

Side-by-side figures in L^AT_EX

Thomas Thurnherr

Abstract

Figures may be placed side-by-side for various reasons, such as comparing results generated under different conditions, because they are part of a bigger picture and therefore belong together, or simply to save vertical space in a document. L^AT_EX knows several ways to align multiple figures neatly. These can generally be divided into standard environments and the more sophisticated packages. This article serves to introduce the different methods and highlight their differences.

1 General remarks on placing figures side-by-side

There are several factors controlling how figures are placed side-by-side. One such is the spacing between figures. By default, the methods described below leave little or no space between two sub-figures. Therefore, horizontal space needs to be added manually (if required) using, e.g., the standard lengths `\quad` and `\qqquad` or the `\hspace` command.

Another factor is how many figures are placed next to each other, or (equivalently) when to break a line. L^AT_EX handles line-breaks automatically, implying that in order to place content side-by-side, one has to control the size of the figures. This is best achieved using `\linewidth` (or a fraction thereof), a dynamic length parameter which adapts to the available width for content. The examples given below illustrate its usage. To force a line-break, it is sufficient to end the paragraph by adding a blank line and L^AT_EX will start a new line.

This article introduces three packages: `subfig`, `subfigure` and `subcaption`. These packages offer many more options than the bare basics described here. They all come with extensive documentation available on your system, as part of the T_EX distribution, or online at CTAN (<http://ctan.org>).

The examples below all show how to arrange figures side-by-side. However, all methods work similarly with tables.

2 The minipage environment

The `minipage` environment is the most basic, and often sufficient, method to place figures side-by-side. Since `minipage` is not a floating environment, all figures have to go inside the `figure` floating environment. L^AT_EX will determine the optimal position for the `figure` environment, which can be influenced through the optional parameter.

The example below illustrates how to align two figures side-by-side using the `minipage` environment:

```
\begin{figure}[ht]
\centering
\begin{minipage}[b]{0.45\linewidth}
\includegraphics...
\caption{Happy Smiley}
\label{fig:minipage1}
\end{minipage}
\quad
\begin{minipage}[b]{0.45\linewidth}
\includegraphics...
\caption{Sad Smiley}
\label{fig:minipage2}
\end{minipage}
\end{figure}
```

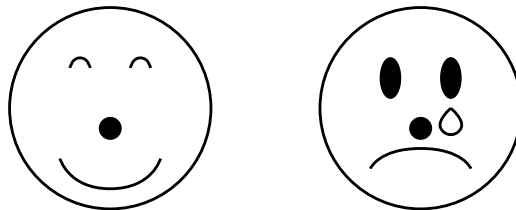


Figure 1: Happy Smiley

Figure 2: Sad Smiley

The `minipage` environment works with figures, tables, lists, and paragraphed text as well as a mix of these content types. This fact implies, however, that the `minipage` environment does not primarily serve to align figures or tables, which is why specific packages like `subfigure` have been developed, providing additional figure- and table-specific functionality.

3 The subfigure package

The `subfigure` package is the oldest of a series of packages implementing commands for placing figures and tables side-by-side. It provides support for captioning and labeling of the sub-figures and sub-tables, which is missing in the `minipage` environment. After loading the package in the preamble, sub-figures and sub-tables are created using:

```
\usepackage{subfigure}% in preamble

\subfigure[⟨lof entry⟩][⟨sub-caption⟩]{%
  ⟨figure⟩}
\subtable[⟨lot entry⟩][⟨sub-caption⟩]{%
  ⟨table⟩}
```

To show the `subfigure` commands in context, here is a complete example aligning four figures side-by-side to illustrate a line break:

```

\begin{figure}[ht]
\centering
\subfigure[Neutral Smiley]{%
  \includegraphics...
  \label{fig:subfigure1}}
\quad
\subfigure[Blush Smiley]{%
  \includegraphics...
  \label{fig:subfigure2}}

\subfigure[Sleepy Smiley]{%
  \includegraphics...
  \label{fig:subfigure3}}
\quad
\subfigure[Angry Smiley]{%
  \includegraphics...
  \label{fig:subfigure4}}
%
\caption{Main figure caption}
\label{fig:figure}
\end{figure}

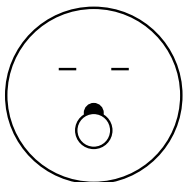
```



(a) Neutral Smiley



(b) Blush Smiley



(c) Sleepy Smiley



(d) Angry Smiley

Figure 3: Main figure caption

3.1 Labeling and referencing with subfigure

Labeling sub-figures can be done by placing a standard label inside the sub-figure command where the figure is loaded, as shown in the code above. For referencing, the package provides two commands: the standard `\ref`, and an additional `\subref` command. `\ref` produces the main figure and the sub-figure number (for example 3(a)), whereas `\subref` only produces the sub-figure number (for example (a)).

3.2 Adding sub-captions to the lists of figures and tables with subfigure

By default, sub-captions are not added to the list of figures (lof) and list of tables (lot). However, the package provides a simple solution to add them to the respective list by setting the value of the counter `lofdepth`, `lotdepth` respectively, to 2 (default: 1).

```

\setcounter{lofdepth}{2}
\setcounter{lotdepth}{2}

```

3.3 hyperref and subfigure

The `subfigure` package supports using `hyperref` to link references and list entries with figures and sub-figures. However, when `\subref` is used, the link jumps to the main caption or sub-caption rather than the figure, which is not desirable. The packages need to be loaded in the correct order, with `hyperref` being last.

3.4 Deprecation of subfigure

The `subfigure` package was marked obsolete or deprecated as it was replaced by `subfig`. This means that the package is neither further developed nor maintained. However, conflicts and other potential issues are well documented and as long as this is kept in mind, nothing speaks against its usage.

4 The subfig package

The more recent `subfig` package was derived from `subfigure`. Therefore, the syntax is very similar, with one exception: It does not distinguish between figures and tables; both are produced by using the `\subfloat` command inside the desired environment.

```

\usepackage{subfig}% in preamble

\subfloat[⟨lof entry⟩][⟨sub-caption⟩]{%
  ⟨figure⟩}
\subfloat[⟨lot entry⟩][⟨sub-caption⟩]{%
  ⟨table⟩}

```

The code below shows the beginning and end of the second example using the `subfig` package (the output is identical, so is not reproduced). The only change is to use `\subfloat` instead of `\subfigure`.

```

\begin{figure}[ht]
\centering
\subfloat[Neutral Smiley]{%
  \includegraphics...
  \label{fig:subfig1}}
\quad
\subfloat[Blush Smiley]{%
  ...
\caption{Main figure caption}

```

```
\label{fig:figure}
\end{figure}
```

4.1 Labeling and referencing with subfig

Similar to `subfigure`, the `subfig` package also implements the standard `\ref` and the `\subref` commands, producing the figure plus sub-figure labels or the sub-figure label only.

4.2 hyperref and subfig

Similar to `subfigure`, the `subfig` package supports the `hyperref` package. And again, `hyperref` needs to be loaded after `subfig`, and references to a sub-figure using `\subref` jump to the caption rather than the sub-figure.

4.3 Adding sub-captions to the lists of figures and tables with subfig

The `subfig` provides a slightly more convenient way to automatically add sub-captions to the `lof` and `lot`. It is sufficient to load the package with the `lofdepth` and `lotdepth` options:

```
\usepackage[lofdepth, lotdepth]{subfig}
```

5 The subcaption package

The `subcaption` package is the most recent of the three packages discussed here. The syntax is somewhat different from the two other packages: handling the size of figures is defined by the figure-enclosing environment, rather similar to `minipage`.

```
\usepackage{subcaption}% preamble

\begin{subfigure}[\langle position \rangle]{\langle width \rangle}
  \langle figure \rangle
\end{subfigure}%
\begin{subtable}[\langle position \rangle]{\langle width \rangle}
  \langle table \rangle
\end{subtable}%
```

And again, the same example, this time using the commands provided by the `subcaption` package:

```
\begin{figure}[ht]
\begin{subfigure}[b]{.45\linewidth}
  \centering
  \includegraphics...
  \caption{Neutral Smiley}
  \label{fig:subcaption1}
\end{subfigure}%
\quad
\begin{subfigure}[b]{.45\linewidth}
  \centering
  \includegraphics...
```

```
\caption{Blush Smiley}
...
\caption{Main figure caption}
\label{fig:figure}
\end{figure}
```

5.1 Labeling and referencing with subcaption

The package handles referencing the same way the two previous packages did, by providing the `\ref` and the `\subref` commands. The former produces a combination of the main plus the sub-label, whereas the latter produces the sub-label only.

5.2 Adding the sub-captions to the lists of figures and tables with subcaption

To add sub-captions to the lists of figures and tables, it suffices to load the package with the option `list=true`:

```
\usepackage[list=true]{subcaption}
```

5.3 hyperref and subcaption

The `subcaption` package is fully compatible with `hyperref`. The `hyperref` package needs to be loaded second and correctly links references and list entries with figures, tables, sub-figures, and sub-tables.

6 The columns environment in beamer

The presentation document-class `beamer` implements its own environment, called `columns`, for side-by-side content, in addition to `minipage`. The results with `columns` and `minipage` are almost exactly the same. Therefore, it is more of a personal preference which one to use. An example for `minipage` was given at the beginning of this article. To end, here is an example of how to place two figures side-by-side using the `columns` environment in `beamer`:

```
\begin{frame}{Frame title}
\begin{columns}
  \begin{column}{0.45\textwidth}
    \includegraphics...
  \end{column}
  \begin{column}{0.45\textwidth}
    \includegraphics...
  \end{column}
\end{columns}
\end{frame}
```

◇ Thomas Thurnherr
 texblog (at) gmail dot com
<http://texblog.org>