Cahiers GUTenberg

UNE MEILLEURE INTÉGRATION DU GRAPHISME MACINTOSH

Laurent SIEBENMANN

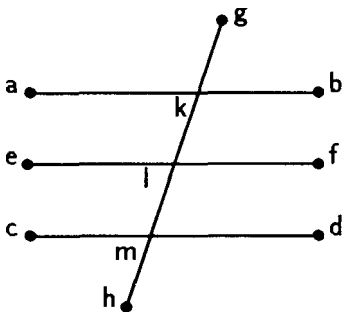
Cahiers GUTenberg, nº 4 (1989), p. 29-38.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_1989___4_29_0>

Une meilleure intégration du graphisme Macintosh

Laurent SIEBENMANN

Mathématique, Bât 425, Université de Paris-Sud, 91405-Orsay, France
Email : siebenma@frlal51.bitnet



Mea industria non sine maximo labore effeci ut qua facilitate litterarum elementa imprimuntur ea etiam geometricae figurae conficerentur.

E. Ratdolt, 1482

Une trentaine d'années après la première impression de la Bible par Gutenberg et ses associés, un dénommé Erhardt Ratdolt a réalisé, à Venise, la première édition imprimée des *Éléments* d'Euclide. Dans les marges généreuses de cette édition, se trouvent les figures[1] géométriques accompagnant les propositions d'Euclide. Ratdolt mentionne, en préface, les difficultés que ce graphisme lui a posé. A l'époque, bien que l'impression de livres d'auteurs modernes et antiques ait été une affaire de tous les jours, rien ou si peu de la Mathématique n'avait été publié. Il explique cela par les difficultés posées par les figures. Sa fière contribution après de grands efforts fut un processus pour d'imprimer les figures aussi facilement que les caractères. Voir [Heath, Vol. 1, Chap. VIII].

Nous voici, un demi millénaire après, avec la Publication Assistée par Ordinateur, de nouveau penchés ce même problème. Donald Knuth est sûrement un Gutenberg de la PAO ; en même temps il est bien connu que ses chefs-d'œuvre TEX et METAFONT laissent en suspens l'intégration du graphisme.

A la décharge de Knuth, rappelons que TEX offre une solution hors pair au problème de la typographie des formules mathématiques ainsi qu'au problème de leur intégration dans la prose. Je crois qu'il est juste de dire qu'après des siècles d'évolution des notations mathématiques depuis la première impression des *Éléments*, le problème des formules abordé par Knuth avait éclipsé le problème posé par les figures.

Si on admet ensuite qu'en mathématiques les figures sont à supprimer, comme le réclament beaucoup de journaux mathématiques (mais bien peu de lecteurs), on se heurte au fait que le graphisme reste incontournable pour la majorité des autres sciences. Le président de notre association GUTenberg, Bernard Gaulle, classe à juste titre le problème de l'intégration du graphisme avant tout autre problème posé

[1] La figure ci-dessus est une refonte par MacDraw de la figure pour la transitivité du parallélisme.

couramment par TEX, à l'exception possible de la césure multilangue.

Le but de cette note est de présenter un mécanisme amélioré qui permet d'intégrer *massivement* le graphisme des normes PICT et EPSF dans les manuscrits TEX. La norme PICT, née avec le Macintosh et son système d'affichage QuickDraw, est appuyée par le logiciel de dessin MacDraw (en premier) et la majorité des autres logiciels de graphisme Macintosh (sauf Illustrator et FreeHand), dont Cricket Draw, Canvas 2.0, ... La norme EPSF (Encapsulated PostScript File [EPSF]) née avec les imprimantes LaserWriter est plus récente et plus puissante ; elle est utilisé par Adobe Illustrator [Illu] (en premier), et par d'autres [Cric], [FH]. PostScript est le langage de description de page créé par Adobe Systems Inc. pour LaserWriter ; il devient un standard *de facto* et passera bientôt dans le domaine public.

Attention ! Presque chaque logiciel utilise en premier lieu son propre format de fichier ; on se permet de dire qu'il appuie PICT (respectivement EPSF) dans la mesure où il supporte des passerelles avec cette norme. Notons d'ailleurs qu'il y a d'importantes différences entre les versions d'un même logiciel, surtout pour MacDraw.

L'article de A. Antoniadis [Ant] lors du congrès GUTenberg'89 décrit bien ces outils graphiques pour Macintosh. Leur excellente interface utilisateur a déchaîné une avalanche de bons graphisque qui justifie mon effort pour une intégration *massive*. La dynamique est bien décrite par Ferguson [Ferg] :

> « L'avantage principal d'utiliser MacDraw est le sens d'accomplissement immédiat (...) tellement vivifiant qu'il engendre un effort et une concentration extraordinaire. Il importe peu qu'il soit difficile d'obtenir des résultats précis et agréables dans certains cas. Le sentiment général est que c'est un résultat satisfaisant et bien mieux que ce qui pourrait être accompli, dans les même limites de temps, par d'autres méthodes. »

L'idée derrière ce mécanisme rejoint sur un point essentiel la majorité des efforts d'intégration de graphismes contemporains. On fait en sorte que l'objet graphique se comporte comme un ou plusieurs caractères — car TEX, comme Gutenberg, est un virtuose en l'art du positionnement de caractères. Malgré le passage d'un demi-millénaire, c'est vraisemblablement là l'idée maîtresse de Ratdolt !

Il y a une distinction technique fondamentale à signaler : A. Violante [Viol] et P. Wilcox [Wilc] mettent les objets graphiques dans des polices graphiques créées avec METAFONT. Je passe par le protocole `\special{...}` conçu par Knuth pour les inclusions graphiques ; ce n'est pas pour autant du TEX standard, car l'interprétation de la commande `\special{...}` est laissée au pilote de l'imprimante (ou de l'épreuvage écran). Les alternatives pour l'exploitation de `\special{...}` évoluent rapidement, voir par exemple [BaSL], [Beebe], [CaBGC], [DeLo], [ReHo].

Le mécanisme que je vais décrire résulte en effet d'une collaboration entre le pilote d'imprimante de Textures (versions postériures à 1.01f), et mon préprocesseur Sweet-teX (versions postfieures à novembre 1989). Rappelons que Textures de Barry Smith [Smit] est la réalisation de TEX sur le Mac qui fut la première à circu-

ler, et reste de loin la plus évoluée, tandis que Sweet-teX est un préprocesseur dont le point fort jusqu'à présent serait la saisie de textes mathématiques très chargés de formules.

Je souligne que ce mécanisme n'élargit pas la gamme des objets graphiques qu'il est possible intégrer. Il s'agit très simplement de faciliter leur intégration de façon à ce qu'une dactylo technique fasse sans problème ce qu'auparavant elle repoussait vers un expert.

Le mécanisme « publication/abonnement » prévu pour System 7.0 du Macintosh (1990+) permettra-t-il bientôt de mieux faire ? Le présent projet a l'avantage de marcher d'ores et déjà, avec tous les systèmes et sur tous les Macintosh.

On ne discutera pas ici d'un éventuel enrobage du graphisme avec la prose (l'enroulement de la prose autour de l'objet graphique) ; voir [Ferg], et le journal TUGboat. Mais on doit noter que les traitements de texte interactifs Microsoft Word et Nisus viennent de progresser nettement dans cette direction.

D'ailleurs, on ne trouvera ici aucune intégration interactive. Par contre, Barry Smith, l'auteur de Textures, a récemment ouvert une telle voie — voir conclusion.

1. Le graphisme pratiqué sous Textures au début de 1989

Pour apprécier le mécanisme amélioré que j'offre, il faut brièvement rappeler le *statu quo* du début de 1989 [Ant].

Parmi les divers mécanismes offerts par Textures pour l'intégration du graphisme, le suivant a pris nettement le dessus. Celui-ci date de la version 1.0 de 1987 et se résume en trois points :

1. Utiliser le presse-papiers et le calepin pour coller les objets graphiques dans un compartiment *pictures* du fichier de Textures approprié, et attribuer à chaque objet un nom.

2. Utiliser dans le fichier `.tex` une commande convenable pour placer l'objet graphique. Cette commande de positionnement, à fournir par l'utilisateur, cf. [DeLo] §3.1, va appeler la commande primitive :

```
\special{picture <nom>
        scaled  <echelle>}
```

   Ensuite, TEX passe cette commande primitive, sans y toucher (sauf pour d'éventuelles expansions de ses arguments), au pilote de l'imprimante, qui pose à son point d'insertion courant l'objet graphique à l'échelle prescrite. Le coin sud-ouest du rectangle minimal qui encadre l'objet se pose au point d'insertion. C'est entièrement la responsabilité de la commande de positionnement de tenir compte de la taille du dessin et de lui réserver la place appropriée. Pour faciliter cela, Textures affiche la largeur et la hauteur de l'objet dans son compartiment *pictures*. Il incombe donc à l'utilisateur de :

3. Inscrire, dans cette commande de positionnement, le nom de l'objet, ses dimensions (largeur et hauteur), et l'échelle voulue.

Ce mécanisme n'est pas du tout difficile à comprendre et, comme première intégration du graphisme, il a réellement fait éclat en lançant une cohabitation de TEX avec le pullulant graphisme du Mac. Le nom Textures vient, en effet, de "TEX" + "pictures".

Plusieurs autres amorces sont fournies par Textures (dans sa version 1.01f) pour l'intégration du graphisme et agissent de façon semblable, mais à partir de données différentes :

```
\special{postscript
        <instructions Postscript>}
\special{postscriptfile
        <nom de fichier Postscript>}
\special{pictfile
        <nom de fichier PICT>}
\special{illustration
        <nom de fichier EPSF>}
```

Nous allons exploiter les deux dernières.

Je trouve que l'intégration du graphisme sous Textures est remarquable par sa généralité et par la qualité des résultats obtenus, mais trop peu ergonomique pour une exploitation massive. Bref, on peut tout faire, mais c'est lourd à gérer dès qu'il s'agit de quantités importantes.

Pour mieux comprendre, revenons à la première recette, qui est la plus ergonomique. Il est vrai qu'une bonne commande de positionnement impose à l'objet graphique le comportement docile et agréable d'un caractère. Une syntaxe possible est :

```
\FramedPicture{<nom>}{<largeur>}
              {<hauteur>}{echelle>}
```

Étant un préprocesseur, Sweet-teX peut aussi offrir un symbolisme un peu plus suggestif, ce qui ne change rien sur le fond.

Après m'avoir donné satisfaction à petite échelle, ce mécanisme de Textures s'est enlisé. Voici un cas que j'ai vu de près : un chapitre de cinquante pages contenant une centaine de petites figures (soit deux par page, une densité modeste) a nécessité une journée de travail pour l'installation du graphisme, puis, à la suite d'une petite erreur de jugement, il a fallu tout refaire. Dans de tels cas défavorables, l'administration du graphisme se révèle aussi coûteuse que sa création, sans offrir en retour la moindre valeur scientifique ou esthétique.

2. Le mécanisme amélioré

Pour répondre à l'appel lancé par GUTenberg pour l'amélioration de l'intégration du graphisme en 1989, je me suis appliqué à éliminer tout le superflu dans le procédé d'intégration 1.,...,3. ci-dessus. L'étape 1 d'importation a ainsi disparue, et le report des dimensions en 3 également. La spécification d'échelle s'est assouplie. La procédure modifiée (encore en évolution !) est la suivante :

Une simple commande du genre[2] :

```
\FP{<nom de fichier>}
    \scale<echelle>\eFP
```

donne une image qui se comportera comme un rectangle (dont la largeur et la hauteur sont celles implicites dans le fichier graphique dénommé `<nom de fichier>` mais corrigées par l'échelle spécifiée et aussi par `\magnification`. L'utilisateur n'a plus besoin de préciser la largeur et la hauteur — pas plus que pour un caractère. Sweet-teX fournira à TEX ces dimensions et les moyens de repérer le fichier (voir plus loin). A l'impression, Textures va insérer le fichier d'art moyennant la commande :

```
\special{<filetype> <nom de fichier>
        scaled <echelle>}
```

où `<filetype>` est `pictfile` ou `illustration`.

[2] Les noms de macro utilisés dans ce texte ainsi que leur syntaxe sont encore provisoires.

Il est également souhaitable de simplifier la spécification de l'échelle. Souvent, beaucoup d'objets graphiques de suite ont la même « échelle de base » ; on a donc le droit de spécifier celle-ci à part, par `\Sc=<echelle>`, et puis d'écrire plusieurs fois de suite seulement :

```
\FP{<nom de fichier>\eFP
```

Puisqu'un mauvais choix de proportions et d'échelle afflige la majorité des débutants avec MacDraw, j'offre des conseils, qui vont simultanément clarifier l'idée d'échelle de base. Normalement une figure ne doit pas être beaucoup plus grande qu'il ne faut pour être lisible, quitte à l'insérer dans le texte si elle est alors assez petite pour peu écarter les lignes. D'ailleurs, les caractères dans les étiquettes (labels) de la figure[3] devraient finalement être d'une taille proche des caractères du texte autour. Essayez donc la politique suivante. Choisir un corps très grand pour les étiquettes à l'écran (soit 36 pt). Dessiner maintenant la figure avec ses proportions en harmonie avec les étiquettes — pas plus grande que nécessaire, mais, à cause des grandes étiquettes, forcément avec des proportions généreuses, ce qui assurera une bonne précision. *L'échelle de base* `\Sc` sera alors 1000/36 ; c'est à dire choisie pour donner des étiquettes de 10 points – comme le texte. (Mille vaut toujours 100%.)

Le débutant va souvent se tromper dans le choix des proportions entre étiquettes et dessin. En 1989[4], la bonne solution pour MacDraw est d'appliquer un facteur d'homothétie convenable aux éléments *graphiques* du dessin. (Pour l'appliquer, il on associe d'abord tous les éléments, y compris les étiquettes.) Une étiquette ne subit pas l'homothétie, seulement son point d'insertion ! Ensuite, il faut ajuster, à la main, la position de quelques étiquettes ; mais, en fait, si on a bien choisi le point d'insertion sur l'étiquette (il y a un choix entre gauche, droite, et centre) beaucoup d'étiquettes vont automatiquement se replacer correctement.

Au stade de l'épreuvage, on peut ajuster l'échelle sans se rappeler l'échelle de base. En effet, on peut introduire un facteur de correction (soit 0.87) comme suit :

```
\FP{LaFigure}\scale=.87\Sc\eFP
```

Pour éviter d'offenser l'œil du lecteur, on maintient ce facteur normalement dans l'intervalle de 0.7 à 1.1. On veut parfois aussi recentrer la figure vis-à-vis de sa boîte en utilisant une commande comme `\Slide { <vecteur>}` posée avant `\eFP` ; elle a le rôle de la petite main dans MacPaint.

En fin de compte, ce qui accélère le plus le rythme de production, c'est que, *la plupart du temps*, on peut correctement insérer un objet graphique, *sans spécifier le moindre paramètre* ; on tape simplement quelque chose comme :

```
\FP{LaFigure}\eFP
```

Peu d'auteurs ont offert pour cela une recette plausible. Le meilleur alternatif me semble celui de Ferguson [Ferg], qui propose de spécifier « au pif » un seul paramètre, le déplacement vertical (pour une figure à mettre en exposition). Ayant, lui aussi, accès à la hauteur à la source

[3] Utiliser une fonte LaserWriter ; un contraste avec les caractères du texte est, en général, un plus.

[4] En l'absence de polices vectorisées « en contour » pour l'écran il n'est pas question d'appliquer une homothétie arbitraire aux étiquettes.

(dans le fichier contenant la figure), il sait d'avance programmer l'échelle qui sera convenable pour caser la figure.

Une seule fois par article , on doit mettre en marche ce nouveau mécanisme en suivant trois étapes :

1. Ranger les objets graphiques dans un « d'art » qui est, en fait, un dossier du système de fichiers du Macintosh, chaque objet graphique étant sauvegardé (option *Save as* de MacDraw) comme un fichier de type `PICT` ou `EPSF`. En pratique, on va créer et maintenir le graphisme sous une de ces deux normes.
2. Mettre en début de fichier `\input RESOURCE`. Dorénavant le fichier `RESOURCE.tex` de commandes pour Sweet-teX (stocké dans le même dossier que Textures) contient un petit jeu de commandes TEX construites pour gérer le graphisme. (Ce jeu est en fait indépendant et portable.)
3. Insérer ensuite la commande suivante de Sweet-teX dans le texte avant de rencontrer le premier objet graphique :

   ```
   :■!EXTART <adresse complète
   du dossier d'art>
   ```

   Cette localisation est souvent compliquée, par exemple :

   ```
   DisqueDur:MonDossier: ...
       :ArticleEnCours:DossierArt:
   ```

   Donc Sweet-teX vous permet de la spécifier par menu ou même de l'omettre. L'omission ne coûte pas cher ; elle est possible dès que Sweet-teX et TEX sont situés dans le même dossier, et les fichiers d'art dans un sous-dossier portant le nom `ARTFOLDER`.

En réponse à cette commande, le préprocesseur Sweet-teX va explorer le dossier d'art désigné et remettre à jour un catalogue de langage TEX affichant les dimensions des objets d'art. Pour cela, il doit essentiellement lire les fichiers d'art de type `PICT` ou `EPSF`, *ce qu'un TEX standard ne sait pas faire* . Par la suite, TEX va consulter ce catalogue (moyennant la commande `\input`) pour pouvoir calculer comment placer les objets graphiques à l'aide du jeu de commandes mentionné.

Je signale deux difficultés qu'il a fallu contourner.

Vitesse. La lecture de douzaines de fichiers peut être une affaire assez lente, et, pourtant, chaque manuscrit est normalement retraité à maintes reprises. Donc le préprocesseur doit prendre soin de ne jamais relire un fichier sauf s'il a été modifié depuis la dernière remise à jour du catalogue.

Capacité. Puisque, actuellement, Textures stocke dans sa tranche de mémoire RAM précieuse, plutôt que sur disque, les informations cueillies dans un catalogue de dossier d'art, on a besoin d'une stratégie pour limiter les dépenses de RAM — lorsqu'il s'agit, par exemple, d'un livre. Il est ainsi permis d'utiliser plusieurs dossiers d'art, quitte à se contenter d'un à la fois. Plus précisément, si vous appelez un deuxième dossier d'art par le protocole `EXTART`, les renseignements sur le premier seront effacés avant que ceux sur le deuxième ne soient chargés à partir du deuxième catalogue. (Une macro commande vous permettra d'utiliser plusieurs dossiers simultanément — au risque de dépasser les capacités de TEX.) Pour un livre, une politique typique serait d'établir un dossier d'art par chapitre.

3. Faut il donner un statut aux Lignes de Base ?

De plus en plus d'experts croient que les inclusions graphiques devraient avoir, en plus de la hauteur, une profondeur spécifiable par le dessinateur ; c'est-à-dire qu'ils devraient posséder, comme un caractère, une *ligne de base* bien désignée. Elle est particulièrement importante pour les petites figures à insérer dans une ligne. Sweet-teX emploie, par défaut, une ligne de base offerte par la construction `\vcenter { ... }` de TEX, ce qui laisse parfois à désirer. Mais on pourra spécifier le niveau de la ligne de base par rapport à la fenêtre (feuille) du logiciel de dessin par une commande comme `\Baseline =<dimension>`. En pratique cela veut dire que le dessinateur va travailler, pour tout un lot de graphiques, sur une fenêtre sur laquelle est tracée une ligne de base ; c'est une ligne temporaire ou imaginaire qui aura valeur pour les rectangles minimaux encadrant les objets graphiques. (Pour Sweet-teX, un tel rectangle n'est spécifié qu'à la translation près ; donc le choix de `\Baseline` est assez arbitraire.)

On aurait raison de se poser la même question pour ce que j'ai appelé ci-dessus *l'échelle de base*, car le dessinateur me semble le mieux qualifié pour la spécifier. C'est important si elle est très variable. Un jour, le dessinateur saura, sans doute, *exécuter d'avance* une réduction à l'échelle de base (sans dégrader les fichiers d'art !), et ainsi la réduire pour Sweet-teX toujours à 1000 (=100%), ce qui résoudra le problème. Pour le moment, il doit se contenter d'une astuce comme l'addition d'une extension numérique au nom de fichier d'art qui indique l'échelle de base.

4. D'autres réalisations de TEX ?

Reste à voir si d'autres réalisations de TEX (éventuellement sur d'autres ordinateurs !) vont s'aligner sur ce mécanisme. Comme nous l'avons noté ci-dessus, la condition prélable principale concerne les commandes `\special {...}` à interpréter par le pilote de l'imprimante. Quant à OzTEX, la norme `EPSF` (plus puissante mais plus rare que `PICT`) seule est reconnue.

5. EPSF : norme encombrante ?

La plupart des fichiers `EPSF` sont inutilement encombrants par la présence d'un « lexique » d'à peu près 10K qui ne varie guère entre deux fichiers de même provenance : une disquette de très petits fichiers `PICT` peut fort bien se traduire en dix disquettes de fichiers `EPSF` ! (L'encombrement minimal, en pratique, d'un fichier `PICT` est de 1K.) Il y a donc un travail de dégraissage à accomplir, de préférence par les concepteurs des logiciels responsables : Illustrator, Cricket Draw, Free Hand ... .

6. Mobilité des manuscrits au-delà du monde Macintosh

Voici un sujet qui exigera des efforts supplémentaires !

Le plus urgent est d'assurer qu'un imprimeur professionnel hors du monde du Mac puisse imprimer *avec le graphisme* ce que TEX produit sur le Mac. S'il s'agit d'exporter, pour impression finale par une imprimante Postscript de haute résolu-

tion, une solution possible va être proposée par Textures. En effet, un ensemble de fontes Computer Modern sous forme *outline* de Postscript sera bientôt en vente (1990 ?). Il sera alors possible d'obtenir une traduction Postscript « universelle » du `.dvi` par l'astuce bien connue d'une impression pendant que la touche « Commande K » est déprimée.

Il serait bon d'assurer la mobilité des sources déjà au niveau des fichiers `.tex`. Pour cela, il faudra deux progrès :

1. *Une bonne norme pour l'interprétation par les pilotes d'imprimantes de la commande* `\special {...}`. Elle doit être établie et suivie. Un comité de TUG en est saisi, et un rapport intérimaire [ReHo] suggère que nos « catalogues d'art » soient des GDF *(Graphics Description Files).*
2. *Le remplacement, en ce qui concerne les fichiers exportés, de la norme* PICT *(non valable hors du Mac) par la norme Postscript* EPSF.

Ce deuxième progrès est aussi en cours ; en effet plusieurs classes de passerelles de PICT vers Postscript existent :

1. pour les besoins de l'imprimante LaserWriter (via LaserWriter et LaserPrep) ; on utilise Commande F ou Commande K pendant l'impression pour intercepter du code Postscript. Ce mécanisme sera vraisemblement remanié pour System 7 du Mac.
2. pour certains programmes de dessin dont Cricket Draw [Cric] et Free Hand [FH].
3. pour les logiciels de traitement d'image en PAO, dont Pixel Paint Professional, Video Paint, Digital Darkroom, MacScan, Zoom, Impression, PhotoMac, Finale.
4. pour les logiciels de mise en page en PAO, dont Ready Set Go, PageMaker, Quark Express, Design Studio, Cortex.

Les passerelles fournies sont souvent imparfaites, et peu rôdées pour une exploitation massive ; mais le professionnel de PAO prêt à improviser ne manque pas de points de départ ! (Je remercie Wm. McBride du journal Mac Fan pour les indications 3 et 4).

Notons que l'approche via METAFONT à l'intégration du graphisme avec TEX [Viol], [Wilc], bien qu'elle soit plus lourde, offre d'ores et déjà une mobilité *universelle* des fichiers de sortie. En effet, ces derniers consistent en fichiers `.tex` et/ou `.dvi` standard accompagnés de polices du format PK. Deux nuances : la mobilité ne serait pourtant pas toujours *parfaite* à cause de problèmes d'alignement de « bitmaps » d'un carrelage auxiliaire recouvrant l'objet graphique. Quant au Macintosh, le format PK n'est pas reconnu par Textures, bien qu'il le soit dorénavant par OzTEX, [Trev].

7. Variabilité des normes

Le rectangle minimal que Sweet-teX exploite pour intégrer un objet graphique est moins fondamental et solide que je n'aie suggéré. Par exemple, après une rotation de l'objet, il doit être recalculé *ab initio.* D'ailleurs, même la définition varie un peu suivant la norme et l'application.

Traditionnellement, dans la structure d'un fichier PICT ou EPSF, ce rectangle a une ou plusieurs places réservées. Malheureusement, pour PICT, ces conventions changent (et se multiplient) ; pour s'en convaincre, il suffit d'exploiter `\special{ pictfile <nom de fichier PICT>}` (de

Textures 1.01f) successivement avec MacDraw 1.xx et MacDraw II. Chaque variation de norme impose un effort supplémentaire — soit au programmeur soit à l'utilisateur. Sweet-teX aplanira les incohérences de MacDraw et de Textures, mais, faute de temps, de bien peu d'autres logiciels.) Ici, la perfection ne sera peut-être jamais atteinte.

8. Pourquoi Sweet-teX ?

Dans cette intégration, TEX a besoin de renseignements qu'il ne sait pas commodément lire tout seul, et *cela donne tout naturellement une tâche pour un préprocesseur* . Il est vrai que n'importe quelle réalisation de TEX pourrait se l'arroger ; mais pas par un protocole établi de TEX comme `\special`.

On va peut-être me reprocher d'avoir caché un petit préprocesseur indépendant dans les entrailles d'un grand. En réalité, le fidèle du langage TEX orthodoxe n'a pas raison de protester, car Sweet-teX offre dorénavant à tous ce petit service graphique sans exiger la moindre connaissance de ses autres possibilités. Ayant exploité Sweet-teX une seule fois pour la corvée de créer un catalogue d'art, on peut très bien utiliser directement ce catalogue (moyennant la commande `\input` et le petit jeu de macros mentionné) sans recourir à Sweet-teX à chaque composition du texte. Autre facilité, *Sweet-teX offre l'option de créer le* catalogue sans la commande `!EXTART` du préprocesseur : on clique sur une icône et on sélectionne le dossier d'art par menu.

9. L'exportation des fichiers `.dvi` de Textures vers Illustrator d'Adobe

Voici, brièvement, une nouveauté qui offre des possibilités *qualitatives* nouvelles pour Textures. Les versions de Textures supérieures à 1.01f du printemps 1989 permettent d'utiliser l'option *Save as* pendant qu'une page d'épreuvage est affichée, dans le but d'exporter la page comme fichier `EPSF`. Elle est alors assimilable dans Adobe Illustrator[5], où elle se décompose en sous-objets graphiques à translater, augmenter, etc. Puisque le fichier `EPSF` qui en résulte peut être réassimilé par TEX, on voit naître une possibilité d'échanges bilatéraux entre TEX et Postscript qui est attendue depuis longtemps et qui aura beaucoup de répercussions.

10. En conclusion

Textures et Sweet-teX offrent maintenant ensemble un mécanisme amélioré d'intégration de graphisme quelque peu semblable à un camion diesel — un peu lent à se mettre en route, mais, une fois en régime de croisière, remarquablement économique et efficace. C'est fort utile, sans être divertissant. Heureusement, par l'exportation du `.dvi` vers Illustrator, on peut ensuite s'offrir les qualités d'une Montgolfière !

Références bibliographiques

[Ant] A. ANTONIADIS, « Graphisme et TEX dans l'environnement Macintosh », *Cahiers GUTenberg*, n° 2 (mai 1989), 63-72.

[BaSL] *M. BALLANTYNE, M. SPIVAK, Y. LEE*, « HI-TEX Cutting and Pasting », *TUGboat*,

[5] Et d'autres logiciels axés sur EPSF, Cf. 1–4 ci-dessus.

Vol. 10 (1989), n ° 2, 164-5.

[Beebe] N. BEEBE, « TEX and graphics: the state of the problem », *Cahiers GUTenberg*, n ° 2 (mai 1989), 13-53.

[Cric] **Cricket Draw** (logiciel), Cricket Software 1987, Tél : (215-) 251-9890, 30 Valley Stream Parkway, Malvern PA19355. *Ce logiciel offre, d'une part, une interface très semblable à MacDraw. Il offre en plus une passerelle bidirectionnelle entre ses propres fichiers et la norme* PICT *; elle est imparfaite mais utile. On a aussi une passerelle en sens unique vers* EPSF. *Une version améliorée, Cricket Draw Master, sera examiné par la revue Mac Fan au printemps 1990.*

[CaBGC] L. CARNES, C. BOHN, I. GRIFFIN, B. CARNES, « Etude de TEX Graphics pour le PC », *Cahiers GUTenberg* n ° 3 (oct. 1989).

[Canv] **Canvas 2.0** (logiciel), Denaba Software, 3305 NW 74th Avenue, Miami FL 33122. *Une variante récente de la conjonction de MacDraw et MacPaint. Plus commode que MacDrawII pour un travail de norme PICT.*

[DeLo] B. DECOUTY et PH. LOUARN, « L'environnement de production de documents TEX à IRISA », *Cahiers GUTenberg*, n ° 2 (Mai 1989), 90-95.

[EPSF] « Encapsulated Postscript files specification, Version 2.0 », *Postscript Developer Tools Group*, PN LPS 5002, Jan. 1989, Adobe Systems Inc., 187 Embarcadero Rd., Suite 200, Palo Alto, CA94303.

[FH] **Free Hand** (logiciel), Aldus Corp. 1988, Altsys Corp. 1988, Version 2.02. *Ce logiciel haut de gamme a des paramètres semblables à [Cric]. (La conversion* vers *PICT manque pourtant.)*

[Ferg] M.G. FERGUSON, « L'incorporation de graphiques dans INRSTEX », *Cahiers GUTenberg*, n ° 2 (mai 1989), 81-89.

[Heath] T.L. HEATH, *Euclid's Elements*, 2nd edition, Cambridge U. Press, 1925, Dover 1956.

[Illu] **Illustrator** (logiciel), Adobe Systems, Inc., 1987-8. *Le premier logiciel de dessin sur Macintosh de norme* EPSF. *Sans passerelle vers la norme* PICT *!*

[MacD] **MacDraw** (logiciel), Apple Computer 1985-1987, and Claris Inc. 1988-89. *Le cheval de bataille du débutant en graphisme. A l'origine, chaque nouveau Macintosh comportait une licence. Sans passerelle vers la norme* EPSF *!*

[ReHo] T. REID and D. HOSEK, « Report from the DVI Driver Standards Committee », *TUGboat*, Vol. 10 (1989), n ° 2, 188-191.

[Sieb] L. SIEBENMANN, **Sweet-teX** (logiciel), *un préprocesseur pour TEX basé sur MacWrite ou WriteNow qui permet à une dactylo de préparer des manuscritspour TEX dans un environnement convivial semblable à d'autres traitements de textes scientifiques.*

[Smit] B. SMITH, **Textures** (logiciel), Blue Sky Research, 534 SW 3rd Avenue, Portland Oregon, 97204. *La première réalisation de TEX pour les ordinateurs Macintosh.*

[Trev] A. TREVORROW, **OzTEX** (logiciel), Kathleen Lumley College, North Adelaide, S.A., 5006 Australia. *Cette nouvelle réalisation de TEX, parue avril 1989, est du domaine public ; la version courante 1.2 d'octobre 1989 est compatible avec SweetteX.*

[Viol] A. VIOLANTE, « DDI dessin DVI », *Cahiers GUTenberg*, n ° 1 (avril 1989).

[Wilc] P. WILCOX, « MetaPlot: Machine independant line graphics for TEX », *TUGoat*, Vol. 10 (1989), n ° 2, 179-187.