ocvu Documentation

Release 0.1.0

Hugo Cachitas

September 01, 2016

Contents

1 Contents: **3**
1.1 Installation 3
1.2 Usage 3
1.3 Reference 3
1.4 Authors 7
1.5 Changelog 7

2 Indices and tables **9**

Python Module Index **11**

OpenCV Utilities

CHAPTER 1

Contents:

1.1 Installation

Dependencies:

- Python 3 (only tested in Python 3.4)
- Numpy
- OpenCV 3.0

Then, at the command line:

```
pip install git+https://github.com/cachitas/ocvu
```

Todo

Deploy on PyPI

1.2 Usage

To use **ocvu** in a project:

```
import ocvu
```

Todo

Add some examples...

1.3 Reference

1.3.1 ocvu

`ocvu.`**`imshow`**(*window_name*, *image*, *border_color=(0, 0, 0)*)

Improves default OpenCV imshow method by adding the possibility to correctly visualize small images without them being stretched. This is accomplished by adding a border around the image.

class `ocvu.Video` (*filepath*, *grayscale=False*)

Bases: `object`

OpenCV Video.

fourcc

4-character code of codec.

fps

Frames per second.

frame_number

Number of the frame that will be read next.

frames

Returns an iterator with all frames.

generate_background_model (*step=None*, *end=None*, *mse_min=50*)

Generates a background model using the median.

Only sufficiently different frames are considered, using the mean squared error method.

Parameters

- **step** – Step to iterate through the video. Default is video FPS rate.
- **end** – Last frame to consider. Default is 2/3 of video length.
- **mse_min** – The minimum error at wich the frame is selected. The lower the error, the more *similar* the two images are.

nframes

Returns the total number of frames.

play (*begin=None*, *end=None*, *step=1*, *window=None*, *wait_time=None*)

read_frame (*number=None*, *grayscale=False*)

Reads the current frame and returns it. You can also ask for a specific frame. Returns a Frame object.

Parameters

- **number** (*int*) – Number of the frame desired. If None, reads the current one.
- **grayscale** (*bool*) – Convert the frame read to grayscale.

show_frame (*number=None*, *window=None*, *resize=False*)

Shows frame *number* in an OpenCV window. Returns the frame read.

size

Returns the size of the video frames: (width, height).

class `ocvu.Contour` (*contour*)

Bases: `object`

Provides a user-friendly object defining a contour in OpenCV. Adapted from: http://answers.opencv.org/question/6043/segmentation-and-contours/

approx

Lorem

area

Contour.area - Area bounded by the contour region

bounding_box

Lorem

centroid

diameter
EquivDiameter: diameter of circle with same area as region

draw(*image*, ***kwargs*)
Draw contour. `kwargs` are passed to `cv2.rectangle` method. Default color is green.

draw_approx(*image*, ***kwargs*)
Draw approximated contour. *kwargs* are passed to *cv2.rectangle* method. Default color is red.

draw_bounding_box(*image*, ***kwargs*)
Draw contour's bounding box on image. *kwargs* are passed to *cv2.rectangle* method. Default color is white.

draw_centroid(*image*, *radius=3*, ***kwargs*)
Draw contour's centroid. *kwargs* are passed to *cv2.circle* method.

draw_ellipse(*image*, ***kwargs*)
Draw contour's ellipse. *kwargs* are passed to *cv2.circle* method.

draw_hull(*image*, ***kwargs*)
Draw convex hull. *kwargs* are passed to *cv2.rectangle* method. Default color is blue.

ellipse
Fits an ellipse and returns the rotated rectangle in which the ellipse is inscribed.

The returned value is the tuple: `((x, y), (major_axis, minor_axis), angle)`

hull
Lorem

moments
Lorem

perimeter
Lorem

position
Top left coordinate of the contour bounding box.

`ocvu.`**find_biggest_contours**(*mask*, *n=1*, *area_min=None*, ***kwargs*)
Returns the *n* biggest contours in *mask*. Returns a list of contours sorted by area in descending order. If you provide a minimal area value, contours returned will be filtered like so.

1.3.2 ocvu.contour module

class `ocvu.contour.`**Contour**(*contour*)
Bases: `object`

Provides a user-friendly object defining a contour in OpenCV. Adapted from: http://answers.opencv.org/question/6043/segmentation-and-contours/

approx
Lorem

area
Contour.area - Area bounded by the contour region

bounding_box
Lorem

centroid

diameter
EquivDiameter: diameter of circle with same area as region

draw(*image*, ***kwargs*)
Draw contour. `kwargs` are passed to `cv2.rectangle` method. Default color is green.

draw_approx(*image*, ***kwargs*)
Draw approximated contour. *kwargs* are passed to *cv2.rectangle* method. Default color is red.

draw_bounding_box(*image*, ***kwargs*)
Draw contour's bounding box on image. *kwargs* are passed to *cv2.rectangle* method. Default color is white.

draw_centroid(*image*, *radius=3*, ***kwargs*)
Draw contour's centroid. *kwargs* are passed to *cv2.circle* method.

draw_ellipse(*image*, ***kwargs*)
Draw contour's ellipse. *kwargs* are passed to *cv2.circle* method.

draw_hull(*image*, ***kwargs*)
Draw convex hull. *kwargs* are passed to *cv2.rectangle* method. Default color is blue.

ellipse
Fits an ellipse and returns the rotated rectangle in which the ellipse is inscribed.

The returned value is the tuple: `((x, y), (major_axis, minor_axis), angle)`

hull
Lorem

moments
Lorem

perimeter
Lorem

position
Top left coordinate of the contour bounding box.

`ocvu.contour.`**find_biggest_contours**(*mask*, *n=1*, *area_min=None*, ***kwargs*)
Returns the *n* biggest contours in *mask*. Returns a list of contours sorted by area in descending order. If you provide a minimal area value, contours returned will be filtered like so.

1.3.3 ocvu.utils module

`ocvu.utils.`**imshow**(*window_name*, *image*, *border_color=(0, 0, 0)*)
Improves default OpenCV imshow method by adding the possibility to correctly visualize small images without them being stretched. This is accomplished by adding a border around the image.

1.3.4 ocvu.video module

Video object

class `ocvu.video.`**Frame**(*number*, *image=None*)
Bases: `object`

image

number

class `ocvu.video.`**`Video`**(*filepath*, *grayscale=False*)

Bases: `object`

OpenCV Video.

`fourcc`

4-character code of codec.

`fps`

Frames per second.

`frame_number`

Number of the frame that will be read next.

`frames`

Returns an iterator with all frames.

`generate_background_model`(*step=None*, *end=None*, *mse_min=50*)

Generates a background model using the median.

Only sufficiently different frames are considered, using the mean squared error method.

Parameters

- **`step`** – Step to iterate through the video. Default is video FPS rate.
- **`end`** – Last frame to consider. Default is 2/3 of video length.
- **`mse_min`** – The minimum error at wich the frame is selected. The lower the error, the more *similar* the two images are.

`nframes`

Returns the total number of frames.

`play`(*begin=None*, *end=None*, *step=1*, *window=None*, *wait_time=None*)

`read_frame`(*number=None*, *grayscale=False*)

Reads the current frame and returns it. You can also ask for a specific frame. Returns a Frame object.

Parameters

- **`number`** (`int`) – Number of the frame desired. If None, reads the current one.
- **`grayscale`** (`bool`) – Convert the frame read to grayscale.

`show_frame`(*number=None*, *window=None*, *resize=False*)

Shows frame *number* in an OpenCV window. Returns the frame read.

`size`

Returns the size of the video frames: (width, height).

1.4 Authors

- Hugo Cachitas - https://github.com/cachitas

1.5 Changelog

1.5.1 0.1.0 (????-??-??)

- Developing...

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

O

ocvu, 3
ocvu.contour, 5
ocvu.utils, 6
ocvu.video, 6

A

approx (ocvu.Contour attribute), 4
approx (ocvu.contour.Contour attribute), 5
area (ocvu.Contour attribute), 4
area (ocvu.contour.Contour attribute), 5

B

bounding_box (ocvu.Contour attribute), 4
bounding_box (ocvu.contour.Contour attribute), 5

C

centroid (ocvu.Contour attribute), 4
centroid (ocvu.contour.Contour attribute), 5
Contour (class in ocvu), 4
Contour (class in ocvu.contour), 5

D

diameter (ocvu.Contour attribute), 5
diameter (ocvu.contour.Contour attribute), 6
draw() (ocvu.Contour method), 5
draw() (ocvu.contour.Contour method), 6
draw_approx() (ocvu.Contour method), 5
draw_approx() (ocvu.contour.Contour method), 6
draw_bounding_box() (ocvu.Contour method), 5
draw_bounding_box() (ocvu.contour.Contour method), 6
draw_centroid() (ocvu.Contour method), 5
draw_centroid() (ocvu.contour.Contour method), 6
draw_ellipse() (ocvu.Contour method), 5
draw_ellipse() (ocvu.contour.Contour method), 6
draw_hull() (ocvu.Contour method), 5
draw_hull() (ocvu.contour.Contour method), 6

E

ellipse (ocvu.Contour attribute), 5
ellipse (ocvu.contour.Contour attribute), 6

F

find_biggest_contours() (in module ocvu), 5
find_biggest_contours() (in module ocvu.contour), 6
fourcc (ocvu.Video attribute), 4
fourcc (ocvu.video.Video attribute), 7
fps (ocvu.Video attribute), 4
fps (ocvu.video.Video attribute), 7
Frame (class in ocvu.video), 6
frame_number (ocvu.Video attribute), 4
frame_number (ocvu.video.Video attribute), 7
frames (ocvu.Video attribute), 4
frames (ocvu.video.Video attribute), 7

G

generate_background_model() (ocvu.Video method), 4
generate_background_model() (ocvu.video.Video method), 7

H

hull (ocvu.Contour attribute), 5
hull (ocvu.contour.Contour attribute), 6

I

image (ocvu.video.Frame attribute), 6
imshow() (in module ocvu), 3
imshow() (in module ocvu.utils), 6

M

moments (ocvu.Contour attribute), 5
moments (ocvu.contour.Contour attribute), 6

N

nframes (ocvu.Video attribute), 4
nframes (ocvu.video.Video attribute), 7
number (ocvu.video.Frame attribute), 6

O

ocvu (module), 3
ocvu.contour (module), 5
ocvu.utils (module), 6
ocvu.video (module), 6

P

perimeter (ocvu.Contour attribute), 5

perimeter (ocvu.contour.Contour attribute), 6
play() (ocvu.Video method), 4
play() (ocvu.video.Video method), 7
position (ocvu.Contour attribute), 5
position (ocvu.contour.Contour attribute), 6

R

read_frame() (ocvu.Video method), 4
read_frame() (ocvu.video.Video method), 7

S

show_frame() (ocvu.Video method), 4
show_frame() (ocvu.video.Video method), 7
size (ocvu.Video attribute), 4
size (ocvu.video.Video attribute), 7

V

Video (class in ocvu), 3
Video (class in ocvu.video), 6