calibtools Documentation

Release 0.4.0dev1

Rich Wareham

April 25, 2014

Contents

CHAPTER 1

Introduction

This package is a repository for some useful scripts for performing camera calibration in Python. It requires the following modules:

- OpenCV's Python bindings
- moviepy
- numpy

1.1 Installation

The easiest way to install `calibtools` is via `easy_install` or `pip`:

```
$ pip install calibtools
```

If you want to check out the latest in-development version, look at the project's GitHub page. Once checked out, installation is based on setuptools and follows the usual conventions for a Python project:

```
$ python setup.py install
```

(Although the *develop* command may be more useful if you intend to perform any significant modification to the library.)

1.2 Further documentation

There is more documentation available online and you can build your own copy via the Sphinx documentation system:

```
$ python setup.py build_sphinx
```

Compiled documentation may be found in `build/docs/html/`.

CHAPTER 2

The calibtools utlity

The calibtools package provides one command-line utility which is named, unimaginatively, `calibtools`. The command-line interface documentation is reproduced below:

```
Usage:
    calibtools (-h | --help) | --version
    calibtools calib [-v... | --verbose...] [--start=INDEX] [--duration=NUMBER]
        [--skip=NUMBER] [--shape=WxH] [--threshold=NUMBER]
        [--no-stop] <video> [<output>]
    calibtools undistort [-v... | --verbose...] [--start=INDEX]
        [--duration=NUMBER] <calibration> <video> <output>

Common options:
    -h --help               Show a command line usage summary.
    -v --verbose            Be verbose in logging progress. Repeat to increase
                            verbosity.
    --version               Output this tool's version number.

    --start=INDEX           Start processing from frame INDEX (0-based).
    --duration=NUMBER       Read at most NUMBER frames from input.

    <video>                 Read input frames from <video>. See section on
                            specifying video input below.

Calibration options:
    --skip=NUMBER           Only process every NUMBER-th frame. Note that this
                            does not affect the interpretation of --duration. A
                            skip of 10 frames with a duration of 20 will result
                            in 2 frames of output. [default: 1]
    --shape=WxH             Checkerboard has WxH internal corners.
                            [default: 8x6]
    --threshold=NUMBER      Skip boards which are not different by NUMBER from
                            what we have previously seen. A value of 0 will
                            include all boards and a value of 1 will *ignore*
                            all boards save the first one. [default: 0.2]
    --no-stop               Don't automatically stop processing when enough
                            variation in board shape has been observed.
    <output>                Write calibration output in JSON format to <output>.
                            The default behaviour is to write to standard
                            output.

Undistort options:
    <calibration>           A file containing calibration information in JSON
                            format as output by calibtools calib.
    <output>                Write raw RGB24 formatted output frames to <output>.
                            Use - to explicitly specify standard output.

Specifying video input:
    When specifying video input (e.g. via <video>) one can use the filename of
    any file in format which OpenCV can understand. If one uses the form
    device:NUMBER then NUMBER is used as a a live video capture device number
    starting from 0. If one uses the form raw:WxH then raw RGB24 frames of
    width W and height H will be read from standard input. This is particularly
    useful if one wants to use ffmpeg to pipe in video from some capture
    device or video format not known to OpenCV.
```

2.1 Recipes

To pipe video directly from ffmpeg, e.g. to have some fine control over a webcam, one can use:

```
$ ffmpeg -f video4linux2 -input_format mjpeg -s 1280x720 -r 30 -i /dev/video1 \
    -vcodec rawvideo -f rawvideo -pix_fmt rgb24 - \
    | calibtools calib raw:1280x720 -v -o foo.json
```

To convert undistorted video on the fly, one can use:

```
$ calibtools undistort calibration.json video.mp4 - | ffmpeg -y -f rawvideo \
    -pix_fmt rgb24 -s 1920x1080 -i - -r 30 output.mp4
```