Strategies for including graphics in \LaTeX\ documents

Klaus Höppner

GuIT meeting 2005
1. Graphics Formats

2. \LaTeX\ graphicx package

3. Supported formats

4. Tools
Overview of graphics formats

Classification of graphics formats:

Vector graphics  set up by geometrical elements like lines, curves, polygons, circles, ...

Bitmap graphics  store image information as a set of colored pixels with a given resolution and color depth. Different compression methods exist:
- bitmaps with data compression only
- bitmaps with lossy compression
Overview of graphics formats

Classification of graphics formats:

**Vector graphics** set up by geometrical elements like lines, curves, polygons, circles, . . .

**Bitmap graphics** store image information as a set of colored pixels with a given resolution and color depth. Different compression methods exist:

- bitmaps with data compression only
- bitmaps with lossy compression
Overview of graphics formats

Classification of graphics formats:

**Vector graphics** set up by geometrical elements like lines, curves, polygons, circles, ... 

**Bitmap graphics** store image information as a set of colored pixels with a given resolution and color depth. Different compression methods exist:

- bitmaps with data compression only
- bitmaps with lossy compression
Overview of graphics formats

Classification of graphics formats:

**Vector graphics** set up by geometrical elements like lines, curves, polygons, circles, . . .

**Bitmap graphics** store image information as a set of colored pixels with a given resolution and color depth. Different compression methods exist:

- bitmaps with data compression only
- bitmaps with lossy compression
Overview of graphics formats

Classification of graphics formats:

**Vector graphics** set up by geometrical elements like lines, curves, polygons, circles, . . .

**Bitmap graphics** store image information as a set of colored pixels with a given resolution and color depth. Different compression methods exist:

- bitmaps with data compression only
- bitmaps with lossy compression
Example: vector drawing

- Vector drawings are fine for geometrical drawings
- Advantage: Easily scaleable
- Advantage: Optimal quality independent of resolution of output device
Example: vector drawing

- Vector drawings are fine for geometrical drawings
- Advantage: Easily scaleable
- Advantage: Optimal quality independent of resolution of output device
Example: vector drawing

- Vector drawings are fine for geometrical drawings
- Advantage: Easily scaleable
- Advantage: Optimal quality independent of resolution of output device
Example: vector drawing

- Vector drawings are fine for geometrical drawings
- Advantage: Easily scaleable
- Advantage: Optimal quality independent of resolution of output device
Example: bitmap

![Bitmap Image]

A bitmap with few colors and sharp borders

- Sometimes you have to use bitmaps when you don’t have a mathematical representation of your drawing (e.g. no data set for land and political borders in shown example)
- Disadvantage: Loss of quality when scaling or zooming
- Disadvantage: Loss of quality when image resolution doesn’t fit to resolution of output device
Example: bitmap

- Sometimes you have to use bitmaps when you don’t have a mathematical representation of your drawing (e.g. no data set for land and political borders in shown example)
- Disadvantage: Loss of quality when scaling or zooming
- Disadvantage: Loss of quality when image resolution doesn’t fit to resolution of output device

A bitmap with few colors and sharp borders
Example: bitmap

- Sometimes you have to use bitmaps when you don’t have a mathematical representation of your drawing (e.g. no data set for land and political borders in shown example)

- Disadvantage: Loss of quality when scaling or zooming

- Disadvantage: Loss of quality when image resolution doesn’t fit to resolution of output device
Example: bitmap

![A bitmap with few colors and sharp borders](image)

- Sometimes you have to use bitmaps when you don’t have a mathematical representation of your drawing (e.g. no data set for land and political borders in shown example)
- Disadvantage: Loss of quality when scaling or zooming
- Disadvantage: Loss of quality when image resolution doesn’t fit to resolution of output device
Example: photo

- A photograph has many colors (typically 16 mio) and smooth transitions
- No mathematical representation
- Again: Loss of quality when zooming into the photo (low resolution photo of big size)
Example: photo

- A photograph has many colors (typically 16 mio) and smooth transitions
- No mathematical representation
- Again: Loss of quality when zooming into the photo (low resolution photo of big size)
Example: photo

A photograph has many colors (typically 16 mio) and smooth transitions.

No mathematical representation.

Again: Loss of quality when zooming into the photo (low resolution photo of big size).
Example: photo

A photograph has many colors (typically 16 mio) and smooth transitions.

No mathematical representation.

Again: Loss of quality when zooming into the photo (low resolution photo of big size).
Comparison

- **vector drawing**
- **low resolution bitmap (pixels visible)**
- **artifacts in a bitmap with lossy compression**
Graphic formats in detail

**EPS**  encapsulated postscript can contain vector drawings and bitmaps

**PNG**  bitmapped portable network graphics format is a successor of GIF, supporting both compression with and without lossy compression

**JPG**  bitmap format with lossy compression, often used for photographs (e.g. digital cameras)

**TIFF**  a bitmap format often used for high quality DTP, supports CMYK color space
Graphic formats in detail

**EPS** encapsulated postscript can contain vector drawings and bitmaps

**PNG** bitmapped portable network graphics format is a successor of GIF, supporting both compression with and without lossy compression

**JPG** bitmap format with lossy compression, often used for photographs (e.g. digital cameras)

**TIFF** a bitmap format often used for high quality DTP, supports CMYK color space
Graphic formats in detail

**EPS**  encapsulated postscript can contain vector drawings and bitmaps

**PNG**  bitmapped portable network graphics format is a successor of GIF, supporting both compression with and without lossy compression

**JPG**  bitmap format with lossy compression, often used for photographs (e.g. digital cameras)

**TIFF**  a bitmap format often used for high quality DTP, supports CMYK color space
Graphics Formats

\LaTeX\ graphicx package

\begin{itemize}
\item \textbf{EPS} encapsulated postscript can contain vector drawings and bitmaps
\item \textbf{PNG} bitmapped portable network graphics format is a successor of GIF, supporting both compression with and without lossy compression
\item \textbf{JPG} bitmap format with lossy compression, often used for photographs (e.g. digital cameras)
\item \textbf{TIFF} a bitmap format often used for high quality DTP, supports CMYK color space
\end{itemize}
Guidelines

- For geometrical drawings (e.g. technical drawings, data plots) use a vector format like EPS or PDF
- If you have a bitmap with sharp borders, use PNG
- For photographs with high color depth and smooth transitions use JPEG (100–150 dpi are enough in most cases)
- In high quality DTP, use TIFF for photographs (especially if you need to support CMYK color space)
Guidelines

- For geometrical drawings (e.g. technical drawings, data plots) use a vector format like EPS or PDF
- If you have a bitmap with sharp borders, use PNG
- For photographs with high color depth and smooth transitions use JPEG (100–150 dpi are enough in most cases)
- In high quality DTP, use TIFF for photographs (especially if you need to support CMYK color space)
Guidelines

- For geometrical drawings (e.g. technical drawings, data plots) use a vector format like EPS or PDF.
- If you have a bitmap with sharp borders, use PNG.
- For photographs with high color depth and smooth transitions use JPEG (100–150 dpi are enough in most cases).
- In high quality DTP, use TIFF for photographs (especially if you need to support CMYK color space).
Guidelines

- For geometrical drawings (e.g. technical drawings, data plots) use a vector format like EPS or PDF
- If you have a bitmap with sharp borders, use PNG
- For photographs with high color depth and smooth transitions use JPEG (100–150 dpi are enough in most cases)
- In high quality DTP, use TIFF for photographs (especially if you need to support CMYK color space)
Including graphics in \LaTeX documents is supported by the packages \texttt{graphics} and \texttt{graphicx}.

\texttt{graphicx} is an extension of \texttt{graphics} supporting key-value-options for e.g. scaling and rotating.

Load \texttt{graphicx} package with
\begin{verbatim}
usepackage{graphicx}
\end{verbatim}

Modern \TeX systems assume \texttt{dvips} as backend when using \texttt{latex} as compiler and \texttt{pdftex} as backend when using \texttt{pdflatex}.

For other backends use
\begin{verbatim}
usepackage[backend]{graphicx}
\end{verbatim}
Including graphics in \LaTeX documents is supported by the packages `graphics` and `graphicx`.

`graphicx` is an extension of `graphics` supporting key-value-options for e.g. scaling and rotating.

Load `graphicx` package with

```
\usepackage{graphicx}
```

Modern \TeX systems assume `dvips` as backend when using `latex` as compiler and `pdftex` as backend when using `pdflatex`.

For other backends use

```
\usepackage[backend]{graphicx}
```
Including graphics in \LaTeX documents is supported by the packages \texttt{graphics} and \texttt{graphicx}.

\texttt{graphicx} is an extension of \texttt{graphics} supporting key-value-options for e.g. scaling and rotating.

Load \texttt{graphicx} package with

\begin{verbatim}
\usepackage{graphicx}
\end{verbatim}

Modern \TeX systems assume \texttt{dvips} as backend when using \texttt{latex} as compiler and \texttt{pdftex} as backend when using \texttt{pdflatex}.

For other backends use

\begin{verbatim}
\usepackage[backend]{graphicx}
\end{verbatim}
Including graphics in \LaTeX\ documents is supported by the packages \texttt{graphics} and \texttt{graphicx}.

\texttt{graphicx} is an extension of \texttt{graphics} supporting key-value-options for e.g. scaling and rotating.

Load \texttt{graphicx} package with

\begin{verbatim}
\usepackage{graphicx}
\end{verbatim}

Modern \TeX\ systems assume \texttt{dvips} as backend when using \texttt{latex} as compiler and \texttt{pdftex} as backend when using \texttt{pdflatex}.

For other backends use

\begin{verbatim}
\usepackage[backend]{graphicx}
\end{verbatim}
Including graphics in \LaTeX documents is supported by the packages `graphics` and `graphicx`.

`graphicx` is an extension of `graphics` supporting key-value-options for e.g. scaling and rotating.

Load `graphicx` package with

\begin{verbatim}
\usepackage{graphicx}
\end{verbatim}

Modern \TeX systems assume `dvips` as backend when using `latex` as compiler and `pdftex` as backend when using `pdflatex`.

For other backends use

\begin{verbatim}
\usepackage[backend]{graphicx}
\end{verbatim}
Including a graphics file

- You can include an image in its natural size with 
  \includegraphics{sample}

- Use options as key-value-pairs (graphicx):
  \includegraphics[key1=opt1,key2=opt2,\ldots]{sample}

- Common options are:
  
  - scale to scale the image by a factor
  - width to scale the image to fit a width
  - height to scale the image to fit a height
  - angle to rotate the image by an angle with the lower left corner as fix point (positive: counter-clockwise)
  - keepaspectratio scale uniquely in x- and y-direction even if both width and height are given
Including a graphics file

- You can include an image in its natural size with \includegraphics{sample}
- Use options as key-value-pairs (graphicx):
  \includegraphics[key1=opt1,key2=opt2,...]{sample}
- Common options are:
  - scale to scale the image by a factor
  - width to scale the image to fit a width
  - height to scale the image to fit a height
  - angle to rotate the image by an angle with the lower left corner as fix point (positive: counter-clockwise)
  - keepaspectratio scale uniquely in x- and y-direction even if both width and height are given
Including a graphics file

- You can include an image in its natural size with \includegraphics{sample}
- Use options as key-value-pairs (graphicx):
\includegraphics[key1=opt1,key2=opt2,...]{sample}
- Common options are:
  - \texttt{scale} to scale the image by a factor
  - \texttt{width} to scale the image to fit a width
  - \texttt{height} to scale the image to fit a height
  - \texttt{angle} to rotate the image by an angle with the lower left corner as fix point (positive: counter-clockwise)
  - \texttt{keepaspectratio} scale uniquely in x- and y-direction even if both width and height are given
Including a graphics file

- You can include an image in its natural size with \includegraphics{sample}
- Use options as key-value-pairs (graphicx):
  \includegraphics[key1=opt1,key2=opt2,...]{sample}
- Common options are:
  - \texttt{scale} to scale the image by a factor
  - \texttt{width} to scale the image to fit a width
  - \texttt{height} to scale the image to fit a height
  - \texttt{angle} to rotate the image by an angle with the lower left corner as fix point (positive: counter-clockwise)
  - \texttt{keepaspectratio} scale uniquely in x- and y-direction even if both width and height are given
Including a graphics file

- You can include an image in its natural size with 
  `\includegraphics{sample}`
- Use options as key-value-pairs (`graphicx`): 
  `\includegraphics[key1=opt1,key2=opt2,...]{sample}`
- Common options are:
  - `scale` to scale the image by a factor
  - `width` to scale the image to fit a width
  - `height` to scale the image to fit a height
  - `angle` to rotate the image by an angle with the lower left corner as fix point (positive: counter-clockwise)
  - `keepaspectratio` scale uniquely in x- and y-direction even if both width and height are given
Including a graphics file

- You can include an image in its natural size with \includegraphics{sample}
- Use options as key-value-pairs (graphicx): \includegraphics[key1=opt1,key2=opt2,...]{sample}
- Common options are:
  - \texttt{scale} to scale the image by a factor
  - \texttt{width} to scale the image to fit a width
  - \texttt{height} to scale the image to fit a height
  - \texttt{angle} to rotate the image by an angle with the lower left corner as fix point (positive: counter-clockwise)
  - \texttt{keepaspectratio} scale uniquely in x- and y-direction even if both width and height are given
Including a graphics file

- You can include an image in its natural size with \includegraphics{sample}

- Use options as key-value-pairs (graphicx):
  \includegraphics[ key1=opt1,key2=opt2,... ]{sample}

- Common options are:
  - scale to scale the image by a factor
  - width to scale the image to fit a width
  - height to scale the image to fit a height
  - angle to rotate the image by an angle with the lower left corner as fix point (positive: counter-clockwise)
  - keepaspectratio scale uniquely in x- and y-direction even if both width and height are given
Including a graphics file

- You can include an image in its natural size with \includegraphics{sample}
- Use options as key-value-pairs (graphicx): \includegraphics[key1=opt1,key2=opt2,...]{sample}
- Common options are:
  - **scale** to scale the image by a factor
  - **width** to scale the image to fit a width
  - **height** to scale the image to fit a height
  - **angle** to rotate the image by an angle with the lower left corner as fix point (positive: counter-clockwise)
  - **keepaspectratio** scale uniquely in x- and y-direction even if both width and height are given
Examples

\includegraphics[width=.3\linewidth]{sample}

\includegraphics[width=.3\linewidth,angle=20]{sample}

\includegraphics[angle=20,width=.3\linewidth]{sample}

\includegraphics[width=.3\linewidth,angle=20]{sample}
More examples

\includegraphics[width=1in,height=1in]{sample}

\includegraphics[width=1in,height=1in,keepaspectratio]{sample}
Supported graphics formats

- Support for graphics file formats and support for features like scaling and rotating depend on the used backend.
- Both `dvips` and `pdftex` support scaling and rotating.
- `dvips` supports EPS.
- `pdftex` supports:
  - PNG
  - PDF
  - JPEG
  - MPS (METAPOST output)
- Include images without extension and the backend driver will look for a supported format (so it’s easy to switch between `latex` and `pdflatex` without changing the document).
Supported graphics formats

- Support for graphics file formats and support for features like scaling and rotating depend on the used backend.

- Both `dvips` and `pdftex` support scaling and rotating.

- `dvips` supports EPS.

- `pdftex` supports:
  - PNG
  - PDF
  - JPEG
  - MPS (METAPOST output)

- Include images without extension and the backend driver will look for a supported format (so it’s easy to switch between `latex` and `pdflatex` without changing the document).
Supported graphics formats

- Support for graphics file formats and support for features like scaling and rotating depend on the used backend.
- Both `dvips` and `pdftex` support scaling and rotating.
- `dvips` supports EPS.
- `pdftex` supports PNG, PDF, JPEG, and MPS (METAPOST output).
- Include images without extension and the backend driver will look for a supported format (so it’s easy to switch between `latex` and `pdflatex` without changing the document).
Supported graphics formats

- Support for graphics file formats and support for features like scaling and rotating depend on the used backend.
- Both `dvips` and `pdftex` support scaling and rotating.
- `dvips` supports EPS.
- `pdftex` supports
  - PNG
  - PDF
  - JPEG
  - MPS (METAPOST output)
- Include images without extension and the backend driver will look for a supported format (so it’s easy to switch between `latex` and `pdflatex` without changing the document).
Supported graphics formats

- Support for graphics file formats and support for features like scaling and rotating depend on the used backend.
- Both `dvips` and `pdftex` support scaling and rotating.
- `dvips` supports EPS.
- `pdftex` supports PNG, PDF, JPEG, and MPS (METAPOST output).
- Include images without extension and the backend driver will look for a supported format (so it’s easy to switch between `latex` and `pdflatex` without changing the document).
Supported graphics formats

- Support for graphics file formats and support for features like scaling and rotating depend on the used backend.
- Both `dvips` and `pdftex` support scaling and rotating.
- `dvips` supports EPS.
- `pdftex` supports:
  - PNG
  - PDF
  - JPEG
  - MPS (METAPOST output)
- Include images without extension and the backend driver will look for a supported format (so it’s easy to switch between `latex` and `pdflatex` without changing the document).
Supported graphics formats

- Support for graphics file formats and support for features like scaling and rotating depend on the used backend.

- Both `dvips` and `pdftex` support scaling and rotating.

- `dvips` supports EPS.

- `pdftex` supports:
  - PNG
  - PDF
  - JPEG
  - MPS (METAPOST output)

- Include images without extension and the backend driver will look for a supported format (so it’s easy to switch between `latex` and `pdflatex` without changing the document).
Supported graphics formats

- Support for graphics file formats and support for features like scaling and rotating depend on the used backend.
- Both `dvips` and `pdftex` support scaling and rotating.
- `dvips` supports EPS.
- `pdftex` supports PNG, PDF, JPEG, and MPS (METAPOST output).
- Include images without extension and the backend driver will look for a supported format (so it’s easy to switch between `latex` and `pdflatex` without changing the document).
Supported graphics formats

- Support for graphics file formats and support for features like scaling and rotating depend on the used backend.
- Both `dvips` and `pdftex` support scaling and rotating.
- `dvips` supports EPS.
- `pdftex` supports:
  - PNG
  - PDF
  - JPEG
  - MPS (METAPOST output)
- Include images without extension and the backend driver will look for a supported format (so it’s easy to switch between `latex` and `pdflatex` without changing the document).
## Converting to a supported format

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Tool</th>
<th>Target</th>
<th>Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPS</td>
<td>✔</td>
<td>–</td>
<td>PDF</td>
<td>epstopdf</td>
</tr>
<tr>
<td>PDF</td>
<td>EPS</td>
<td>gs</td>
<td>✔</td>
<td>–</td>
</tr>
<tr>
<td>PNG</td>
<td>EPS</td>
<td>ImageMagick</td>
<td>✔</td>
<td>–</td>
</tr>
<tr>
<td>JPEG</td>
<td>EPS</td>
<td>ImageMagick</td>
<td>✔</td>
<td>–</td>
</tr>
<tr>
<td>TIFF</td>
<td>EPS</td>
<td>ImageMagick</td>
<td>or tif2eps</td>
<td>PDF tif2eps+epstopdf</td>
</tr>
</tbody>
</table>


**Tools for graphics conversion**

**ImageMagick** command line tool for graphics conversion and manipulation (changing size, gamma correction, ...), available for Unix and Windows

**netpbm** command line conversion tools, mainly on Unix but Windows binaries exist

**gs** Ghostscript is a PostScript interpreter available for various OS

**epstopdf** is a Perl script to convert EPS to PDF using gs

**tif2eps** by Bogusław Jackowski et al. uses gs to convert TIFF to EPS. Nice tool, also supporting CMYK color space.

**GUI tools** like Gimp, Adobe Photoshop, Corel Draw, ...
Tools for graphics conversion

**ImageMagick** command line tool for graphics conversion and manipulation (changing size, gamma correction, ...), available for Unix and Windows

**netpbm** command line conversion tools, mainly on Unix but Windows binaries exist

**gs** Ghostscript is a PostScript interpreter available for various OS

**epstopdf** is a Perl script to convert EPS to PDF using gs

**tif2eps** by Bogusław Jackowski et al. uses gs to convert TIFF to EPS. Nice tool, also supporting CMYK color space.

**GUI tools** like Gimp, Adobe Photoshop, Corel Draw, ...
Tools for graphics conversion

**ImageMagick** command line tool for graphics conversion and manipulation (changing size, gamma correction, ...), available for Unix and Windows

**netpbm** command line conversion tools, mainly on Unix but Windows binaries exist

**gs** Ghostscript is a PostScript interpreter available for various OS

**epstopdf** is a Perl script to convert EPS to PDF using gs

**tif2eps** by Bogusław Jackowski et al. uses gs to convert TIFF to EPS. Nice tool, also supporting CMYK color space.

**GUI tools** like Gimp, Adobe Photoshop, Corel Draw, ...
Tools for graphics conversion

- **ImageMagick** command line tool for graphics conversion and manipulation (changing size, gamma correction, ...), available for Unix and Windows

- **netpbm** command line conversion tools, mainly on Unix but Windows binaries exist

- **gs** Ghostscript is a PostScript interpreter available for various OS

- **epstopdf** is a Perl script to convert EPS to PDF using gs

- **tif2eps** by Bogusław Jackowski et al. uses gs to convert TIFF to EPS. Nice tool, also supporting CMYK color space.

- **GUI tools** like Gimp, Adobe Photoshop, Corel Draw, ...
Tools for graphics conversion

**ImageMagick** command line tool for graphics conversion and manipulation (changing size, gamma correction, ...), available for Unix and Windows

**netpbm** command line conversion tools, mainly on Unix but Windows binaries exist

**gs** Ghostscript is a PostScript interpreter available for various OS

**epstopdf** is a Perl script to convert EPS to PDF using gs

**tif2eps** by Bogusław Jackowski et al. uses gs to convert TIFF to EPS. Nice tool, also supporting CMYK color space.

**GUI tools** like Gimp, Adobe Photoshop, Corel Draw, ...
## Tools for graphics conversion

<table>
<thead>
<tr>
<th>Tools for graphics conversion</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ImageMagick</strong></td>
</tr>
<tr>
<td><strong>netpbm</strong></td>
</tr>
<tr>
<td><strong>gs</strong></td>
</tr>
<tr>
<td><strong>epstopdf</strong></td>
</tr>
<tr>
<td><strong>tif2eps</strong></td>
</tr>
<tr>
<td><strong>GUI tools</strong></td>
</tr>
</tbody>
</table>
Additional tools: potrace

- potrace is a tool to trace a pure black and white bitmap and produce a vector drawing
- potrace is a command line tool, binaries available for Unix, Mac OSX and Windows
- input formats are PBM, PGM, PPM
- output format is EPS
- Cool!
Additional tools: potrace

- potrace is a tool to trace a pure black and white bitmap and produce a vector drawing.
- potrace is a command line tool, binaries available for Unix, Mac OS X and Windows.
- Input formats are PBM, PGM, PPM.
- Output format is EPS.
- Cool!
Additional tools: potrace

- potrace is a tool to trace a pure black and white bitmap and produce a vector drawing
- potrace is a command line tool, binaries available for Unix, Mac OSX and Windows
- input formats are PBM, PGM, PPM
- output format is EPS
- Cool!
Additional tools: potrace

- potrace is a tool to trace a pure black and white bitmap and produce a vector drawing
- potrace is a command line tool, binaries available for Unix, Mac OS X and Windows
- input formats are PBM, PGM, PPM
- output format is EPS
- Cool!
Additional tools: potrace

- potrace is a tool to trace a pure black and white bitmap and produce a vector drawing.
- potrace is a command line tool, binaries available for Unix, Mac OSX and Windows.
- Input formats are PBM, PGM, PPM.
- Output format is EPS.
- Cool!
Example

original bitmap

traced vector drawing
Additional tool: package overpic

- \LaTeX{} package written by Rolf Niepraschek
- overlays an image with a \LaTeX{} picture environment
- you can add new elements to the picture (text, symbols, ...)

Example:

```latex
\begin{overpic}[grid,tics=5]{map}
  \put(32,74){\includegraphics[width=.3]\busstop.mps}
  \put(32,77){\llap{\scriptsize\colorbox{back}{Windm"uhle}}}
  \put(28,63){\small\textcolor{red}{\ding{55}}}
  ... 
  \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}
```
Additional tool: package overpic

- \LaTeX{} package written by Rolf Niepraschtk
- overlays an image with a \LaTeX{} picture environment
- you can add new elements to the picture (text, symbols, ...)

Example:

\begin{overpic}[grid,tics=5]{map}
\put(32,74){\includegraphics[scale=\textcolor{red}{.3}]{busstop.mps}}
\put(32,77){\llap{\textcolor{red}{\scriptsize \%}}\colorbox{back}{Windm"uhle}}
\put(28,63){\small \textcolor{red}{\ding{55}}} \ldots
\put(6.3,13){\colorbox{back}{\Pisymbol{ftsy}{68}}} \ldots
\put(29.8,61.4){\color{blue}\vector(-1,-3){2}} \ldots
\put(38.6,63){\color{blue}\vector(1,3){2}} \ldots
\end{overpic}
Additional tool: package overpic

- \LaTeX{} package written by Rolf Niepraschh
- overlays an image with a \LaTeX{} picture environment
- you can add new elements to the picture (text, symbols, ...)

Example:

\begin{overpic}[grid,tics=5]{map}
\put(32,74){\includegraphics[scale=.3]{busstop.mps}}
\put(32,77){\llap{\scriptsize%\colorbox{back}{Windm"uhle}}}
\put(28,63){\small\textcolor{red}{\ding{55}}}
...\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}
Additional tool: package overpic

- \LaTeX\ package written by Rolf Niepraschck
- overlays an image with a \LaTeX\ picture environment
- you can add new elements to the picture (text, symbols, ...)

Example:

```latex
\begin{overpic}[grid,tics=5]{map}
  \put(32,74){\includegraphics[scale=.3]{busstop.mps}}
  \put(32,77){\llap{\scriptsize\colorbox{back}{Windm"uhle}}}
  \put(28,63){\small\textcolor{red}{\ding{55}}}
  \...\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}
```
Example

original

with grid

final