LaTeX Graphics and Tables

David M. Auslander

Mechanical Engineeering
University of California at Berkeley

September 15, 2007

NOTE: This slide set has been done using LaTeX (documentclass: beamer). Where appropriate, the LaTeX code for each slide will follow the slide.
This is accomplished through the use of the "verbatim" environment,

```
\begin{verbatim}
% stuff printed literally ...
\ end{verbatim} % The space after \ is so
  % the end-verbatim command is not executed!
```

```
\begin{frame}[containsverbatim]
NOTE: This slide set has been done using {\LaTeX} (document

This is accomplished through the use of the ‘‘verbatim’’ en
\begin{verbatim}
\begin{verbatim}
% stuff printed literally ...
\ end{verbatim} % The space after \ is so
  % the end-verbatim command is not executed!
\ end{verbatim}


\end{frame}
```

Graphical Representations

- Bit Mapped
- Vector
- Hybrid (mix of bit mapped and vector)

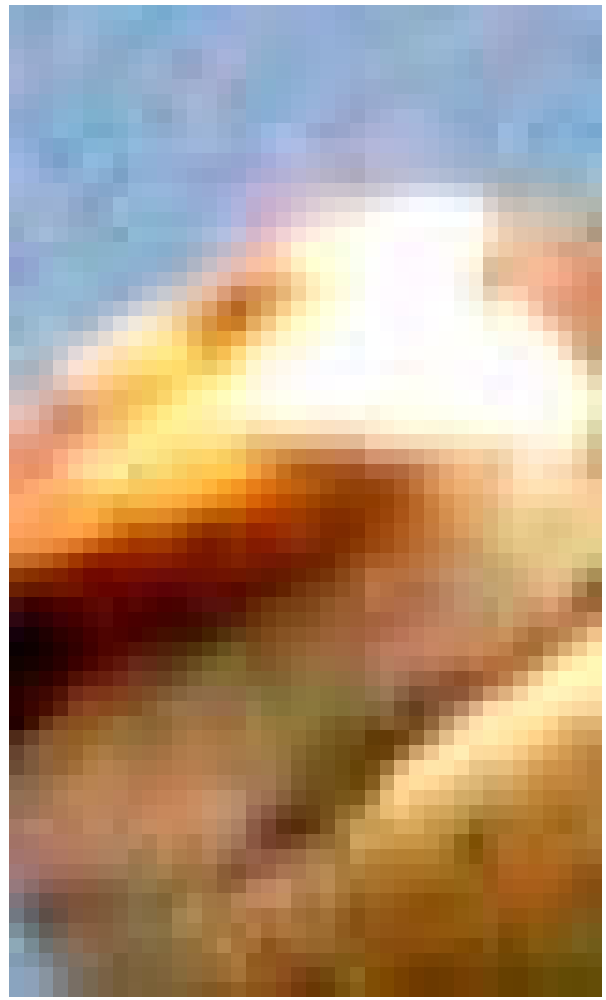
LaTeX Code

```
\begin{frame}{Graphical Representations}
\begin{itemize}
  \item Bit Mapped
  \item Vector
  \item Hybrid (mix of bit mapped and vector)
\end{itemize}
\end{frame}
```

Bit Mapped Graphics

Bit mapped graphics represent each pixel of the image explicitly.

The pelican looks OK, but enlarging its head shows the pixelation. Common file types for bit-mapped files: jpg, bmp, tif, png



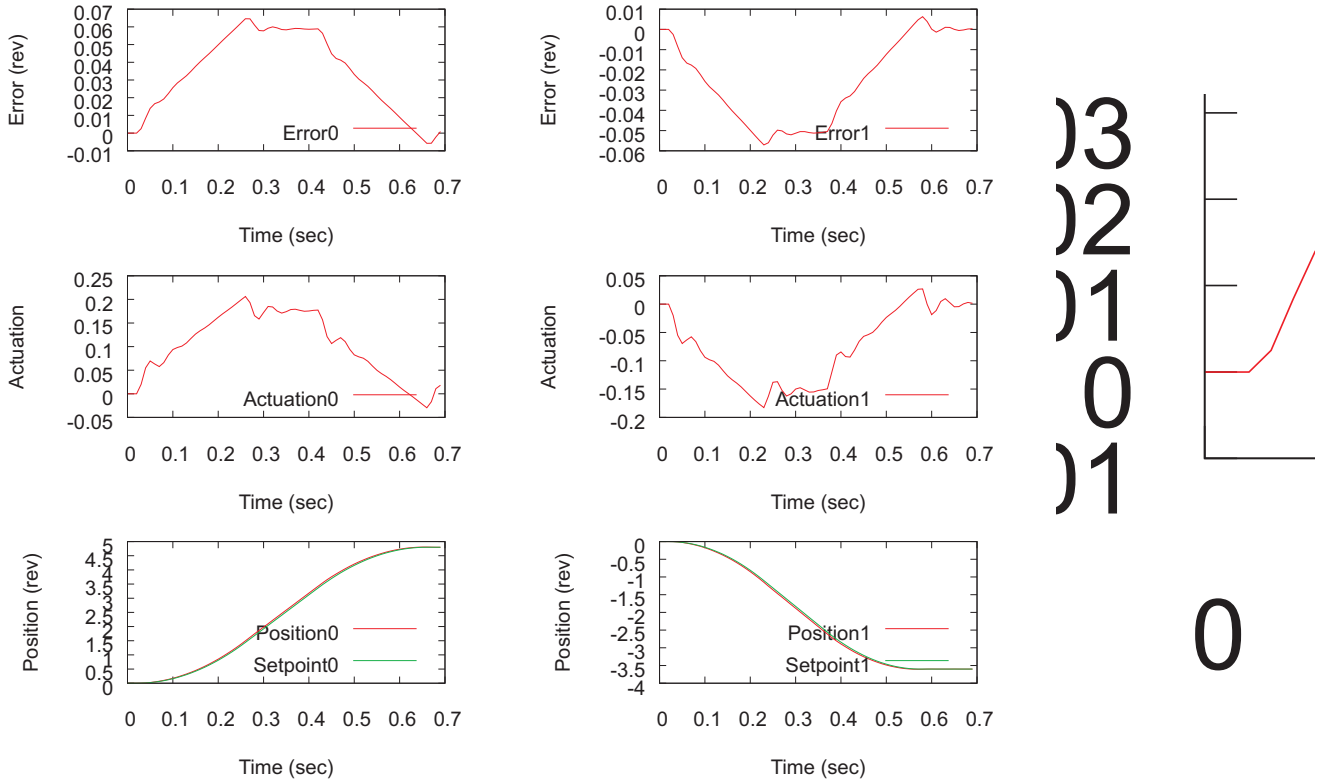
LaTeX Code

```
\begin{columns}
\column{1.5in}
Bit mapped graphics represent each pixel of the image expli
\vspace{0.2in}
The pelican looks OK, but enlarging its head shows the pixe

Common file types for bit-mapped files: jpg, bmp, tif, png
\column{1.5in}
\includegraphics[height=2.5in]{Pelican.pdf}
\column{1.5in}
\includegraphics[height=2.5in]{PelicanCropped.pdf}
\end{columns}
```

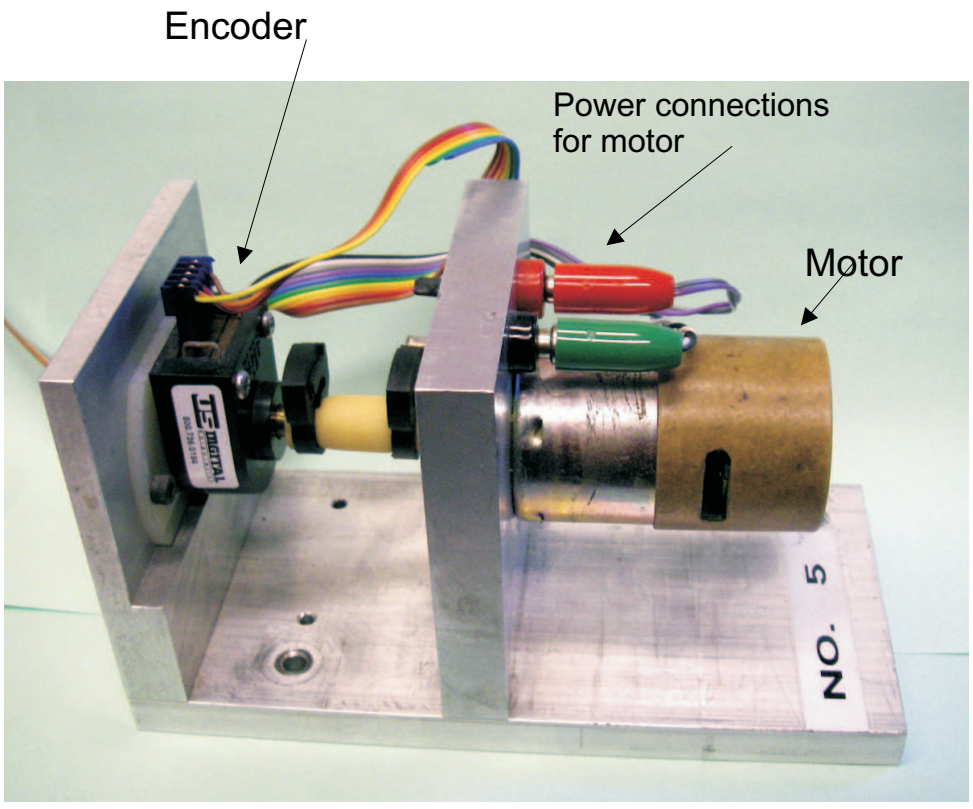
Vector (Object) Graphics

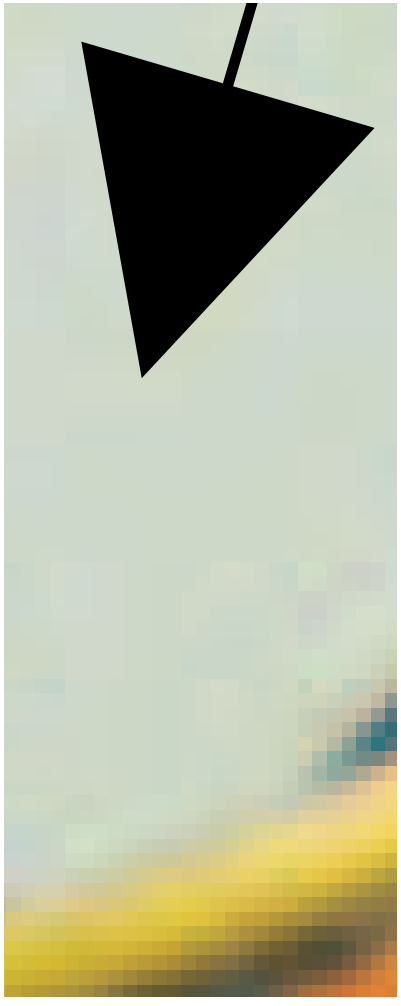
Vector graphics give instructions on how to draw the figure so images match the resolution of the display (or printer) regardless of cropping.



Hybrid Graphics

A mixture of vector and bit-mapped graphics in the same image.





Placing Graphics in a Document

- 'includegraphics' is the basic command for getting graphics into LaTeX documents
- The package 'graphicx' must be used to support it ('usepackage')
- Will place the figure at the indicated point in the document
- Has a variety of options for sizing and placement of graphics

Options for includegraphics

scale	Scale the figure by a factor
width	Specify width (with units: in, cm, pt)
height	Specify height (with units: in, cm, pt)
angle	Rotation angle (degrees, counterclockwise)
origin	Origin for the rotation (bl for bottom left, etc.)

Plus several other less frequently used options.

LaTeX Code

```
\begin{description}
  \item[scale] Scale the figure by a factor
  \item[width] Specify width (with units: in, cm, pt)
  \item[height] Specify height (with units: in, cm, pt)
  \item[angle] Rotation angle (degrees, counterclockwise)
  \item[origin] Origin for the rotation (bl for  bottom lef
\end{description}
\vspace{0.25in}
Plus several other less frequently used options.
```

Creating Graphical Objects

- Camera, scanner - bit mapped images
- Bit map image processors - Photoshop, Irfanview, many others
- Paint programs - bit mapped images
- Illustration programs (Adobe Illustrator, Corel Draw, ...) - vector and hybrid images
- "Office" suite simple graphics (Microsoft, Open Office, Wordperfect) – vector images plus some bit map image processing
- LaTeX has a "picture" environment for drawing commands. This predates most current computer graphics although some attempts at graphic front ends have been made.

LaTeX and Postscript

- LaTeX and Postscript grew up together. LaTeX depends very much on Postscript
- Three abstraction levels:

  Postscript Description language that is very close to the actual device that puts ink to paper (or pixels to screen)

  Tex Intermediate level to describe basic typography

  LaTeX High level, describes document structure
- LaTeX produces Postscript output for printing and uses it for graphics input

LaTeX Code

```
\frametitle{{\LaTeX} and Postscript}
\begin{itemize}
  \item {\LaTeX} and Postscript grew up together. {\LaTeX}
  \item Three abstraction levels:
  \begin{description}
    \item[Postscript]Description language that is very clos
    \item[Tex]Intermediate level to describe basic typograp
    \item[{\LaTeX}]High level, describes document structure
  \end{description}
  \item {\LaTeX} produces Postscript output for printing ar
\end{itemize}
```

EPS versus PDF in LaTeX

- EPS — encapsulated postscript. Pure PS but designed for single-page items with bounding information
- Early LaTeX required EPS for all graphics; produced a device independent output (DVI) and then postscript
- PDF — portable document format, also from Adobe (inventor of Postscript)
- It is possible to produce PDF output from EPS-based LaTeX
- But, hyper-reference information as in the table-of-contents is lost
- PDF-LaTeX was developed to preserve structural cross-reference information
- PDF-LaTeX requires graphics files in PDF (or JPEG or PNG for bit-mapped files but I don't know how portable LaTeX files with JPEG/PNG are)

Creating EPS and/or PDF Graphics Files: "Laundering" Graphics Files

- Graphics rarely starts as either EPS or PDF
- Other programs are often needed to get them into EPS or PDF form
- For EPS, in Windows I use Corel Draw (not free) or Open Office Draw (free) for converting as well as creating new graphics — many others would work also
- Clipboard transfer can get almost any graphic into one of these programs
- For PDF-Windows, more of a problem! Print-to-PSD can be used from almost any application (natively, with Acrobat (not free) or other programs (some free))
- PDF creation is usually aimed at full page. White space must be removed. Adobe Acrobat can do that. OO-Draw can also do it, but awkwardly. I don't know of others.

Exercise

1. Start with a file of random text (about 10 pages worth). Using document class 'report' divide it up into chapters and sections.
2. Use 'includegraphics' to add a graphics file (remember to `\usepackage{graphicx}`)
3. Use options to change its size (including very big and very small)
4. Move it around, top of page, middle, bottom
5. Add several more graphics. Put them close together, far apart.

LaTeX Code

```
\begin{enumerate}
  \item Start with a file of random text (about 10 pages wo
  \item Use ‘includegraphics’ to add a graphics file (remem
\verb!\usepackage{graphicx}!)
  \item Use options to change its size (including very big
  \item Move it around, top of page, middle, bottom
  \item Add several more graphics. Put them close together
\end{enumerate}
```

Floating Objects in a Document

- As last exercise shows, it is very difficult to do manual figure placement for any significantly sized document (over 2 pages!)
- LaTeX handles this with "floats"
- Floats are typographical objects that fit on a single page and do not have critical placement requirements
- Most common: figures, tables

Where Does LaTeX Put Floating Objects?

- Be prepared to give up control of where your figures will go!
- LaTeX will put them "where it thinks best"
- You have some influence, but not control
- Given its druthers, LaTeX likes to put floats at the top and bottom of pages
- If there are a lot of them, LaTeX will fill a page with floats
- The float can be placed ahead of where it is located in the text — probably the most annoying placement issue
- This works very well for reports, articles, papers, books, but not for brochures, flyers, newsletters, etc., where float placement is an important design decision

Parts of a Float

A typical float:

```
\begin{figure}[h]
\centering
\includegraphics[height=3.0in]{CVI-NewProject.pdf}
\caption{\label{fig:newproject}‘New Project’ Initial Dialog
\end{figure}
```

- Float options, ‘h’ in this case for ‘here’, give placement preference
- Centering causes the float object to be horizontally centered
- Figure floats most commonly are graphics files but they don't have to be. *Anything* can be put inside the figure environment.
- ‘Caption’ places the caption and also does automatic figure numbering (will be above or below the graphic depending on where you put it)
- ‘Label’ provides for a cross-reference

Float Placement Options

t	Put the figure at the top of a page
b	... bottom
p	on a page of all floats
h	here
!	add this to try to force a preference
default	Most say 'tbp' but some say the default is 'htbp' (??)

- Figures will be kept in order (can cause problems if one is especially large)
- Figure numbers will be assigned automatically
- Refer to a figure with the command: `\ref{fig:newproject}` where the argument is a label

Types of Floating Objects

- Most LaTeX documents get by with just Figures and Tables
- Custom floats can be defined for other types of floating objects
- Use the package "float"
- Set up the names, etc. ...

```
\floatstyle{ruled} % Caption style
\newfloat{program}{thp}{lop}
  % float name; placement options; file extension
\floatname{program}{Program}
  % name to use with numbering (Program 1)
```

Exercise

Redo the previous exercise using floats for figure placement.

Tabular Information

- Needed for a wide variety of circumstances
- LaTeX can produce excellent tables, but uses quite a bit of syntax to do it!
- Any valid LaTeX commands can be used for cell contents (although plain text is most common)
- The standard table environment ('tabular') can only handle tables that fit on a single page
- Tables can be put inside floats or not, as needed
- There are packages to handle long tables with similar syntax (but they can't be floated, at least, not easily)

Defining Columns

- Columns are defined as arguments to the tabular environment
- For simple tables, column widths are set automatically based on the widest item in that column
- For these types of tables the justification for the column and whether the columns should be separated by lines is all that needs to be specified:

```
\begin{tabular}{| c | l | r |}
```

- This specifies three columns, separated by lines (the bar character, `|`; use two bars, `||`, for double lines), center, left and right justified, respectively

Adding Content

- Items in a row are separated with ampersands; rows are ended with double backslash
- Here's a simple table based on the column specification from the previous slide:

City	Population, 1790	Population, 1990
New York	33,131	7,322,564
Baltimore	18,320	736,014
Richmond (VA)	3,761	203,056

LaTeX Code

```
\begin{tabular}{| c | l | r |}
\textbf{City} & \textbf{Population,  1790}
  & \textbf{Population, 1990}\\
New York & 33,131 & 7,322,564\\
Baltimore & 18,320 & 736,014\\
Richmond (VA) & 3,761 & 203,056\\
\end{tabular}
```

Adding Horizontal Lines

This needs some added horizontal lines ...

City	Population, 1790	Population, 1990
New York	33,131	7,322,564
Baltimore	18,320	736,014
Richmond (VA)	3,761	203,056

LaTeX Code

```
This needs some added horizontal lines ...
\vspace{0.5in}
\begin{tabular}{| c | l | r |}
\hline
\textbf{City} & \textbf{Population,  1790}
  & \textbf{Population, 1990}\\
\hline \hline
New York & 33,131 & 7,322,564\\
Baltimore & 18,320 & 736,014\\
Richmond (VA) & 3,761 & 203,056\\
\hline
\end{tabular}
```

Text Wrapping in Tables

- LaTeX will not wrap text in tables unless the width of the column is explicitly specified — the table will just go off the right margin!
- Instead of c, l or r, use p{width}, m{width}, or b{width} for vertical alignment as:

  p Vertical alignment at the top

  m Middle

  b Bottom

Table With Text Wrap

Here's what that table looks like with text wrap in columns two and three:

City	**Population, 1790**	**Population, 1990**
New York	33,131	7,322,564
Baltimore	18,320	736,014
Richmond (VA)	3,761	203,056

LaTeX Code

```
Here's what that table looks like with text wrap
in columns two and three:
\\[0.5in]  % the \\ adds a line break
% to avoid text wrapping around the side of the table
\begin{tabular}{| c | p{0.8in} | p{0.8in} |}
\hline
\textbf{City} & \textbf{Population,  1790}
  & \textbf{Population, 1990}\\
\hline \hline
New York & 33,131 & 7,322,564\\
Baltimore & 18,320 & 736,014\\
Richmond (VA) & 3,761 & 203,056\\
\hline
\end{tabular}
```

Floating Tables

- Tables are a predefined floating environment
- Use ‘table’ instead of ‘figure’
- Everything else is the same including captions, table numbering, cross-referencing

Tables, More

- There are several more things that can be done with tables:
- Multi-column and multi-row spanning
- Special formatting between rows (favorite example – lining up decimal points)
- Table width control
- Long tables (more than a page)

Exercise 1

Add the following table to the document you've been working with. Use it plain and within a float environment.

Item	**Credit**	**Number**	**Total Credit**
Assigned reading (weekly)	1	14	14
Small reports	2	7	14
Regular reports	10	5	50
Project report	1	20	20
Exam	10	2	20
Total			**118**

Grading Matrix

Exercise 2

Add a column at the right for comments and provide for text wrap in that column. Make up some comments to put there and then continue as in Exercise #1.