Stratégies pour inclure des graphiques dans des documents en LaTeX*

Klaus Höppner
Nieder-Ramstädter Str. 47
64283 Darmstadt
Germany
klaus.hoeppner@gmx.de

Résumé

Cette exposé présente des stratégies pour inclure des graphiques dans des documents en LaTeX. Il montre l'utilisation des *packages* graphiques standards de LaTeX, et introduit les différents formats de graphiques. Quelques outils externes de conversion des formats de graphiques sont discutés.

Présentation des formats

En général, les graphiques existent sous deux types de formats : les vectoriels et les matriciels [1]. Il y a différentes sortes d'images matricielles : sans compression (qui peut rendre vos fichiers vraiment énormes, tributaires de la résolution et de la profondeur de couleur, et je ne les traiterai pas ici), avec des méthodes de compression qui préservent complètement la qualité de l'image tout en réduisant la taille des données, et des méthodes de compression avec pertes *(lossy compression)* [2] qui provoquent une réduction conséquente de la qualité de l'image.

Voyons cela plus en détail :

Les graphiques vectoriels sont mis en place en dessinant ou en remplissant des objets géométriques tels que des lignes, des courbes de Bézier, des polygones, des cercles, etc. Les propriétés de ces objets sont stockées mathématiquement. Les graphismes vectoriels sont en général des dispositifs indépendants. Il est facile de les mettre à l'échelle ou de les tourner sans perte de qualité, car leur rastérisation [3] est effectuée par l'imprimante ou son pilote.

Les *bitmaps* sans pertes de compression mémorisent les informations de l'image sous forme de pixels, chaque pixel étant d'une couleur donnée. En principe, la qualité d'une image est meilleure avec une résolution et une profon-

* Cet article est paru en anglais : « Strategies for including graphics in LaTeX documents », *TUGboat*, vol. 26, No 1, p. 59-62, 2005. Il a été traduit en français par René FRITZ et reproduit avec l'aimable autorisation de l'auteur et de la rédaction de TUGboat.

1. Dont l'information est décrite sous la forme d'une matrice de points : *bitmap* en anglais. Le terme *bitmap* est un anglicisme à éviter.

2. Type de compression qui élimine les informations qui ne sont pas absolument nécessaires pour l'appréciation visuelle d'images numérisés afin d'obtenir le meilleur taux de compression possible.

3. Procédé qui consiste à convertir une image vectorielle en une image matricielle destinée à être affichée sur un écran ou imprimée par un matériel d'impression.

deur d'échantillonnage accrues (par exemple, les fichiers GIF utilisent une profondeur d'échantillonnage de 8 bits, ce qui conduit à indexer 256 couleurs différentes tandis qu'une image matricielle avec une profondeur d'échantillonnage de 24 bits peut avoir environ 16 millions de couleurs). La mise à l'échelle et la rotation des images matricielles produira une perte de qualité, et l'impression de telles images avec un dispositif ayant une autre résolution peut produire de mauvais résultats. La FIG. 1 montre la différence entre des images matricielles et vectorielles proportionnées.

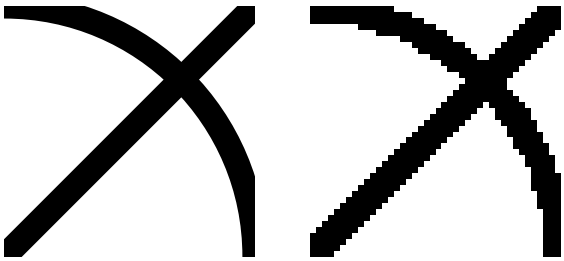


FIGURE 1 – *Zoom sur des parties d'images : vectorielle (à gauche) et matricielle (à droite).*

Les *bitmaps* avec pertes de compression utilisent le fait que l'œil humain distingue assez bien de petites différences de luminosité sur une zone relativement vaste, mais ne perçoit pas exactement la force d'une variation de luminosité à fréquence élevée. Pour cette raison, les composants d'une région de haute fréquence peuvent être réduits, conduisant à des fichiers de plus petites tailles. Ceci fonctionne bien pour des photos qui contiennent généralement des transitions en douceur des couleurs, mais pour les graphiques avec une frontière, des artefacts peuvent se produire, comme le montre la FIG. 2. Le format de graphiques le plus marquant qui utilise une compression avec pertes est JPEG.

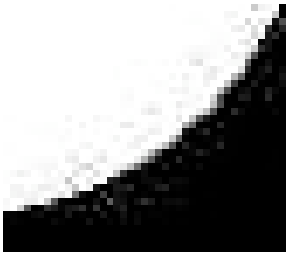


FIGURE 2 – *Image JPEG montrant des artefacts à la transition entre le noir et le blanc.*

Formats dans la pratique

Il existe de très nombreux formats de graphiques, et je ne m'intéresserai surtout qu'à ceux qui sont le plus souvent utilisés :

EPS ou *encapsulated PostScript* est surtout utilisé pour les graphiques vectoriels, mais peut aussi contenir des images matricielles.

PNG ou *Portable Network Graphics* a été introduit en raison du problème posé suite à la demande de brevet d'Unisys pour l'algorithme de compression utilisé pour le format GIF. Pour cette raison, il est souvent utilisé de nos jours sur des pages Web. PNG est un format matriciel qui prend en charge la compression avec ou sans pertes de qualité d'image.

JPEG sigle de *Joint Photographic Experts Group* est un format matriciel avec pertes de compression. Il est souvent utilisé pour les photographies (la plupart des appareils photo numériques produisent des fichiers JPEG).

TIFF ou *Tag(ged) Image File Format* est un format matriciel parfois utilisé pour des images de haute qualité, en partie parce qu'il supporte l'espace colorimétrique CMYK[4] important notamment pour l'impression commerciale.

4. Palette de couleurs CMJN : cyan, magenta, jaune, noir (CMYK en anglais : cyan, magenta, yellow, black).

Maintenant, la question est : Quel format doit-je utiliser pour quel but ? Bien qu'il n'y ait pas une véritable réponse à cette question, mon conseil est le suivant :

1. Pour les dessins (par exemple, des dessins techniques ou distribution de points (de données placettes)) utilisez des graphiques vectoriels. Ils vous donnent le maximum de liberté pour manipuler une image que, lors de son inclusion dans un document, vous avez souvent besoin de redimensionner pour l'adapter à votre présentation. En outre, ils sont indépendants du périphérique de sortie, et donc vous pouvez zoomer sur l'image dans votre visualiseur de document sans voir chacun des pixels.

   Les outils de dessin offerts par les distributions TEX – notamment PSTricks et MetaPost, peuvent habituellement produire de façon native des sorties en EPS. La plupart des programmes de dessin vectoriels comme xfig et Corel Draw offrent également une fonctionnalité d'exportation pour produire des sorties en EPS (même si parfois bogués).

2. Si vous êtes coincé avec des images matricielles, utilisez le format PNG pour les images avec de fortes transitions de couleur, comme le passage du noir au blanc.

3. Pour les photographies, vous pouvez utiliser le format JPEG dans la plupart des cas, depuis que la perte de qualité par compression est normalement imperceptible à l'impression. Sur la plupart des dispositifs, une résolution de 100 à 200 dpi [5] sera suffisante (rappelez-vous que la résolution de l'écran est normalement d'environ 75 à 100 dpi, et que les imprimantes couleur prétendent avoir à haute résolution, mais ont des tirages couleur lissés [6], de sorte que vous aurez du mal à remarquer la différence par rapport au format JPEG avec une meilleure résolution).

Package `graphics` de LaTeX

Depuis l'introduction de LaTeX, l'ensemble `graphics` fait partie du set de packages standard qui accompagnent la distribution LaTeX 2ε de base [1]. Il se compose de deux fichiers de style, `graphics.sty` et `graphicx.sty`. Alors que `graphics.sty` nécessite l'utilisation de `\scalebox` et de `\rotatebox` pour la mise à l'échelle ou la rotation de graphiques, l'extension de style `graphicx.sty` permet ces deux fonctions à l'aide du paquet `keyval`, qui fournit une interface pour la spécification des paramètres. En général, il n'y a pas de raison de ne pas toujours utiliser `graphicx.sty`.

Ainsi, la première étape consiste à charger le fichier de style `graphicx` après la déclaration `\documentclass{}` :

```
\usepackage{graphicx}
```

En fait, le compilateur TeX ne sait rien sur les graphiques, et leur inclusion est réalisée par le pilote DVI. Ainsi, le paquet `graphicx` doit faire deux choses :

1. trouver la *BoundingBox* [7] de l'image (ce qui peut être gênant lorsque vous avez par exemple un fichier EPS créé par une application qui a écrit un mauvaise commentaire [8] de *BoundingBox* – dans ce cas, il peut être utile de mettre la commande `\includegraphics` dans

5. *Dot per inch* en anglais ou « ppp » (point par pouce), indiquant la résolution d'un scanner, d'un imageur, d'une imprimante…

6. Par tramage.

7. Boîte des limites définie dans l'en-tête d'un fichier PostScript.

8. Comme en LaTeX, le caractère % introduit les commentaires situés dans l'en-tête d'un fichier PostScript.

une \fbox pour savoir ce que graphicx pense de la *BoundingBox*) ;

2. produire la commande \special convenable pour le pilote de sortie ; de sorte que l'utilisation de l'ensemble des graphiques soit dépendant du pilote.

Aujourd'hui, il existe deux principaux flux de travail pour la production de documents : utiliser latex pour produire un fichier DVI et dvips pour le convertir à PostScript, ou utiliser pdflatex pour produire un fichier PDF. La plupart des systèmes modernes de TEX sont configurés de manière à vérifier automatiquement si vous utilisez latex ou pdflatex et la production de dvips \spécial dans le premier cas et les commandes \pdfimage dans le deuxième cas. Donc, si vous utilisez un de ces flux de travail, vous ne devriez pas avoir à préciser explicitement votre sortie finale. Si vous utilisez un autre processus final, vous devez l'indiquer en option, par exemple :

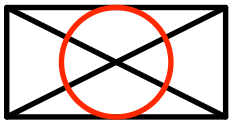
```
\usepackage[dvipsone]{graphicx}
```

pour le pilote Y & Y dvipsone, mais sachez que d'autres modes de sortie (backends) souvent ne prennent pas en charge la mise à l'échelle ou la rotation. Par exemple, les visualiseurs de DVI comme xdvi ou windvi essaient d'interpréter les commandes spéciales des dvips, mais les rotations ne peuvent pas être affichées correctement dans l'aperçu du DVI.

Pour inclure une image il suffit d'utiliser la commande :

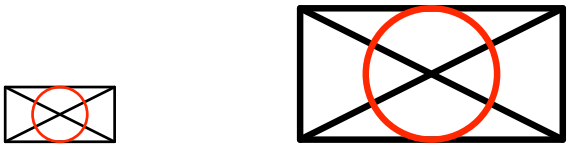
```
\includegraphics{image}
```

après que le paquet soit chargé.



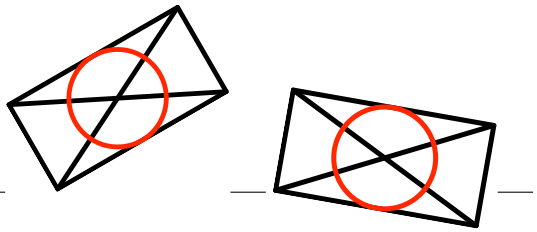
Notez qu'aucune extension n'a été donnée pour le fichier. L'explication sera donnée plus tard. Quand \includegraphics est utilisé sans options, comme indiqué ci-dessus, l'image a sa taille naturelle. En utilisant le style de graphicx, vous pouvez ajuster la taille de votre image par un facteur :

```
\includegraphics[scale=0.5]{image}
\includegraphics[scale=1.2]{image}
```



Une autre option permet la rotation d'une image :

```
\includegraphics[angle=30]{image}
\includegraphics[angle=-10]{image}
```



Les nombres positifs donnent des rotations dans le sens inverse des aiguilles d'une montre, les négatifs dans le sens contraire. L'origine de la rotation se situe dans le coin inférieur gauche de l'image, de tel sorte que le résultat de la rotation dans le sens des aiguilles d'une montre (voir ci-dessus) ne possède pas seulement une hauteur mais aussi une profondeur située sous la ligne de base représentée par les traits.

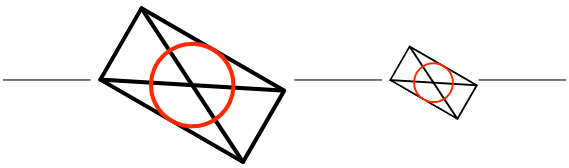
La taille des images ne peut pas seulement être proportionnée à un facteur donné, vous pouvez aussi bien en spécifier la hauteur ou la largeur :

```
\includegraphics[width=2cm]{image}
\includegraphics[height=1.5cm]{image}
```

height est la hauteur au-dessus de la ligne de base. Si votre image a une profondeur, vous pouvez alors utiliser totalheight, c'est-à-dire la somme de la hauteur et de la profondeur qui sera ainsi à dimension de la longueur donnée :

```
\includegraphics[angle=-30,
                 height=1cm]{sample}
```

```
\includegraphics[angle=-30,
                 totalheight=1cm]{sample}
```



Vous pouvez spécifier la largeur et la hauteur. Dans ce cas, votre image sera mise à l'échelle différemment dans direction horizontale et verticale, sauf si vous utilisez l'option `keepaspectratio` :

```
\includegraphics[width=1.5cm,
                 height=1.5cm]{sample}
\includegraphics[width=1.5cm,
        height=1.5cm,keepaspectratio]
                                {sample}
```

Notez que les appels des options d'angle, de largeur ou de hauteur sont sensibles à l'ordre dans lequel ils sont passés. Si vous spécifier d'abord l'angle cela fera tourner votre image en premier et ensuite l'image tournée sera mise aux dimensions désirées de largeur et de hauteur, tandis que si vous précisez d'abord la largeur et la hauteur, l'image de départ sera mise à ces dimensions en premier puis sera ensuite tournée.

Formats graphiques supportés

Pour rendre les choses un peu plus compliquée, `latex` avec `dvips` et `pdflatex` supportent différents formats de graphiques :

- `latex` + `dvips` : EPS ;
- `pdflatex` : PDF, PNG, JPEG, MPS.

Le TABLEAU 1 montre les moyens de convertir les formats graphiques standard dans des formats pris en charge. En particulier, la conversion de graphiques EPS utilisé avec `latex` + `dvips` au format PDF avec le flux de production `pdflatex` est très facile, il suffit d'exécuter le script Perl `epstopdf`, qui utilise Ghostscript pour convertir les fichiers EPS en PDF.

TABLEAU 1 – *Conversion des formats de graphiques supportés par* `latex` + `dvips` *et* `pdflatex`.

Origine	But	Moyen
`latex` + `dvips`		
EPS		directement supporté
PNG	EPS	ImageMagick/netpbm
JPEG	EPS	ImageMagick/netpbm
TIFF	EPS	ImageMagick/netpbm/ tif2eps
`pdflatex`		
PDF		directement supporté
EPS	PDF	`epstopdf`
PNG		directement supporté
JPEG		directement supporté
TIFF	PNG	ImageMagick/netpbm
TIFF	PDF	`tif2eps` + `epstopdf`

Cela explique aussi pourquoi il est généralement préférable de donner les noms des fichiers sans leur extension dans les commandes `\includegraphics`. Dans ce cas, le package `graphics` cherche automatiquement le format graphique supporté. Donc, si vous avez une image à la fois au format EPS et au format PDF (par exemple), vous pouvez utiliser aussi bien `latex` + `dvips` que `pdflatex` sans modifier le source.

Un autre cas particulier utile : inclure un fichier produit par METAPOST est également facile ; même si, techniquement, c'est un fichier EPS, il utilise seulement un petit ensemble de commandes PostScript. Ainsi, `pdflatex` peut supporter directement l'inclusion d'un tel fichier. La seule chose que vous ayez à faire est de changer l'extension

du fichier de sortie en `.mps` [9].

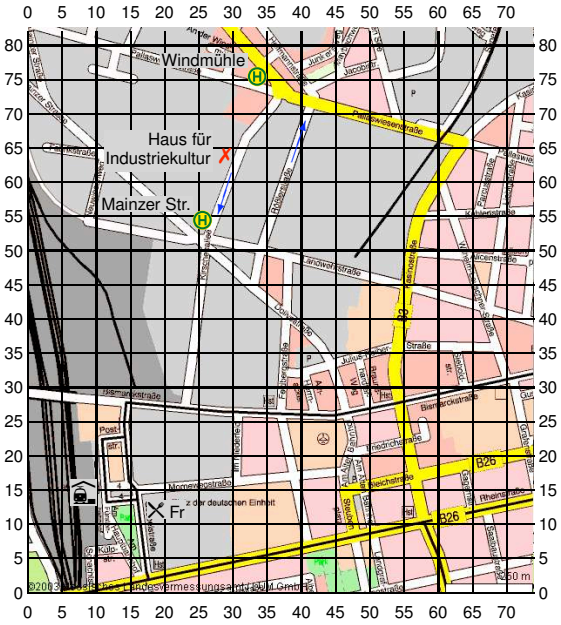


FIGURE 3 – *Carte avec des marquages produits avec overpic.*

Outils de conversion d'image

Il existe plusieurs outils pour la conversion des formats de graphiques, à la fois libres et commerciaux. Outre des outils libre basés sur GUI comme Gimp sur les systèmes Unix, il y a deux outils en ligne de commande disponibles pour Unix et Windows : ImageMagick [2] et netpbm [3].

ImageMagick permet de convertir des images directement, par exemple en tapant :

```
convert sample.gif sample.png
```

tandis que netpbm utilise le format pnm comme format intermédiaire :

```
giftopnm sample.gif | pnmtopng - > sample.png
```

Un autre bon outil est `tif2eps` par Bogus law Jackowski *et al.* [4] qui utilise Ghostscript pour convertir un fichier TIFF en EPS, par exemple :

```
gs -- tif2eps.ps sample.tif sample.esp -rh
```

qui produit un fichier EPS hexadécimal et compressé par RLE [10]. Selon mon expérience les fichiers EPS produits avec `tif2eps` sont plus petits que ceux produits par ImageMagick. De plus, les fichiers TIFF CMYK sont supportés sans problème.

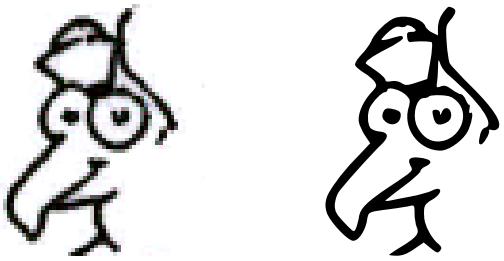


FIGURE 4 – *Zoom sur une image : matricielle (à gauche), vectorielle (à droite).*

Autres outils

Il existe de nombreux autres outils utiles. J'en citerai deux que j'utilise très souvent.

`overpic` est un package LaTeX écrit par Rolf Niepraschk [5]. Il inclut une image dans un environnement `picture`, vous donnant la possibilité d'ajouter de nouveaux éléments dans l'image avec les commandes normales de `picture`. La FIG. 3 montre la superposition de symboles et de textes sur une carte. Le code source de cette image ressemble à :

```
\usepackage[abs]{overpic}
...
\begin{document}
\begin{overpic}[grid,tics=5]{map}
\put(32,74){\includegraphics[scale=.3]
{busstop.mps}}
```

9. Les versions récentes de METAPOST permettent spécifier directement l'extension `.mps`. Les fichiers `.mps` sont aussi utilisables directement dans la chaîne `latex` + `dvips` : il suffit pour cela de préciser `\DeclareGraphicsExtensions{.mps,.eps,.ps} \DeclareGraphicsRule{mps}{eps}{*}{}` dans le préambule du document.

10. *run-length encoding*, appelé en français le codage par plages.

```
\put(32,77){\llap{\scriptsize
\colorbox{back}{Windm\"uhle}}}
\put(28,63){\small\textcolor{red}{%
\ding{55}}}
...
\put(17.5,11){\scriptsize\colorbox{%
back}{{\Pisymbol{ftsy}{65} Fr}}}
\put(6.3,13){\colorbox{back}%
{{\Pisymbol{ftsy}{68}}}}
\put(29.8,61.4){\color{blue}%
\vector(-1,-3){2}}
\put(38.6,63){\color{blue}%
\vector(1,3){2}}
\end{overpic}
\end{document}
```

potrace est un outil pour convertir une image matricielle d'un noir et blanc pure en une image vectorielle [6]. La FIG. 4 montre un exemple d'image matricielle convertie en image vectorielle.

Références

[1] CTAN:macros/latex/required/graphics

[2] http://www.imagemagick.org

[3] http://netpbm.sourceforge.net

[4] CTAN:support/pstools/tif2eps

[5] CTAN :macros/latex/contrib/overpic

[6] http://potrace.sourceforge.net

Traduit par René FRITZ
28 avril 2009
Les notes de bas de page ont été ajoutées par le traducteur

Version du 20 mai 2009 améliorée grâce aux corrections suggérées par Jacques André, Barbara Beeton, François Pétiard et Denis Roegel que je remercie vivement.