ADAMS

Advanced **D**ata mining **A**nd **M**achine learning **S**ystem

Module: adams-imaging

Peter Reutemann

February 8, 2012

DRAFT

©2009-2012

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

Contents

1 Java Advanced Imaging **7**

2 ImageJ **9**

3 ImageMagick **11**

4 Movies **15**

Bibliography **17**

List of Figures

3.1 ImageMagick flow for processing (resizing) a single image. 12
3.2 ImageMagick commands to resizing. 12
3.3 The original image to be resized with ImageMagick. 13
3.4 The output of ImageMagick, after resizing the image 13

4.1 Screenshot of the mandelbrot movie 15

Chapter 1

Java Advanced Imaging

Chapter 2

ImageJ

Chapter 3

ImageMagick

ImageMagick® is a software suite to create, edit, compose, or convert bitmap images ([4]). In order to process images with ImageMagick, the tools need to be present in the system's path.

There are three ImageMagick actors available:

- `transformer.ImageMagickReader` – for reading any image file that ImageMagick supports and forwarding a `BufferedImage` object.
- `transformer.ImageMagickTransformer` – performs any ImageMagick command on the incoming image that the `convert` tool [1] supports and outputs another image again.
- `sink.ImageMagickWriter` – for writing a `BufferedImage` to a file format that ImageMagick supports. If the image type cannot be determined based on the extension, you can also specify which type to generate.

The example flow in Figure 3.1 loads a single photo from disk and then uses ImageMagick to resize it to 90 by 90 pixels and scaling it by 200% (see 3.2). Finally, the modified image is displayed in the image viewer.

[1] `http://www.imagemagick.org/script/convert.php`

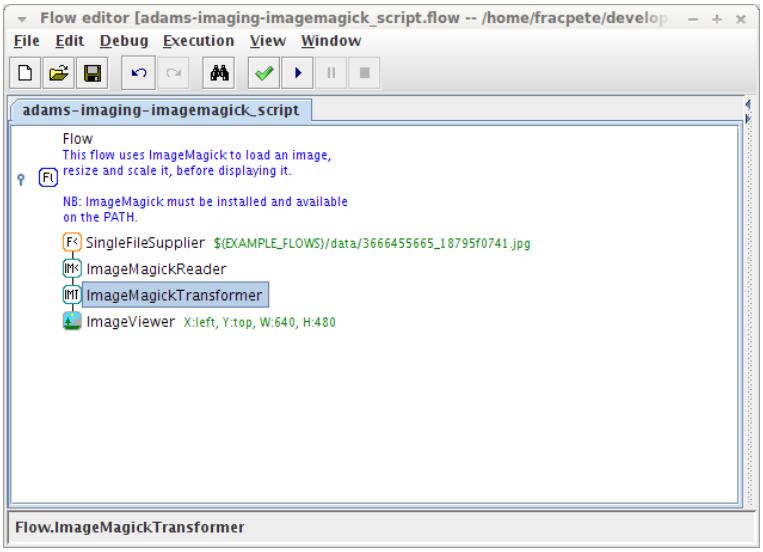

Figure 3.1: ImageMagick flow for processing (resizing) a single image.

```
# resizing image
-resize 90x90
# scaling it up again
-scale 200%
```

Figure 3.2: ImageMagick commands to resizing.

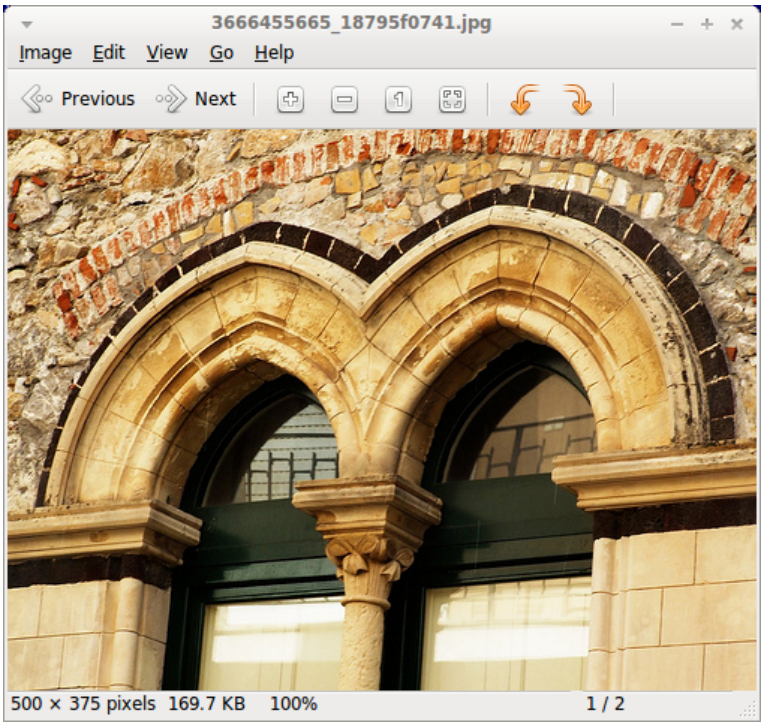

Figure 3.3: The original image to be resized with ImageMagick.

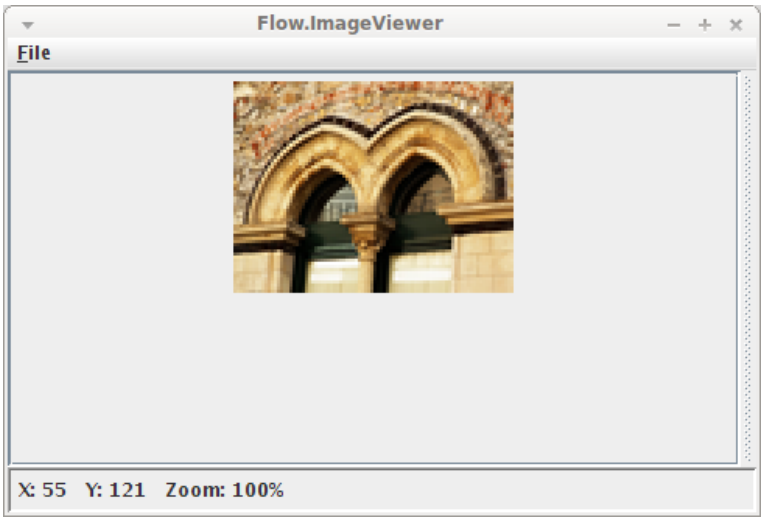

Figure 3.4: The output of ImageMagick, after resizing the image

Chapter 4

Movies

It is possible to create movies within ADAMS. Since it is a very experimental feature, only very limited support is currently available. At the time of writing, you can only generated QuickTime®[1] movies from still images. The `MovieWriter` sink simply takes an array of file names (have to be JPG files) as input and generates a movie using the Java Media Framework[2].

The screenshot in Figure 4.1 shows a movie generated with the flow [3] while calculating the Mandelbrot[4] set. After each pixel is calculated, a screenshot of the plot is taken and saved as JPG image. After the calculation is complete, the `MovieWriter` actor is used to generate the movie.

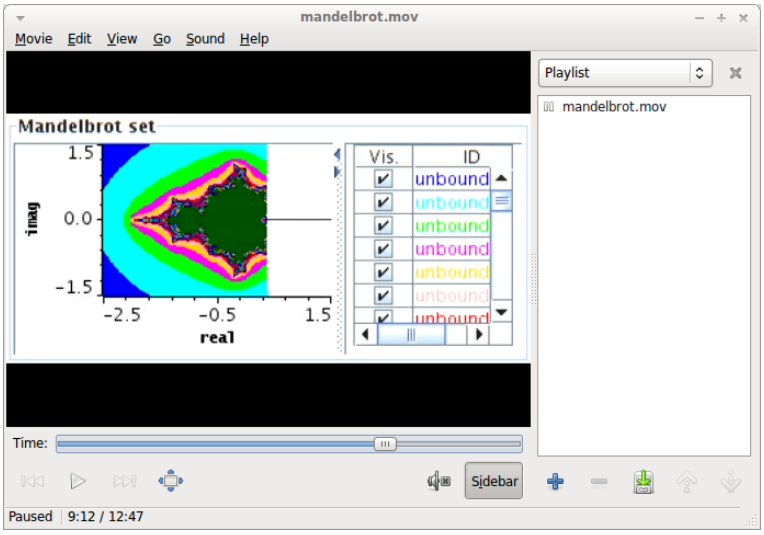

Figure 4.1: Screenshot of the mandelbrot movie

[1] `http://en.wikipedia.org/wiki/QuickTime`
[2] `http://en.wikipedia.org/wiki/Java_Media_Framework`
[3] adams-imaging-mandelbrot_colored-movie.flow
[4] `http://en.wikipedia.org/wiki/Mandelbrot_set`

Bibliography

[1] *ADAMS* – Advanced Data mining and Machine learning System
`http://adams.cms.waikato.ac.nz/`

[2] *JAI* – Java Advanced Imaging API
`http://java.sun.com/javase/technologies/desktop/media/jai/`

[3] *ImageJ* – Image Processing and Analysis in Java
`http://rsbweb.nih.gov/ij/`

[4] *ImageMagick* – Software suite to Convert, Edit, and Compose Images
`http://www.imagemagick.org/`