# Batch-oriented MPEG generation with GMV in background mode

# Release 3.0

Sven H.M. Buijssen

Institute of Applied Mathematics, University of Dortmund
Vogelpothsweg 87, 44227 Dortmund, Germany
Email: featflow@featflow.de
Homepage: www.featflow.de

October 2005

**Abstract**

For automatisation purposes I developed a few years ago a Perl script called `gmvmpeg` that simplifies the generation of MPEG movies from a sequence of visualisation data files in GMV format[1]. Right from the beginning, the script has been able to create movies "in background" on (remote) computers, i.e. without the need for X Windows or even an interactive shell. Over the years, the script has been extended to create either MPEG-1 or MPEG-2 format, to run in parallel, to determine the path for every helper application it relys on automatically, in short: to be more user-friendly and foolproof.

This paper is intended to present the current state of development.

## What it does

There are numerous software packages for the visualisation of numerical data. For the visualisation of data resulting from simulations with FEAST [Bec04, BGTW04] and FEATFLOW [BT98] GMV (General Mesh Viewer) of Los Alamos National Laboratory belongs to our favourite tools. For an overview of the features of this program please see its documentation [Ort05].

Mainly two features of GMV are exploited by the script `gmvmpeg`: GMV comes with the possibility to save arbitrary settings into an attribute file. Thus, you only have to choose what to visualise (cutplanes, isosurfaces, isovolumes, particles etc.) and put this to disk. Next, we can use the batch mode of GMV together with this attribute file to visualise all files of a sequence. The images we get are finally passed to an MPEG encoder to create a video.

[1] "General Mesh Viewer" of the Los Alamos group

## How to use it

Let us assume that you have generated a bunch of GMV files – be it with FEAST, FEATFLOW or some other code. They represent discrete snapshots of a time-dependent computational problem (from CFD, CSM or whatever). Most likely, you then want to visualise your calculations with help of a movie. With gmvmpeg all you need to do in order to generate an MPEG movie out of a sequence of files containing data from a numerical simulation is the following:

1. Start GMV
2. Load a (more or less) arbitrary file from your sequence
3. Make a decision on the subset of data to be displayed.[2]
4. Save your settings into a so-called attribute file.
5. Finally run the perl-script gmvmpeg with appropriate command lines options (see below).

## Installation and configuration

The installation process is as simple as copying this script (which can be obtained from the FEATFLOW homepage) to a directory you like and adjust your $PATH variable to include this directory.
You might want to adjust a few variables within the script. The script relys on the following programs:

- GMV for the visualisation process; can be downloaded from the GMV homepage (see below).

  GMV 2.7 or above is required.

- `sgitopnm` from the `netpbm` package for converting the GMV images into a format the MPEG encoder can handle

  Alternatively, you can use the `convert` program that ships with ImageMagick [ima05]. It is a substitute for the common Unix `convert` program with support for many more file formats. Especially, the SGI RGB and PPM format are supported. The MPEG encoder needs the latter as input format.

- `mpeg2encode` for the encoding process (see below).

If the paths to these programs are invalid on your system, the script tries to automatically determine correct paths. If found, it will give a warning telling you which program's path information has been overridden and continue using this path information. To get rid of this warning, adjust the line in gmvmpeg the script lists in the warning.

Additionally, if you want on-the-fly decompression of compressed GMVfiles during MPEG generation, you have to specify the program locations of the following supported compression utilities: `gzip`, `compress` and `bzip2`.

[2] The manuals [Ack98], [AT99] and especially the complete GMVmanual [Ort05] will help you in doing so.

In case you would like to use the *parallel mode* of `gmvmpeg`, ensure that you have the Perl module Parallel::ForkManager installed on your system. If not, get it from CPAN [cpa] or a nearby mirror site.

## Command line options

`gmvmpeg` has the following command line options to control the noninteractive generation of MPEG movies.

attribute file: **-a <filename>**, **--attributes <filename>**

- Path to the `GMV` attribute file to be used.
- Default: "default.attr"

input file pattern: **-i <file pattern>**, **--input <file pattern>**

- `gmvmpeg` will look for `GMV` data files that match the file pattern you specify here. Use the % character as wildcard. Several %'s mean padding the numbers with zeros.

  Example:
  a) Let's assume that you have the following file sequence:

  u.2.gmv, u.3.gmv, u.4.gmv, u.5.gmv, ...

  Then, the <file pattern> would be: u.%.gmv.

  b) If your file sequence is like:

  file02.gmv, file03.gmv, file04.gmv, file05.gmv, ...

  Then, the <file pattern> would be: file%%.gmv.

  If this mechanism is not flexible enough for your needs, look for the definition of the variable $filename in `gmvmpeg` and change it as required.
- Default: "u.%.gmv"

file name of MPEG movie (output): **-o <filename>**, **--output <filename>**

- Basename of the MPEG movie

  When creating MPEG-1 format, the extension ".mpeg" will be added, for MPEG-2 it's ".m2v".
- Default: "movie"

creation of MPEG-1 or MPEG-2 files: **--mpeg1, --mpeg2**

- Instruct the encoder to respectively produce movies in MPEG-1 or MPEG-2 format.
- Default: MPEG-1

indices of input files: **-fls <number1>,<number2>[,<number3>]**

- The sequence of input files starts with number <number1> and ends with <number2> with a stride of <number3>. If <number3> is omitted or zero, time stepping is adaptive and all files available are taken.
  ("fls" stands for "first", "last", "stride")
- Default: "1,100000,0"
  (Thus, every existing GMV file in the range 1 till 100.000 is used for creating a movie.)

invisible mode: **-I, --invisible**

- Both the OpenGL and the Mesa version of GMV will pop up a window for each file processed and make a snapshot of it. Invoking `gmvmpeg` with this flag causes the use of the batch version of GMV such that the generation process is done in background mode. You will notice nothing but an increasing computer load and progress being made as subsequent files are processed. No X server display is needed.

  Using this option, the complete MPEG generation of one or several videos can be transferred to an arbitrary computer in a network. This means, you can even log in via modem and start this "visualisation process" in a VT100 emulation.
- Default: not set.

parallel mode: **-j <jobs>**, **--jobs <jobs>**

- Specify the number of jobs, i.e. instances of GMV, to run simultaneously. If there is more than one -j option, the last one is effective.

  *Note:* This works only if you have installed the Perl module Parallel::ForkManager! If it is missing, the number of jobs is reset to 1.
- Default: set to 1.

working directory: **--wd <path>**

- Specify an alternative working directory. Any file given without an absolute path will be searched for in <path>. This holds for attribute files, GMV data files and the movie to be created.
- Default: set to current directory.

on-the-fly decompression of files: **-Z, -z, -gzip, --bzip2**

- Look for GMV files with suffix .Z, .gz or .bz2 and decompress them on-the-fly using `compress`, `gzip` and `bzip2` respectively.
- Default: not set.

These command line options are the ones that you will most likely change each time. The following you will probably use rarely:

help screen: **-h**, **--help**

- Show a help screen explaining all possible command line options.

check paths: **--checkonly**

- Verify that all program paths are set correctly within the script.

bitrate: **-b, --bitrate**

- Specify bitrate for the MPEG movie.
- Default: 5000000 for MPEG-1, 3500000 for MPEG-2

keep PPM and RGB files: **-k, --keep-files**

- The snapshots are converted to PPM "rawbits image" format and are, by default, deleted when the MPEG encoding has finished. The same happens to the MPEG encoder's configuration file. If you want to manually adjust the `gmvmpeg`'s default MPEG encoder setting, specify this option to avoid unnecessary regeneration of the image files.
- Default: not set.

maximum size of MPEG movie: **-m number, --max number**

- Obsolete. Just provided for compatibility reasons. Use -b/--bitrate instead.
- Tells the MPEG encoder to limit the file size to <number> MB.
- Default: not set.
  (Thus, by default there is no file size limitation.)

verbose: **-V, --verbose**

- Verbose mode.
- Default: not set.
  (Thus, by default `gmvmpeg` will swallow all output from `GMV` and the MPEG encoder.)

version information: **--version**

- Prints version information.
- Default: not set.

window size: **-x <number>, -y <number>**

- Resolution in x- and y-direction of the movie to be generated.
- Default: window size 800x600.

## Where to get the programs mentioned

(see also our homepage)

- gmvmpeg: `http://www.featflow.de/download/gmvmpeg3`
- GMV: `http://www-xdiv.lanl.gov/XCM/gmv/GMVHome.html`
- NetPBM: `http://wuarchive.wustl.edu/graphics/graphics/packages/NetPBM/`
- ImageMagick: `http://www.imagemagick.org/`
- MPEG-2 encoder: `ftp://ftp.mpeg.org/pub/mpeg/mssg/mpeg2vidcodec\_v12.tar.gz`
- compress: shipped with every Un*x flavour
- gzip: `ftp://ftp.gnu.org/pub/gnu/gzip/`
- bzip2: `http://sources.redhat.com/bzip2/`

## Example

Finally, we want to show an invocation of gmvmpeg. We will give the same example as in [Ack98] where we visualised a pressure distribution.

- Start GMV and adjust its settings for displaying a pressure distribution.
- Save your adjustments in an attribute file named "pressure.attr".
- To create an MPEG movie called "pressure.mpeg" with a 400x320 resolution from the data files "u.1.gmv" to "u.99.gmv", just type:

```
gmvmpeg3 -a pressure.attr -i u.%.gmv -fls 1,99 -o
               pressure -x 400 -y 320
```

- If you have prepared additional attribute files "streamfunction.attr" and "temperature.attr", the batch oriented MPEG generation in "invisible" mode (i.e. with exploitation of the batch version of GMV) can be started with the following shell script (only every second file is processed):

```
#!/bin/sh
gmvmpeg3 -i u.%.gmv -fls 1,99,2 -x 400 -y 320 --invisible \
         -a pressure.attr       -o pressure

gmvmpeg3 -i u.%.gmv -fls 1,99,2 -x 400 -y 320 --invisible \
         -a streamfunction.attr -o streamfunction

gmvmpeg3 -i u.%.gmv -fls 1,99,2 -x 400 -y 320 --invisible \
         -a temperature.attr    -o temperature
# End sample.sh
```

## Remarks

Unlike in the first version of gmvmpeg, there is no more need for the virtual framebuffer X server called Xvfb from the XFree86 Project, Inc.
Starting from GMV version 2.7, there is a batch version of GMV that does the rendering in background and saving it to disk. Thus, the annoying peculiarity about the visualisation process in former versions of gmvmpeg where for each file GMV a window popped up on your screen and took a screenshot of it is no longer existing. Hence, our workaround with Xvfb (side-track the output of GMV to this X server which emulates a dumb framebuffer using virtual memory) has become obsolete.

## References

[Ack98] Jens F. Acker. *Working with GMV under FEATFLOW*. Preprint 98 - 50 (SFB 359), October 1998.

[AT99] Jens F. Acker and Stefan Turek. *3D Presentation of FEATFLOW Data with GMV*. Preprint 99 - 19 (SFB 359), April 1999.

[Bec04] Ch. Becker. *FEAST – The Realisation of Finite Element Software for High–Performance Applications*. PhD thesis, Universität Dortmund, 2004. to appear.

[BGTW04] S. H. M. Buijssen, M. Grajewski, S. Turek, and H. Wobker. High performance fem simulation. Research report, nrw graduate school of production engineering and logistics, Universität Dortmund, Leonhard-Euler-Str. 5, 44221 Dortmund, September 2004. p.52–55.

[BT98] Ch. Becker and S. Turek. Featflow finite element software for the incompressible Navier–Stokes equations – user manual release 1.1. Technical report, Universität Heidelberg, 1998.

[cpa] *Comprehensive Perl Archive Network*. http://www.cpan.org/.

[ima05] *ImageMagick®*, 2005. http://www.imagemagick.org/.

[Ort05] Frank A. Ortega. *GMV 3.8. General Mesh Viewer Uses's Manual*. Los Alamos National Laboratory, 2005. http://www-xdiv.lanl.gov/XCM/gmv/.

All these papers are available at http://www.featflow.de/documentation.html.