4	7,6	9,4	3,6
K	151,8	82,6	76,2	41,2	38,8	20,6	20,4	10,6	11,2	5,0
L	83,6	58,5	42,4	29,2	21,4	14,8	11,2	7,4	6,0	3,8
M	196,2	99,3	98,4	49,8	50,0	24,8	25,6	12,6	13,2	6,4
N	123,3	72,7	62,6	36,2	32,0	18,2	16,6	9,0	9,0	4,6
O	285,8	103,0	142,2	51,6	71,6	26,0	36,6	13,0	19,8	6,4
P	149,6	76,2	75,8	38,0	39,0	19,0	20,8	9,6	12,2	4,6
Q	304,8	119,0	153,2	59,6	77,8	29,8	40,0	14,8	22,0	7,4
R	180,2	86,2	91,2	43,2	46,8	21,6	25,0	10,8	15,0	5,4
S	293,8	137,0	148,2	68,4	75,6	34,0	40,0	17,2	22,8	8,6
T	104,4	73,4	52,8	36,6	27,0	18,2	14,0	9,2	7,6	4,6
U	157,4	74,2	79,2	37,0	40,0	18,4	20,8	9,4	11,4	4,8
V	147,4	79,4	74,2	39,6	37,8	19,8	19,4	10,0	10,2	5,2
W	237,4	124,2	119,4	62,0	60,6	31,2	31,2	15,4	16,6	7,8
X	207,4	112,8	105,0	56,4	53,6	28,0	28,2	14,2	15,2	7,2
Y	143,3	82,8	72,4	41,4	37,2	20,6	19,0	10,4	10,2	5,2
Z	199,0	131,4	100,6	65,4	51,4	32,8	26,8	16,6	14,6	8,2
Tot.	4612,8	2378,0	2325,2	1188,4	1186,4	594,2	618,8	299,2	341,0	147,0
Moy.	177,4	91,5	89,4	45,7	45,6	22,9	23,8	11,5	13,1	5,6

LOIS D'EVOLUTION DES TEMPS

1°) Première étape du décodage

Intéressons nous aux caractères les plus simples ; c'est-à-dire à ceux ne comportant que des droites. Le graphe ci-dessous montre l'évolution du temps d'exécution en fonction du nombre de droites.



Nous nous apercevons que les paramètres des lettres ne comportant qu'un seul contour et ne possédant qu'un seul point de naissance sont sur une même droite.

La loi suivie pour ces caractères est :

$$T = 0.260 + 0.067 \cdot ND$$

T temps total en milli-secondes

ND ... nombre de droites

Notons que le facteur de corrélation des données est 0.999989

Les "lettres" s'écartant de la droite comportent plusieurs points de naissance, de plus, le 'A' possède 2 contours.

Ceci nous permet d'établir la formule suivante :

$$T = 0.260 + 0.067ND + 0.145 \cdot NC + 0.03 \cdot PST + 0.04 \cdot NCT$$

T.... temps total en milli-secondes

ND... nombre de droites

NC... nombre de courbes

PST.. nombre de points de naissance moins 1

NCT.. nombre de contours moins un.

L'examen d'une police pour laquelle de nombreuses primitives ont été définies, nous permet d'établir la formule globale suivante :

$$T = TC + TP$$

T TEMPS TOTAL EN MILLI-SECONDES

$$TC = 0.26 + 0.07 \cdot ND + 0.14 \cdot NC + 0.03 \cdot PST + 0.04 \cdot NCT$$

$$TP = 0.14 \cdot NP + 0.07 \cdot ND^2 + 0.15 \cdot NC^2 + (ND^3 + 2.17 \cdot NC^3) \cdot 0.01$$

ND nombre de droites directement définies

NC nombre de courbes directement définies

PST.... nombre de points de naissance moins un

NCT ... nombre de contours moins un

NP nombre des primitives

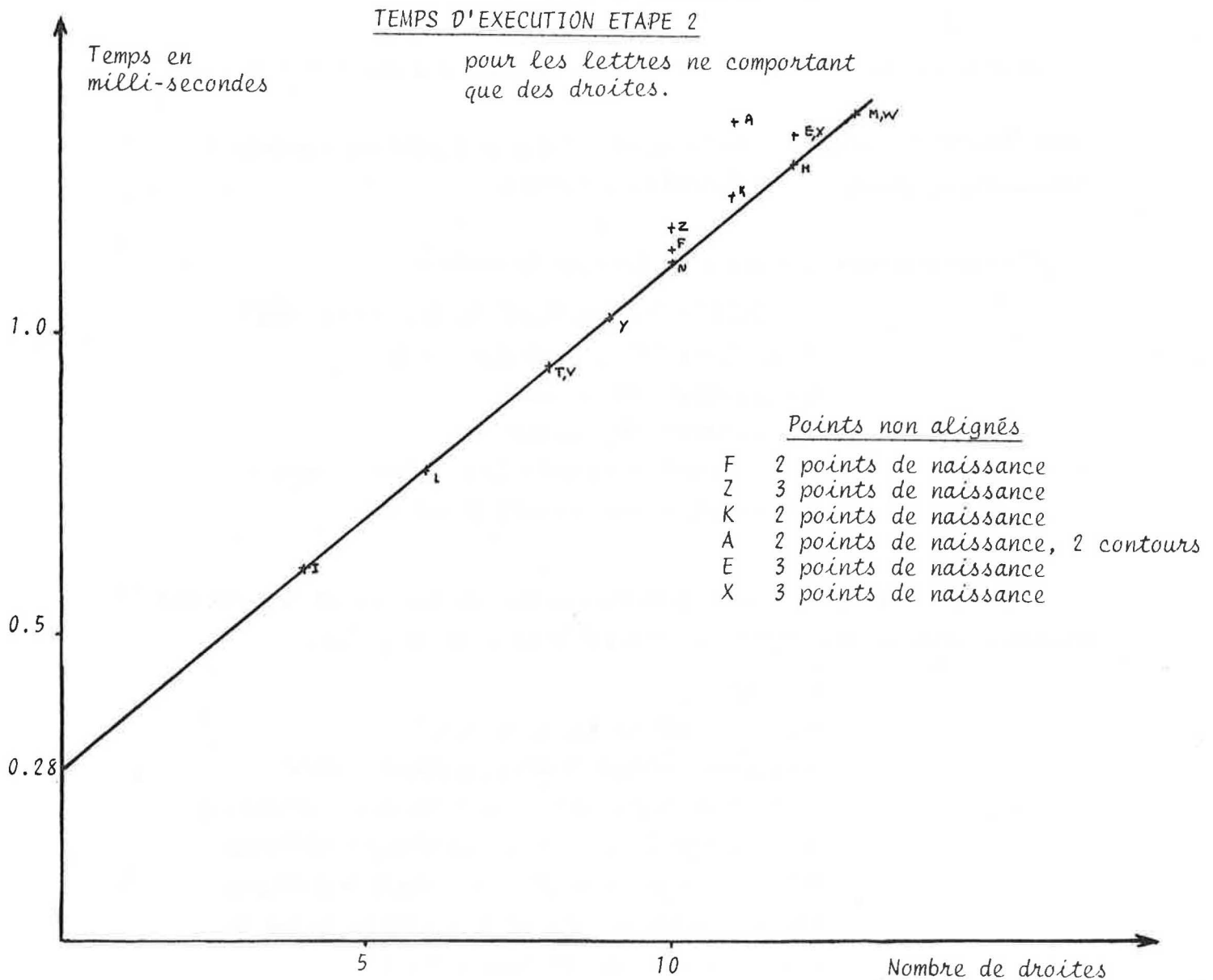
ND2 ... nombre de droites définies par l'ensemble des primitives

ND3 ... nombre de droites définies dans l'ensemble des primitives faisant l'objet d'une inversion de fichiers.

NC3 ... même chose que précédemment pour les courbes

2°) Deuxième étape du décodage

Opérons comme précédemment. Le graphe ci-dessous montre l'évolution du temps de traitement pour les caractères ne comportant que des droites.



Nous constatons que les points correspondant aux lettres ne possédant qu'un seul point de naissance et un seul contour sont situés sur une droite. L'équation de cette dernière est :

$$T = 0.28 + 0.08 \cdot ND$$

T temps total en milli-secondes

ND ... nombre de droites

Le facteur de corrélation est 0.99997.

L'examen des lettres restantes permet d'établir la formule globale :

$$T=0.28+0.08*ND+5.04*NC+0.03*PST+0.14*NCT$$

T.... Temps total en milli-secondes

ND.... nombre de droites

NC ... nombre de courbes

PST .. nombre de points de naissance moins un

NCT .. nombre de contour moins un

3°) Troisième étape de décodage

L'établissement de la loi d'évolution du temps est ici relativement complexe ; ceci est dû à plusieurs facteurs :

- du nombre moyen de processeurs de suivie de contour mis en jeu par lettre.
- du nombre moyen de points situés sur les courbes et les droites d'une lettre donnée
- du nombre moyen de données à trier.
- de la largeur de la lettre.

De plus, les variations par rapport à ces paramètres ne sont pas linéaires, ainsi, le temps de tri suit une loi du type :

$$T=K1*NC1+K2*NC2+K3*NC2*NC2+...$$

T temps total de tri

NC1... nombre de colonnes ne comportant qu'un seul segment

NC2... nombre de colonnes comportant deux segments

K1, K2, K3 ... constantes multiplicatives

A titre d'information, les valeurs approximatives des constantes sont :

K1=0.1013 coefficient de corrélation 0.987

K2=0.0410 coefficient de corrélation 0.901

K3=0.0024 coefficient de corrélation 0.753

Les lois que l'on peut établir ne sont guère représentatives du système car trop liées aux choix spécifiques effectués dans les algorithmes. Seules les moyennes globales ont une signification, car elles effacent le "bruit de fond" dû aux choix particuliers. De ce fait, les courbes tracées à partir des moyennes présentent un facteur de corrélation très bon.

Le temps général d'exécution est régi par :

$$T=K1*NCOL$$

T..... Temps total en milli-secondes

NCOL..... Nombre moyen de colonnes pour la police considérée

Le temps de tri est régi par la loi :

$$T=K2*NCOL$$

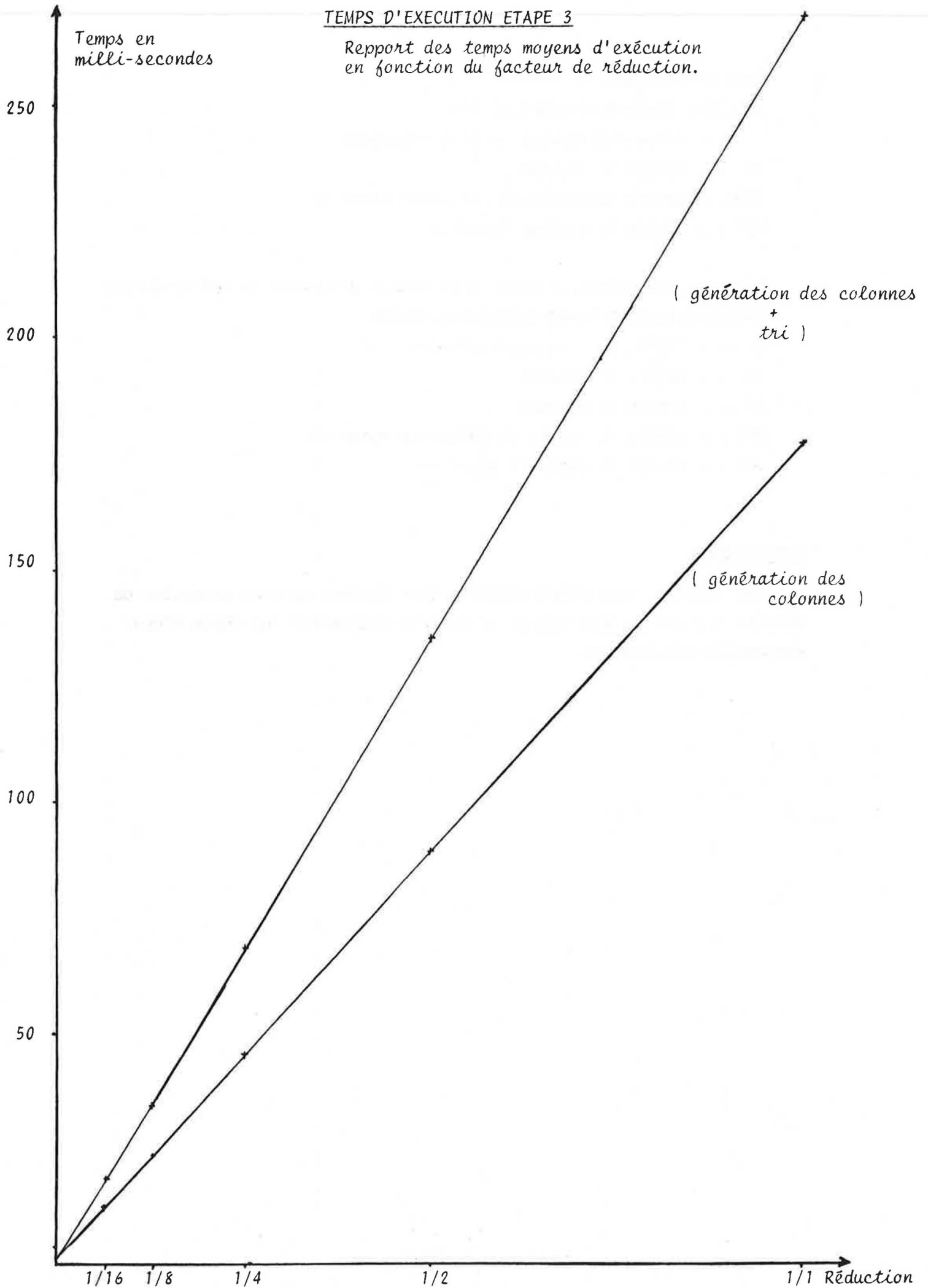
4°) Etape intermédiaire

L'examen des résultats préliminaires montre que le temps ne dépend pas du rapport de réduction.

Ainsi que nous l'avons fait pour les deux premières étapes du décodage traçons l'évolution du temps pour les lettres ne comportant que des droites.

TEMPS D'EXECUTION ETAPE 3

Report des temps moyens d'exécution
en fonction du facteur de réduction.



Nous en déduisons :

$$T=0.31+0.16*ND+0.04*PST+0.05*NCT$$

T..... temps d'exécution en milli-secondes

ND..... nombre de droites

PST.... nombre de points de naissance moins un

NCT ... nombre de contour moins un

L'examen des autres lettres, nous permet d'obtenir la loi globale :

$$T=0.31+0.16*ND+0.86*NC+0.04*PST+0.05*NCT$$

T..... temps total en mill-secondes

ND..... nombre de droites

NC..... nombre de courbes

PST.... nombre de points de naissance moins un

NCT ... nombre de contours moins un

CONCLUSIONS

Les lois que nous avons établies vont maintenant nous permettre de dégager les points délicats et de ce fait d'orienter les choix d'une éventuelle réalisation.

VERS UNE REALISATION

Les outils dont nous disposons sont :

- le simulateur, et les résultats obtenus grâce à ce dernier
- un cross-assembleur du micro-processeur Z80 de ZILOG
- un simulateur de ce même micro-processeur
- les notices de circuits spéciaux

Nous allons examiner les différentes étapes du décodage afin de déterminer les performances de diverses solutions.

PREMIERE ETAPE DE DECODAGE :

La loi d'évolution du temps met en relief les principaux paramètres :

- temps d'initialisation
- temps de transfert d'une droite
- temps de transfert d'une courbe
- temps d'initialisation d'une primitive

Les algorithmes responsables de ces temps ont fait l'objet d'une simulation (μP : Z80). Pour une horloge simulée de 2.5 MHz, nous avons obtenu les résultats suivants (variations en pour-cent du temps obtenu par rapport à celui du PDP11/40) :

- initialisation..... + 30 %
- traitement d'une droite..... - 10 %
- traitement d'une courbe..... - 15 %
- initialisation d'une primitive..... + 30 %
- traitement d'une droite issue d'une primitive..... + 20 %
- traitement d'une courbe issue d'une primitive..... + 25 %
- traitement d'une inversion de primitive - 10 %

*ou assembleur croisé : Assembleur d'une machine exécuté sur une autre machine.

Nous constatons peu de variations de performances ceci est dû au fait que le traitement se situe au niveau de l'octet, ce qui avantage le micro-processeur. Notons qu'il existe une version plus rapide du Z80, le Z80A travaillant avec une horloge de 4 MHz. Cette version permettrait d'obtenir un temps moyen de traitement par caractère, pour une police du type univers, inférieur à la milli-seconde.

DEUXIEME ETAPE DU DECODAGE :

L'ensemble de base étudié est l'assemblage d'un Z80 et du circuit de calcul d'AMD : le 9511. La simulation de cet ensemble étant relativement complexe, les variations de temps ont été estimées à l'exception de la phase d'initialisation et du traitement de droites.

Les résultats sont :

- initialisation + 30 %
- traitement d'une droite..... + 20 %
- traitement d'une courbe.....+ 100 %

Le dernier résultat montre qu'il est nécessaire d'effectuer le traitement de courbes en parallèle, l'utilisation d'un ensemble composé de quatre Z80 et de quatre 9511 permettrait une amélioration du temps de traitement de l'ordre de 30 %. Alors la loi serait :

$$T=0.35+0.10*ND+3.5*NC+...$$

ce qui donne un temps moyen de traitement par caractère d'environ 12 milli-secondes. Ce temps est excessif, nous pouvons le réduire par deux voies :

- utilisation de micro-processeurs 16 bits type 8086, Z8000
- utilisation d'une ou plusieurs unités spécialisées de calcul

A elle seule l'utilisation d'un processeur 16 bits permettrait des temps comparables à celui du PDP11/40. L'assemblage de quatre processeurs réaliserait un gain de temps pour le traitement de courbes de 60 %. Nous aurions alors :

$$T=0.3+0.1*ND+2.*NC+...$$

Le temps moyen de traitement d'un caractère serait de 8 milli-secondes.

L'utilisation de circuits de calculs rapides pour réaliser la multiplication (assemblage de circuits type MPY-24A5 de TRW) permettrait de réduire le temps de calcul d'une courbe de :

- 50 % avec un Z80 associé à un 9511
- environ 75 % avec un microprocesseur 16 bits

TROISIEME ETAPE DE DECODAGE :

L'étude faite au paragraphe précédent montre qu'il est souhaitable de dissocier la phase de génération des données de la phase de tri. Par suite, nous allons traiter les deux problèmes séparément.

A) Phase de génération des données :

Le gain obtenu par rapport au temps global sur PDP 11/40 en utilisant comme processeurs de suivi de contour des LSI 11-II de DEC est de : - 60 %. Le temps de traitement de 1400 colonnes est de 100 milli-secondes.

Ce temps peut être amélioré en câblant les boucles de calcul (un processeur de suivi de contour contient deux séquences de boucles). La réalisation câblée pose peu de problèmes étant donnée la simplicité des algorithmes. Le gain de temps par processeur est estimé à plus de 60 % (60 % à 85 % suivant les données). Le temps moyen de traitement de 1400 colonnes chute alors à moins de 40 milli-secondes.

B) Phase de tri

L'algorithme réalisant le tri est simple. De ce fait, la boucle mineure de contrôle peut être aisément câblée. Le gain de temps estimé est de 60 %. Ce qui nous donne un temps moyen de traitement de moins de 40 milli-secondes.

ETAPE INTERMEDIAIRE

Le temps moyen de traitement obtenu sur PDP 11/40, étant de 4 milli-secondes, il suffit d'employer pour la réalisation pratique un micro-processeur 16 bits de la dernière génération, ce qui permettrait d'obtenir un temps moyen similaire.

CONCLUSION

L'ensemble constitué par :

- Pour la première étape d'un Z80 A.
- Pour la seconde étape d'un Z8000 ou 8086 associé à un 9511 et un circuit de multiplication rapide.
- Pour la troisième étape de 8 LSI 11-II associé à des circuits câblés d'un Z8000 ou 8086 associé à un circuit câblé d'un micro-processeur de supervision type Z80A
- pour l'étape intermédiaire d'un Z8000 ou 8086

....

- et un micro processeur permettant une supervision de l'ensemble. permettrait d'engendrer 200 caractères par seconde ; caractères définis dans des matrices 200*200. La vitesse minimale (génération de matrices 2048*2048) serait de 24 caractères par seconde .

Les données que nous avons fournies représentent les performances minimales que l'on peut espérer ; l'optimisation des algorithmes peut faire reculer ces limites. Afin de chiffrer les gains possibles, certains algorithmes ont fait l'objet d'optimisation. Les gains obtenus sont les suivants :

- étape 3, tracé de droite - 60 %
durée du travail d'optimisation : 3 jours
- étape 3, tracé de cercle - 60 %
durée du travail : 3 jours
- étape 2, extraction de racine carrée - 50 %
durée du travail : 1 jour
- étape 2, intersection de droites - 15 %
durée du travail : 1 jour

Ceci montre que pour une telle réalisation, les performances possibles ne sont pas figées ; on peut raisonnablement espérer entre la machine brute et celle supportant un code optimisé, une amélioration de plus de 50 %.

CONCLUSION

CONCLUSION

Le but initial du travail était de définir un organe capable d'engendrer des caractères à haute définition, à la vitesse de mille signes par seconde. Ce but n'est que partiellement atteint dans la mesure où à cette vitesse la réalisation proposée ne permet pas une définition supérieure aux générateurs actuels (les performances sont alors équivalentes). Par contre, elle permet, effectivement, de créer à faible vitesse des caractères à haute définition. De plus, à complexité et coût équivalents, elle supprime le problème de l'encombrement mémoire des générateurs existants.

Une autre voie d'étude est la réalisation d'un générateur monoprocesseur (donc lent) capable de concurrencer en coût et qualité les photocomposeuses "photographiques".

Etant donné les intérêts industriels mis en jeu, il n'a pas été possible de décrire les algorithmes suivants :

- découpage d'une courbe évoluée en arcs de cercle,
- générateur point par point d'arcs de cercle,
- préciser la "structure d'arbre".

Je le regrette et prie le lecteur de m'en excuser.

Sur le plan théorique cette thèse montre qu'il est possible de créer, en une seule passe et en temps réel, à partir de la description de contour(s) et de leur "structure d'arbre", les taches correspondantes sans avoir recours à des mémoires considérables. Le calcul de cette "structure d'arbre" par l'éditeur est, actuellement, un goulot d'étranglement vis-à-vis d'une utilisation interactive du système global, éditeur et générateur.

L'expérience acquise permet d'envisager la restructuration de l'éditeur et par suite de minimiser le goulot en question.

BIBLIOGRAPHIE

BIBLIOGRAPHIE

PRINCIPE DE LA TYPOGRAPHIE

LIVRE DE VULGARISATION (on trouvera en particulier une bibliographie plus étendue).

- [1] PEIGNOT Jérôme
"De l'écriture à la typographie"
Paris, Gallimard, Collection Idées - 1967.

PETIT DICTIONNAIRE DES TERMES COURANTS

- [2] AVIS F.C.
"Type face terminology"
London, Glenwien Press - 1965

CATALOGUE DE POLICES

- [3] Specimen book of Monotype printing types (vol I et II)

ARTICLES DE RECHERCHE SUR LA STRUCTURE DES CARACTERES

- [4] COUEIGNOUX Philippe
"Generation of Roman Printed Fonts"
Massachusetts Institute of Technology, Department of Electrical Engineering,
Ph.D Thesis 1975
- [5] COX C.H., III, COUEIGNOUX, B.A. BLESSER, M. EDEN
"Skeletons : a link between theoretical and physical letters descriptions"
Department of Electrical Engineering, M.I.T.
- [6] MERGLER H.W., VARGO P.M.
"An approach to computer assisted letter design"
The journal of Typographic Research - Vol 3 - Number 4 - 1968
- [7] SHILLMAN P.J.
"Character recognition based on phenomenological attributes : theory and Methods"
P.h. D. Thesis, Department of electrical engineering, M.I.T. - 1974

BIBLIOGRAPHIE

LA PHOTOCOMPOSITION PROGRAMMEE

LE POINT DE VUE D'UN ARTISTE

- [1] FRUTIGER A.
"The IBM selectric composer-The evolution of composition technology"
I.B.M. Journal - Janvier 1968

ARTICLE DE SYNTHESE

- [2] LARUE José
"Composition automatique des textes"
Technique de l'ingénieur - Informatique N° H21 - Septembre 1975

COURS EXPOSANT L'ETAT ACTUEL DES TECHNIQUES

- [3] BACCHUS P. - ERABAND P. - FRETIN M. - LHEUREUSE B. - PROFIZI J.C.
"La composition programmée des textes"
Séminaire I.R.I.A. - Avril 1977

BIBLIOGRAPHIE

MICRO- INFORMATIQUE

- [1] DANIEL R. - GLYNN M.
"Microprocessors : Technology, architecture and applications"
(Disponible aux éditions Lavoisier) 1976
WILEY - Interscience publication
- [2] SAINT.PATRICE Alain
"Etude d'un système de développements pour applications multiprocesseurs"
Rapport de D.E.A. ; Ecole Nationale Supérieure des Mines de Saint-Etienne
1978

BIBLIOGRAPHIE

LE GENERATEUR DE CARACTERES

OUVRAGE DE REFERENCE

- [1] MORVAN Pierre, LUCAS Michel
"Image et ordinateur - Introduction à l'infographie interactive"
Librairie Larousse 1976

TECHNIQUES DE CODAGE

- [2] MURAT KUNT
"Comparaison de techniques d'encodage pour la réduction de redondance d'images"
Thèse 183, Département d'électricité
Ecole Polytechnique de Lausanne - 1974
- [3] COUEIGNOUX Ph.
"Comparaison of type face by contour encoding"
S.M. Thesis Department of electrical engineering, M.I.T. - 1973
- [4] COUEIGNOUX Ph.

Ph.D Thesis, Department of electrical engineering, M.I.T. - 1975
Chapitre 4
- [5] WURDLE Caroline
"Artspeak : A graphics language for Artists"
ACM Sigplan notices - Vol II, N° 6, Juin 1976
Pages 32 à 38

ARTICLES TRAITANT DES METHODES DE GENERATIONS D'ARCS DE CERCLE

- [6] WON L. CHUNG
"Notes on circle generation algorithms"
Computer Graphics and Image Processing - N° 6 - 1977
Pages 196 à 198
- [7] DENERT Ernst
"A method for computing point of a cercle using only integers"
Computer Graphics and Image Processing - N° 2 1973 - Pages 83 à 91
- [8] JORDAN Bernard W., LENON William J., HOLM Barry D.
"An improved algorithm for generation of non parametric curves"
I.E.E.E. - Transactions on Computers - Vol C22 - Numéro 12 - 1973
- [9] HORN Berthol K.P.
"Circles generators for display devices"
Computer Graphics and Image Processing N° 5 - 1976 - Pages 280 à 288

SUR LE DECODAGE DE COURBES EVOLUEES

- [10] MERGLER H.W. and VARGO P.M.
"On approach to computer assisted letter design"
The journal of Typographic Research, Vol II, N° 4, Octobre 1968
Pages 299 à 322

- [11] COUEIGNOUX Ph.
Ph. D Thesis - M.I.T.
Chapitre 3

- [12] COUEIGNOUX Ph.
Ph.D. Thesis - M.I.T.
Appendice II

BIBLIOGRAPHIE

EDITION DE CARACTERES

RUDNIK Antoine

"Réalisation d'une base de données sur mini-ordinateur"

Rapport de 3ème année - Ecole Nationale Supérieure des Mines de Saint-Etienne - 1978

LUCAS Michel

"La programmation des consoles de visualisation"

Université Scientifique et Médicale de Grenoble
Laboratoire d'Informatique

AUTORISATION DE SOUTENANCE

VU les dispositions de l'article 3 de l'arrêté du 16 Avril 1974,

VU les rapports de présentation de Messieurs :

- Philippe COUEIGNOUX
- Michel LUCAS

Monsieur Marc HOURDEQUIN

est autorisé à présenter une thèse en soutenance pour l'obtention du diplôme de DOCTEUR-INGENIEUR, spécialité "Systèmes et Réseaux Informatiques".

Grenoble, le 17 Octobre 1978

le Directeur de l'ENSM

le Président de l'INPG



Ph. TRAYNARD
Président
de l'Institut National Polytechnique

