Making 35 mm Colour Slides with SLITEX

Ken Yap

When I first read about SLITEX, and the colour layers feature, I envisioned being able to generate pretty 35 mm slides for talks from SLITEX input. Some conferences suggest that 35 mm slides be used on account of the large lecture theatres. I was disappointed to discover that slides meant page-size overhead transparencies, and colour meant that one had to obtain special colour materials.

When I saw a colour film recorder, I realized that this was the output device I was looking for. These devices can have resolutions as high as 4096 by 4096 pixels within a 24 mm by 36 mm frame. So I decided to write a dvi-to-35 mm slide converter.

1 Overview of SLITEX

To produce colour slides in SLITEX, a list of legal colours is declared in the root file thus:

```
\colors{red,green,yellow}
```

Next the slide file is included with the command sequence

```
\colorslides{slides.tex}
```

In the slide file, the list of colours that will be used on a given page is specified thus:

```
\begin{slide}{red,yellow}
```

Any of the legal colours may be used as command sequences or as environment arguments, similar to typeface or size commands. The colour names are meaningful to the author only; SLITEX has no way of checking what colours will be on the output media.

The reason for the separation of the root file and the slide file becomes clear when one examines the dvi output from a run of SLITEX: SLITEX iterates through the colour list with the slide file, so in the example above, first the red pages are generated, then the green pages and finally the yellow pages. Each page appears as many times as colours that it uses. The entities that appear on a layer page are those of the matching colour, or those that are outside the scope of any colour declaration, i.e. pervasive entities. This means that several colour layers have to be combined to yield a slide for each page.

2 Implementation

Our film recorder differs from a printer in one major respect: there is no font cache. This device takes an array of pixels and renders that onto the film. The converter has to do the page rasterization.

The backend comprises two programs: dviras and rascombine. The former converts dvi input into a monochrome Sun format *rasterfile*, and the latter combines monochrome rasters, assigning colours, to produce a colour rasterfile.

For dviras, I started with the Nelson Beebe suite of dvi converters because the rasterization code was already written, since Beebe's dvi family includes backends for dot matrix printers. The only hitch encountered was when I realized that the film recorder needed landscape format rasters. A 90° rotation had to be done on portrait pages. I did not want to use a naïve bit-by-bit copying algorithm on a raster of some 2 million bits, so I sought help. Alan Paeth of the University of Waterloo kindly showed me a fast rotation algorithm for 1-bit deep rasters which took under a second for a 2000 by 1000 raster on a Sparcstation I. Problem solved.

The second program, rascombine, assigns colours to each layer, then merges those layers into a single colour raster. The layer colours should be those in the colour list of the root file. A database converts colour names to red/green/blue values, so one can paint layers by name, e.g. vermilion. It is also possible to assign the colours of the background and of entities outside the scope of colour declarations. So "black" lettering can be made purple, if desired!

3 Example

```
dviselect 1 < root.dvi > 1.dvi
dviras 1.dvi
rascombine -b gray -o x.ras
    red,green,yellow 1.ras
```

Chris Torek's excellent dviselect program extracts all the layers of page 1 into 1.dvi. Next, the dviras command creates the rasterfile 1.ras with three consecutive rasters. Finally the rascombine command creates a colour rasterfile x.ras with a background of gray and with red, green and yellow text.

4 Results

We have just made our first batch of slides with this backend and they look gorgeous. One advantage of using slides for talks is that pictures of experimental results can accompany text slides and one does not have to flitter between the slide projector and the overhead projector.

I hope to try the backend on a colour printer next. This software will be available as part of Nelson Beebe's forthcoming release of his dvi driver family.

5 Conclusion

A more elegant and general solution is to merge the pages at the dvi level, inserting \specials between entities from different layers. This would have two advantages: the amount of data to be handled is smaller before rasterization, and non-raster devices can be handled by minor modifications to existing drivers. For example, the injected \specials could direct the output device to change colours. This approach would also produce a method for including non-TeX graphics in slides; any graphics accepted by the driver and printer could be included, whereas currently, with rasterfiles, only rasters can be merged into the final image, and that with difficulty.

◇ Ken Yap
Department of Computer Science,
University of Rochester
Internet: ken@cs.rochester.edu

An Easy Way to Make Slides With LaTeX

Georg Denk

Introduction

In the following, a style option for the article style will be presented which makes it easy to produce slides. It does not support an overlay structure as SLiTeX, but it enables the user to change an article to a sequence of slides by simply copying and rearranging. Therefore, it is merely an option for the article style and not a new style.

The design of this option fits into the philosophy of LaTeX: The logical structure of a text is created by the user and not the details of how to put this into a nice output.

The style option described here supports a standard layout which is nevertheless easy to change if necessary. The user is able to think in "normal" dimensions and font sizes as the proper magnification of the slide is done by the output device.

Some Commands

The style option eslides — which stands for *easy slides* and is used to distinguish this option from SLiTeX — makes several commands and environments available to the user. These are described in the following. This style option will not conflict with other style options such as 12pt or german. The option file is not listed here but is available from the author. Send a short request to the e-mail address given below.

The Magnification of the Slide

The \magnification command gives the global magnification of the slide. As the dvi file should be magnified by the same factor, only the quantities 1000, 1095, 1200, 1440, 1728, 2074 and 2488, resp., should be used. A magnification factor of 1440 or 1728 will give good results. Thus one says, e. g.,

```
\magnification{1440}
```

Leslie Lamport has written that LaTeX should not worry about a magnification of a document. As the output device, however, magnifies everything on the page but the sheet of paper, it is necessary that some of the lengths for the page layout such as \textwidth have to be scaled properly. This is done by the \magnification command. As this procedure is hidden away from the user, he is able to think in "real" dimensions. The previewing will show a correct picture of the slide, usually only somewhat smaller.

If the output device is not able to magnify the slide, it can be done with a photocopying machine, too. The calculation of the various lengths of the page differs as the fixed point of the mapping has changed from the offset point to the middle of the upper edge of the sheet. The eslides.sty file contains the necessary commands to handle this.

Page Layout

The eslides style makes the pagestyle myslide available to the user:

```
\pagestyle{myslide}
```

Every slide will have a head line and a foot line, separated from the text by a horizontal rule. The head consists of a centered running head which is set with the \markright or \markboth command. The foot line contains a logo, the "name of the conference" and the page number of the slide.

The logo can be anything, e. g., the logo of the university or of the company. In the following examples it is the logo of the Technical University of Munich which is drawn by some picture commands. The logo is changed by

\renewcommand{\logo}{*your logo*}

As the logo will not change often, probably the best place for the definition will be in the eslides.sty file. Perhaps, the logo can be taken from your special letter.sty.