Cahiers GUTenberg

VERS L'INTRODUCTION DE LA COULEUR DANS TEX

Christophe Cérin, Benoît Lemaire

Cahiers GUTenberg, nº 5 (1990), p. 8-16.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_1990___5_8_0>

Vers l'introduction de la couleur dans TEX

Christophe CÉRIN et Benoît LEMAIRE

Université Paris-Sud - LRI - Bât 490, 91405 ORSAY Cedex, tél : 69-41-70-82.
e-mail : `cerin@lri.lri.fr` *et* `lemaire@frlri61.bitnet`

Résumé Nous assistons, depuis un an, à l'arrivée sur le marché d'imprimantes couleurs. Si elles sont bien destinées à envahir nos bureaux, il conviendrait que TEX prenne en compte leurs nouvelles possibilités pour une meilleure publication assistée par ordinateur.

Dans cet article, nous discutons, en premier lieu, de la couleur en général, comment la spécifier, la représenter, puis, au travers d'exemples, nous essayons de dégager des classes de problèmes qui appellent des classes de solutions communes, des algorithmes communs. Cela nous conduit à proposer un ensemble minimal de primitives permettant de résoudre les problèmes identifiés suivants :

- action sur une entité syntaxique (lettre, mot, phrase, paragraphe, ...). Nous distinguons alors deux sous-cas :
  - simple mise en couleur de l'entité syntaxique avec des constructeurs du type `\begincolor ...\endcolor`.
    Par exemple, POUCE.
  - action plus complexe : la couleur d'une lettre dépend de celles de ses voisines ou, plus généralement, de paramètres spécifiant l'interaction des couleurs. Par exemple :

    ceci est un autre exemple de dégradé.
- action sur une entité graphique (souligné, cadre, vecteur ou flèche de l'environnement `picture`).

L'implantation des constructeurs initiaux proposée est réalisée en partie avec PostScript, par l'intermédiaire de la commande `special` dans la syntaxe reconnue par le translateur `dvips` de Tom Rokicki, et en TEX. L'ensemble des macros est écrit pour fonctionner avec LATEX. Deux jeux sont disponibles dont un pour la QMS ColorScript ainsi que trois primitives, d'un niveau d'abstraction élevé, permettant de réaliser des dégradés de couleurs.

1. Introduction

Lorsque les coûts d'inclusion dans nos documents d'éléments graphiques ou typographiques en couleur seront plus faibles, nous pouvons penser que l'extrémité de la chaîne de la micro-édition électronique (saisie–mise en forme–impression) sera constituée d'une imprimante laser couleur haute résolution.

Il faut bien reconnaître qu'actuellement, l'utilisateur de la micro informatique perçoit mal la dimension réelle de l'usage de la couleur et considère volontiers celle-ci comme, d'une part, un paramètre supplémentaire à gérer et d'autre part comme un luxe.

1.1. Intérêts et effets de la couleur

Si la deuxième affirmation s'avère encore exacte, nous pouvons tout de même nous demander si le mode et la vitesse d'extraction de l'information chez un sujet, ne se trouvent pas améliorés lorsqu'on lui présente un support de communication visuelle couleur (feuille de papier ou écran d'ordinateur) au lieu d'un support noir et blanc. Nous pouvons également nous interroger des effets de la couleur sur la forme et la présentation d'une information. Celle-ci devient-elle plus simple à capter, plus complexe (on a rajouté du bruit)? Son extraction devient-elle plus précise?

Des études montrent [Hoadley90] que la couleur amplifie les distinctions et les relations entre les différents contenus d'une information; elle facilite aussi les mécanismes d'apprentissage et de compréhension de cette même information. Nous pouvons résumer ces différents apports par :

- la couleur améliore les performances de mémorisation.

- la couleur améliore les performances dans les phases de recherche et de localisation d'informations.
- la couleur améliore la rétention d'informations fugitives.
- la couleur autorise un meilleur jugement pour la prise de décisions.

De plus, l'utilisation de représentations graphiques couleurs augmente la précision de l'information extraite [Hoadley90].

Dans les paragraphes qui suivent, nous allons voir que le paramètre couleur est lié à d'autres paramètres comme le contraste, la luminosité. Nous rappelerons alors les différents modèles de représentation des couleurs et les possibilités de base offertes aux programmeurs afin de piloter les imprimantes. Enfin, pour répondre aux problèmes du rendu des couleurs nous considèrerons la technologie des imprimantes couleur.

2. Spécification des couleurs

2.1. Les modèles de couleur

On a montré expérimentalement que l'on pouvait reconstituer toute lumière visible à partir de trois lumières (monochromatiques) bien choisies. Les trois couleurs ainsi déterminées sont appelées *couleurs primaires*. Le phénomène de perception, quant à lui, est purement psychophysiologique : deux couleurs ne provoquent pas forcément les mêmes sensations chez deux individus différents. Une couleur est définie par sa *teinte* (vert, jaune, bleu ...), sa *saturation* qui permet de mesurer la proportion de couleur pure par rapport au blanc, sa *luminance* (quotient de l'intensité lumineuse d'une surface par l'aire de cette surface). On peut alors définir trois teintes fondamentales : le rouge, le vert et le bleu, qui mélangées donnent le blanc. Si l'on combine deux de ces trois couleurs on obtient le complémentaire de la troisième (synthèse additive). Ainsi : *bleu+vert* $\rightarrow$ *cyan*, *rouge+vert* $\rightarrow$ *jaune*, *rouge* + *bleu* $\rightarrow$ *magenta*. Les trois modèles fondamentaux utilisés en informatique sont donc :

- le modèle RGB (red-green-blue).
- le modèle CMY (cyan-magenta-yellow) qui devient parfois CMYK (K pour black) lorsque, pour recréer la couleur originale on ajoute un peu de noir pour obtenir des effets de contraste nécessaires.
- le modèle HSB (hue–saturation–brightness).

Sur micro-ordinateur, l'utilisateur peut désigner une couleur, soit en cliquant directement sur un pixel d'une couleur particulière (comme par exemple pour le cône du macintosh – cf figure 1), soit en spécifiant chacune des valeurs des composantes de base comme avec Adobe Illustrator ou Aldus Freehand.

2.2. PostScript et les modèles CMYK, RGB et HSB

Les dernières versions de PostScript permettent de représenter une couleur selon un des trois précédents modèles. Les premières versions reconnaissaient les opérateurs `setrgbcolor` et `sethsbcolor` qui, parce qu'ils étaient implantés pour des imprimantes noir et blanc, utilisaient les niveaux de gris pour simuler les couleurs. On dispose désormais d'un opérateur `colorimage` pour gérer les images bitmap couleurs et d'un opérateur `setcmykcolor` pour désigner une couleur dans le modèle CMYK. Nous utiliserons dans nos macros

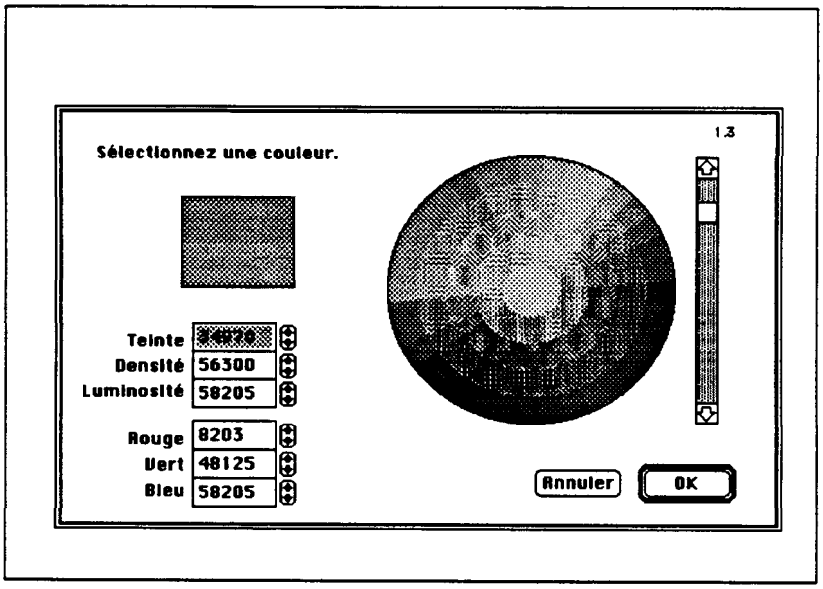

Figure 1: *cône des couleurs sur Mac*

TEX un appel à un autre opérateur PostScript : `currentcmykcolor` qui dépose sur la pile utilisateur les quatres couleurs actuelles.

2.3. Problèmes des rendus des niveaux de gris ... et des couleurs

Pour donner l'impression des couleurs avec une imprimante noir et blanc, on imprime des motifs constitués de points plus ou moins espacés et plus ou moins nombreux ou gros. Ce mécanisme (halftoning) autorise la création des ombres et des gris plus ou moins foncés.

Pour rendre la couleur sur une imprimante couleur, nous pouvons aussi utiliser le même procédé : juxtaposer les trois couleurs primaires du modèle. Nos yeux reconstituent alors une couleur solide. D'autre part, une impression sur une feuille blanche de points colorés extrémement petits donnera l'illusion d'une teinte claire.

Cependant, la mise côte à côte de motifs provoque des effets de *moiré* (aspect ondulé). Pour atténuer ce phénomène on oriente les points suivant des angles particuliers.

2.4. Technologie des imprimantes couleur

Les deux grands procédés de reproduction des couleurs sont aujourd'hui, le jet d'encre et le transfert thermique. Des interpréteurs PostScript fonctionnent aussi bien avec l'une des deux technologies. Par exemple, les imprimantes QMS ColorScript utilisent une technologie à transfert thermique ce qui nécessite un rouleau et un film de quatre couleurs (pour CMYK) facile à changer. L'impression se déroule alors en quatre phases pour le mode par transfert thermique, suivant les quatres couleurs fondamentales qui ont été séparées. Ces machines possèdent aussi un interpréteur PostScript (licence Adobe) directement intégré aux cartes électroniques

qui pilotent l'imprimante; l'impression est accélérée.

Dans le mode par transfert thermique, la feuille d'impression est coincée entre le film spécial d'une couleur (un film pour chaque couleur de base) et une tête d'impression. Cette dernière sélectionne les points particuliers, puis par chauffage, autorise le dépôt de la couleur sur la feuille constituée d'un papier spécial. Pour un mécanisme à jet d'encre, la feuille de papier s'enroule autour d'un tambour. Au fur et à mesure du déroulement du papier, il y a projection des quatres couleurs de base provenant de quatres réservoirs différents. A chaque projection, il faut maintenir propre la tête d'impression. Pour de plus amples renseignements le lecteur pourra consulter [Cheryl90] et [Bauman89] ainsi que le numéro spécial de la revue Publish' de mai 1989.

3. La couleur dans TEX

Pour notre part, nous avons travaillé à l'élaboration de macros où la spécification des couleurs s'effectue, à bas niveaux, aux moyens des opérateurs PostScript vus au paragraphe 2.2. Le choix de PostScript induit les conséquences suivantes :

1. utilisation de la commande TEX `special` (cf [Beebe89] et [Cerin89] pour un exemple d'utilisation d'inclusion de graphiques PostScript)

2. utilisation d'un driver dvi → ps gérant l'inclusion de textes PostScript. Nous nous servons de *dvips* de Tom Rokicki. Si vous ne disposez pas de ce translateur, en cas de portage, vous devrez sans doute retoucher au code de nos macros pour rendre la syntaxe du `special` compatible avec votre translateur. Ceci se réalise très simplement.

La brique de base est constituée par la macro `\begincolor{c}{m}{y}{k}` (on supose que l'on travaille avec la QMS ColorScript) qui se charge de la sauvegarde de la couleur courante, puis impose comme couleur actuelle la couleur formée par les quatres paramètres. La macro `\endcolor` restitue la couleur précédemment sauvegardée. Les macros que l'on va maintenant examiner consistent à placer convenablement des ordres `\begincolor` et `\endcolor` entre des unités syntaxiques ou graphiques.

4. Quelques Macros

La création des deux opérateurs de base `\begincolor` et `\endcolor` a permis d'obtenir d'autres macros selon deux approches :

- reprise de macros existantes.
- création de nouvelles macros.

En reprenant le source de macros déjà définies dans *latex.tex* et en y incluant les opérateurs `\begincolor` et `\endcolor` aux endroits appropriés, nous obtenons de nouvelles macros qui produisent un résultat similaire, mais en couleur! Cette méthode simple permet de définir, par exemple :

- une boîte rouge
- un souligné bleu
- un trèfle vert irlandais

L'introduction de la couleur permet également de créer de nouveaux *types de macros*. Nous nous sommes intéressés, jusq'à présent, à la définition de macros permettant la mise en couleur de chacun des caractères d'une chaîne en fonction de divers paramètres. Nous présentons, dans cet article, deux exemples de ce type :

La macro **echelon**

La couleur d'un caractère dépend uniquement de celle du caractère précédent. La gamme de couleur choisie se répète sur la longueur de la chaîne suivant une fonction en dent de scie. La plage de couleurs est spécifiée en déclarant pour chaque couleur primaire, une valeur minimale, une valeur maximale et un incrément.

Exemple de rendu avec la macro echelon

Une autre macro fonctionne suivant le même procédé, mais la gamme de couleurs spécifiée est alternativement croissante et décroissante.

Exemple de rendu avec une variante de echelon

La macro **degrade**

La couleur d'un caractère dépend de la couleur du caractère précédent mais aussi de la longueur de la chaîne. La plage de couleurs est, en effet, distribuée uniformément sur l'ensemble de la chaîne. Nous associons pour chaque couleur primaire, une valeur minimale et une valeur maximale, afin de former la gamme. L'incrément est calculé pour chaque couleur primaire et vaut :

$$\frac{increment\_couleur = couleur\_max - couleur\_min}{longueur\ de\ la\ chaine}$$

Exemple de rendu avec la macro degrade

Nous présentons maintenant les algorithmes et l'implantation en TEX de ces deux macros. Ceux-ci consistent à construire une liste contenant les caractères de la chaîne encadrés par des ordres de changement de couleur. Pour les décrire, nous employons une notation fonctionnelle avec les opérateurs classiques **car**, qui prend le premier élément d'une liste, **cdr**, qui retourne la liste privée de son premier élément et **cons** qui est le constructeur de listes.

4.1. La macro echelon

Nous supposons, dans un premier temps, que la couleur est représentée par un seul paramètre. Pour une chaîne de caractères c, **echelon** retourne une liste vide si c est vide; sinon, **echelon** retourne une liste constituée de l'instruction **\begincolor**, du premier caractère de c, de l'instruction **\endcolor** et de l'application récursive de **echelon** sur la chaîne privée de son premier caractère. La valeur de la couleur est incrémentée à chaque nouvel appel en contrôlant le dépassement éventuel de la borne supérieur.

```
def echelon:chaine → liste d'instructions
  echelon(c)← l;
    where l =
      if l = ∅ then ← ∅;
      else
        let l1 =
          (\begincolor{couleur}
          (car c)
          \endcolor);
        couleur ← couleur + increment;
        if couleur > couleurmax then
          couleur ← couleurmin;
        endif;
        ← (cons l1 echelon (cdr c));
      endif
```

```
;;
```

Ainsi avec *couleurmin* = 20, *couleurmax* = 50, *increment* = 15, *couleur* initialisée à *couleurmin* et c valant "exemple", on obtient la liste :

```
(
  \begincolor{20} e \endcolor
  \begincolor{35} x \endcolor
  \begincolor{50} e \endcolor
  \begincolor{20} m \endcolor
  \begincolor{35} p \endcolor
  \begincolor{50} l \endcolor
  \begincolor{20} e \endcolor
)
```

La conversion de cet algorithme dans la syntaxe TEX est loin d'être triviale. Il faut, en effet, pouvoir accéder à chacun des caractères d'une chaîne et gérer la récursivité. Le premier point est résolu en balisant le paramètre avec des crochets, par exemple. On trouvera une description de cette subtilité dans [Seroul89b]. Cette méthode n'autorise cependant que le passage d'un seul paramètre. Nous gérons également la récursivité comme en [Seroul89b].

Nous donnons ici le programme pour la version RGB (LaserWriter). Les variables globales sont :

1. la couleur de départ : `\sauverouge`, `\sauvevert`, `\sauvebleu`
2. la couleur maximale : `\maxrouge`, `\maxvert`, `\maxbleu`
3. la couleur courante : `\rouge`, `\vert`, `\bleu`
4. l'incrément pour chaque couleur : `\increrouge`, `\increvert`, `\increbleu`
5. la mémorisation de la liste d'instructions : `\file` (initialisé à {})

```
\def\ECHELON#1{
  \ifx#1]%
    \def\next{\file}%
  \else{%
    \global\advance\rouge by \increrouge%
    \global\advance\vert by \increvert%
    \global\advance\bleu by \increbleu%
    \ifnum\rouge>\maxrouge{%
      \global\rouge=\sauverouge%
    }\fi%
    \ifnum\vert>\maxvert{%
      \global\vert=\sauvevert%
    }\fi%
    \ifnum\bleu>\maxbleu{%
      \global\bleu=\sauvebleu%
    }\fi%
    \edef\file{%
       \file\begincolor{\number\rouge}
       {\number\vert}{\number\bleu}%
       #1\endcolor}%
       \let\next=\ECHELON%
  }\fi%
  \next
}
```

L'appel se fait par

```
\ECHELON[exemple]
```

Le test `\ifx#1]` est la condition d'arrêt : lorsque le crochet droit est rencontré, la macro retourne la variable `\file` qui contient la séquence 'instructions voulue. Dans le cas général, les valeurs des trois couleurs sont incrémentées et des tests vérifient que les bornes ne sont pas dépassées — auquel cas, les couleurs sont ramenées à leur valeur d'origine —. Le caractère (`#1`), encadré par les `\begincolor` et `\endcolor` appropriés, est ensuite ajouté à la variable `\file`. L'instruction `\let \next= \ECHELON` suivie de `\next` réalise l'appel récursif.

Cette macro n'est pas utilisable telle quelle, elle doit être encadrée d'une macro réalisant l'appel. De plus, nous avons greffé la possibilité d'imprimer plusieurs caractères dans la même couleur, ce qui permet d'étaler la plage de variation. L'initialisation des variables globales s'effectue au moyen de la nouvelle macro `\initcolor`. Les valeurs des couleurs varient entre 0 et 255 pour le standard RGB et entre 0 et 100

pour le mode CMYK. Il est donc recommandé de programmer une mise en couleur de la façon suivante :

```
{% debut de groupe
% rouge = bleu = vert = 20
% maxrouge = maxvert = maxbleu = 210
% chaque caractere change de couleur
\initcolor{20}{20}{20}{210}{210}{210}{1}
% increrouge = increvert = increbleu = 40
\initinc{40}40{40}
\echelon{exemple}
}% fin du groupe
```

4.2. La macro degrade

Au lieu de répartir la gamme des couleurs sur un intervalle de longueur prédéfini, cette macro la distribue sur l'ensemble de la chaîne. Elle procède en trois étapes :

1. calcul de la longueur de la chaîne (macro `longueur`).
2. calcul de l'incrément pour chaque couleur primaire.
3. appel de la macro `echelon`

On notera que le procédé de création de macros est incrémental : on se sert de la définition d'anciennes macros pour en générer de nouvelles.

L'appel à `longueur` retourne dans la variable globale `\lg` la longueur de la chaîne à colorier.

La définition de cette macro est :

```
\def\degrade#1{%
  \countdef\lg=220%
  \longueur[#1]%
  \ifnum\lg=0{%
    \typeout{macro degrade :%
             parametre vide}}
  \else{%
    \ifnum\lg>1{%
      %astuce des piquets et des barrieres
      \global\advance\lg by -1%
      \increrouge=\maxrouge%
      \global\advance\increrouge by%
          -\rouge%
      \global\divide\increrouge by \lg%
      \increvert=\maxvert%
      \global\advance\increvert by -\vert%
      \global\divide\increvert by \lg%
      \increbleu=\maxbleu%
      \global\advance\increbleu by -\bleu%
      \global\divide\increbleu by \lg%
     }\fi%
  \echelons[#1]%
  }\fi}%

\def\longueur#1{%
  \lg=0%
  \def\LONGUEUR##1{%
    \ifx##1]%
      \def\next{}%
    \else%
      \global\advance\lg by 1%
      \let\next=\LONGUEUR%
    \fi
    \next}
  \LONGUEUR}
```

5. Conclusion

Nous avons été amené à classer ces quelques macros suivant le type d'objet sur lequel elles agissent : entité syntaxique (`\echelon` et `\degrade` appartiennent à ce groupe) ou entité graphique (`\fboxcolor`, `\underlinecolor`, ...). Cependant, cette taxinomie n'est pas exhaustive si l'on pense, par exemple, au problème de la mise en couleur avec une unité de coloriage égale au point et non plus égale au caractère : on pourrait alors effectuer un dégradé de couleur sur l'intérieur d'une lettre. Peut-on vraiment agir à ce niveau dans TEX actuellement ?

L'écriture de ces macros soulève un problème qui apparaît peu approfondi dans la littérature : celui de la *récursivité* en TEX. Comment réaliser un appel récursif avec plusieurs paramètres ? Y a-t-il un empilement véritable des contextes comme dans les langages récursifs classiques ? TEX est-il bien adapté à l'implantation immédiate d'algorithmes fonctionnels comme ceux présentés dans cet article ? Si cela n'était pas le cas, que faudrait-il ajouter ?

Remerciements

Nous sommes vivement reconnaissants à la société QMS[1] qui nous a permis de valider nos implantations sur leur imprimante couleur Colorscript Model 10.

Références bibliographiques

[Adobe86] ADOBE SYSTEMS INCORPORATED, *PostScript Language Reference Manual*, Addison-Wesley, oct. 1986.

[Adobe87] ADOBE SYSTEMS INCORPORATED, *PostScript Language Tutorial and Cookbook*, Addison-Wesley, mar. 1987.

[Adobe88] ADOBE SYSTEMS INCORPORATED, *PostScript Language Program Design*, Addison-Wesley, feb. 1988.

[Bauman89] Keith BAUMAN, *Experiment In. Publish' review*, nov. 1989, page 76–80.

[Beebe89] Nelson H.F. BEEBE, « TEX and Graphics: The State of the Problem ». *Cahiers GUTenberg* numéro 2, Mai 1989, pages 13–53.

[Cerin89] Christophe CÉRIN, « GiTEX, Paps : deux logiciels manipulant PostScript et LATEX », *Cahiers GUTenberg* numéro 2, Mai 1989, pages 73–80.

[Cheryl90] CHERYL ENGLAND SPENCER, *Hot Wax Cold Ink. Macworld review*, feb. 1990, page 162–172.

[Hoadley90] Ellen D. *Hoadley*, « Investigating the Effects of Color », *Communication of the ACM*, feb. 1990, vol. 33 number 2, page 120–125.

[Knuth86] Donald E. KNUTH, *The TEXbook*, Addison-Wesley, 1986.

[Lamport86] Leslie LAMPORT, LATEX *user's guide & reference Manual*, Addison-Wesley, 1986.

[Seroul89a] Raymond SEROUL, *Le petit livre de TEX*, InterEdition, Mai 1989.

[Seroul89b] Raymond SEROUL, « Le coin des Gourous », *Cahiers GUTenberg* numéro 3, oct. 89, page 57–59.

[1] QMS : 1bis rue du Petit Clamart, Velizy Plus, 78147 VELIZY Cedex

La couleur dans TEX

- une boîte rouge
- un souligné bleu
- un trèfle vert irlandais ♣

Fonction	**Primitive**
$\dfrac{1}{1+x^2}$	$\operatorname{Arctg}(x)+C$

$$g = \sum_{n=0}^{\infty} x^n \left(\sum_{i=1}^{r} a_i (1+x)^{\mu_i} \right)$$

Vers l'introduction de la couleur dans TEX

Exemple de rendu avec la macro degrade