

---

# labvision

*Release 1.0.0*

Feb 12, 2023



---

## Contents

---

<b>1</b>	<b>Contents</b>	<b>3</b>
1.1	Installation . . . . .	3
1.1.1	Initial Installation . . . . .	3
1.1.2	Update . . . . .	3
1.2	Camera . . . . .	3
1.2.1	Camera . . . . .	3
1.2.2	Camera Config . . . . .	3
1.2.3	Digital Cameras . . . . .	4
1.2.4	Quick Timer . . . . .	4
1.3	Images . . . . .	4
1.3.1	Basics . . . . .	4
1.3.2	Contours . . . . .	4
1.3.3	cropping . . . . .	4
1.3.4	Draw . . . . .	4
1.3.5	Feature Detection . . . . .	6
1.3.6	Geometric . . . . .	6
1.3.7	Gui . . . . .	6
1.3.8	Gui Base . . . . .	6
1.3.9	Morphological . . . . .	6
1.3.10	Smoothing . . . . .	6
1.3.11	Thresholding . . . . .	6
1.4	Video . . . . .	6
1.4.1	OpenCV Video . . . . .	6
1.4.2	FFMPEG Video . . . . .	6
1.4.3	Shape and Size . . . . .	6
<b>2</b>	<b>Indices and tables</b>	<b>7</b>
	<b>Python Module Index</b>	<b>9</b>
	<b>Index</b>	<b>11</b>



Lab vision provides a range of tools to collect, manipulate and store images / video.



## 1.1 Installation

The project can be installed and maintained using pip

### 1.1.1 Initial Installation

```
pip install git+https://github.com/MikeSmithLabTeam/labvision
```

### 1.1.2 Update

```
pip install --upgrade git+https://github.com/MikeSmithLabTeam/labvision
```

## 1.2 Camera

Camera contains classes for controlling webcams through opencv and digital cameras via the linux tool gphoto2. The timer provides a simple way to perform timelapse imaging.

### 1.2.1 Camera

### 1.2.2 Camera Config

Web camera settings All settings are accessible via self.cam\_type in CameraBase class Only required settings are 1 resolution and 1 fps. Can add any optional info you want.

```
labvision.camera.camera_config.MIKELAPTOP = {'fps': (20.0,), 'res': (640, 480, 3)}
```

Digital Camera settings Acquisition settings for digital cameras need their own script. See nikon\_config.sh for example in scripts folder. You may need to change permissions to make this file executable.

### 1.2.3 Digital Cameras

### 1.2.4 Quick Timer

## 1.3 Images

Images contains a host of methods for interacting with and manipulating images.

### 1.3.1 Basics

### 1.3.2 Contours

### 1.3.3 cropping

### 1.3.4 Draw

`labvision.images.draw.draw_filled_polygon(im, points, color=(0, 0, 255))`

Adds a filled polygon to an image

#### Parameters

- **im** – np.ndarray of dtype np.uint8
- **points** – N, 2 ndarray of corner points
- **color** – BGR color tuple or images library color

**Returns** im: ndarray

`labvision.images.draw.draw_circle(im, cx, cy, rad, color=(0, 255, 255), thickness=2)`

Draw Circle NEEDS DOCSTRING

#### Parameters

- **im** –
- **cx** –
- **cy** –
- **rad** –
- **color** –
- **thickness** –

**Returns**

`labvision.images.draw.draw_circles(im, circles, color=(0, 255, 255), thickness=2)`

NEEDS DOCSTRING

#### Parameters

- **im** –
- **circles** –
- **color** –
- **thickness** –

**Returns**



labvision.images.draw.**draw\_circles\_with\_scale**(*im*, *circles*, *values*,  
*cmap=<sphinx.ext.autodoc.importer.\_MockObject object>*, *thickness=2*)

NEEDS DOCSTRING

#### Parameters

- **im** –
- **circles** –
- **values** –
- **cmap** –
- **thickness** –

#### Returns

labvision.images.draw.**draw\_delaunay\_tess**(*im*, *points*)

Draws the delaunay tessellation for a set of points on an image

#### Parameters

- **im**(*input image*) – Any number of channels
- **points**(*array of N points*) – Shape (N, 2). *points[:, 0]* contains x coordinates *points[:, 1]* contains y coordinates

**Returns in** – Same shape and type as input image

**Return type** annotated image

labvision.images.draw.**draw\_polygon**(*im*, *vertices*, *color=(0, 0, 255)*, *thickness=1*)

Draws a polygon on an image from a list of vertices

#### Parameters

- **im**(*input image*) – Any number of channels
- **vertices**(*array of N vertices*) –  
**Shape (N, 2) where** *vertices[:, 0]* contains x coordinates *vertices[:, 1]* contains y coordinates
- **color**(*BGR tuple*) – if input image is grayscale then circles will be black
- **thickness**(*int*) – Thickness of the lines

**Returns out** – Same shape and type as input image

**Return type** output image

labvision.images.draw.**draw\_polygons**(*im*, *polygons*, *color=(0, 0, 255)*)

Draws multiple polygons on an image from a list of polygons

#### Parameters

- **im**(*input image*) – Any number of channels
- **polygons**(*array containing coordinates of polygons*) – shape is (P, N, 2) where P is the number of polygons, N is the number of vertices in each polygon. *[:, :, 0]* contains x coordinates, *[:, :, 1]* contains y coordinates.
- **color**(*BGR tuple*) –

**Returns img** – Same shape and type as input image

**Return type** annotated image

`labvision.images.draw.draw_voronoi_cells(im, points)`

Draws the voronoi cells for a set of points on an image

**Parameters**

- **im** (*input image*) – Any number of channels
- **points** (*array of N points*) – Shape (N, 2). `points[:, 0]` contains x coordinates  
`points[:, 1]` contains y coordinates

**Returns** **im** – Same shape and type as input image

**Return type** annotated image

`labvision.images.draw.draw_contours(im, contours, col=(0, 0, 255), thickness=1)`

NEEDS DOCSTRING

**Parameters**

- **im** –
- **contours** –
- **col** –
- **thickness** –

**Returns**

## 1.3.5 Feature Detection

## 1.3.6 Geometric

## 1.3.7 Gui

## 1.3.8 Gui Base

## 1.3.9 Morphological

## 1.3.10 Smoothing

## 1.3.11 Thresholding

## 1.4 Video

Video contains an implementation of OpenCV and FFMPEG to read and write videos. Shape

### 1.4.1 OpenCV Video

### 1.4.2 FFMPEG Video

### 1.4.3 Shape and Size

## CHAPTER 2

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### I

`labvision.camera.camera_config`, [3](#)  
`labvision.images.draw`, [4](#)



## D

`draw_circle()` (*in module labvision.images.draw*), 4  
`draw_circles()` (*in module labvision.images.draw*), 4  
`draw_circles_with_scale()` (*in module labvision.images.draw*), 4  
`draw_contours()` (*in module labvision.images.draw*), 6  
`draw_delaunay_tess()` (*in module labvision.images.draw*), 5  
`draw_filled_polygon()` (*in module labvision.images.draw*), 4  
`draw_polygon()` (*in module labvision.images.draw*), 5  
`draw_polygons()` (*in module labvision.images.draw*), 5  
`draw_voronoi_cells()` (*in module labvision.images.draw*), 6

## L

`labvision.camera.camera_config` (*module*), 3  
`labvision.images.draw` (*module*), 4

## M

`MIKELAPTOP` (*in module labvision.camera.camera\_config*), 3