

ADAMS

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

Module: adams-video



Peter Reutemann

November 18, 2015

©2015



THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-sa/3.0/>

Contents

1	Flow	7
2	Tools	9
2.1	Trail viewer	9
3	Data	11
3.1	.trail format	11
	Bibliography	13

List of Figures

2.1	Trail viewer displaying a mice trail overlayed on a background image.	9
-----	---	---

Chapter 1

Flow

The video module offers some actors for basic video display and processing support.

Available sources:

- *ListWebcams* – lists the names of all available webcams attached to the computer¹.
- *WebcamImage* – outputs images from the selected webcam attached to the computer².
- *WebcamInfo* – outputs images from the selected webcam ³.

Available transformers:

- *AddTrailBackground* – adds a image as trail background⁴
- *AddTrailStep* – adds an additional step to the trail passing through⁵.
- *ExtractTrackedObject* – extracts a tracked object in an image and forwards it as new image. container⁶.
- *GetTrailBackground* – retrieves the trail background image, if any.
- *MjpegImageSequence* – generates an image sequence from MJPEG movies, one frame at a time⁷.
- *MovieImageSequence* – generates an image sequence from movies, one frame at a time (uses Xuggle[3]⁸).
- *TrackObjects* – tracks objects in images sequences, e.g., from movies⁹.
- *TrailFileReader* – reads a trail from disk.
- *TrailFileWriter* – writes a trail to disk¹⁰.
- *TrailFilter* – applies a filter to the trail passing through.

¹adams-video-list_webcams.flow

²adams-video-webcam.flow

³adams-video-webcam_info.flow

⁴adams-video-track_objects-predefined3.flow

⁵adams-video-track_objects-predefined3.flow

⁶adams-video-track_objects-user_selected_object.flow

⁷adams-video-play_mjpeg_video.flow

⁸adams-video-play_mp4_video.flow

⁹adams-video-track_objects-predefined.flow, adams-video-track_objects-predefined2.flow

¹⁰adams-video-track_objects-predefined3.flow

- *TransformTrackedObject* – transforms a tracked object in an image with a callable transformer, e.g., for blurring a face¹¹.

Available sinks:

- *AnimatedGifFileWriter* – generates an animated GIF from an array of image files or images.
- *FFmpeg* – actor for processing videos using ffmpeg[2]¹².
- *TrailDisplay* – displays trail objects¹³.

Available conversions:

- *QuadrilateralLocationCenter* – outputs a Point2D object that is the center of the rectangle surrounding the quadrilateral coordinates.
- *QuadrilateralLocationToString* – turns the quadrilateral coordinates into a string.
- *StringToQuadrilateralLocation* – turns a string into quadrilateral locations.

¹¹adams-video-track_objects-predefined2.flow

¹²adams-video-ffmpeg.flow

¹³adams-video-display_trail.flow, adams-video-track_objects-predefined3.flow

Chapter 2

Tools

2.1 Trail viewer

The *Trail viewer* is a simple tool for viewing trails of objects that have been tracked using a flow. It can be used to apply filters to trails and save those trails back to disk. Figure 2.1 shows a screenshot.

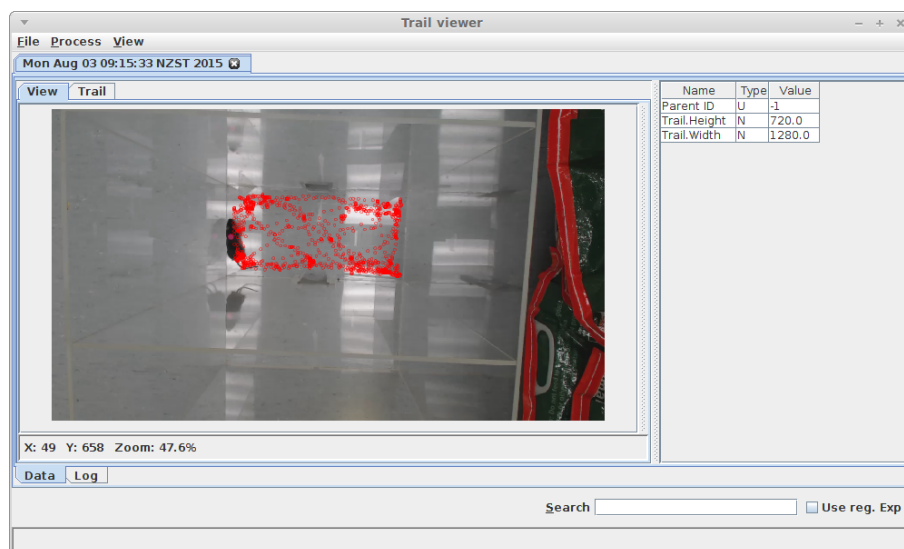


Figure 2.1: Trail viewer displaying a mice trail overlaid on a background image.

Chapter 3

Data

3.1 .trail format

The *.trail* format is a simple text format. In its essence it is spreadsheet-like format with a header and a body.

The *header* contains the meta-data for the trail. Most importantly, the width and the height of the trail. The format is in Java Properties format, each line prefixed with “# ”. Each stored value also defines in a separate property what data type it is (N = numeric, S = string, B = boolean).

It is possible to store an optional background image in the header. The image data is stored as RGBA signed bytes, row-by-row. In order to produce smaller files, the data is compressed using gzip[4]. The compressed bytes are then stored in a upper-case hexadecimal notation, with a maximum of 1000 bytes per line. The background data lines are prefixed with “% ”.

The *body* consists of four columns: timestamp (with milli-seconds), X position, Y position, meta-data for that step. The meta-data column is a blank-separated list of key-value pairs (“key=value”).

Here is an example file, without a background:

```
# #Tue Jul 28 09:24:20 NZST 2015
# Trail.Height\tDataType=N
# Trail.Width\tDataType=N
# Trail.Height=720.0
# Parent\ ID=-1
# Trail.Width=1280.0
Timestamp,X,Y,Meta-data
"1970-01-01 12:00:00.127",424,292,""
"1970-01-01 12:00:00.224",423,285,""
"1970-01-01 12:00:00.304",423,277,""
```


Bibliography

- [1] *ADAMS* – Advanced Data mining and Machine learning System
<https://adams.cms.waikato.ac.nz/>
- [2] *FFmpeg* – a complete, cross-platform solution to record, convert and stream audio and video
<http://ffmpeg.org/>
- [3] *xuggle* – a free open-source library for Java developers to uncompress, manipulate, and compress recorded or live video in real time
<http://www.xuggle.com/>
- [4] *gzip* – compression/decompression algorithm based on the DEFLATE algorithm, which is a combination of LZ77 and Huffman coding.
<https://en.wikipedia.org/wiki/Gzip>