The movie15 Package

Alexander Grahn
a.grahn@web.de

13th October 2005

Abstract

A LaTeX package for inclusion of movies, sounds and 3D objects into PDF documents with PDF-1.5/1.6 compatibility.

Contents

1 Introduction

The way multimedia content, i. e. movies and sounds, is included into PDF has changed with Adobe's PDF specification, version 1.5. PDF-1.5 supports a larger variety of movie and sound formats, limited only by the number of plugins available for Adobe Reader®. With PDF-1.6, support for 3D objects, stored in the U3D file format, has been added. See Table 1 for a list of viable media formats. The specification allows media file contents to be completely embedded into the PDF output, thus producing selfcontained PDF documents.

This package provides an interface to embed movies, sounds and 3D objects into PDF documents for use with LaTeX as well as pdfLaTeX. Media file contents is incorporated into PDF output by default. This is done either directly during source processing using pdfLaTeX or during conversion from Postscript to PDF using Ghostscript's `ps2pdf`.

The final PDF output can be viewed with Windows® and Mac OS® versions of Adobe Reader. However, embedded media file data can also be extracted and saved to disk from within Readers which support file attachments. This makes PDF documents a little more portable to Readers which do not ship with a multimedia plugin.

2 Installation

The file 'movie15.sty' should be stored in a place where LaTeX can find it.

3 Using the package

Invoke the package by putting the line

```
\usepackage[<package options>]{movie15}
```

to the preamble of your document, i. e. somewhere between `\documentclass` and `\begin{document}`.

'movie15' takes the option '`3D`'. It enables the 3D feature from the PDF-1.6 specification. Inclusion of 3D files into PDF is discussed separately in section 5.2.

Currently, there are no further options.

If PDF is generated via DVI and Postscript by the usual `latex` → `dvips` → `ps2pdf` sequence of commands, the 'graphicx' and 'hyperref' packages are required and must be loaded *after* 'movie15'.

In the case of pdfLaTeX, the 'graphicx' and 'hyperref' packages are not required.

Note, that more than one run may be necessary to resolve internally created cross-references. Appropriate warnings will be emitted in such cases.

4 Requirements

pdfLaTeX, version $\geq$ 1.20, is needed for direct PDF output.

If the package option '`3D`' is set, 'movie15' loads Michael Mehlich's 'fp' package for fixed point arithmetic. Since it does not belong to the core packages of most TeX distributions it must be installed prior to selecting the '`3D`' option.

Adobe Reader, version 6, is required for playing movies and sounds, version 7 for rendering embedded 3D content. Unfortunately, only Windows and Mac versions of Adobe Reader provide the necessary plugins.

5 The user interface

5.1 Media inclusion

Movies, sounds and 3D objects are embedded into the document using the command

```
    \includemovie[<options>]{<width>}{<height>}{<media file>}
```

Unless left empty (cf. option '`text`') the `<width>` and `<height>` arguments must be given in valid TEX dimensions. Horizontal and vertical dimensions of the media clip are scaled independently to fit `<width>` and `<height>`. `<media file>` specifies the file name of the media clip. If the media file is embedded as part of the final PDF output, which is the default, it may reside wherever TEX or Ghostscript search for input files, depending on the PDF producing method.

Below, common options to `\includemovie` are listed. Options specific to embedding of 3D content are discussed separately in section 5.2.

`attach[=false]`

By default, the embedded data stream is re-used as embedded file attachment. This makes 'movie15' a little more portable to PDF Readers which do not have a multimedia plugin because it gives the user the opportunity to extract the media file and to save it to disk for later playback with an external player. However, support of file attachments is required. Re-use as file attachment can be suppressed with option '`attach=false`'. Note, that '`attach=false`' is a poor method to prevent the user from obtaining a copy of the media file. There are several (free) tools for decompressing and extracting stream objects from PDF files.

`autopause`

Pause playing when the page is closed.

`autoplay`

Put this option to the options list if you want the playback of the media clip to be started automatically after the page has opened. This is especially useful if there are multiple movie clips on the current page to be played at the same time. Also resumes playback of previously paused media. In the case of embedded sound being started this way, `<width>` and `<height>` can be safely set to '0pt' since no further user interaction is necessary to start playing.

`autostop`

Stop playing when the page is closed. This is default behaviour. Override with option ‘`continue`’.

`autoresume`

Resume previously paused playback when the page is opened again.

`continue`

Continue playback in background on premature page change. However, paused media remain paused. So do media that have reached the end or that still show the poster.

`controls[=true|false]`

Player specific controls are displayed during playback of the media clip, if available. For movies and sounds the default is ‘`false`’; controls are not shown. For 3D content, the default is ‘`true`’, since ‘`false`’ would make the 3D content non-interactive, i. e. not responsive to mouse interaction.

`depth=<depth>`

Specifies how far the media display area should extend below the bottom line of the running text. Any valid TEX dimension is accepted.

`externalviewer`

Opens and plays the media in an external application.

`inline[=false]`

Media file *data* are embedded into PDF output by default. This can be suppressed with option ‘`inline=false`’.

`label=<label spec>`

The media is labelled with `<label spec>` such that it can be referenced by the `\movieref` command elsewhere in the document. The `\movieref` command creates a hyperlink to either stop, pause, resume the media, to play it with different settings, such as frame rate or volume, or to show a different part of it. `<label spec>` may consist of any sequence of letters, digits or punctuation characters.

`mimetype=<mimetype>`

According to the PDF specification, the viewer application needs to know the MIME type of `<media file>`. Package ‘movie15’ tries to make a guess from the file name extension. If, however, a media file is of unknown type or has a non-standard file name extension, its MIME type must be explicitly specified using this option. See Table 1 for a list of known file name extensions.

`mouse[=true|false]`

Enable mouse interaction: Clicking onto the media (giving it the focus) pauses the playback while clicking outside resumes it. Alternatively, once the media has got the focus, repeated pressing of key 'P' on the keyboard switches between Play and Pause (at least with some players).

`palindrome[=true|false]`

The media is played forward and backward, if supported by the media player. In combination with option '`repeat`' this forward-and-reverse playback repeats as many times as specified. Each complete forward and reverse playback counts as one repeat.

`poster`

The first frame of the movie or the frame at the beginning of the movie section specified by the '`startat`' and '`endat`' options is shown.

`poster=<image>`

Inserts still image from file `<image>` to be shown when the media clip is not activated. The file type of `<image>` depends on the output driver: PS or EPS when using LaTeX together with `dvips`; PDF, PNG or JPEG when using pdfLaTeX. This option is obsolescent and should not be used any longer. Instead, use option '`text`' if a still image from an external file is to be inserted. See option '`text`' for a possibility to scale the media display area according to the poster image dimensions.

`rate=<rate>`

`<rate>` specifies the playback rate, where 1 is normal playback, 0.5 is halfspeed, 2 is doublespeed, -1 is normal speed in reverse etc.

`repeat[=<repeats>]`

Playback of the media clip will be repeated `<repeats>` times. Without argument this option causes the player to repeat forever.

`startat=<offset>`
`endat=<offset>`

Start/stop playback at the specified offsets. `<offset>` may be an absolute time or frame value, or a named marker, or a marker plus a time or frame. In the latter case, time or frame values are regarded as relative to the named marker. Support of markers, time and frame offsets is media type dependent. Media incompatible offsets are ignored. The following syntax applies to `<offset>`:

```
time:<time in seconds>
frame:<frame number>
marker:<quoted string>
```

Some `<offset>` examples:

```
    time:20.5
    frame:100
    {marker:"Chapter 1", time:60}
```

`text=<text>`

`<text>` is typeset onto and centred within the media display area which has been specified by the `<width>` and `<height>` arguments as well as the '`depth`' option. If either or both of the `<width>` and `<height>` arguments are left empty, any unspecified dimension of the media display is scaled to match the corresponding dimension, i. e. width and height, of `<text>`. Likewise, the depth of the media display is adjusted to the depth of `<text>` if the optional '`depth`' has not been given. Therefore, option '`text`' can be used to scale the media display to the natural or scaled dimensions of a poster image file, such as (requires package 'graphicx'):

```
 \includemovie[text={\includegraphics[scale=2]{path/to/poster}}]
   {}{}{path/to/movie}
```

`textoverposter[=false]`

Set this option to '`false`' if you wish `<text>` to be obscured by the external poster image which has been inserted with the poster option. Deprecated.

`url`

Treat `<media file>` as URL. Sets option '`inline`' to '`false`' automatically.

`volume=<percentage of original volume>`

The meaning of this option should be self explaining.

5.2 Inclusion of 3D objects

The PDF-1.6 specification, which was introduced with the advent of Adobe Acrobat/Reader 7, allows embedding of 3 dimensional graphic objects, such as CAD models or 3D scientific data, and lets the user interactively manipulate them. At time of writing this documentation, the only supported file format is U3D [2], and only one commercial software [3] for exporting into U3D format, yet from a number of CAD and 3D vector formats, including DXF and VRML, was known. However, a tryout version of [3] can be downloaded from the Web site.

Selection of the '`3D`' package option enables the 3D feature. Most of the command options listed in section 5.1 do what they are supposed to do in the case of embedded 3D as well. Other options are ignored, in particular '`open`', '`repeat`', 'palindrome', 'startat', 'endat', 'rate' and 'volume', because they do not make sense, 'inline' and 'url', because 3D files are always fully embedded and finally 'mouse', because 3D objects are interactive by definition.

There are a few options to \includemovie which define how the 3D object is positioned within the viewport of a virtual camera, or conversely, how the virtual camera is positioned and oriented within a coordinate system, called 'The World', which bears the 3D object at a fixed position. Fig. 1 should help to visualise the scenery: The virtual camera is orbiting at a distance of ROO around the centre of orbit, specified by the position vector $\overrightarrow{COO}$; $\sphericalangle AAC$ is the camera's aperture angle. In addition, the direction vector $\overrightarrow{C2C}$ is needed to specify the initial camera position.

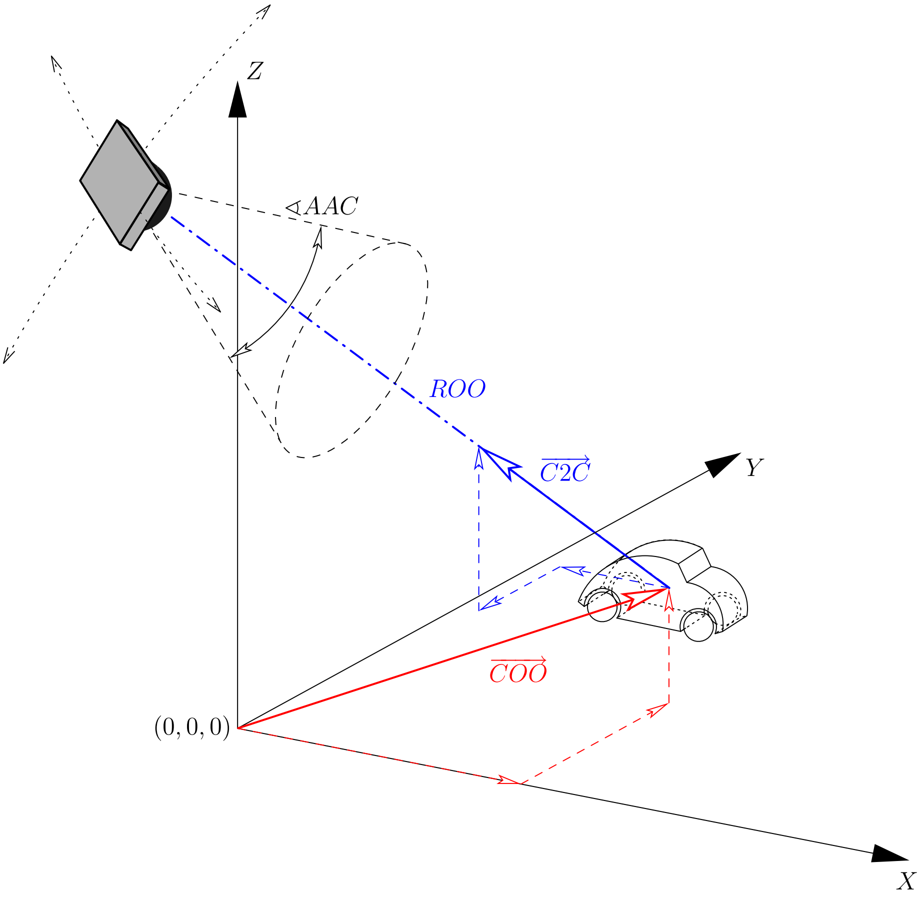

Figure 1: Camera and 3D object in the World System XYZ; centre of orbit position vector $C\vec{O}O$, centre of orbit to camera direction vector $C\vec{2}C$, radius of orbit ROO, aperture angle of camera $\sphericalangle AAC$.

The *default view*, i. e. the view that is shown initially after activating the 3D object in the Reader, can be set using the options '3Dcoo' for the centre of orbit, '3Dc2c' for the centre of orbit to camera vector, '3Droo' for the orbital radius and '3Daac' for the aperture angle of the camera:

`3Daac=<angle>`

This option sets the aperture angle of the camera, measured in degrees. Fixed point real numbers between 0 and 180 are admissible. A sensible value of 30 is preset by default. Larger values can be used to achieve wide-angle or fish-eye effects, cf. example 2 in section 6.

`3Dc2c=<x> <y> <z>`

`<x> <y> <z>` specify a direction vector $\overrightarrow{C2C}$ of arbitrary length, originating in the centre of orbit and pointing to the virtual camera; cf. option '`3Dcoo`' for the number format of `<x> <y> <z>`. Note, that *top* (`3Dc2c=0 0 1`) and *bottom views* (`3Dc2c=0 0 -1`) as well as views with *very small but larger than zero inclination angles* relative to the world axes should be avoided when going the `latex` → `dvips` → `ps2pdf` route. Due to a bug in Ghostscript they lead to unusable 3D inclusions. This restriction does not apply to pdfLaTeX!

`3Dcoo=<x> <y> <z>`

`<x> <y> <z>` specify the positional vector $\overrightarrow{COO}$ of the centre of orbit. Only fixed point numbers in the range from -999999999999999999.999999999999999999 to +999999999999999999.999999999999999999 are allowed.

`3Droo=<r>`

`<r>` (always positive!) specifies the radius of orbit *ROO* of the virtual camera; cf. option '`3Dcoo`' for the number format

Without the above options the immovable camera sits at the origin $(0, 0, 0)$ of the World, and looks in the positive Y direction, i. e. default settings of `3Droo=0`, `3Dcoo=0 0 0` and `3Dc2c=0 -1 0` are assumed. (Note, that $\overrightarrow{C2C}$ is the opposite of the view vector!)

Therefore, in order to get a 'front view' of the 3D object it would be sufficient to gradually increase the radius *ROO* from LaTeX run to LaTeX run until the object or group of objects can be fully seen in the viewport. Since the object size in terms of world coordinate units may differ in orders of magnitude between U3D files, the optimal '`3Droo`' setting might be found by experimenting with a series of geometrically increasing numbers, such as 1, 10, 100, 1000, 10000, 100000 etc. to get a rough approximation and by a subsequent fine tuning using smaller changes.

Ideally, there should be no need for changing the default setting of `3Dcoo=0 0 0` for the centre of orbit. However, this requires that the 3D object or group of objects be centred around the world origin $(0, 0, 0)$ during file export into the U3D format. (This is the default setting with [3].) If the 3D object apparently pivots around an orbital centre that lies outside of the object's cubature, either the file export could be repeated with appropriate settings or the centre of orbit could be moved into the object centre using the '`3Dcoo`' option. Regarding the latter method, this can be easily achieved if the object is examined from two different directions, say from left (`3Dc2c=-1 0 0`) in order to adjust the `<y>` and

<z> coordinates of ‘3Dcoo’ and from front (3Dc2c=0 -1 0) in order to adjust the remaining <x> coordinate.

```
3Dviews=<views file>
```

Instead of or in addition to the default view, further *named, predefined views* can be set in an auxiliary file specified by the ‘3Dviews’ option. The predefined views can later be selected from a drop down list in the tool bar that is associated with every activated 3D object in the Reader. The file <views file> contains lines which obey the syntax:

```
[<name>]{<coo_x> <coo_y> <coo_z>}{<c2c_x> <c2c_y> <c2c_z>}{<roo>}{<aac>}
```

The <name> entry is optional. If <name> is not given, a default name consisting of ‘View’ followed by the number of the current entry in the list is formed. For <coo_x>, <coo_y>, <coo_z>, <c2c_x>, <c2c_y>, <c2c_z>, <roo> and <aac> the same rules as for the corresponding options ‘3Dcoo’, ‘3Dc2c’, ‘3Droo’ and ‘3Daac’ apply. Trailing whitespace or comment signs (%) are allowed. Reading of the file stops either at its end, at the first empty line encountered or at the first line containing nothing but whitespace and/or a comment sign followed by arbitrary stuff.

```
3Dbg=<r> <g> <b>
```

This option sets the background colour of the canvas. Only fixed point real numbers in the range from 0 to 1 are allowed for the colour components.

```
3Djscript=<JavaScipt file>
```

Things like animation, lighting, background colour etc. may also be script driven. Option ‘3Djscript’ associates a <JavaScipt file> with the 3D object. The script will be executed upon activation of the object. Refer to the Acrobat 3D JavaScript Reference [4] for details.

5.3 Media hyperlinks

A movie or sound may serve as the destination of hyperlinks, which are inserted into the document using the command

```
  \movieref[<options>]{<label spec>}{<text>}
```

It makes <text> a hyperlink to a movie or sound that has been labelled with <label spec> using the ‘label’ option of the \includemovie command. Media hyperlinks can be inserted at any location within the document. Clicking onto the hyperlink causes the Reader to open the page containing the media it is pointing to. Referencing of 3D objects has not been implemented yet.

The primary purpose of media hyperlinks, however, is to provide a means to control the playback of media. Depending on the <options> being in effect, clicking onto the hyperlink causes the media to stop, pause or resume, or to restart at different settings, such as frame rate, volume, starting and ending positions.

Options that control the playback *characteristics* are: '`controls`', '`endat`', '`mouse`', '`palindrome`', '`rate`', '`repeat`', '`startat`' and '`volume`'. Their meaning is the same as for the `\includemovie` command and can be looked up in section 5.1. They take effect in combination with the '`play`' option only.

The options which follow let the hyperlink change the playback *state* of media. If none of them is specified, '`play`' is assumed.

`play`

Immediately closes the media and restarts playback. If new playback settings have been specified through the options given above, they override the original settings being in effect during media inclusion.

`stop`

Stops and rewinds the media to its beginning or to the '`startat`' position.

`resume`

Resumes paused media. In fact, this option makes the hyperlink a toggle switch: repeated clicking switches between paused and resumed state.

`pause`

Has the same effect as option '`resume`'. Added for completeness only.

`reset`

Revert to the original settings which were in effect during media inclusion.

`close`

Immediately closes the media player.

5.4 Compatibility commands

Two user commands have been provided that make 'movie15' a replacement for the 'multimedia' package which is part of Till Tantau's 'Beamer' Class:

```
\movie[<options>]{<poster text>}{<media file>}
```

```
\hyperlinkmovie[<options>]{<label spec>}{<text>}
```

All of the optional arguments to `\includemovie` and `\movieref` can be used with the compatibility macros as well. In addition, they honor the following options:

```
height=<height of display>
width=<width of display>
```

Set the horizontal and vertical dimensions of the media display area (`\movie` only).

```
autostart
```

The same as ‘`autoplay`’.

```
loop[=<repeats>]
```

The same as ‘`repeat`’.

```
once
```

The same as ‘`repeat=1`’.

```
showcontrols
```

The same as ‘`controls`’.

```
start=<offset in seconds>s
```

Start playback at the specified temporal offset. The trailing ‘`s`’ is mandatory.

```
duration=<duration in seconds>s
```

The duration of the media segment to be played. The trailing ‘`s`’ is mandatory. Note, that ‘`duration`’ cannot be used together with option ‘startat’ from `\includemovie`. The following options can be combined in order to define a media segment: ‘`startat`’/‘`endat`’, ‘`start`’/‘`endat`’, ‘`start`’/‘`duration`’.

6 Examples

1. A short circular MPEG movie, taken from `http://www.linux-video.net/Samples/`

```
\includemovie[
  poster,
  text=(random.mpg),
  mouse,
  repeat
]{
  .5\linewidth
}{
  .375\linewidth
}{random.mpg}
```

(random.mpg)

2. Embedded U3D file. It is based on a VRML model by Peter Whitehouse `<pwhitehouse@optusnet.com.au>`. Conversion to U3D has been made with DeepExploration$^{®}$[3]. A file ‘`dice.vws`’ that provides predefined views and a JavaScript ‘`dice.js`’ defining light sources have been associated with the 3D object.

```
\includemovie[
  poster,
  text=(dice.u3d),
  3Droo=28,
  3Dc2c=1 -1 1,
  3Dbg=1 1 1,
  3Dviews=dice.vws,
  3Djscript=dice.js
]{
  .5\linewidth
}{
  .5\linewidth
}{dice.u3d}
```

(dice.u3d)

References

[1] Adobe Systems Inc.: *PDF Reference, fifth edition: Adobe Portable Document Format version 1.6*, 2004, Implementation note 144, Appendix H.3, available at: `http://partners.adobe.com/public/developer/en/acrobat/sdk/pdf/pdf_creation_apis_and_specs/PDFReference.pdf`.

[2] `http://www.3dif.org/`

[3] `http://www.righthemisphere.com/products/dexp/`

[4] Adobe Systems Inc.: *Adobe Acrobat 7.0 3D JavaScript Reference*, 2005, available at `http://partners.adobe.com/public/developer/en/acrobat/sdk/pdf/javascript/Acro3DJS.pdf`

Table 1: Media formats for use with Adobe Reader (partially from [1])

Extension	MIME type	Description
aif, aifc, aiff	audio/aiff	Audio Interchange File Format
au, snd	audio/basic	NeXT/Sun Audio Format
avi	video/avi	Audio/Video Interleaved
mid, rmi	audio/midi	Musical Instrument Digital Interface
mov, qt	video/quicktime	Apple QuickTime
mp3, m3u	audio/x-mp3	MPEG Audio Layer-3
mp4	video/mp4	MPEG-4 Video/Audio
mpeg, mpg	video/mpeg	MPEG-2 Video
ra, ram, rm	audio/x-pn-realaudio	Real Media
smil	application/smil	Synchronized Multimedia Integration Language
swf	application/x-shockwave-flash	(Adobe?) Macromedia Flash
u3d	model/u3d	Universal 3D File Format
wav	audio/wav	MS Audio Format
wma	audio/x-ms-wma	Windows Media Audio
wmv	video/x-ms-wmv	Windows Media Video