Figures in LaTeX

Contents

1 Introduction

This guide describes how to use external graphics in your LaTeX document. Section 2 covers the macros that you need. The remainder of the guide is about the graphics themselves, and how to get them into LaTeX: section 3 discusses types of graphics, with examples; section 4 lists some programs and methods for creating and editing graphics, and section 5 explains how to convert them into formats that LaTeX can handle.

This guide does not go into creating graphics with LaTeX macros. You can read about it in Oetiker *et al.* (2008), chapter 5. For a much more complete treatment, consult Mittelbach and Goossens (2004), chapter 10.

This edition of the manual reflects the 2008 RuG TeX Live installation.

2 The macros

Graphics inclusion is not built into the LaTeX core. You need to load a package which provides this facility[1]:

```
\usepackage{graphicx}
...
\includegraphics{mouse}
```

or, if you want to put the graphic inside a `figure` environment:

1. The examples here use the `graphicx` package. For old documents which use `epsf` or `psfig` you can load the `epsfig` package which reimplements the macros from these old packages in terms of the same core package, `graphics`, as the `graphicx` package. There should be no reason to keep using the old `epsf` or `psfig` packages.

```
\usepackage{graphicx}
...
\begin{figure}[h]
\centering
\includegraphics{blocks}
\caption{Example figure environment}%
  \label{figenv}
\end{figure}
See Figure~\ref{figenv}.
```

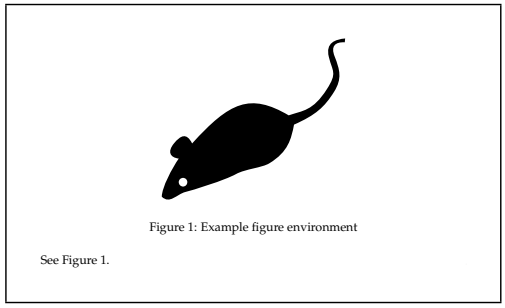


Figure 1: Example figure environment

See Figure 1.

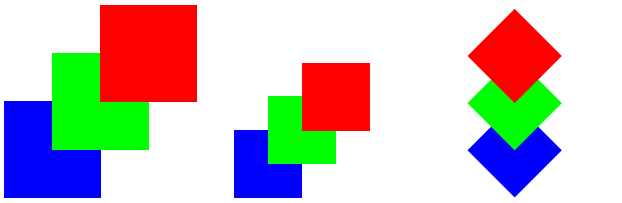
You can read more about the figure- and other floating environments in Oetiker *et al.* (2008) Section 2.12.

2.1 Optional parameters

You can scale and rotate your graphics with optional parameters:

```
\includegraphics[height=...,angle=...,...]{figure file}
```

The examples below illustrate scaling to a specified height or width, scaling by a factor and rotating:



`[width=1in]` `[height=.7in]` `[angle=45,scale=.5]`

2.2 Tips

- If you want to resize your graphic to text width, use `[width=\linewidth]`. `\linewidth` is the current text width. It takes e.g. column typesetting and indenting within lists into account.
- Most of the time, you can omit the extension, e.g. .eps or .pdf: LaTeX and pdflatex automatically pick file types they can handle; more about this in section 5 about file formats for LaTeX and pdflatex.
- Don't use an absolute path such as `c:/course/graphics/figure`. Absolute paths make it hard to move your documents to another location or another system. You can put your graphics in the same directory as your LaTeX source and then just use the filename, or you can put them in a subdirectory and use a relative path such as `graphics/figure`.
- Don't use the TeXnicCenter Insert Picture... menu item. This generates absolute paths and explicit extensions.

3 Types of graphics

Pictures can be defined in different ways, depending on the type of information they contain and on the software with which they have been created. Figures 1–6 contain some examples, each together with an enlarged detail.

Bitmaps are built up as a grid of pixels. Figures 1 and 2 show a photograph and a screenshot respectively. The grid structure is obvious in the enlarged detail.

Vector graphics are defined in terms of lines, circles, curves and other geometric shapes. They keep their sharpness at any scale; see figures 3–5.

Some file formats can contain both bitmapped and vector data. In figure 6, the bitmapped background becomes fuzzy when enlarged, but the text on top remains sharp.

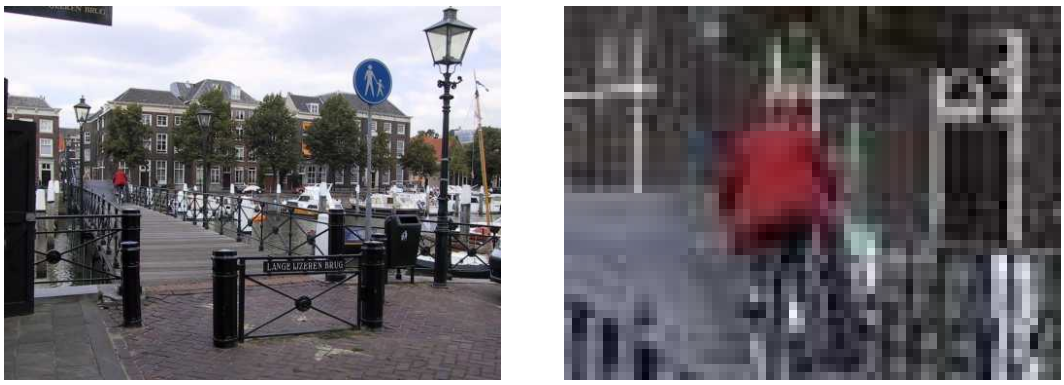


Figure 1. Bitmapped art: a photograph

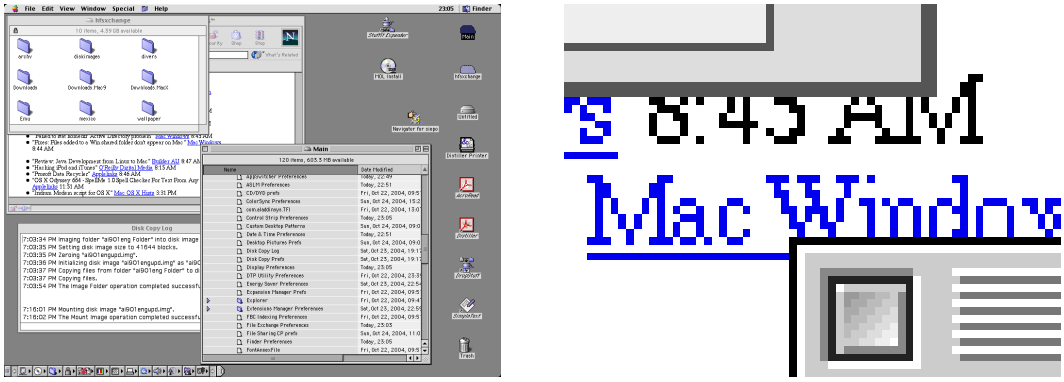


Figure 2. Bitmapped art: screenshot

3.1 Jpeg compression

High-resolution bitmapped files can get very big. There are various ways to reduce those file sizes.

Lossless compression works by storing information in a more compact way. A very simplified example: instead of enumerating a thousand identical white pixels one by one, you can say at once that the next one thousand pixels are white. Lossless compression can be quite effective when there are large areas of solid colors or regular patterns. Png is a lossless bitmapped format that can be processed directly by pdflatex.

Lossless compression doesn't work so well with photographic images. When we no longer insist on exactly preserving every bit of information and accept *lossy* compression, then very good results van be reached with jpeg. At medium to high quality settings, the loss of information is essentially invisible but the compression rate is easily 10:1. When you save in jpeg format you can usually make your own tradeoff between file size and image quality.

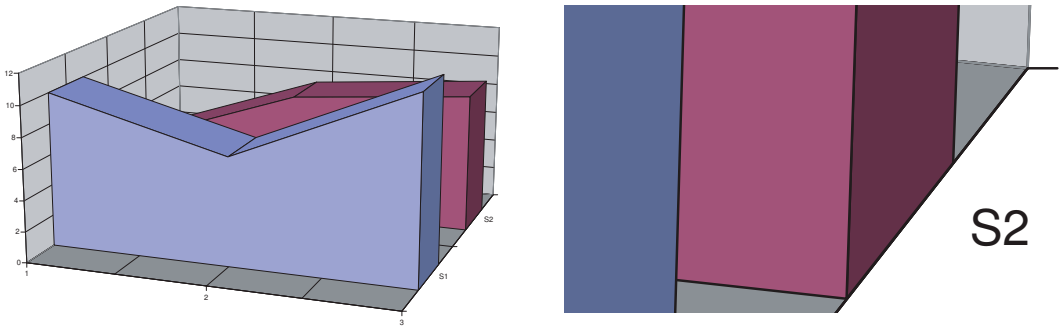


Figure 3. Vector art: Excel graph

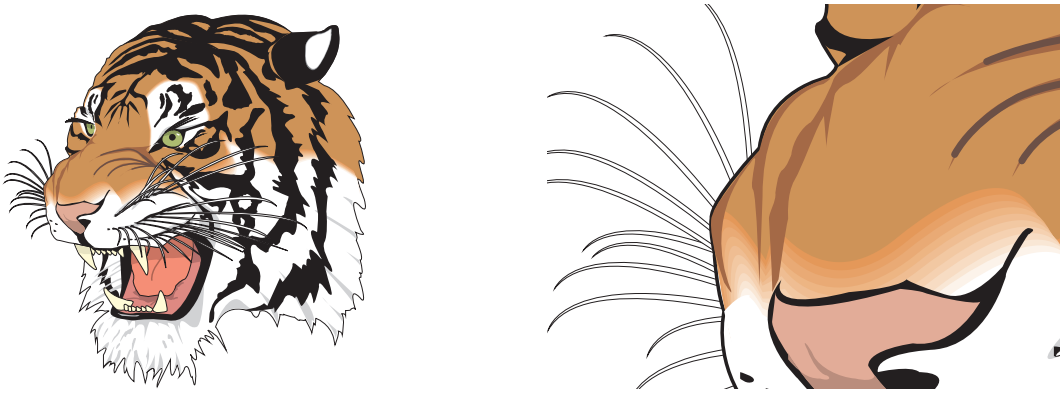


Figure 4. Vector art: adapted from a Ghostscript example file

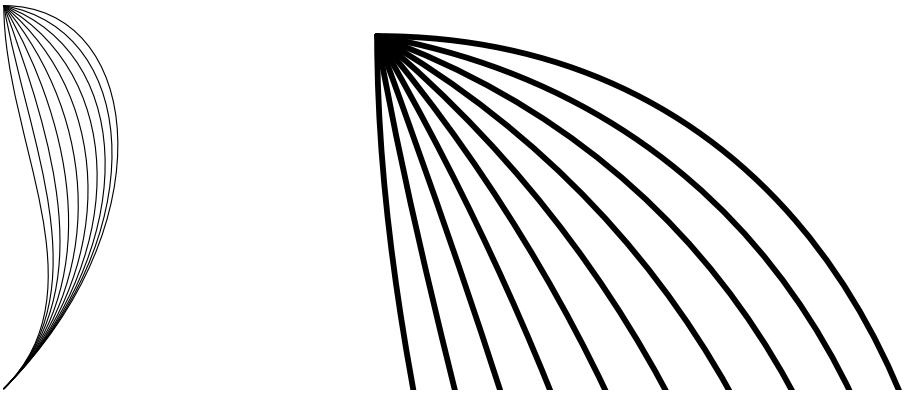


Figure 5. Vector art: generated with MetaPost

For non-photographic bitmapped images such as screenshots or logos, jpeg compression produces visible artifacts; see figure 7. To be fair, for the right image quality was set very low in order to make the artifacts more obvious. Compression isn't all that good either compared to png. Nevertheless, many people use jpeg compression for any bitmap that comes along, even when png would have been much better.[2]

2. Jpeg2000 (extension .jp2 or .j2c) is a successor of the jpeg format. This type of compression supports both lossy and lossless compression. Pdflatex can use pdf graphics which internally use jpeg2000 compression, but at the moment you cannot use jpeg2000 images directly.

Figure 6. Bitmapped and vector combined

Figure 7. Don't use jpeg compression for screenshots.

3.2 Bitmap resolutions

The resolution of a bitmap should be high enough to look sharp, but not higher than necessary, in order to keep file size and loading times within bounds.

For screen viewing, the ideal resolution would be exactly one bitmap pixel per screen pixel, but of course you may not know at what screen resolution and zoom level your document will be viewed.

For printing, good resolutions are:

Photographs. 150–300 DPI (dots per inch) depending on the output device. Most printers and imagesetters simulate grays and other tints with dot patterns or halftone screens. As a consequence, the effective printed resolution of a photograph is much lower than the resolution of the output device, no matter how high the resolution of the original photograph.

Charts and diagrams. 600–1200 DPI. 600 is enough to avoid visible blockiness. Higher resolutions can mean finer detail if the printer resolution is also high. But it is better to use a vector format.

Screenshots. Leave at original resolution.

Of course I am talking about resolution after scaling; if you print a 5 cm wide, 300 DPI image at a width of 10 cm then the effective resolution is 150 DPI.

If you only have a low-resolution picture and want to use it anyway, then use it as-is, because increasing resolution artificially won't make it better.

3.3 Problems with vector graphics

Missing fonts. If some standard fonts (Times, Helvetica, Courier, Symbol, Zapf Dingbat) are not embedded then epspdftk (see section 5.2) can help. If the pdf output target is set to prepress then fonts will be embedded during conversion to pdf. Epspdftk can convert back and forth between eps and PostScript on the one hand and pdf on the other. If other fonts are missing then you have a real problem.

Zero-width lines. If your graphic has some very thin lines, then try to zoom in to see whether the lines stay thin or really have some positive width. If they stay thin, try to set a thicker line width, e.g. 0.3pt, in the program which created the graphic.

Transparency and fill patterns. These features are not well supported by the graphics- and conversion software included in our TEX installation. Save or convert such graphics with a program that does support these features.

General fixes. You may be able to import and fix problems in a draw program: substituting fonts, changing line widths, replacing pattern fills with something else; see section 4 on draw programs.

As a last resort, you can convert your graphic to a bitmapped png file of sufficiently high resolution.

4 Creating graphics

In computer graphics, the term drawing means vector art and painting means bitmapped art.

4.1 Drawings and diagrams

Although these can be created in Word or PowerPoint, you may want to check out some free alternatives such as Inkscape, which is a specialized drawing program resembling CorelDRAW, or the Draw module of OpenOffice. OpenOffice.org is fairly good at reading and writing MS Office files, and has a pdf export option. In any case, investigate at an early stage how to get your drawings into LaTeX, see section 5.

4.2 Charts

These are normally generated as a byproduct from spreadsheets or mathematical or statistical or econometric software.

4.3 Bitmaps: paint programs and image editors

There exists a large selection of free and inexpensive paint programs and image editors. Paint is a very basic paint program which is included with Windows. IrfanView is a small, simple and free image viewer and converter. It is included in the RuG menu. If you need something more substantial, have a look at Gimp, which was originally developed for Linux. Yet another option is Adobe PhotoShop Elements, which may be easier to work with. It is commercial, but costs a fraction of its professional big brother, PhotoShop, which is the favorite of professional designers.

4.4 Screenshots

You can make screenshots without specialized software. The PrtScrn key will copy the entire screen to the clipboard, and Alt-PrtScrn the active window. All paint- and image-editing programs listed above can retrieve the screenshot from the clipboard, usually with Edit / Paste. All of them also have tools to crop images.[3]

5 Converting to LaTeX- and pdflatex-compatible formats

I'll only discuss graphics usage for the two most popular output options: generating PostScript with LaTeX and dvips, and generating pdf with pdflatex. The `graphicx` package will automatically detect these two cases. In the LaTeX-plus-dvips case, it will look for graphics in eps format, and in the pdflatex case it will look for graphics in pdf, jpg and png format.[4]

These are stable formats, with little room for ambiguity. Eps and pdf can contain just about any kind of graphics information. This small selection of file formats is therefore not a real limitation, but it can mean extra work.

5.1 Converting bitmaps to png and jpg

Many image editors and paint programs, including IrfanView, can convert to png and jpg.

3. With IrfanView this is not quite obvious, but if you just start to drag the cursor then it changes into a selection tool, and then you can apply Edit / Crop selection.
4. If you specify a non-default output driver as optional parameter to the graphicx package, then different graphics formats may be supported.

5.2 Converting between PostScript, eps and pdf

The Epspdf utility can convert between PostScript, eps and pdf, often with no loss of information. It can also remove unwanted borders (compute tight boundingbox option). Epspdf itself is the command-line version, and Epspdftk is a GUI program. You can find it via *Start / Programs / TeX Live 2008 / Utilities / PostScript-Pdf conversions*.

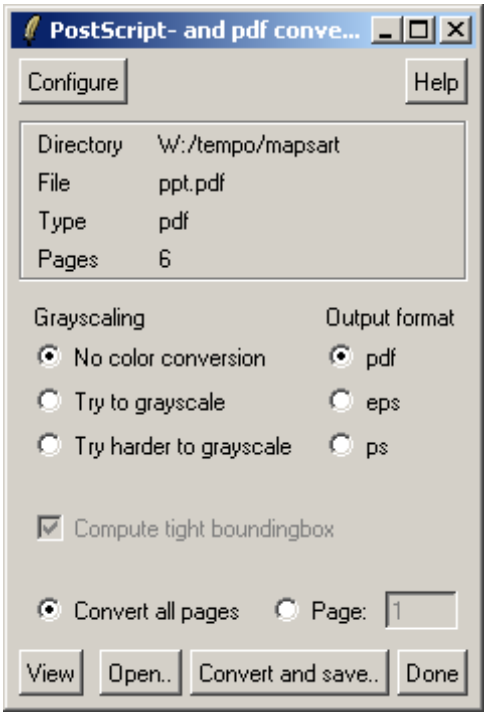


Figure 8. Epspdftk main screen

5.3 Converting bitmaps to eps

Pdflatex can handle the .png- and .jpg bitmapped formats, so for pdflatex there is no need for such a conversion.

Simpler paint programs and image editors such as Paint and IrfanView don't convert to eps or pdf, at least last time I checked, but more advanced ones, such as the free Gimp program and the commercial PhotoShop and PhotoShop Elements programs do.

5.4 Wmf, emf and the clipboard

Wmf and emf are native vector formats for Windows, and can be read by most Windows graphics programs. Another option is Wmf2eps. This shareware program does exactly what its name implies, and its conversions are quite accurate. It uses Windows' native PostScript printerdriver in the background.[5]

If you cannot even save as wmf or emf then again Wmf2eps can copy the clipboard content to a file.

5.5 Exporting eps and PostScript from Windows programs

A fairly reliable way to get eps or PostScript from a Windows program is by 'printing' to a PostScript file. This is approximately what the above wmf2eps program does.

For this, you need to have a PostScript printer driver. If you don't have one installed *and* you have Administrator privileges, go to 'Printers' and start up the Add Printer wizard. Choose Local Printer and uncheck automatic detection. As printer port, you can pick FILE, otherwise you

5. Most Windows PostScript drivers use the same core. Individual PostScript drivers add to this core a ppd- or PostScript Printer Definition file which is basically an enumeration of the printer's features and properties. Wmf2eps comes with its own ppd.

would have to manually check 'Print to File' anytime you print. A good choice for manufacturer and model would be 'Generic' and 'MS Publisher Imagesetter' respectively.

Pay attention to printer settings: in the Print dialog, click 'Properties', then 'Advanced' (on either tab). In the 'Advanced Document Settings' tree, navigate first to 'Document Options' then to 'PostScript Options'.

For 'PostScript Output Option' the default setting is 'Optimize for speed'. Change that to 'Optimize for Portability' or 'Archive Format', or, for single pages only, 'Encapsulated PostScript'. These non-default options presumably produce cleaner PostScript code, without printer-specific hacks. Experiment with this and other options if you run into problems (e.g. bad-looking screen output, or part of a graphic getting cut off, or conversion to bitmap).

What works best may depend on your Windows version: under Windows 2000, Archive worked best for me, but I have been warned that this option was unusable in older Windows versions.

Also look at the setting 'TrueType Font Downloading Option'. Pick 'Outline', not 'Automatic' or 'Bitmap'.

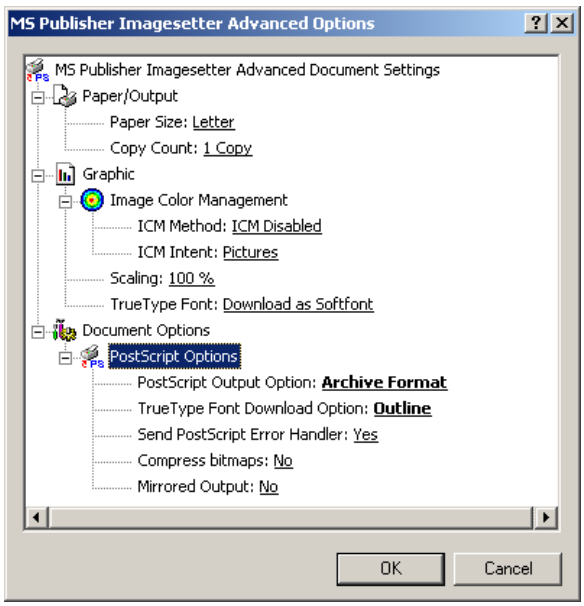


Figure 9. configuring a PostScript printer driver

References and urls

Epspdf and epspdftk, http://tex.aanhet.net/epspdf/, available from CTAN.

Gimp for Windows, http://gimp-win.sourceforge.net/, image editor.

Inkscape, http://inkscape.org/, draw program.

IrfanView, http://www.irfanview.com/, image viewer and converter for Windows.

Mittelbach, F. and M. Goossens, The LaTeX Companion. Addison-Wesley, second edn., 2004.

Oetiker, T., H. Partl, I. Hyna and E. Schlegl, The Not So Short Introduction to LaTeX2$_\varepsilon$, 2008, CTAN:info/lshort/, included in most free TeX distributions.

OpenOffice.org, http://www.openoffice.org/, office suite.

Wmf2eps, http://www.wmf2eps.de.vu/, available from CTAN.

Siep Kroonenberg
N.S.Kroonenberg at rug dot nl