

Cahiers **GUT**enberg

☞ LES CONCEPTS GRAPHIQUES DE POSTSCRIPT

☞ Bruno BORGHI

Cahiers GUTenberg, n° 1 (1989), p. 45-51.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_1989__1_45_0>

© Association GUTenberg, 1989, tous droits réservés.

L'accès aux articles des *Cahiers GUTenberg*

(<http://cahiers.gutenberg.eu.org/>),

implique l'accord avec les conditions générales

d'utilisation (<http://cahiers.gutenberg.eu.org/legal.html>).

Toute utilisation commerciale ou impression systématique
est constitutive d'une infraction pénale. Toute copie ou impression
de ce fichier doit contenir la présente mention de copyright.

Les concepts graphiques de PostScript

Bruno BORGHI

METASOFT — 13, rue Duhamel - 35000 RENNES (France) — Tél : 99 31 25 88

Résumé

Cet article est consacré à une analyse des concepts graphiques mis en œuvre par le langage de description de pages PostScript. Les principes fondamentaux du modèle graphique de PostScript sont ainsi dégagés tout à fait indépendamment des aspects proprement langage (syntaxe, opérateurs généraux).

1. Introduction : Terminologie

L'auteur d'un article en français traitant d'infographie se trouve toujours confronté à l'éternel dilemme : comment trouver un équilibre entre l'emploi de termes anglais (ce qui est interdit par nos lois), et la recherche systématique d'un mot français ? J'ai eu tendance, pour ma part, à privilégier les termes français.

Afin de lever les ambiguïtés, lorsqu'un terme français est employé pour la première fois, il est suivi du terme anglais correspondant tel qu'on le trouve dans le manuel PostScript. Les principaux choix terminologiques sont les suivants :

- *caractère* est utilisé dans le sens de *signe typographique, glyphe* ;
- *résolution* est employé, improprement, en traduction de *resolution*. Résolution s'applique normalement à un appareil de mesure comme un microscope alors que le terme correct est *définition* pour les dispositifs de sortie comme les moniteurs vidéo. L'usage en infographie, auquel je me

suis conformé, veut cependant qu'on emploie *résolution* ;

- *appareil* ou *dispositif* sont utilisés en traduction de *device* ;
- *motif* est utilisé plutôt que *pattern* ;
- *trame* ou *image discrète* traduisent *bitmap*, quand c'est possible ;
- *découpe* est utilisé plutôt que *fenêtrage* pour parler du *clipping* ; fenêtrage a été un temps le terme consacré, mais depuis l'émergence des systèmes multi-fenêtres, ce terme est ambigu. Découpe est une traduction pratiquement littérale de *clipping* ;
- *tracé* est utilisé là où on utilise souvent improprement *raster*.

2. Les composants graphiques élémentaires

Les systèmes graphiques usuels décrivent un objet puis le placent sur la page (modèle d'instantiation). Dans le langage PostScript, il n'y a pas d'objets graphiques à proprement parler : il n'y a que des procédures. Les composants graphiques d'une page sont construits incrémentalement par l'exécution d'opérateurs appropriés, et aucune manipulation n'est possible sur les composants eux-mêmes (modèle procédural). Ce modèle simplifie les choses énormément du point de vue de la réalisation de l'interpréteur, car on a un système sans mémoire : un opérateur graphique quand il s'exécute a accès à toutes

les informations dont il a besoin au moment de l'exécution même (pas d'évaluation retardée); ces informations se trouvent dans une structure appelée *état graphique* (*Graphic State*).

Les composants graphiques sont au nombre de trois : les dessins, les images et le texte.

Les dessins

On construit un dessin en construisant un parcours (*path*). Un parcours est un ensemble de polygones orientés. Chaque polygone est formée de segments de droite (*lineto*) et de courbes de Bézier du 3ème degré (*curveto*); une polygone peut être ouverte, ou se refermer sur son point d'origine (*closepath*). Un parcours peut contenir plusieurs polygones (*moveto*). Le parcours le plus général n'est donc pas connexe, les polygones qu'il définit ne sont pas convexes, deux segments ou courbes quelconques du parcours peuvent avoir une intersection non vide.

Une fois un parcours défini, on peut le tracer sous forme de ligne épaisse (*stroke*) ou en colorier l'intérieur (*fill*).

Les images

PostScript permet de placer sur la page n'importe quelle image rectangulaire échantillonnée en niveaux de gris, ayant éventuellement subi une transformation affine quelconque.

Il existe une forme d'image particulière, appelée masque : c'est une image binaire, dont les pixels à 1 sont tracés dans la couleur courante.

Le texte

Le composant élémentaire du texte est le caractère. Du point de vue graphique, un

caractère est défini par les opérateurs standard de PostScript : ce peut être un dessin ou une image (c'est en fait une procédure). Les caractères natifs des implémentations d'Adobe sont, en première approximation, des contours remplis. On change de corps par homothétie sur ces contours¹.

Les caractères sont organisés en polices². Une police contient un petit nombre de données et un pointeur vers les descriptions des caractères. Les descriptions de caractères sont donc partagées par plusieurs polices. Parmi les données propres à chaque police d'une même fonte, la plus importante d'un point de vue graphique est la matrice de transformation de la police. Cette matrice détermine la taille des caractères au moment du tracé.

Le gros problème du tracé de caractères apparaît quand l'appareil de sortie est à basse résolution et que le corps du caractère est petit. Les contours fournissent une description continue du caractère. La simple projection de cette description sur la trame discrète ne permet pas un rendu correct. La présence ou l'absence d'un pixel dans ces conditions modifie très nettement l'aspect visuel du caractère sur le papier ou l'écran. Un traitement purement géométrique ne permet pas d'éviter les phénomènes de crénelage (*aliasing*), et d'inhomogénéité dans les jambages et les graisses. Pour lutter contre ça, Adobe a

¹C'est véritablement une homothétie quand la police n'est pas "magique", sinon c'est presque une homothétie. En toute rigueur typographique, le dessin d'un caractère n'est valable que pour une certaine gamme de corps. Cependant, comme en photocomposition optique, pour PostScript, pour changer de dessin, il faut changer de police.

²Suivant la terminologie proposée par Jacques André, j'appelle *fonte* l'ensemble des caractères pour tous les corps pour un style donné, et *police* le sous-ensemble d'une fonte correspondant à un seul corps. PostScript manipule uniquement des polices, pas des fontes. Il y a seulement des fontes virtuelles, grâce aux mécanismes qui permettent de transformer une police en n'importe quelle autre police de la fonte.

ajouté une poudre de perlimpinpin (*“font magic”*) : la description des caractères n'est pas purement contour, mais contient également des informations sur la “sémantique” du dessin³.

Par ailleurs, le tracé d'un caractère à partir de ses contours est un processus coûteux en CPU (quelques dixièmes de seconde). Un système de cache permet de conserver l'image discrète de chaque caractère quand elle est calculée pour la première fois. Cette image est ensuite utilisée chaque fois que possible pour éviter un nouveau calcul. L'image est simplement dupliquée : dès la deuxième fois, le tracé d'un caractère devient très peu coûteux. Une image dans le cache correspond à un caractère dans une police donnée, pour une matrice de transformation donnée.

3. Le modèle PostScript

Une page est construite en déposant **successivement** de l'encre aux endroits sélectionnés. L'encre peut-être noire, blanche, grise ou colorée.

Ce modèle porte le nom de **modèle du peintre**. Chaque coup de pinceau recouvre ce qu'une couche précédente aurait pu laisser.

Ce modèle est complété par la notion de découpe : l'encre n'est déposée sur la page qu'à l'intérieur du parcours de découpe courant (*clipping path*). La découpe est prise en compte par toutes les opérations de tracé : caractères, lignes épaisses, surfaces remplies, images ; c'est véritablement un pochoir posé sur la page.

Le parcours de découpe est un parcours ordinaire, construit par les opérateurs standard. Il est donc absolument

³D'autres compagnies, comme Bitstream ou URW, ont maintenant mis à leur catalogue des polices “magiques”.

quelconque et ne présente aucune bonne propriété.

4. Les systèmes de coordonnées

PostScript travaille dans un espace affine réel de dimension 2. Sur cet espace sont définis deux systèmes de coordonnées : le repère utilisateur (*user space*) et le repère appareil (*device space*). La relation entre ces deux systèmes est exprimée par une matrice de passage du repère utilisateur vers le repère appareil : la matrice de transformation courante (*Current Transformation Matrix, CTM*). C'est une matrice de changement de repère affine, donc 3 lignes par 3 colonnes, la dernière colonne valant (0, 0, 1). Cette matrice peut être quelconque pourvu qu'elle reste inversible ; elle n'est pas forcément orthogonale.

On peut considérer le repère appareil comme un repère fixe, et le repère utilisateur comme un repère variable. Les coordonnées et mesures contenues dans un fichier PostScript sont toutes exprimées dans le repère utilisateur. Les transformations affines (translation, rotation, affinité) modifient le repère utilisateur ; en multipliant la CTM par ces matrices de changement de repère, on obtient une nouvelle CTM qui est utilisée pour la suite des calculs.

Le tracé des caractères est une application de ce principe. La matrice de transformation de la police (*FontMatrix*) est composée avec la CTM pour former la matrice de passage du repère propre de la description des caractères vers le repère appareil. Les caractères des polices natives d'Adobe sont décrits dans un repère propre où ils sont de corps 1000. Le changement de corps est donc simplement

un changement de repère. Pour être magique, l'algorithme de tracé du caractère peut avoir accès à la CTM, et donc se comporter différemment selon les coefficients de cette matrice.

En interne, l'interpréteur PostScript travaille uniquement dans le repère appareil. C'est un repère orthonormé dépendant des caractéristiques physiques de l'appareil de visualisation; un pixel physique y est un carré de côté 1.

On obtient les coordonnées appareil d'un point en multipliant ses coordonnées utilisateur par la CTM.

5. La construction de parcours

Il n'y a à tout moment qu'un seul parcours, qui fait partie de l'état graphique. Le parcours est construit incrémentalement par les opérateurs *moveto*, *lineto*, *curveto*, *closepath*.

Les arcs de cercles sont réalisés par une suite de courbes de Bézier lors de l'exécution des opérateurs *arc*, *arcn*, *arcto*. Ils n'existent pas dans la structure interne du *path*⁴ (cf. opérateur *pathforall*).

Les opérateurs de construction de *path* transforment à la volée les coordonnées utilisateur en coordonnées appareil. Une modification de la CTM ne peut donc pas avoir d'effet rétroactif; tout changement de repère n'est pris en compte que pour les prochains segments.

Au moment du tracé, que ce soit en remplissage ou en ligne épaisse, le parcours

⁴En effet, la notion d'arc de cercle n'est définie que dans des espaces euclidiens, et pas dans les espaces affines. Par contre, les courbes de Bézier et les segments de droite sont des notions affines. Or, les transformations d'espace de PostScript sont affines et non euclidiennes, ce qui interdit l'usage généralisé des arcs de cercle.

est d'abord linéarisé : les courbes de Bézier sont approximées par une suite de segments de droite. L'approximation est optimale pour la résolution de l'appareil. En interne, le contour élémentaire sur lequel on fait des calculs est donc un polygone quelconque.

La linéarisation des courbes de Bézier est une opération purement géométrique. Cette linéarisation est obligatoire pour deux raisons :

1. on ne connaît pas d'algorithme simple de tracé pour remplir directement une zone délimitée par des courbes de Bézier du 3ème ordre⁵; la linéarisation est un processus assez simple par les techniques de subdivision;
2. pour les lignes épaisses, il faudrait calculer une représentation analytique de la courbe formant le contour de ligne épaisse; or cette forme analytique n'est pas polynomiale⁶; les calculs induits sont coûteux : on les évite par la linéarisation.

6. Le tracé

Les opérations de tracé utilisent les données de l'état graphique telles qu'elles sont au moment du tracé. Par exemple, si au cours de la construction du parcours on a changé la largeur de ligne, seule la dernière valeur est prise en compte, et ce pour tout le parcours.

Le remplissage

L'opérateur de tracé fondamental est *fill*, l'opérateur de remplissage. Pour remplir

⁵Bien qu'on en connaisse pour des courbes paraboliques.

⁶Polynôme divisé par la racine carrée d'un autre polynôme.

l'intérieur d'un parcours, il faut se fixer une règle pour discriminer l'intérieur et l'extérieur. La règle standard de PostScript, et dans les faits **la plus employée**, est la règle de la tangente tournante (*Non Zero Winding Rule*).

C'est une règle topologique, où le sens du parcours revêt une importance primordiale. Soit un point de l'espace, et un contour simple connexe et non croisé. On peut attribuer à ce point trois valeurs :

1. 0 si le contour n'est pas autour du point (point extérieur);
2. 1 si le contour est autour du point, et orienté dans le sens direct;
3. -1 si le contour est autour du point, et orienté dans le sens indirect.

Si le point est sur le contour, la valeur peut être quelconque dans l'intervalle $[-1, +1]$; on considère que le point est intérieur.

Sachant qu'un contour croisé n fois est équivalent à $n + 1$ contours non croisés, on calcule la valeur de la règle pour chaque point de l'espace en additionnant les valeurs correspondant à chaque contour connexe. Si la valeur finale est 0, le point est extérieur au parcours; sinon le point est intérieur au parcours. Selon cette règle, les points du parcours sont intérieurs au parcours.

PostScript permet d'employer également la règle pair-impair (*Even-Odd Rule*), par l'opérateur *eofill*. On effectue le même calcul. Si la valeur finale est paire, le point est extérieur au parcours; sinon le point est intérieur au parcours. Ici, l'orientation des contours n'a pas d'importance, c'est simplement leur nombre. Cette règle est en général la seule disponible dans les systèmes graphiques, mais elle est très rarement utilisée dans des fichiers PostScript.

Le description des règles ci-dessus en donne un modèle externe, conceptuel; on peut trouver un modèle équivalent basé sur des observations locales, à partir duquel une implémentation est possible. L'algorithme pair-impair est classique, et on peut l'adapter assez simplement pour la règle de la tangente tournante.

La découpe

La découpe est définie par l'opérateur *clip* ou *eoclip*. Bien que la découpe ne se traduise pas par un tracé dans la page, c'est conceptuellement le tracé d'un pochoir dans une feuille recouvrant la page.

Les problèmes qui se posent pour définir le pochoir sont exactement des problèmes de remplissage, puisqu'il faut définir un intérieur et un extérieur. Comme pour le remplissage, on peut choisir entre les deux règles tangente-tournante ou pair-impair.

Les lignes épaisses

L'opérateur de tracé de ligne épaisse est *stroke*. Il trace le parcours courant avec l'épaisseur courante.

L'épaisseur courante est définie dans le **repère utilisateur**. Cela veut dire que la ligne est d'épaisseur constante dans le repère utilisateur. Si la CTM n'est pas orthonormée, la ligne est d'épaisseur variable dans le repère appareil, selon l'orientation du segment; en effet dans le cas général, le cercle du repère utilisateur de diamètre l'épaisseur de trait devient dans le repère appareil une ellipse quelconque.

On peut tracer les lignes épaisses en pointillé. Un motif de pointillé est défini dans le repère utilisateur et appliqué cycliquement le long du parcours, en fonction de l'abscisse curviligne du parcours dans le repère utilisateur (cf. opérateur *setdash*).

En interne, l'interpréteur PostScript en

fonction de l'épaisseur, de la CTM, des attributs de pointillé, de forme des jointures et des bouts, transforme le parcours en un nouveau parcours qui, rempli par la règle de la tangente-tournante, donnera la ligne épaisse (cf. opérateur *strokepath*). Cette transformation de parcours à épaisir en parcours à remplir est purement géométrique.

Les images

Chaque pixel d'une image est transformé en un parallélogramme quelconque qui est rempli avec la couleur voulue. Les images sont en niveaux de gris sur 8 bits au maximum, soit 256 niveaux.

La différence entre les opérateurs *image* et *imagemask* réside simplement dans le choix de couleur. Dans le cas d'*image*, tous les pixels ont leur couleur propre, noir, blanc ou gris, et sont tracés dans cette couleur. Dans le cas d'*imagemask*, les pixels "noirs" ont la couleur courante, et les pixels "blancs" ne sont pas tracés.

7. Les couleurs

Le modèle PostScript est un modèle en couleurs; le noir, le blanc et les gris ne sont que des cas particuliers de couleurs. Il n'y a pas de notion de transparence.

L'état graphique contient la spécification de la couleur courante. C'est elle qui est utilisée par tous les opérateurs de tracé : *fill*, *stroke*, *show*, *imagemask*.

Le cas de la vraie couleur est le plus simple : les surfaces sont encrées dans la couleur courante, qui peut être spécifiée selon le modèle RVB (rouge, vert, bleu) ou HSB (teinte, saturation, luminance : *hue*, *saturation*, *brightness*).

La plupart des systèmes de visualisation (en particulier d'impression) sont mono-

chromes. Dans ces systèmes, la visualisation de niveaux de gris est obtenue par le procédé des demi-teintes. Un niveau de gris est simulé visuellement par une texture formée d'une suite de points noirs et blancs : la densité de points noirs donne, en première approximation, le niveau de gris; l'œ il joue le rôle d'intégrateur spatial.

La texture peut être spécifiée très précisément pour chaque niveau de gris par les opérateurs *setscreen* et *settransfer*. Le motif de la texture peut être absolument quelconque, et il n'y a pas de limitation théorique aux dimensions de ce motif.

Dans le cas de la demi-teinte, les surfaces sont encrées avec le motif de texture; les points noirs sont encrés en noir, les points blancs sont encrés en blanc⁷.

Si on compare ce modèle au modes vidéo des systèmes graphiques classiques, seul le mode *STORE* est disponible; en particulier, il n'y a pas le mode *XOR* si prisé. En effet, le modèle est un modèle purement couleur; or *XOR* en couleurs n'a pas de signification établie. D'autre part, le modèle du peintre est un modèle "sans mémoire" : le résultat de l'encrage d'une zone est indépendant de ce qui pouvait être auparavant sur cette zone.

8. Les caractères

En pratique, le tracé des caractères utilise les opérateurs standard de tracé évoqués plus haut. Cependant, au lieu que le tracé soit effectué dans la mémoire de page, il est effectué en général dans la mémoire cache caractères.

Tout ce qui précède s'applique donc de façon identique aux caractères, pour leur tracé dans le cache, sauf que dans le cache,

⁷C'est le principe de la peinture écossaise.

un caractère est toujours tracé en noir.

Pour finalement placer des caractères sur la page, il faut donc pouvoir prendre une image discrète de caractère dans le cache et la poser dans la mémoire de page avec la couleur courante.

C'est conceptuellement le cas favorable de l'opérateur *imagemask* : la résolution de l'image est la résolution de l'appareil et l'orientation de l'image est parallèle aux axes.

Comme pour tous les opérateurs de tracé, la trace effective laissée par un caractère dépend du parcours de découpe courant.

9. Conclusion

Le modèle graphique de PostScript s'appuie sur un petit nombre de concepts, simples et cohérents, que nous avons passés en revue dans cet exposé. Cet ensemble de concepts est à la fois minimal et complet.

Il faut cependant souligner que la force véritable de PostScript est d'avoir su rassembler ces concepts au sein d'un langage rigoureux, où les opérateurs graphiques et les opérateurs généraux sont utilisés de façon strictement équivalente, permettant une extensibilité totale. C'est cette vertu qui donne à PostScript sa richesse, sa puissance et sa simplicité d'emploi.

PUBLICITÉ

Vient de paraître



Coéditeur français Editions Universitaires
77 rue de Vaugirard
F-75006 PARIS France

Auteur(s) ANIS Jacques, avec la collaboration de J.L. CHISS et de Chr. PUECH

Titre L'ECRITURE, théorie et description

Collection Prisme 10, série : problématiques

Public Etudiants en lettres, de sciences humaines et des arts graphiques ; enseignants de tout niveaux ; professionnels des arts graphiques.

Contenu Les développements de l'informatique (desktop publishing, E.A.O.) remettent, paradoxalement, l'écriture au premier plan. Ce livre réalise une analyse très complète des problèmes de l'écrit. (ethnologie, linguistique, pédagogie). Les auteurs dégagent le rôle autonome de la langue écrite par rapport à la parole, notamment dans les systèmes graphiques ; ils présentent ensuite une étude des espaces graphiques avec études de cas portant sur le journal, le poème et le vidéotex.

Nombreuses suggestions pédagogiques et analyses de documents concrets.

