autoscrub Documentation

Release 0.3.0

Russell Anderson, Philip Starkey

Jan 29, 2018

Contents

1 Contents 3

Python Module Index 17

autoscrub is a command line tool that automates the production of educational videos by increasing the playback speed during silent segments.

CHAPTER 1

Contents

1.1 Installation

These installation instructions assume you already have Python installed. If you do not already have a copy of Python, we recommend you install Anaconda Python.

Note: autoscrub's Python 3 support is experimental, and only works with 3.6+. If you run into any troubles with Python 3, please let us know via the issue tracker, and then try Python 2.7 instead.

1.1.1 FFmpeg

Autoscrub requires that you have FFmpeg installed in your system, and that it is accessible from the directory autoscrub is run in (usually by appending the location of the FFmpeg executables to the system PATH).

Linux

FFmpeg can usually be installed via your linux package manager. This typically also ensures that the location of FFmpeg is added to the system PATH.

Windows

Windows binaries can be downloaded here. We recommend using the stable version (the one with a 3 digit version number). You will need to select the version that matches your system architecture (usually 64-bit) and with static linking. We have tested with FFmpeg v3.3.3 (x64-static) but any newer version should also work.

autoscrub requires the ffmpeg.exe and ffprobe.exe files be accessible from the directory that autoscrub is run from. You can either place the executables in a common location and add that location to the windows PATH environment variable or place the executables in the folder you will run autoscrub from (for example the folder containing your media files you wish to convert).

Mac OSX

FFmpeg builds can be found here.

1.1.2 Autoscrub

PyPi

We recommend installing autoscrub from the Python Package Index. To do this, open a terminal (linux/OSX) or command prompt (Windows) window, and run:

```
pip install autoscrub
```

Upgrading autoscrub

To upgrade to the latest version of autoscrub, run:

```
pip install -U autoscrub
```

To upgrade to a specific version of autoscrub (or, alternatively, if you wish to downgrade), run:

```
pip install -U autoscrub==<version>
```

where `<version>` is replaced by the version you wish (for example `pip install -U autoscrub==0.1.3`).

Development Version

If you wish to use the latest development version, you can obtain the source code from our mercurial repository. Once you have cloned our repository, you should run `python setup.py install` in order to build and install the autoscriub package.

1.2 Example command line usage

The installation process of autoscrub automatically creates a command line utility for you to use. Here we will show examples of the most common usage. To see the full set of available commands and options run `autoscrub --help` from a terminal or look in the *Command line usage reference* documentation.

1.2.1 autoprocess

The most common command you will use is `autoscrub autoprocess`. The autoprocess command accepts a variety of options, however they have been preconfigured with sensible defaults that will suit many users.

To use the default options, run (replace paths as appropriate):

```
autoscrub autoprocess /path/to/input/file.mp4 /path/to/output/file.mp4
```

The most common options you are likely to want to adjust are:

- `-d` (or `--silence-duration`): This specifies the minimum length that autoscrub will use in detecting a silent segment (which will be sped up). Silent segments of audio that are shorter than this time will not be sped up. Adjust this to match your presentation style.

- `-t` (or `--target-threshold`): This specifies the audio level used to determine whether there is silence or not. If the audio is below this level, for at least the length of time specified by `--silence-duration`, then autoscrub will speed up this segment. Adjust this to compensate for a noisy background or quite speaking volume. The units are specified in decibels (dB).

Here we specify custom values for both these options (5 second silent duration, -20dB silence threshold) as an example:

```
autoscrub autoprocess -d 5 -t -20 /path/to/input/file.mp4 /path/to/output/file.mp4
```

To see all available options for autoprocess, run:

```
autoscrub autoprocess --help
```

1.3 Command line usage reference

1.3.1 autoscrub

Welcome to autoscrub!

If you're unsure which command you want to run, you likely want:
    autoscrub autoprocess <input video path> <output video path>
To view the additional options for autoprocessing, run:
    autoscrub autoprocess –help

To see command line arguments for the other autoscrub commands, run:
    autoscrub COMMAND –help
where the available commands are listed below.

```
autoscrub [OPTIONS] COMMAND [ARGS]...
```

autoprocess

automatically process the input video and write to the specified output file

```
autoscrub autoprocess [OPTIONS] input_filepath output_filepath
```

Options

-d, --silence-duration <silence_duration>
The minimum duration of continuous silence (in seconds) required to trigger speed up of that segment. [default: 2.0]

-h, --hasten-audio <hasten_audio>
The method of handling audio during the speed up of silent segments. [default: tempo]

-l, --target-lufs <target_lufs>
The target LUFS for the output audio [default: -18.0]

-p, --pan-audio <pan_audio>
Copies the specified audio channel (left|right) to both audio channels.

-r, --rescale <rescale>
rescale the input video file to the resolution specified [usage: -r 1920 1080]

-s, --speed <speed>
The factor by which to speed up the video during silent segments [default: 8]

-t, --target-threshold <target_threshold>
The audio threshold for detecting silent segments in dB [default: -18.0]

-v, --silent-volume <silent_volume>
The factor to scale the audio volume during silent segments [default: 1.0]

--delay <delay>
The length of time (in seconds) to delay the start of the speed up of a silent segment. This also ends the speedup early, by the same amount, and must satisfy the condition 2*delay < silence-duration [default: 0.25]

--debug
Retains the generated filtergraph file for inspection

Arguments

input_filepath
Required argument

output_filepath
Required argument

display-video-properties

Displays properties about the input file

```
autoscrub display-video-properties [OPTIONS] input_filepath
```

Arguments

input_filepath
Required argument

identify-silences

Displays a table of detected silent segments

```
autoscrub identify-silences [OPTIONS] input_filepath
```

Options

-d, --silence-duration <silence_duration>
The minimum duration of continuous silence (in seconds) required to trigger speed up of that segment. [default: 2.0]

-t, --target-threshold <target_threshold>
The audio threshold for detecting silent segments in dB [default: -18.0]

Arguments

input_filepath
Required argument

loudness-adjust

Adjusts the loudness of the input file

```
autoscrub loudness-adjust [OPTIONS] input_filepath output_filepath
```

Options

-l, --target-lufs <target_lufs>
The target LUFS for the output audio [default: -18.0]

Arguments

input_filepath
Required argument

output_filepath
Required argument

make-filtergraph

Generates a filter-graph file for use with ffmpeg.

This command is useful if you want to manually edit the filter-graph file before processing your video.

```
autoscrub make-filtergraph [OPTIONS] input_filepath
```

Options

-d, --silence-duration <silence_duration>
The minimum duration of continuous silence (in seconds) required to trigger speed up of that segment. [default: 2.0]

-h, --hasten-audio <hasten_audio>
The method of handling audio during the speed up of silent segments. [default: tempo]

-l, --target-lufs <target_lufs>
The target LUFS for the output audio [default: -18.0]

-p, --pan-audio <pan_audio>
Copies the specified audio channel (left|right) to both audio channels.

-r, --rescale <rescale>
rescale the input video file to the resolution specified [usage: -r 1920 1080]

-s, --speed <speed>
The factor by which to speed up the video during silent segments [default: 8]

-t, --target-threshold <target_threshold>
The audio threshold for detecting silent segments in dB [default: -18.0]

-v, --silent-volume <silent_volume>
The factor to scale the audio volume during silent segments [default: 1.0]

--delay <delay>
The length of time (in seconds) to delay the start of the speed up of a silent segment. This also ends the speedup early, by the same amount, and must satisfy the condition 2*delay < silence-duration [default: 0.25]

Arguments

input_filepath
Required argument

process-filtergraph

Processes a video file using the filter-graph file created by the autoscrub make-filtergraph command

```
autoscrub process-filtergraph [OPTIONS] input_filepath output_filepath
```

Arguments

input_filepath
Required argument

output_filepath
Required argument

trim

removes unwanted content from the start and end of the input file

```
autoscrub trim [OPTIONS] input_filepath output_filepath
```

Options

--start <start>
Content before this time is removed [default: 0]

--stop <stop>
Content after this time is removed

--re-encode <re_encode>
Re-encode the file with the codec specified

Arguments

input_filepath
Required argument

output_filepath
Required argument

1.4 autoscrub API

Advanced users may wish to build their own Python programs that use the autoscrub API. To use the autoscrub API, include `import autoscrub` at the top of your Python file and call the below functions as required.

`autoscrub.`**concatFileList**(*concat_path*, *output_path*, *overwrite=None*)

Take a file list for the ffmpeg concat demuxer and save to `output_path`. The concat file (located at `concat_path`) must contain lines of the form:

```
file '/path/to/file1'
file '/path/to/file2'
file '/path/to/file3'
```

This avoids a re-encode and can be used with formats that do not support file level concatenation.

Parameters

- **concat_path** – the filepath containing the list of media files to concatenate
- **output_path** – the filepath at which to write the result of the concatenation

Keyword Arguments **overwrite** – If `True`, overwrites the `output_path` with no prompt. If `False`, the function will fail if the `output_path` exists. Defaults to `None` (prompts user for input)

Returns `output_path` if successful or `None`.

`autoscrub.`**concatSegments**(*segment_paths*, *output_path=None*, *overwrite=None*)

Concatenate a list of inputs (`segment_paths`) using the ffmpeg concat demuxer. A concat file will be created of the form:

```
file '/path/to/file1'
file '/path/to/file2'
file '/path/to/file3'
```

This avoids a re-encode and can be used with formats that do not support file level concatenation.

Parameters **segment_paths** – A list of filepaths to concatenate

Keyword Arguments

- **output_path** – the filepath at which to write the concat file. If left as the default (`None`) it appends `_concat` to the end of the filename and preserves the input file extension.
- **overwrite** – If `True`, overwrites the `output_path` with no prompt. If `False`, the function will fail if the `output_path` exists. Defaults to `None` (prompts user for input).

Returns The `output_path` where the output of ffmpeg was written.

`autoscrub.ffmpeg`(*filename*, *args=[]*, *output_path=None*, *output_type=None*)

Runs ffmpeg on filename with the specified args.

Parameters **filename** – The filepath passed to the `-i` option of ffmpeg.

Keyword Arguments

- **args** – A list of additional arguments to pass to ffmpeg.
- **output_path** – The filepath to append to the end of the ffmpeg command, designating the output file for the ffmpeg result. If left as the default (`None`) it appends `_processed` to the end of the filename and preserves input file extension unless `output_type` is specified.
- **output_type** – Determines the output file type. Specify as a string containing the required file extension. This is ignored if `output_path` is specified.

Returns The `output_path` where the output of ffmpeg was written.

`autoscrub.ffmpegComplexFilter`(*input_path*, *filter_script_path*, *output_path='/dev/null'*, *run_command=True*, *overwrite=None*)

Executes the ffmpeg command and processes a complex filter

Prepare and execute (if run_command) ffmpeg command for processing input_path with an ffmpeg filter_complex string (filtergraph) in filter_script_path, and save to output_path. As this requires re-encoding, video and audio settings are chosen to be compliant with YouTube's 'streamable content' specifications, available at (as of April 2017) https://support.google.com/youtube/answer/1722171

Parameters

- **input_path** – The path to the video file to process.
- **filter_script_path** – The path to the filter script.

Keyword Arguments

- **output_path** – The path to save the processed video (defaults to os.devnull).
- **run_command** – If False, simply prepare and return the command for debugging or later use (default: True).
- **overwrite** – Optionally specify addition of -y or -n flag to ffmpeg which is useful for unattended scripting (default None).

Returns `output_path` if `run_command=True` otherwise returns the fully formatted ffmpeg command to run as a string.

`autoscrub.ffprobe`(*filename*)

Runs ffprobe on `filename` and returns the log output from stderr.

Parameters **filename** – The filepath passed to ffprobe.

Returns The output of the ffprobe command.

`autoscrub.findDuration`(*log_output*)

Finds the duration in seconds from ffprobe log_output.

Parameters **log_output** – The output of ffprobe, as returned by `autoscrub.ffprobe()`.

Returns A float containing duration in seconds or None if the duration could not be determined.

`autoscrub.findLoudness`(*log_output*)

Extract loudness (key, value) pairs from ffmpeg log_output when using the ebur128 filter.

Parameters **log_output** – The output of the ffmpeg ebur128 filter, as returned by `autoscrub.getLoudness()`.

Returns

A loudness dictionary with keys:

```
I:   integrated loudness in dBLUFS
LRA: loudness range in dBLUFS
LRA high:
LRA low:
Threshold:
```

`autoscrub.`**`findSampleRate`**(*log_output*)

Finds the audio sample rate in Hz from ffprobe log_output.

Parameters **`log_output`** – The output of ffprobe, as returned by `autoscrub.ffprobe()`.

Returns A float containing audio sample rate in Hz or None if the sample rate could not be determined.

`autoscrub.`**`findSilences`**(*log_output*)

Extract silences from ffmpeg log_output when using the silencedetect filter.

Parameters **`log_output`** – The output of the ffmpeg silencedetect filter, as returned by `autoscrub.getSilences()`.

Returns

a list of silence dictionaries, with keys:

```
silence_start: the timestamp of the detected silent interval in␣
↪seconds
silence_end:   the timestamp of the detected silent interval in␣
↪seconds
silence_duration:  duration of the silent interval in seconds
```

`autoscrub.`**`generateFilterGraph`**(*silences*, *factor*, *delay=0.25*, *rescale=True*, *pan_audio='left'*, *gain=0*, *audio_rate=44100*, *hasten_audio=None*, *silent_volume=1.0*)

Generate a filtergraph string (for processing with the -filter_complex flag of ffmpeg) using the trim and atrim filters to speed up periods in the video designated by a list of silence dictionaries. This function calls `autoscrub.silenceFilterGraph()`, `autoscrub.resizeFilterGraph()` and `panGainAudioGraph()` as appropriate.

Parameters

- **`silences`** – A list of silence dictionaries generated from `autoscrub.getSilences()`
- **`factor`** – to speed up video during (a subset of) each silent interval

Keyword Arguments

- **`delay`** – to omit from silent intervals when changing speed (default 0.25s)
- **`rescale`** – Scale and pad the video (pillar- or letter-box as required) for 1920 x 1080 display (default True)
- **`pan_audio`** – 'left', 'right', or None/False specify whether to duplicate a stereo channel of input audio stream (default 'left')
- **`gain`** – in dB to apply when pan_audio is 'left' or 'right'
- **`audio_rate`** – Sample rate of audio input (in Hz, default 44100) used in asetrate/aresample filters when `hasten_audio=True`.

- **hasten_audio** – None, 'pitch' or 'tempo'. Speed up audio during silent segment by either increasing pitch (with asetrate and aresample filters) or tempo (with atempo filter).
- **silent_volume** – scale the volume during silent segments (default 1.0; no scaling).

Returns The generated filtergraph as a string.

`autoscrub.getDuration(filename)`

Runs ffprobe on filename and extracts duration in seconds.

Parameters **filename** – The filepath of the media file you wish to process.

Returns A float containing duration in seconds or None if the duration could not be determined.

`autoscrub.getLoudness(filename)`

Runs the ffmpeg ebur128 filter on filename.

Parameters **filename** – the path to the video file to examine.

Returns

A loudness dictionary with keys:

```
I:   integrated loudness in dBLUFS
LRA: loudness range in dBLUFS
LRA high:
LRA low:
Threshold:
```

`autoscrub.getSampleRate(filename)`

Runs ffprobe on filename and extracts audio sample rate in Hz.

Parameters **filename** – The filepath of the media file you wish to process.

Returns A float containing audio sample rate in Hz or None if the sample rate could not be determined.

`autoscrub.getSilences(filename, input_threshold_dB=-18.0, silence_duration=2.0, save_silences=True)`

Runs the ffmpeg filter silencedetect with the specified settings.

Parameters **filename** – the path to the video file to examine

Keyword Arguments

- **input_threshold** – instantaneous level (in dB) to detect silences with (default -18).
- **silence_duration** – seconds for which level mustn't exceed threshold to declare silence (default 2).
- **save_silences** – print the above timestamps to CSV file (default = True).

Returns

a list of silence dictionaries, with keys:

```
silence_start: the timestamp of the detected silent interval in␣
↪seconds
silence_end:   the timestamp of the detected silent interval in␣
↪seconds
silence_duration:  duration of the silent interval in seconds
```

`autoscrub.hhmmssd_to_seconds(s)`

Convert a `'[hh:]mm:ss[.d]'` string to seconds.

Parameters **s** – A string in the format `'[hh:]mm:ss[.d]'`. The hours and decimal seconds are optional.

Returns The number of seconds as a float.

`autoscrub.matchLoudness`(*filename*, *target_lufs=-18*, *output_path=None*)

Applies the volume ffmpeg filter in an attempt to change the audio volume to match the specified target.

Parameters **filename** – the path to the video file to examine.

Keyword Arguments

- **target_lufs** – The target LUFS for the output audio (default: -18)
- **output_path** – the filepath at which to write the resultant file. If no output path is specified, it follows the conventions of `autoscrub.ffmpeg()`.

Returns The `output_path` where the output of ffmpeg was written.

`autoscrub.panGainAudioGraph`(*a_in='[0:a]'*, *duplicate_ch='left'*, *gain=0*, *a_out='[a]'*)

Generate a filtergraph string (for processing with the -filter_complex flag of ffmpeg) using the pan and volume filters to duplicate audio from one stereo channel to another, and optionally change the volume by gain.

Keyword Arguments

- **a_in** – The named filtergraph audio input pad. Defaults to `[0:a]` (see FFmpeg filter documentation).
- **duplicate_ch** – 'left', 'right', or None/False specify whether to duplicate a stereo channel of input audio stream (default 'left').
- **gain** – to apply (in dB) to the audio stream using the volume filter.
- **a_out** – The named filtergraph audio output pad. Defaults to `[a]` (see FFmpeg filter documentation).

Returns The generated filtergraph as a string.

`autoscrub.resizeFilterGraph`(*v_in='[0:v]'*, *width=1920*, *height=1080*, *pad=True*, *mode='decrease'*, *v_out='[v]'*)

Generate a filtergraph string (for processing with the -filter_complex flag of ffmpeg) using the scale and pad filters to scale & pad the video for width x height display, with optional padding.

Keyword Arguments

- **v_in** – The named filtergraph video input pad. Defaults to `[0:v]` (see FFmpeg filter documentation).
- **width** – of display on which the output stream must fit (default 1920).
- **height** – of display on which the output stream must fit (default 1080).
- **pad** – add letter- or pillar-boxes to the output as required to fill width x height.
- **mode** – argument of ffmpeg scale filter (default 'decrease').
- **v_out** – The named filtergraph video output pad. Defaults to `[v]` (see FFmpeg filter documentation).

Returns The generated filtergraph as a string.

`autoscrub.set_terminal_encoding`(*encoding*)

Sets the encoding used for communicating with ffmpeg and ffprobe

Sets the default value for the `encoding` keyword argument passed to `subprocess.Popen`. This chould match your system encoding, and is unlikely to need changing.

`autoscrub.silenceFilterGraph`(*silences*, *factor*, *delay=0.25*, *audio_rate=44100*, *hasten_audio=None*, *silent_volume=1.0*, *v_in='[0:v]'*, *a_in='[0:a]'*, *v_out='[v]'*, *a_out='[a]'*)

Generate a filtergraph string (for processing with the -filter_complex flag of ffmpeg) using the trim and atrim filters to speed up periods in the video designated by a list of silence dictionaries, where each silence dictionary contains keys:

```
silence_start: the timestamp of the detected silent interval in seconds
silence_end:   the timestamp of the detected silent interval in seconds
silence_duration:  duration of the silent interval in seconds
```

Parameters

- **silences** – A list of silence dictionaries generated from getSilences
- **factor** – to speed up video during (a subset of) each silent interval

Keyword Arguments

- **delay** – to omit from silent intervals when changing speed (default 0.25s)
- **audio_rate** – Sample rate of audio input (in Hz, default 44100) used in asetrate/aresample filters when hasten_audio=True
- **hasten_audio** – None, 'pitch' or 'tempo'. Speed up audio during silent segment by either increasing pitch (with asetrate and aresample filters) or tempo (with atempo filter).
- **silent_volume** – scale the volume during silent segments (default 1.0; no scaling)
- **v_in** – The named filtergraph video input pad. Defaults to `[0:v]` (see the FFmpeg filter documentation).
- **a_in** – The named filtergraph audio input pad. Defaults to `[0:a]` (see the FFmpeg filter documentation).
- **v_out** – The named filtergraph video output pad. Defaults to `[v]` (see the FFmpeg filter documentation).
- **a_out** – The named filtergraph audio output pad. Defaults to `[a]` (see the FFmpeg filter documentation).

Returns The generated filtergraph as a string

`autoscrub.trim`(*input_path*, *tstart=0*, *tstop=None*, *output_path=None*, *overwrite=None*, *codec='copy'*, *output_type=None*)

Extract contents of input_path between tstart and tstop.

Parameters **input_path** – The path to the media file to process

Keyword Arguments

- **tstart** – A integer/float in seconds, or a '[hh:]mm:ss[.d]' string (default 0)
- **tstop** – A integer/float in seconds, or a '[hh:]mm:ss[.d]' string (default None)
- **output_path** – Defaults to appending '_trimmed' to input_path
- **overwrite** – Optionally specify addition of -y or -n flag to ffmpeg
- **codec** – Specify the codec to use in the encoding of the output file (default: copy).
- **output_type** – Determines the output file type. Specify as a string containing the required file extension. This is ignored if `output_path` is specified.

Returns The `output_path` where the output of ffmpeg was written.

`autoscrub.`**`trimSegments`**(*input_path*, *trimpts*, *output_path=None*, *output_type=None*, ***kwargs*)

Extract segments of a file using a list of `(tstart, tstop)` tuples. Each segment is saved as a file of the same type as the original.

Parameters **`input_path`** – The path to the media file to process

Keyword Arguments

- **`trimpts`** – A list of `(tstart, tstop)` tuples. See *`trim()`* for the supported formats of `tstart` and `tstop`.
- **`output_path`** – The folder in which to save the segments. Defaults to the folder 'temp' in the current working directory.
- **`output_type`** – Determines the output file type. Specify as a string containing the required file extension.
- **`kwargs`** – A list of additional keyword arguments to pass to *`trim()`*. Note that `tstart`, `tstop` and `output_path` cannot be specified as additional keyword arguments as they are already specified explicitly when `trimSegments` calls `trim`.

Returns A list of paths to each segment created.

`autoscrub.`**`writeFilterGraph`**(*filter_script_path*, *silences*, *factor*, ***kwargs*)

Generates a filtergraph string (using *`autoscrub.generateFilterGraph()`*) and writes it to a file.

Parameters

- **`filter_script_path`** – Path to save the filter script .
- **`silences`** – A list of silence dictionaries generated from *`autoscrub.getSilences()`*.
- **`factor`** – to speed up video during (a subset of) each silent interval.

Keyword Arguments **`kwargs`** – Accepts keyword arguments of *`autoscrub.generateFilterGraph()`*.

Returns `None`

Python Module Index

a

autoscrub, 9

Index

Symbols

–debug
autoscrub-autoprocess command line option, 6
–delay <delay>
autoscrub-autoprocess command line option, 6
autoscrub-make-filtergraph command line option, 8
–re-encode <re_encode>
autoscrub-trim command line option, 8
–start <start>
autoscrub-trim command line option, 8
–stop <stop>
autoscrub-trim command line option, 8
-d, –silence-duration <silence_duration>
autoscrub-autoprocess command line option, 5
autoscrub-identify-silences command line option, 6
autoscrub-make-filtergraph command line option, 7
-h, –hasten-audio <hasten_audio>
autoscrub-autoprocess command line option, 5
autoscrub-make-filtergraph command line option, 7
-l, –target-lufs <target_lufs>
autoscrub-autoprocess command line option, 5
autoscrub-loudness-adjust command line option, 7
autoscrub-make-filtergraph command line option, 7
-p, –pan-audio <pan_audio>
autoscrub-autoprocess command line option, 5
autoscrub-make-filtergraph command line option, 7
-r, –rescale <rescale>
autoscrub-autoprocess command line option, 6
autoscrub-make-filtergraph command line option, 8
-s, –speed <speed>
autoscrub-autoprocess command line option, 6
autoscrub-make-filtergraph command line option, 8
-t, –target-threshold <target_threshold>
autoscrub-autoprocess command line option, 6
autoscrub-identify-silences command line option, 6
autoscrub-make-filtergraph command line option, 8
-v, –silent-volume <silent_volume>
autoscrub-autoprocess command line option, 6
autoscrub-make-filtergraph command line option, 8

A

autoscrub (module), 9
autoscrub-autoprocess command line option
–debug, 6
–delay <delay>, 6
-d, –silence-duration <silence_duration>, 5
-h, –hasten-audio <hasten_audio>, 5
-l, –target-lufs <target_lufs>, 5
-p, –pan-audio <pan_audio>, 5
-r, –rescale <rescale>, 6
-s, –speed <speed>, 6
-t, –target-threshold <target_threshold>, 6
-v, –silent-volume <silent_volume>, 6
input_filepath, 6
output_filepath, 6
autoscrub-display-video-properties command line option
input_filepath, 6
autoscrub-identify-silences command line option
-d, –silence-duration <silence_duration>, 6
-t, –target-threshold <target_threshold>, 6
input_filepath, 7
autoscrub-loudness-adjust command line option
-l, –target-lufs <target_lufs>, 7
input_filepath, 7
output_filepath, 7
autoscrub-make-filtergraph command line option
–delay <delay>, 8
-d, –silence-duration <silence_duration>, 7
-h, –hasten-audio <hasten_audio>, 7
-l, –target-lufs <target_lufs>, 7
-p, –pan-audio <pan_audio>, 7
-r, –rescale <rescale>, 8
-s, –speed <speed>, 8
-t, –target-threshold <target_threshold>, 8
-v, –silent-volume <silent_volume>, 8
input_filepath, 8
autoscrub-process-filtergraph command line option
input_filepath, 8
output_filepath, 8

autoscrub-trim command line option
–re-encode <re_encode>, 8
–start <start>, 8
–stop <stop>, 8
input_filepath, 9
output_filepath, 9

C

concatFileList() (in module autoscrub), 9
concatSegments() (in module autoscrub), 9

F

ffmpeg() (in module autoscrub), 10
ffmpegComplexFilter() (in module autoscrub), 10
ffprobe() (in module autoscrub), 10
findDuration() (in module autoscrub), 10
findLoudness() (in module autoscrub), 10
findSampleRate() (in module autoscrub), 11
findSilences() (in module autoscrub), 11

G

generateFilterGraph() (in module autoscrub), 11
getDuration() (in module autoscrub), 12
getLoudness() (in module autoscrub), 12
getSampleRate() (in module autoscrub), 12
getSilences() (in module autoscrub), 12

H

hhmmssd_to_seconds() (in module autoscrub), 12

I

input_filepath
autoscrub-autoprocess command line option, 6
autoscrub-display-video-properties command line option, 6
autoscrub-identify-silences command line option, 7
autoscrub-loudness-adjust command line option, 7
autoscrub-make-filtergraph command line option, 8
autoscrub-process-filtergraph command line option, 8
autoscrub-trim command line option, 9

M

matchLoudness() (in module autoscrub), 13

O

output_filepath
autoscrub-autoprocess command line option, 6
autoscrub-loudness-adjust command line option, 7
autoscrub-process-filtergraph command line option, 8
autoscrub-trim command line option, 9

P

panGainAudioGraph() (in module autoscrub), 13

R

resizeFilterGraph() (in module autoscrub), 13

S

set_terminal_encoding() (in module autoscrub), 13
silenceFilterGraph() (in module autoscrub), 13

T

trim() (in module autoscrub), 14
trimSegments() (in module autoscrub), 14

W

writeFilterGraph() (in module autoscrub), 15