A Crash Course on SimpleCV

Katherine Scott

SightMachine

kat@sightmachine.com anthony@sightmachine.com

February 27, 2013

Overview

Quick Start!

What is SimpleCV?

- What makes up SimpleCV?

SimpleCV

- Getting Started
- SimpleCV Shell
- iPython Web Notebook

Image Basics

Get Started!

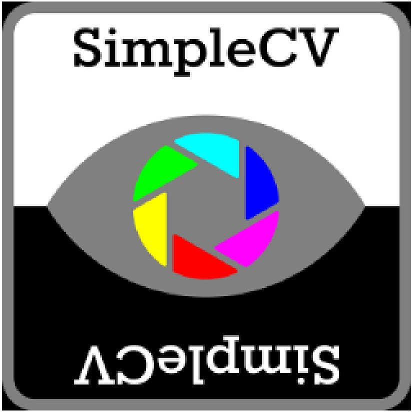
There are a lot of dependencies for SimpleCV and it is a bit tough for beginners. We've brought disks that are ready to go!

- Windows / Linux
  - Boot from USB drive.
  - Alternatively install VirtualBox and the image.
  - `https://www.virtualbox.org/`
- Macs
  - Newer macs are persnikety about booting from a USB drive.
  - Install virtual box and the ISO and go to town.
- When you get home install from SuperPack or preferably source libs.
  - take awhile and is not a perfect science.
  - `https://github.com/ingenuitas/SimpleCV`
  - If you want to contribute this is a great place to start.

About the tutorial

- It will be a lot of live coding. I'll lead, you follow along.
- If you have a question feel free to interrupt.
- If you are having an issue raise a flag. Anthony will help you.

What Makes Up SimpleCV?



SimpleCV != OpenCV

- OpenCV is really busy, we help by wrapping python.
- We add lots of other fun stuff (OCR, Barcodes, etc.)
- We are not competing, we are complementing.
- Purposes are different. Python is great for prototyping. C++ great for embedded.

Core Dependencies

- OpenCV Python Bindings
- Numpy
- SciPy
- SciKits Learn and Orange
- PyGame (this is going away)
- Python Imaging Library (PIL)
- ipython
- PIL (Python Imaging Library)

Optional Dependencies

- Barcodes- Zebra Crossing ZXIng
- Optical Character Recognition (OCR) - Tesseract
- Beautiful Soup
- Kinect Support - freenect
- Unit Tests - nose
- Web Stuff - flask / CherryPy
- Arduino - pyfirmata
- Many Many Many more.

This is why we put everything in a superpack / virtual box / bootable drive

- Just get to the core library functions.
- We encourage you to install the full library when you get home.
- Help is available if you need it.

Getting Help after the tutorial.

- Primary Source: http://help.simplecv.org/questions/
- Documentation http://www.simplecv.org/docs/
- Tweet at us: @Simple_CV
- Another Good Resource: http://www.reddit.com/r/ComputerVision

On the Printed Page

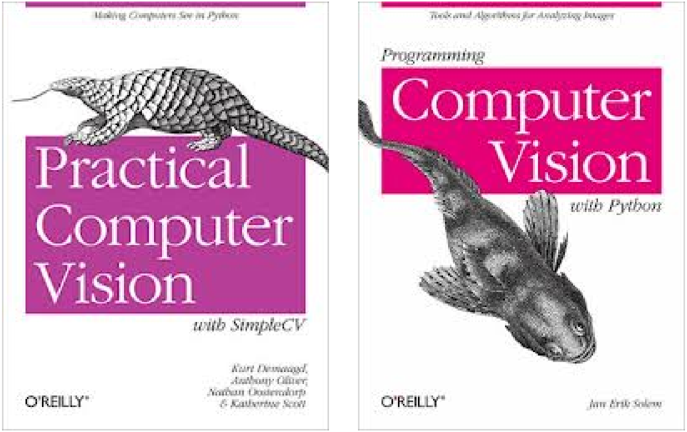

Making Computers See in Python
Practical Computer Vision
with SimpleCV
Kurt Demaagd, Anthony Oliver, Nathan Oostendorp & Katherine Scott
O'REILLY
Tools and Algorithms for Analyzing Images
Programming Computer Vision
with Python
O'REILLY
Jan Erik Solem

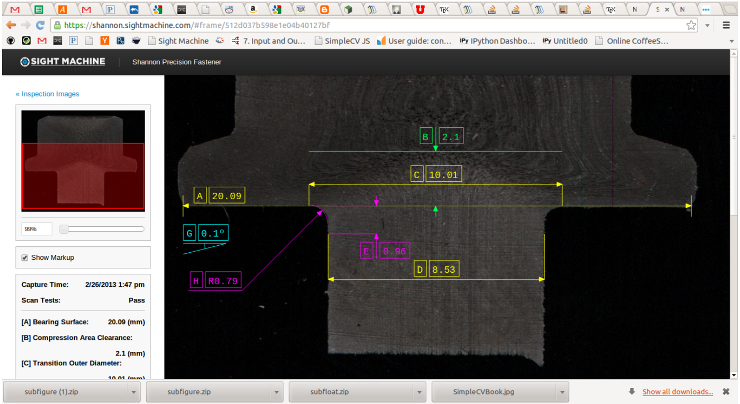
Figure: Two books about using Python for Computer Vision

So why are we doing this?

- We are really nice people who believe in Python and Open Source.
- We are trying to disrupt industrial quality control systems.

SIGHT MACHINE

Early Prototypes



☑ Show Markup

Capture Time: 2/26/2013 1:47 pm

Scan Tests: Pass

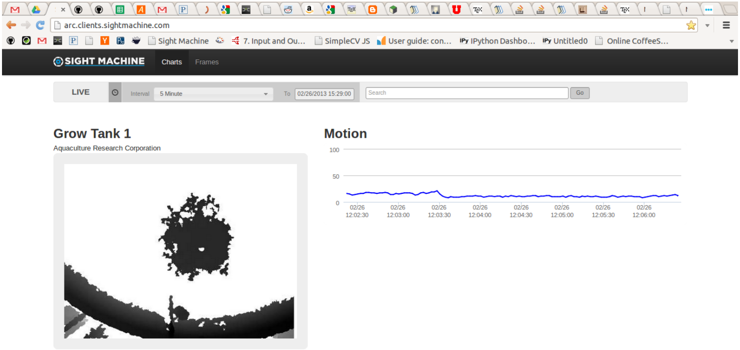
[A] Bearing Surface: 20.09 (mm)

[B] Compression Area Clearance: 2.1 (mm)

[C] Transition Outer Diameter: 10.01 (mm)

Figure: Early Customer - Industrial Fastener Morphology and Metallurgy

Early Prototypes



arc.clients.sightmachine.com

Interval 5 Minute

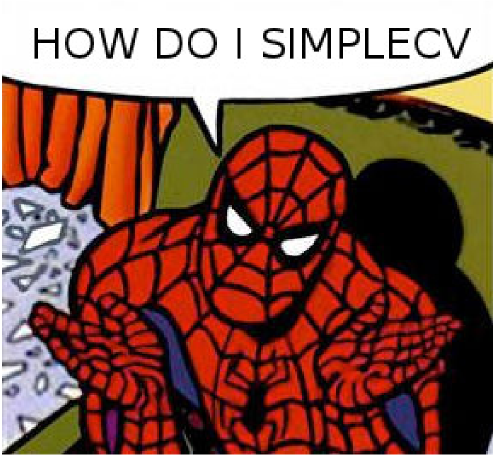
To 02/26/2013 15:29:00

Grow Tank 1

Aquaculture Research Corporation

Figure: Early Customer - Aquaponics Research Facility

How do I SimpleCV?



Where do I write my code?

So how do I SimpleCV?

- In a python file, just like any other library.
- In a command line REPL like iPython.
- In the browser using iPython Notebooks (we'll use this today).

We really like iPython. It is kinda like using Matlab without the $ 5000 per seat license cost.

How does fit into a work flow?

At SightMachine we roughly use these three tools for different parts of our workflow.

Tool	Uses
iPython REPL	Prototypes, Sanity Checks, Etc
iPython Web Notebook	Testing and Development
Python Files	Deployment Code

Table: SimpleCV Workflow

SimpleCV Hello World as a Script

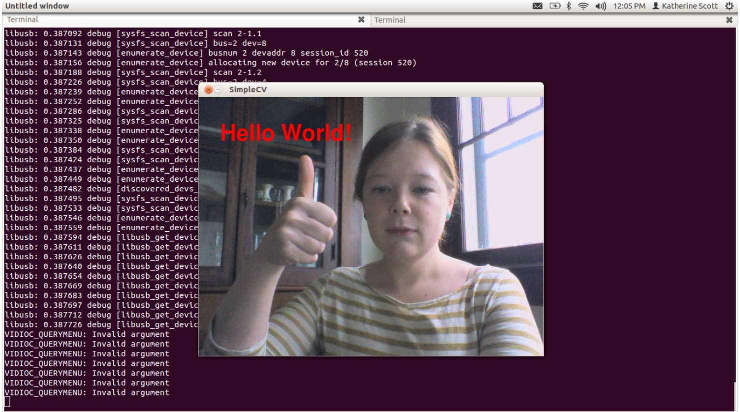
Example (HelloWorld.py)

```python
from SimpleCV import Image, Display, Color, Camera
cam = Camera(0) #Get the first camera
disp = Display((640,480)) # Create a 640x480 Display
while( disp.isNotDone() ):
    img = cam.getImage() # get an image
    # write text at 40,40 font_size 60pts, color is red
    img.drawText("Hello World!",40,40,
                 fontsize=60,color=Color.RED )
    img.save(disp) # show it

```

How do I run Hello World?

- Run the py file with *python HelloWorld.py* in the command.
- Close it by pressing *esc* or *ctrl* – *c*



```
libusb: 0.387092 debug [sysfs_scan_device] scan 2-1.1
libusb: 0.387131 debug [sysfs_scan_device] bus=2 dev=8
libusb: 0.387143 debug [enumerate_device] busnum 2 devaddr 8 session_id 520
libusb: 0.387156 debug [enumerate_device] allocating new device for 2/8 (session 520)
libusb: 0.387188 debug [sysfs_scan_device] scan 2-1.2
libusb: 0.387226 debug [sysfs_scan_device] bus=2 dev=4
libusb: 0.387239 debug [enumerate_device
libusb: 0.387252 debug [enumerate_device
libusb: 0.387286 debug [sysfs_scan_devic
libusb: 0.387325 debug [sysfs_scan_devic
libusb: 0.387338 debug [enumerate_device
libusb: 0.387350 debug [enumerate_device
libusb: 0.387384 debug [sysfs_scan_devic
libusb: 0.387424 debug [sysfs_scan_devic
libusb: 0.387437 debug [enumerate_device
libusb: 0.387449 debug [enumerate_device
libusb: 0.387482 debug [discovered_devs_
libusb: 0.387495 debug [sysfs_scan_devic
libusb: 0.387533 debug [sysfs_scan_devic
libusb: 0.387546 debug [enumerate_device
libusb: 0.387559 debug [enumerate_device
libusb: 0.387594 debug [libusb_get_devic
libusb: 0.387611 debug [libusb_get_devic
libusb: 0.387626 debug [libusb_get_devic
libusb: 0.387640 debug [libusb_get_devic
libusb: 0.387654 debug [libusb_get_devic
libusb: 0.387669 debug [libusb_get_devic
libusb: 0.387683 debug [libusb_get_devic
libusb: 0.387697 debug [libusb_get_devic
libusb: 0.387712 debug [libusb_get_devic
libusb: 0.387726 debug [libusb_get_devic
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument
```

The SimpleCV Shell - Custom iPython REPL

Sometimes you just want to test an idea without writing a full script. For this reason we created the SimpleCV shell, which is a custom ipython instance. The SimpleCV shell will allow you to:

- Test your ideas in a REPL similar to Matlab.
- Access the SimpleCV documentation.
- Import modules that you are working with to test.
- Run through an interactive tutorial.

Starting the SimpleCV Shell

In OSX and Linux just type *simplecv* at the command line. On Windows you just click on the SimpleCV icon.

Example (Shell Basics)

```
+-----------------------------------------------------------+
 SimpleCV 1.3.0 [interactive shell] - http://simplecv.org
+-----------------------------------------------------------+
Commands:
"exit()" or press "Ctrl+ D" to exit the shell
"clear" to clear the shell screen
"tutorial" to begin the SimpleCV interactive tutorial
"example" gives a list of examples you can run
"forums" will launch a web browser for the help forums
"walkthrough" will launch a web browser with a walkthrough
Usage:
dot complete works to show library
for example: Image().save("/tmp/test.jpg") will dot complete
just by touching TAB after typing Image().
Documentation:
help(Image), ? Image, Image ? , or Image() ? all do the same
"docs" will launch webbrowser showing documentation
```

SimpleCV Shell Like a Boss

- Putting a ? in front of a class or method will give you documentation. The "/" key will let you search.
- iPython has tab completion for methods.
- Up arrow will give you previous commands.
- %paste will let you paste formatted code.
- Other cool stuff can be found by googling iPython magic

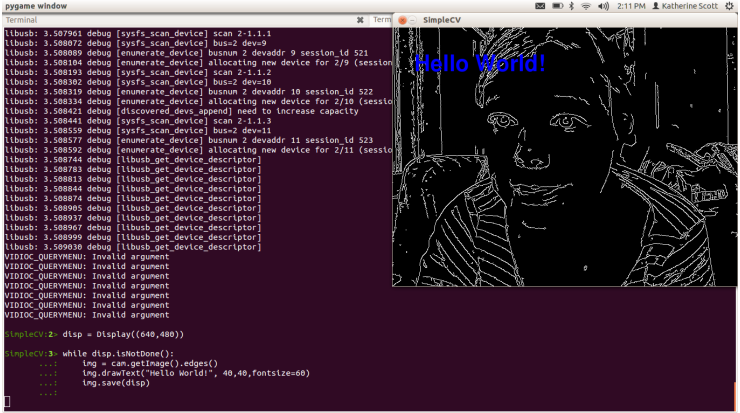
Let's repeat Hello World in SimpleCV Shell

Example (In the SimpleCV shell)

```
SimpleCV:1> cam = Camera()
SimpleCV:2> disp = Display((640,480))
SimpleCV:3> while disp.isNotDone():
       ...:     img = cam.getImage().edges()
       ...:     img.drawText("Hello World!",40,40,fontsize=60)
       ...:     img.save(disp)
       ...:
SimpleCV:4> exit
```

- Just push return after each line.
- iPython will do tabbing in the while loop.
- esc to quit or $ctrl - c$.
- type “exit” to quit.

Yes, it really is that simple.

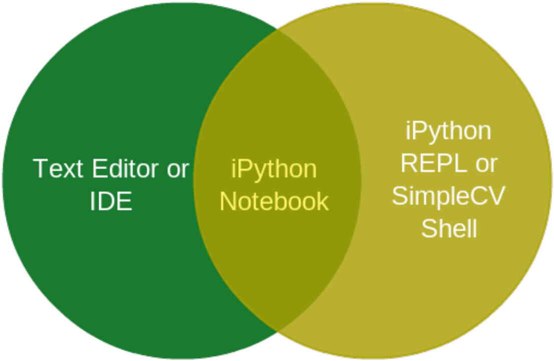


```
libusb: 3.507961 debug [sysfs_scan_device] scan 2-1.1.1
libusb: 3.508072 debug [sysfs_scan_device] bus=2 dev=9
libusb: 3.508089 debug [enumerate_device] busnum 2 devaddr 9 session_id 521
libusb: 3.508104 debug [enumerate_device] allocating new device for 2/9 (session
libusb: 3.508193 debug [sysfs_scan_device] scan 2-1.1.2
libusb: 3.508302 debug [sysfs_scan_device] bus=2 dev=10
libusb: 3.508319 debug [enumerate_device] busnum 2 devaddr 10 session_id 522
libusb: 3.508334 debug [enumerate_device] allocating new device for 2/10 (sessio
libusb: 3.508421 debug [discovered_devs_append] need to increase capacity
libusb: 3.508441 debug [sysfs_scan_device] scan 2-1.1.3
libusb: 3.508559 debug [sysfs_scan_device] bus=2 dev=11
libusb: 3.508577 debug [enumerate_device] busnum 2 devaddr 11 session_id 523
libusb: 3.508592 debug [enumerate_device] allocating new device for 2/11 (sessio
libusb: 3.508744 debug [libusb_get_device_descriptor]
libusb: 3.508783 debug [libusb_get_device_descriptor]
libusb: 3.508813 debug [libusb_get_device_descriptor]
libusb: 3.508844 debug [libusb_get_device_descriptor]
libusb: 3.508874 debug [libusb_get_device_descriptor]
libusb: 3.508905 debug [libusb_get_device_descriptor]
libusb: 3.508937 debug [libusb_get_device_descriptor]
libusb: 3.508967 debug [libusb_get_device_descriptor]
libusb: 3.508999 debug [libusb_get_device_descriptor]
libusb: 3.509030 debug [libusb_get_device_descriptor]
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument

SimpleCV:2> disp = Display((640,480))

SimpleCV:3> while disp.isNotDone():
       ...:     img = cam.getImage().edges()
       ...:     img.drawText("Hello World!", 40,40,fontsize=60)
       ...:     img.save(disp)
       ...:
```

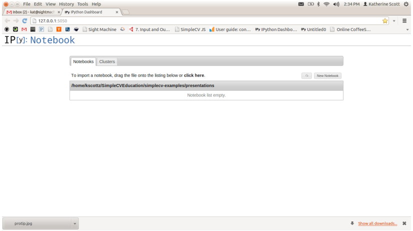
Why use iPython Web Notebooks



[online diagramming & design] creately.com

- Web notebooks give you the best features of an IDE and a REPL!

How do I use the notebook?



IP[y]: Notebook

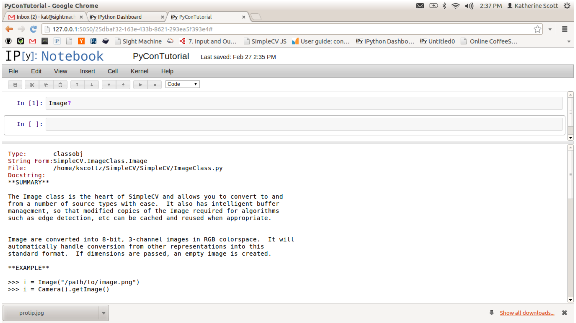
To import a notebook, drag the file onto the listing below or **click here**.

/home/kscottz/SimpleCVEducation/simplecv-examples/presentations

Notebook list empty.

- From the shell just type *simplecv notebook*.
- You will get to a dashboard to create a new notebook.
- By default notebooks are in the path where you start ipython.

How do I use the notebook?



```
In [1]: Image?

In [ ]:
```

```
Type:       classobj
String Form:SimpleCV.ImageClass.Image
File:       /home/kscottz/SimpleCV/SimpleCV/ImageClass.py
Docstring:
**SUMMARY**

The Image class is the heart of SimpleCV and allows you to convert to and
from a number of source types with ease.  It also has intelligent buffer
management, so that modified copies of the Image required for algorithms
such as edge detection, etc can be cached and reused when appropriate.


Image are converted into 8-bit, 3-channel images in RGB colorspace.  It will
automatically handle conversion from other representations into this
standard format.  If dimensions are passed, an empty image is created.

**EXAMPLE**

>>> i = Image("/path/to/image.png")
>>> i = Camera().getImage()
```

- Everything we mentioned about the SimpleCV shell still holds.
- Magic commands, inline documentation, etc. still work.
- *enter* starts a new line.
- *ctrl* — *enter* executes a line.

Caveats about iPython Web Notebooks

- iPython Web Notebooks are still version 0.1.4
- **There is no auto-save. Get in the** $ctrl - s$ **save habit.**
- If you edit a module you import you must restart the core.
- Minimal editing support. No find/replace.
- The core can sometimes crash on large images.
- The notebooks hold on to data by default. This can fill up your version control system fast. Try the download as python command from the gui.

Image Basics

- Most functionality lives in the image class.
- Image takes a single parameter, the file name.
- Takes everything you would like. PNG, BMP, JPG, many more.
- Can also take a URL to an image.
- A couple of built-ins like lenna and simplecv.
- *img.show*() will display the image.

Image Basics

- You can get the image file name using *img.filename*
- Images can also come from appropriately shaped numpy arrays.
- PIL and OpenCV images can also be passed into the image.
- Can also take a URL to an image.
- The img.getEXIFData() command can show jpg EXIF data.

Paragraphs of Text

Sed iaculis dapibus gravida. Morbi sed tortor erat, nec interdum arcu. Sed id lorem lectus. Quisque viverra augue id sem ornare non aliquam nibh tristique. Aenean in ligula nisl. Nulla sed tellus ipsum. Donec vestibulum ligula non lorem vulputate fermentum accumsan neque mollis.

Sed diam enim, sagittis nec condimentum sit amet, ullamcorper sit amet libero. Aliquam vel dui orci, a porta odio. Nullam id suscipit ipsum. Aenean lobortis commodo seDerkt commodo leo gravida vitae. Pellentesque vehicula ante iaculis arcu pretium rutrum eget sit amet purus. Integer ornare nulla quis neque ultrices lobortis. Vestibulum ultrices tincidunt libero, quis commodo erat ullamcorper id.

Bullet Points

- Lorem ipsum dolor sit amet, consectetur adipiscing elit
- Aliquam blandit faucibus nisi, sit amet dapibus enim tempus eu
- Nulla commodo, erat quis gravida posuere, elit lacus lobortis est, quis porttitor odio mauris at libero
- Nam cursus est eget velit posuere pellentesque
- Vestibulum faucibus velit a augue condimentum quis convallis nulla gravida

Multiple Columns

Heading

1. Statement
2. Explanation
3. Example

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer lectus nisl, ultricies in feugiat rutrum, porttitor sit amet augue. Aliquam ut tortor mauris. Sed volutpat ante purus, quis accumsan dolor.

Table

Treatments	Response 1	Response 2
Treatment 1	0.0003262	0.562
Treatment 2	0.0015681	0.910
Treatment 3	0.0009271	0.296

Table: Table caption

Theorem

Theorem (Mass–energy equivalence)

$E = mc^2$

Verbatim

Example (Theorem Slide Code)

```
def doStuff(a,b,c=[1,2,3]):
    a = 5
    b = a
    c.reverse()

derp = [1,2,3,4]
for i in derp:
   doStuff()
   pass
print deep
```

Verbatim

Example (Theorem Slide Code)

```
$E = mc^2$
```

Figure

Uncomment the code on this slide to include your own image from the same directory as the template .TeX file.

Citation

An example of the `\cite` command to cite within the presentation:

This statement requires citation [Smith, 2012].

References

John Smith (2012)
Title of the publication
Journal Name 12(3), 45 – 678.

The End