

The background image shows an aerial view of a city during sunset or sunrise. The sky is filled with warm orange and yellow hues. Below, a large body of water is visible, with several boats and small islands. The city's buildings are densely packed, particularly along the waterfront. In the foreground, there's a mix of urban structures, green spaces, and what appears to be an industrial or port area with various facilities and roads.

# Balloon Mapping with Raspberry Pi

Getting Started...

# Special thanks...



p o l a k v a n b e k k u m

# Introductions - Me



# Introductions - You

- What is your name?
- What do you?
- Why are you here?
- What are you looking to walk away with?



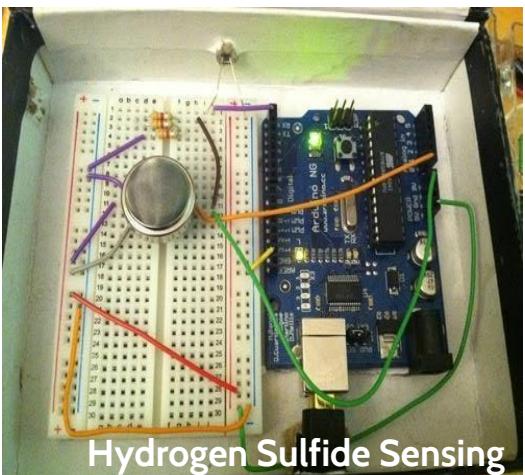
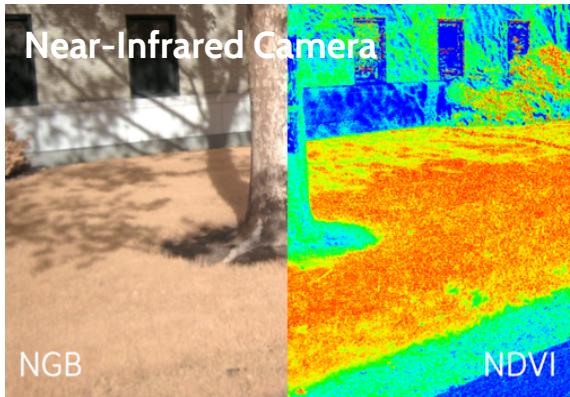


Image Credits: Public Lab



Infragram is an online tool for analyzing plant health with near-infrared imagery! To learn more, [read about our open source cameras](#), or [buy a modded camera](#) from the Public Lab non-profit.

Select an Infragblue image:

[Browse...](#)

No file selected.

Or use a USB webcam [\[tutorial\]](#):

[Activate webcam](#)

Presets:

Raw

NDVI

NIR

Infragrammar:

RGB

R

G

B

[Run](#)

[Save](#)

Color maps:

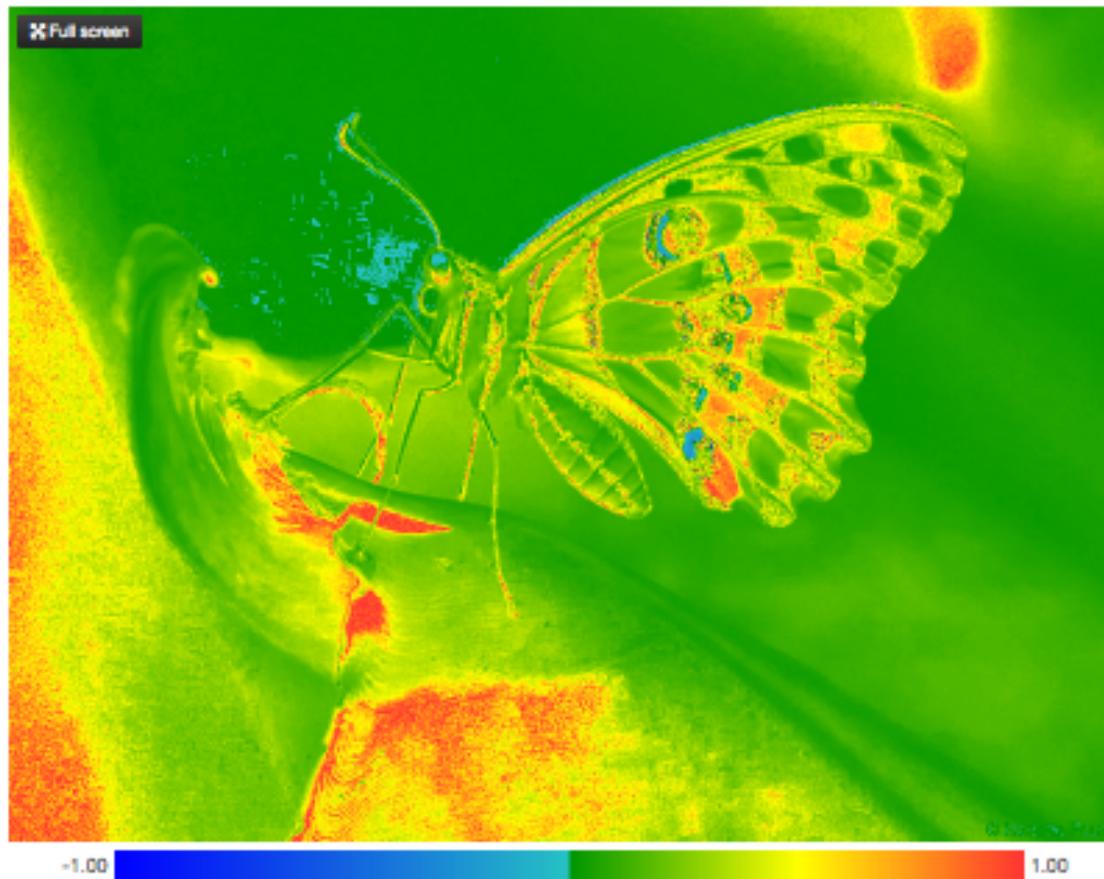
[Greyscale](#)

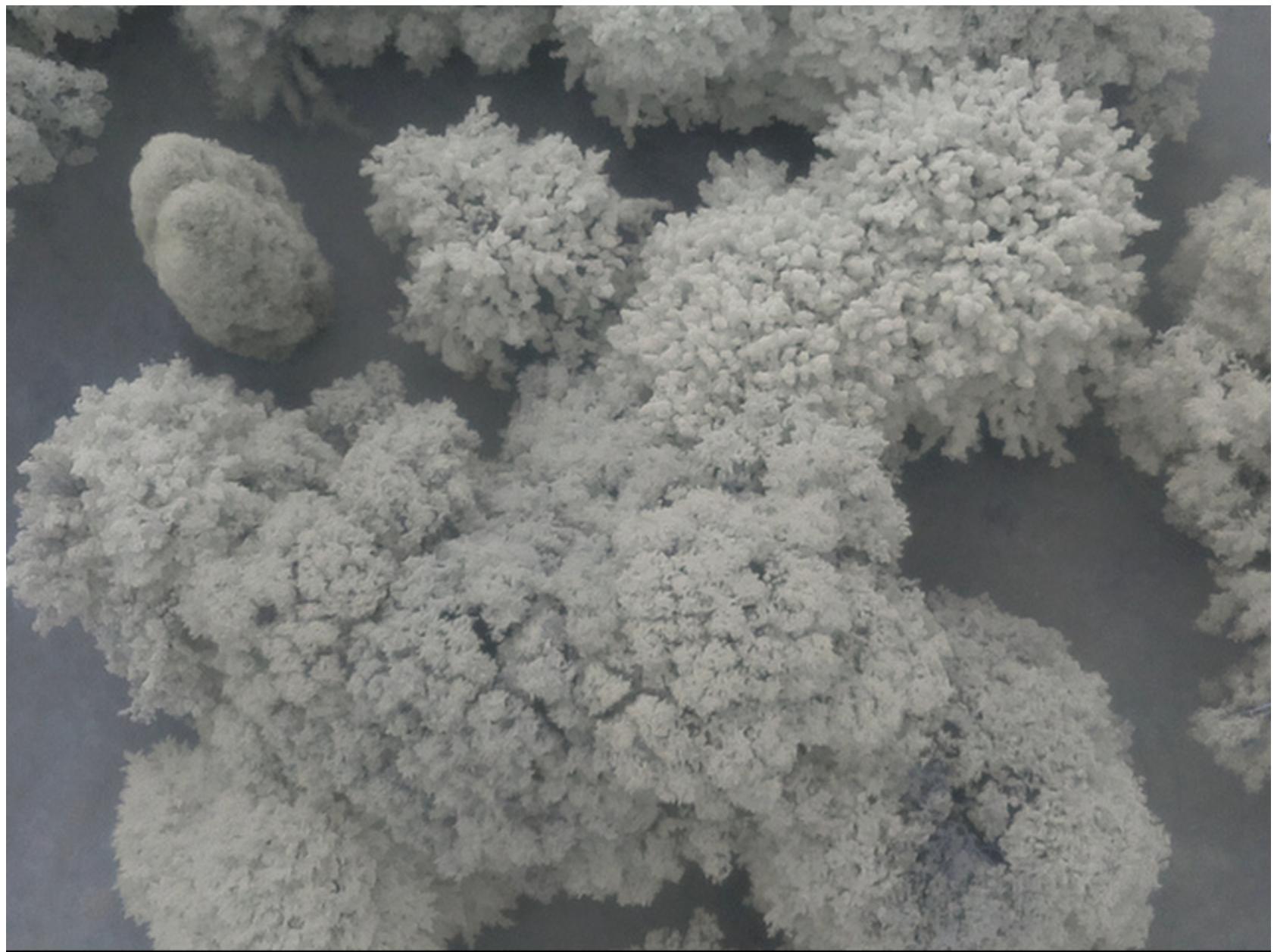
[Color](#)

Slider (X):



[Help](#)





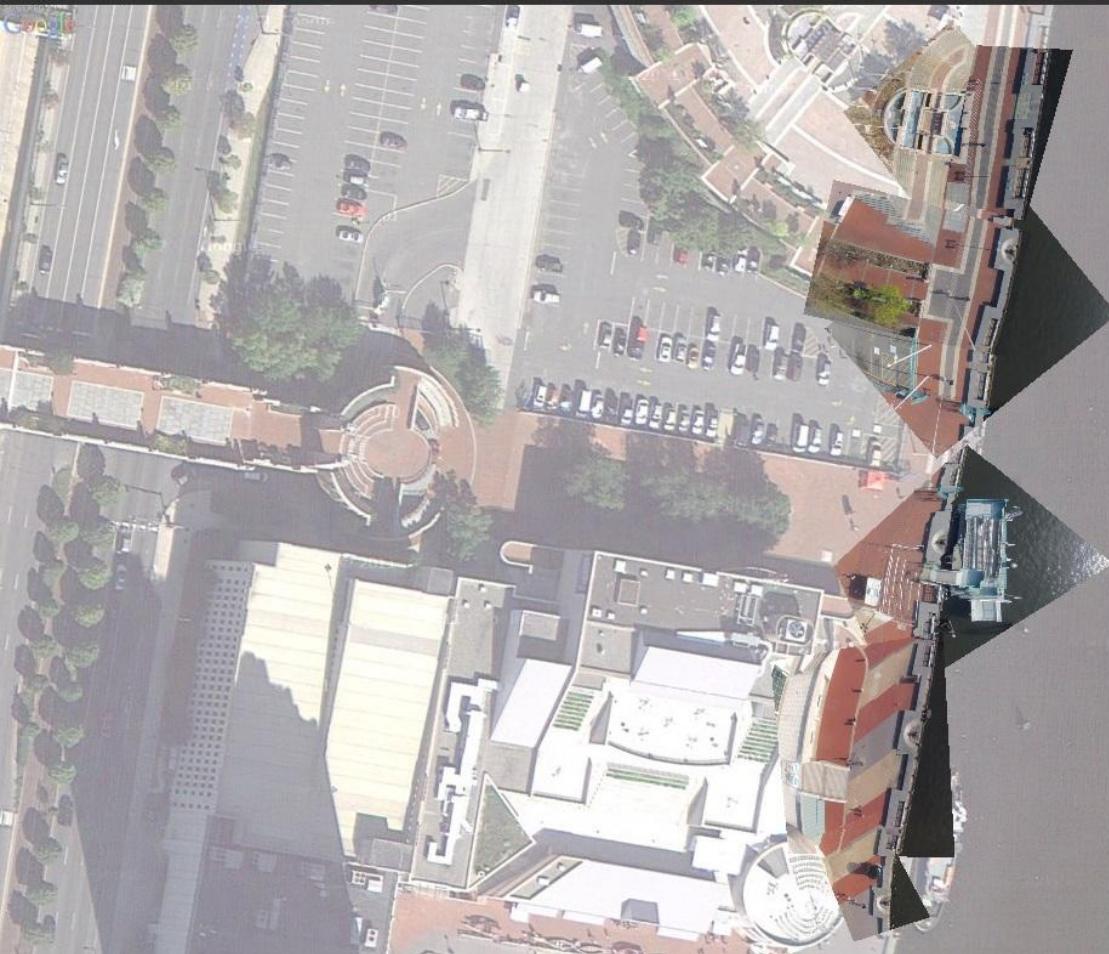


# MapKnitter.org





Save



Imagery ©2013 DigitalGlobe - Terms of Use

+

-



Image Source: Dana Bauer/Public Labs



Image Source: Dana Bauer/Public Labs



Grassroots mappers monitor  
burnoffs during the Deepwater  
Horizon oil spill



Wendy Lady

LA  
1230  
FL

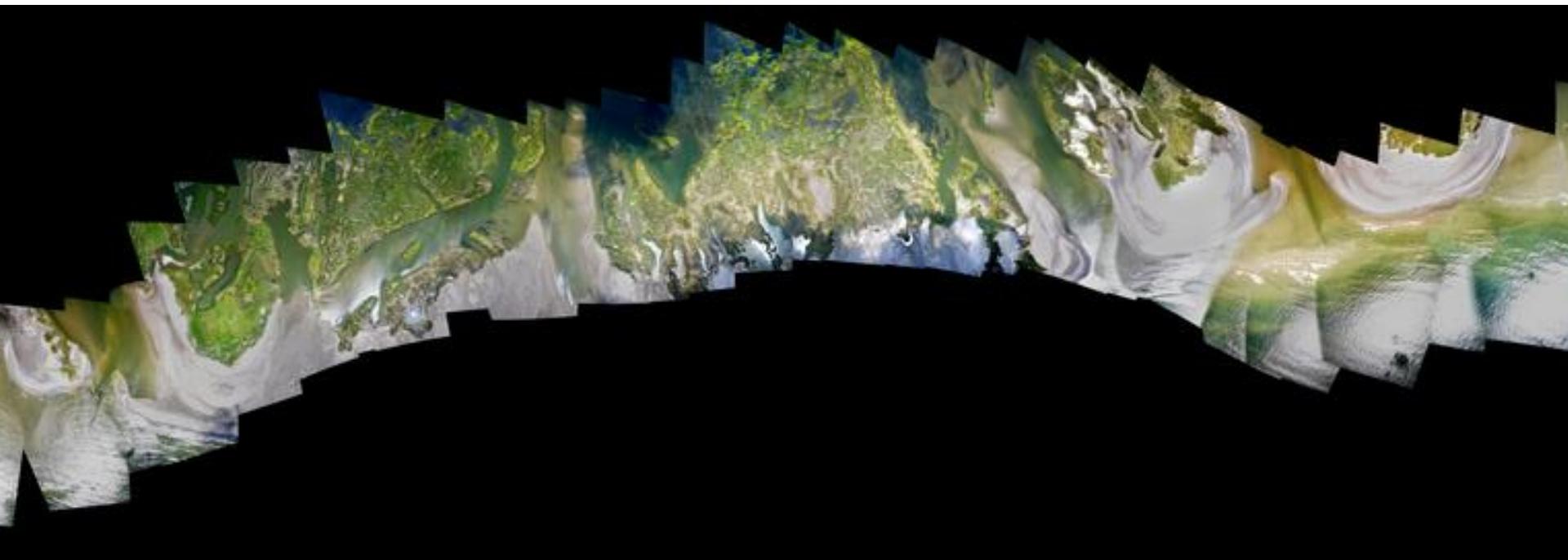
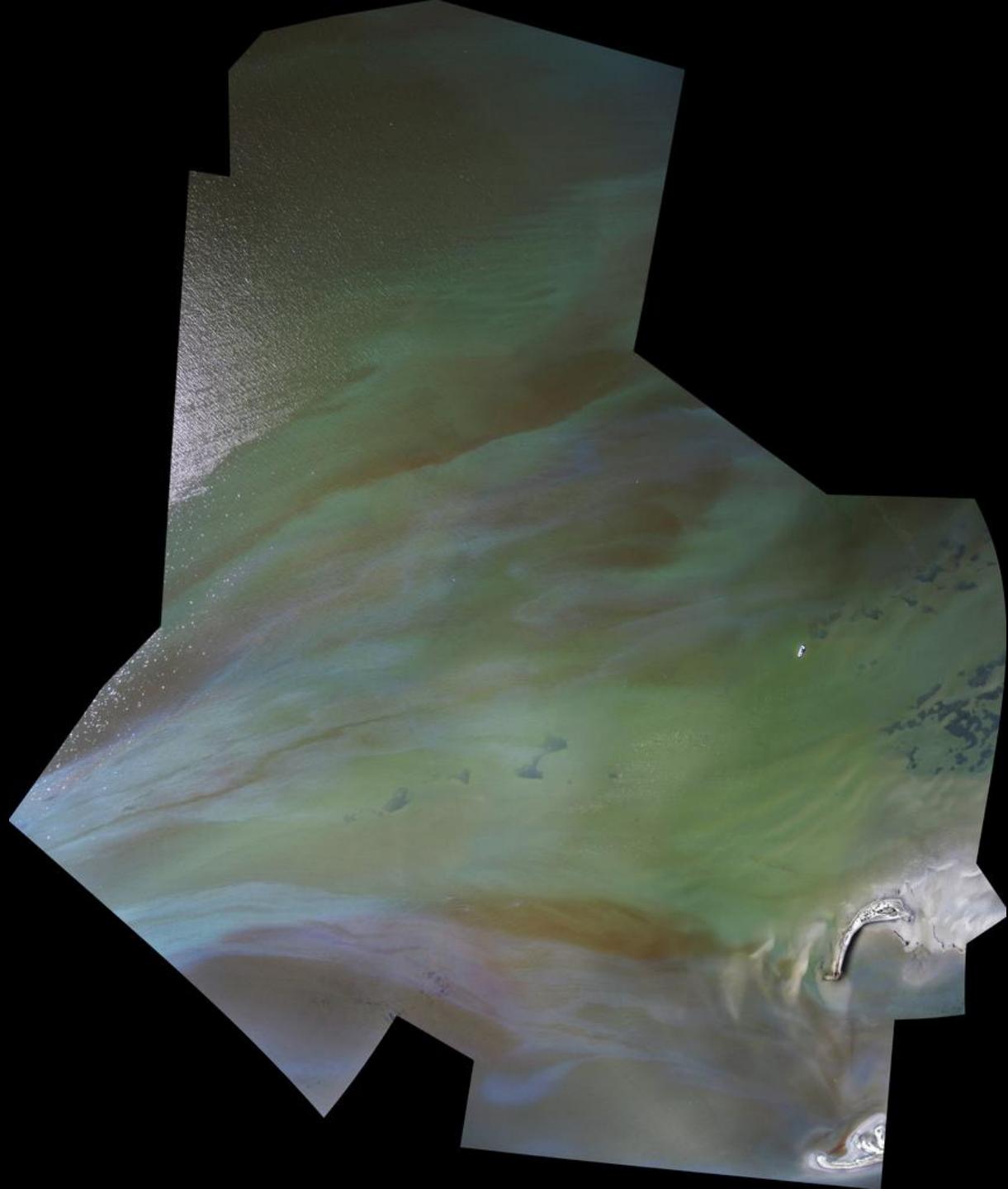


Image Source: [Stewart Long](#)

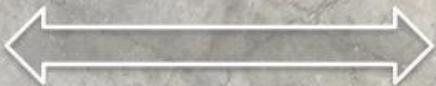


Image Source: [Stewart Long](#)





**+/- 8 feet  
between cracks**







OccupyOakland, November 2, 2011: 10am



# Flight Platform - Balloon



# Flight Platform - Balloon

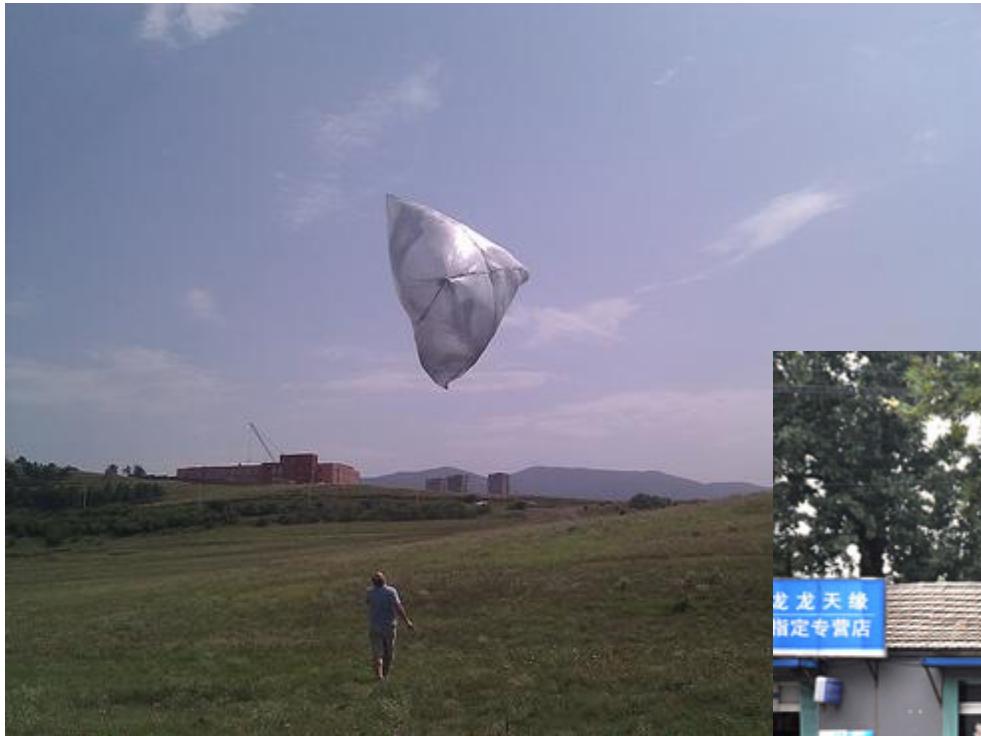


Image Source: [Grassroots Mapping](#)

# Flight Platform - Kites



Image Source: [Public Lab](#)

# Flight Platform - DIY Kites



Image Source: [Public Lab](#)

# Flight (?) Platform - Pole



Image Source: [250 Miles Crossing Philadelphia](#)



Image Source: [250 Miles Crossing Philadelphia](#)

# Camera Housing



# Camera Housing



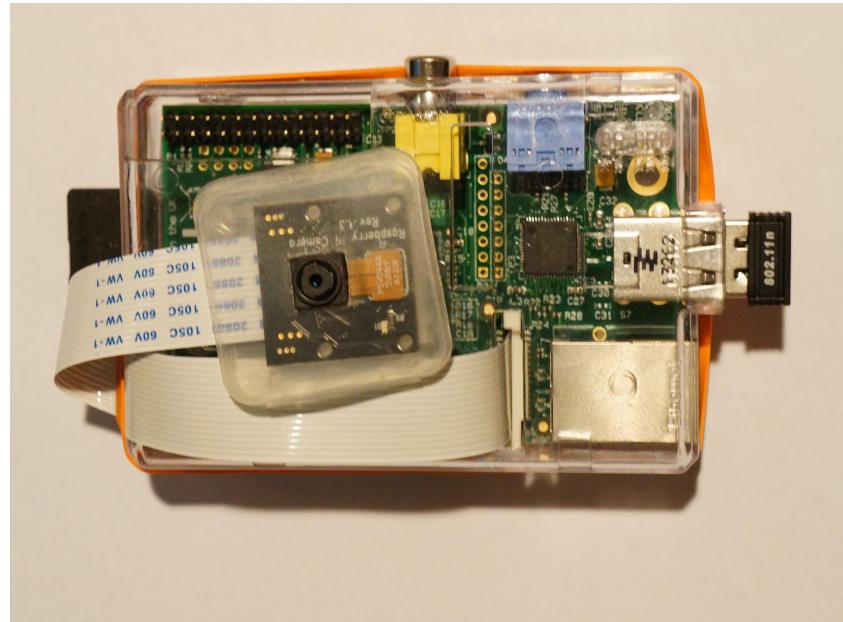
Image Source: [Public Lab](#)

# Camera Housing



Image Source: [Public Lab](#)

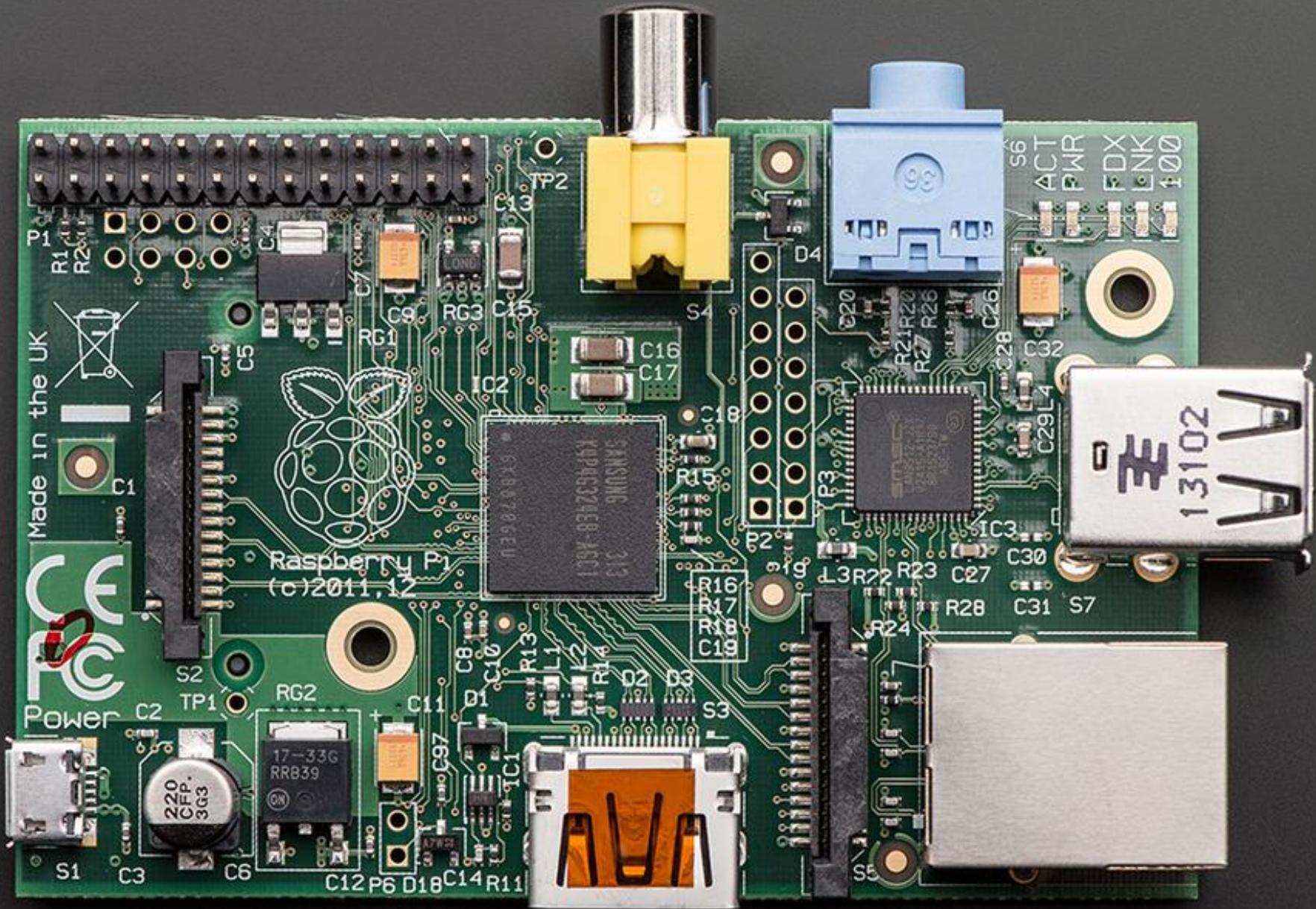
# Cameras



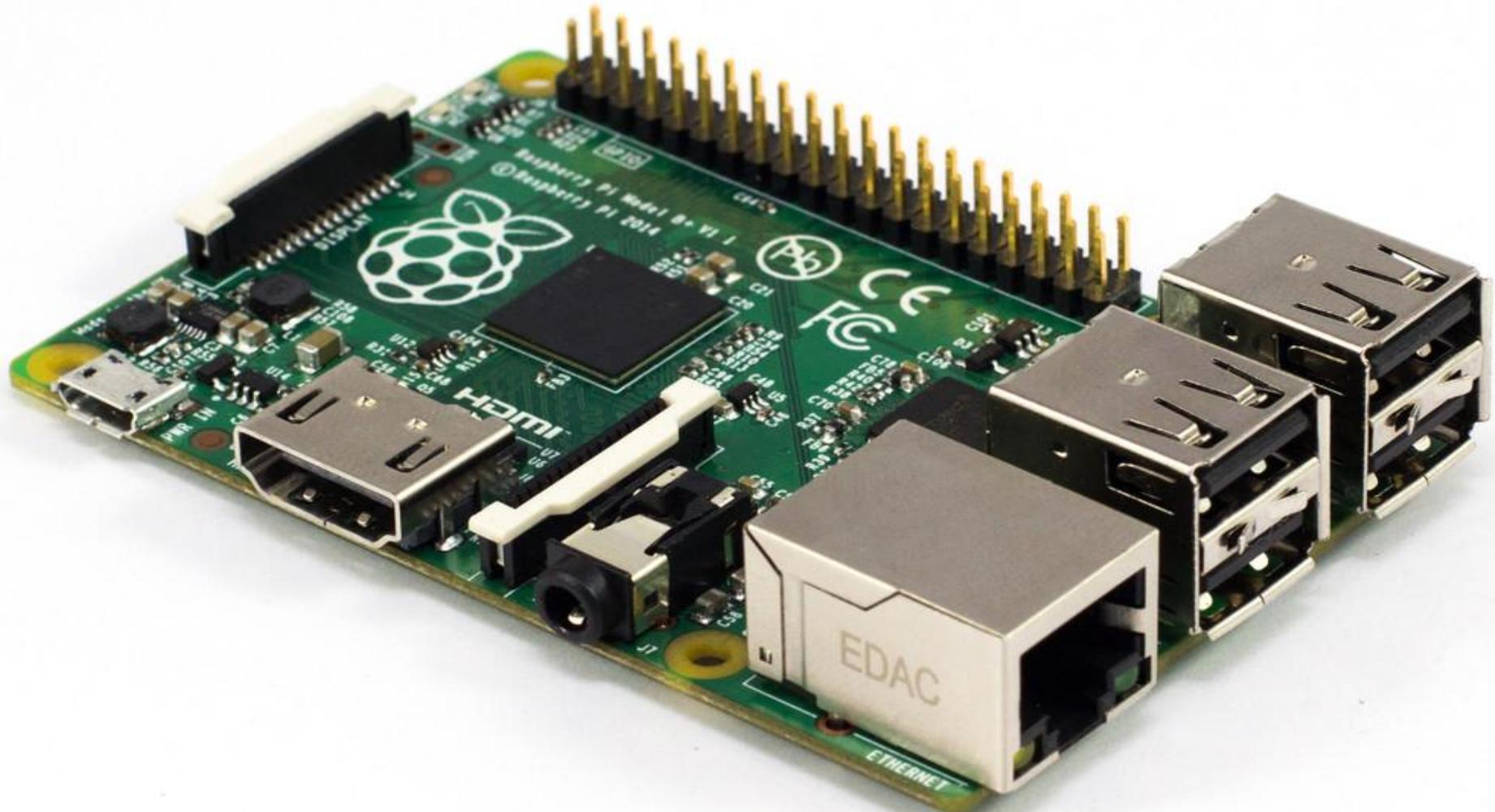




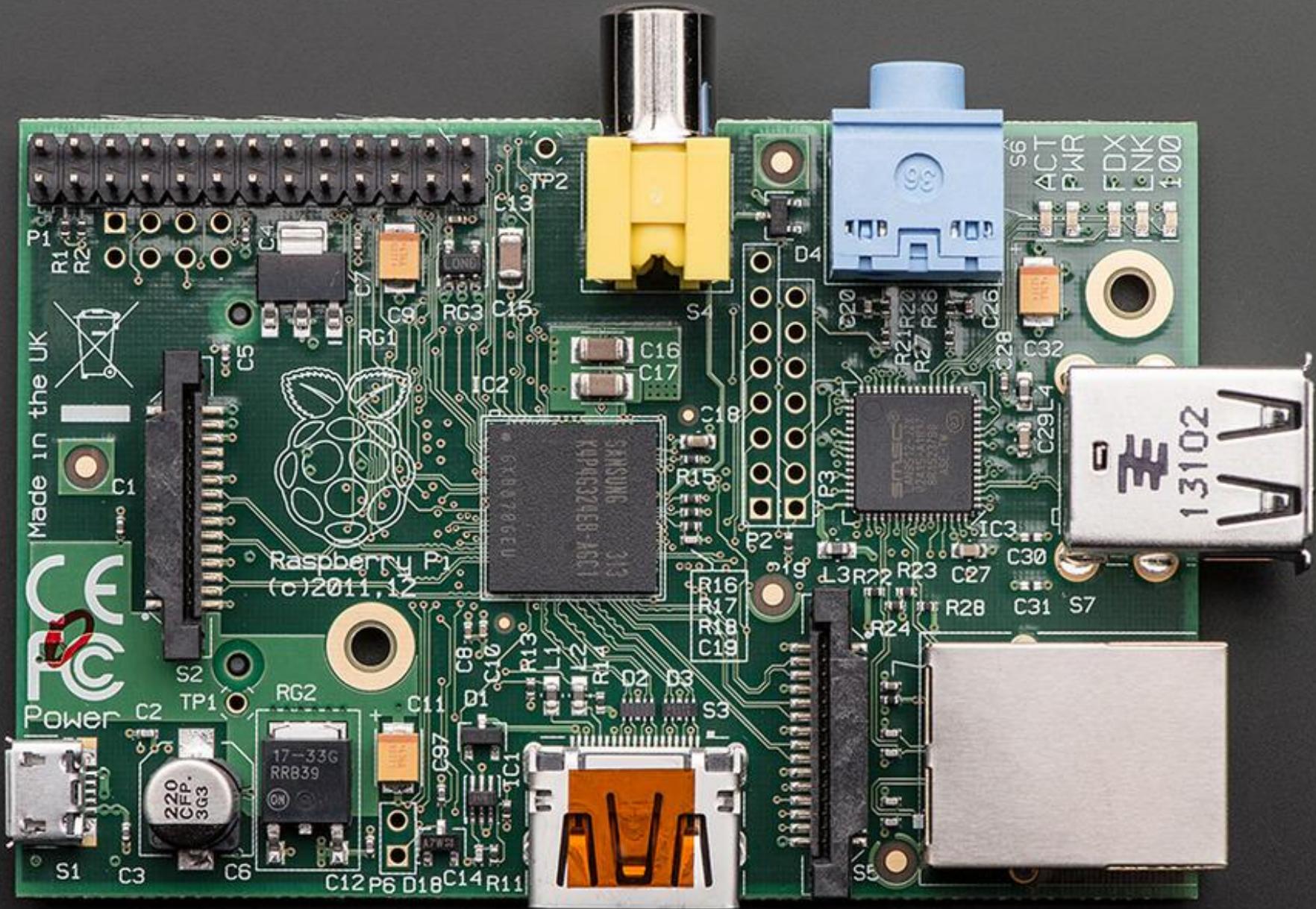
# Raspberry Pi - Model A



# Raspberry Pi - Model B



# Raspberry Pi - Model B+

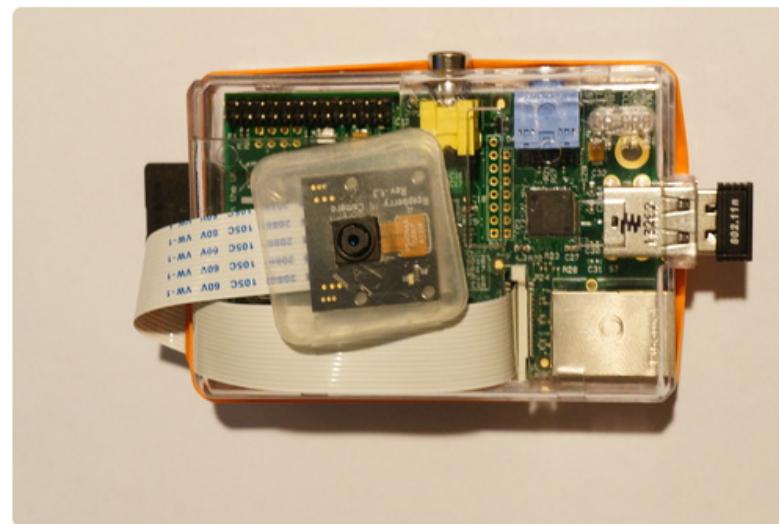


# Raspberry Pi - Model B



[Write a research note](#) [Post related work »](#)[The Homebrew Oil Testing ...](#)

29% funded    \$7,415 pledged    5 days to go



## Raspberry Pi Camera Rig

 [Follow](#)  [Like \(2\)](#) ▾

by SeanKMcGinnis | March 12, 2014 22:31 | 2,516 views | 3 comments

I have been playing with the Raspberry Pi for a while now and I thought it was about time to document what I have been working on. The following steps will take you from getting a brand new Raspberry Pi to having a camera rig to take images of whatever you are looking to monitor.

### Parts & Components %

The following is a list of components you can purchase and build the unit I document here. I had a few of the components, but have provided links to them so you can get an estimate of what you might need.

As a note: I did not include a monitor/display, keyboard and mouse, but you'll need them for obvious reasons.

[Raspberry Pi Basic Starter Kit](#) - This had all of the components I needed to get started including:

- Raspberry Pi B (*I have used a Raspberry Pi A, but the extra memory of the B makes for a more robust platform*)

### Recent topic contributors

- [manetomapping](#)
- [geraldmc](#)
- [liz](#)
- [cfastie](#)
- [fliperbaker](#)

### Related wiki pages

[How to Make a Raspberry Pi Camera Rig](#)

<http://bit.ly/RaspiCameraRig>

# How To Capture Images



- 1. Python 2.7+ Installation
- 2. Python 3.2+ Installation
- 3. Quick Start
- 4. Basic Recipes
- 5. Advanced Recipes
- 6. Frequently Asked Questions (FAQ)
- 7. Camera Hardware
- 8. Deprecated Functionality
- 9. API Reference
- 10. Array Extensions
- 11. Change log
- 12. License

## picamera

This package provides a pure Python interface to the [Raspberry Pi camera](#) module for Python 2.7 (or above) or Python 3.2 (or above).

## Links

- The code is licensed under the [BSD license](#)
- The [source code](#) can be obtained from GitHub, which also hosts the [bug tracker](#)
- The [documentation](#) (which includes installation, quick-start examples, and lots of code recipes) can be read on ReadTheDocs
- Packages can be downloaded from [PyPI](#), but reading the installation instructions is more likely to be useful

## Table of Contents

- [1. Python 2.7+ Installation](#)
  - [1.1. Firmware upgrades](#)
  - [1.2. Raspbian installation](#)
  - [1.3. User installation](#)
  - [1.4. System installation](#)
  - [1.5. Virtualenv installation](#)
  - [1.6. Development installation](#)
  - [1.7. Test suite](#)
- [2. Python 3.2+ Installation](#)
  - [2.1. Firmware upgrades](#)

PyScripter - C:\Users\Sean\Documents\GitHub\raspberry-pi-camera\picamera-time-lapse.py

File Edit Search View Project Run Tools Help

File Explorer

Computer OS (C:) DVD RW D Microsoft

```
• import time
• import picamera

#####
## Capture Images with Picamera
## delay      - REQUIRED - time delay to start taking pictures after initializing
## flight_duration - REQUIRED - estimated length of flight in minutes
## image_interval - REQUIRED - delay between pictures
#####

def captureImages(delay, flight_duration, image_interval):
    camera = picamera.PiCamera()
    true_interval = image_interval - 3
    image_count = (flight_duration * 60)/image_interval
    print 'Going to Capture %s images over %s minutes with %s seconds between images' % (image_count, flight_duration, image_interval)
    time.sleep(delay)
    try:
        for x in range(image_count):
            timestr = time.strftime("%Y%m%d-%H%M%S")
            camera.capture('ImageOutput/img' + timestr + '.jpg')
            time.sleep(true_interval)
    finally:
        camera.close()
        print 'Image capture session ended'
```

picamera-time-lapse.py

Python Interpreter

Call Stack Variables Watches Breakpoints Output Messages Python Interpreter



# SeanKMcGinnis / raspberry-pi-camera

[Unwatch 1](#)[Star 0](#)[Fork 0](#)**Description****Website**

Short description of this repository

Website for this repository (optional)

[Save](#) or [Cancel](#)

4 commits

1 branch

0 releases

1 contributor



branch: master

[raspberry-pi-camera](#) / **Include documentation**

SeanKMcGinnis authored 3 minutes ago

latest commit 67814ad23e

[.gitattributes](#)

Initial Commit

7 months ago

[.gitignore](#)

Initial Commit

7 months ago

[picamera-time-lapse.py](#)

Include documentation

3 minutes ago

We recommend [adding a README](#) to this repository to help give people an overview of your project.

[Add a README](#)**HTTPS clone URL**<https://github.com/>

You can clone with [HTTPS](#), [SSH](#), or [Subversion](#).

[Clone in Desktop](#)[Download ZIP](#)

<https://github.com/SeanKMcGinnis/raspberry-pi-camera>

# Call Script in the Field

- Tethering
- JuiceSSH



07:40 Sat Oct 18, 2014



```
software;
the exact distribution terms for each program are described in th
e
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Oct 18 10:38:55 2014 from 192.168.43.1
pi@raspberrypi ~ $ dir
Desktop  image  ocr_pi.png  python_games  RaspiCamera
pi@raspberrypi ~ $ cd RaspiCamera
pi@raspberrypi ~/RaspiCamera $ dir
ImageOutput  RaspiTimelapse.py  RaspiTimelapse.pyc
pi@raspberrypi ~/RaspiCamera $ python
Python 2.7.3 (default, Mar 18 2014, 05:13:23)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more informa
tion.
>>> import RaspiTimelapse as rt
>>> rt.captureImages(1,1,5)
Going to Capture 12 images over 1 minutes with 5 seconds between
images
Image capture session ended
>>> █
```



# Upcoming - 3D Printed Case & Housing

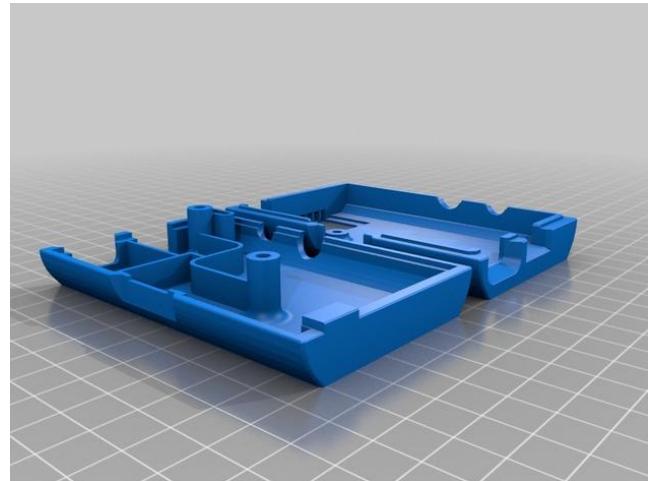
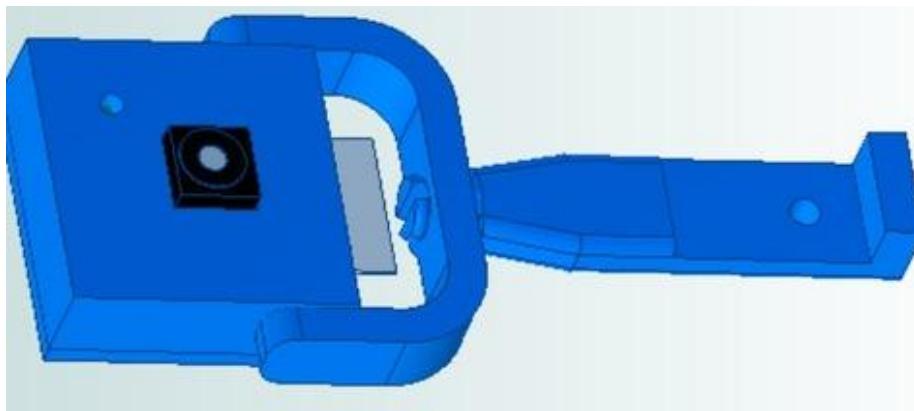
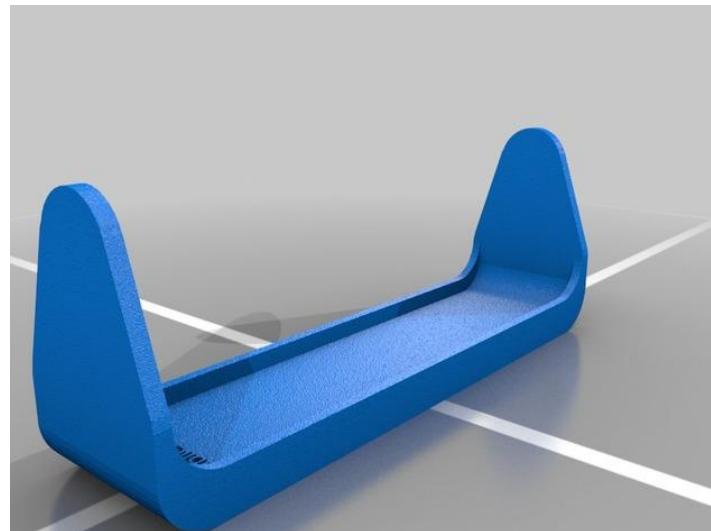
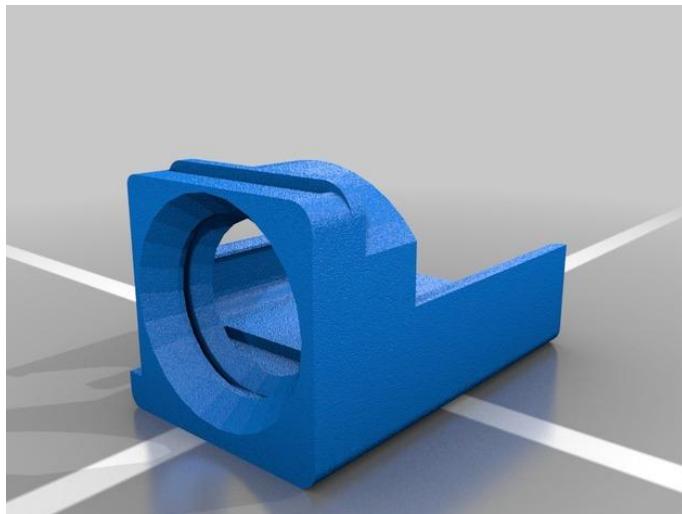


Image Source: [Thingiverse](#)

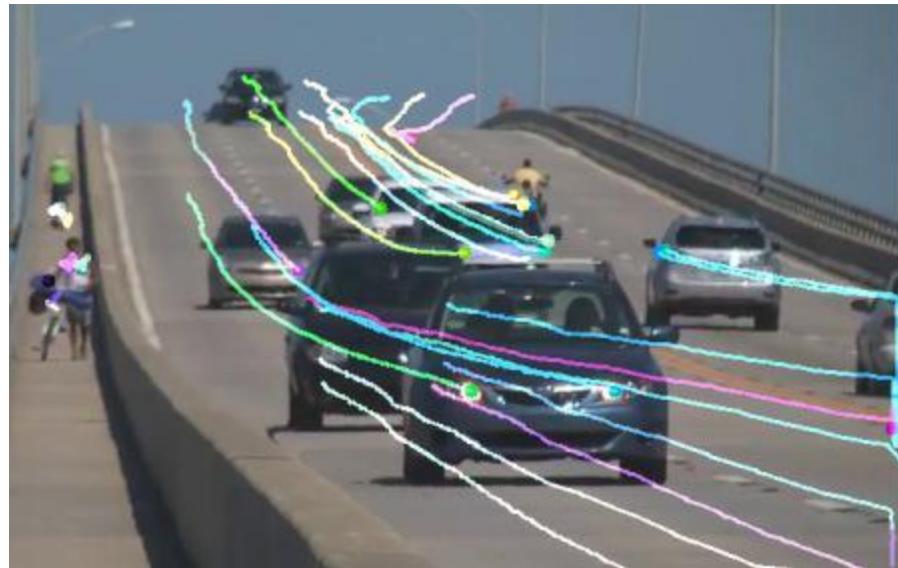
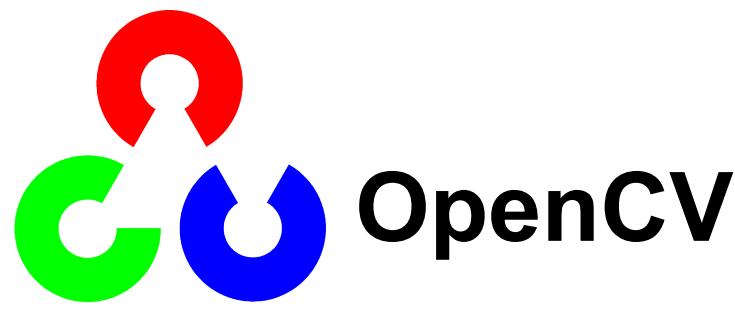


Image Source: [OpenCV](#)

# Global Positioning System (GPS)





# FAA Guidelines

- 101.1a (1) Except as provided for in §101.7, any balloon that is moored to the surface of the earth or an object thereon and that has a diameter of more than 6 feet or a gas capacity of more than 115 cubic feet.
- (2) Except as provided for in §101.7, any kite that weighs more than 5 pounds and is intended to be flown at the end of a rope or cable.”

# FAA Guidelines w/in 5 miles of Airport

When you are within 5 miles of an airport, you need to file a Notice to Airmen (NOTAM), for instance , fill out this [PDF request for FAA Form 7711-2 -- a waiver for moored balloons and kites.](#)

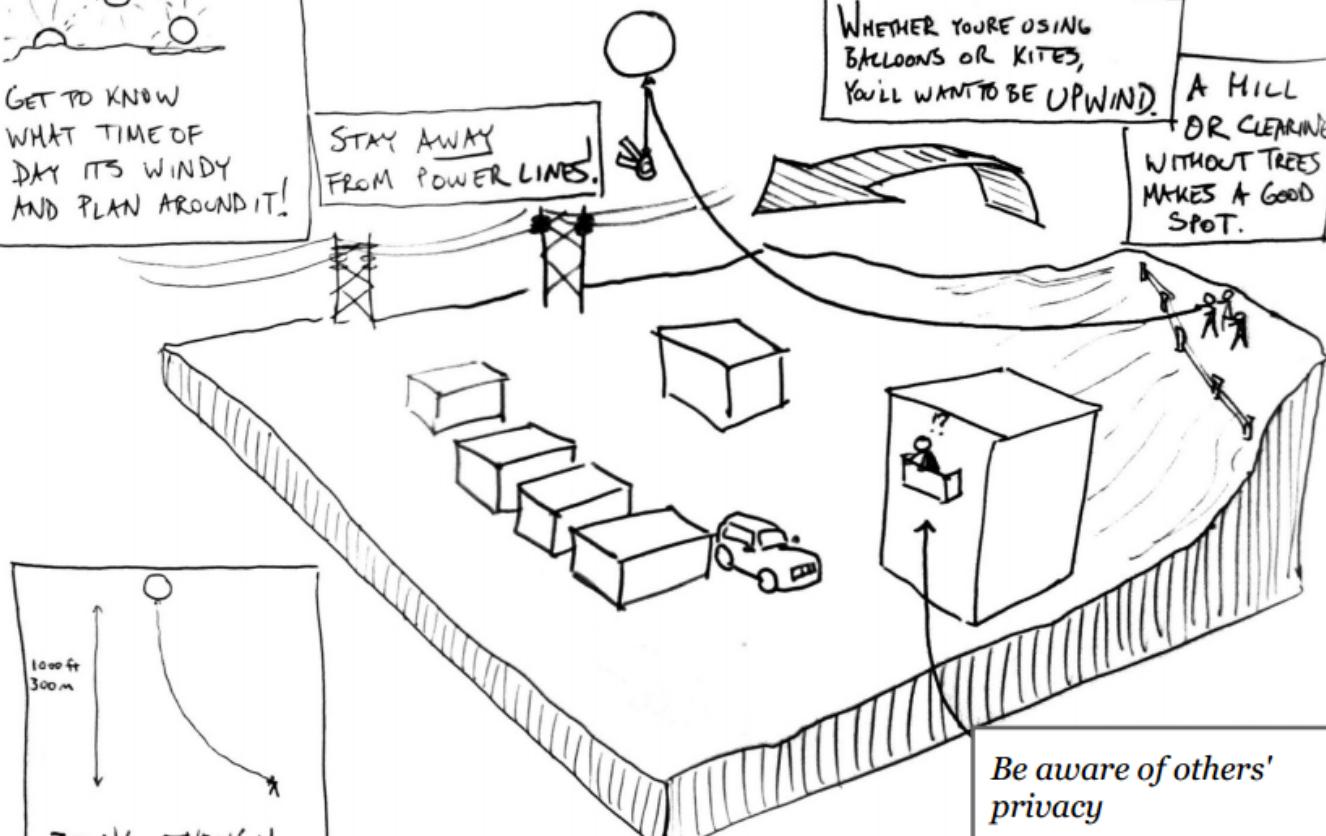
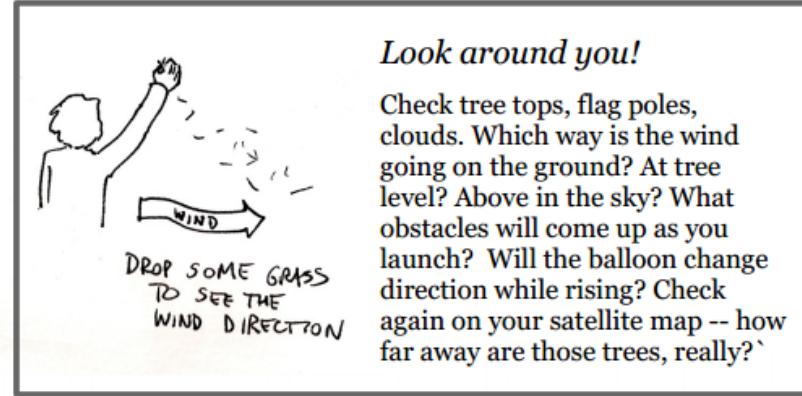
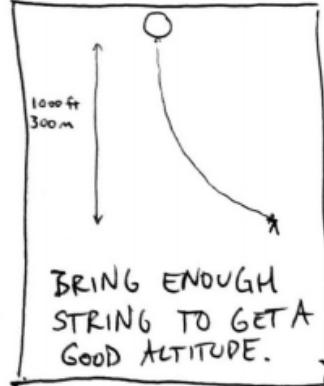
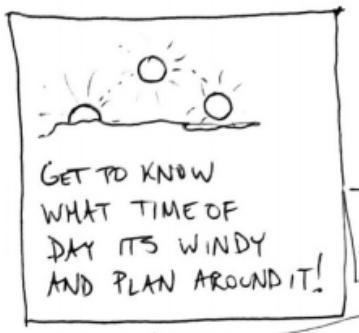
This requires attaching a marked-up 7.5 series USGS Topographic Quadrangle Map. Then call your [local Flight Standards District Office](#) to make contact with someone to email it to for approval.

# Getting Helium

High Pressure Cylinders								
Size	R	RR	Q	LD	S	K	T	KHP
Height (In)	14	17	32	43	47	51	55	51
Weight (Lbs)	11	24	46	58	61	113	139	188
Nominal Volume (CU FT)	20	40	80	122	150	244	330	N/A

# Balloon Mapping Quick Start Guide

## Safely launching your balloon and camera



### Be aware of others' privacy

Make sure to talk to your neighbors first. Maybe they'll want to help!

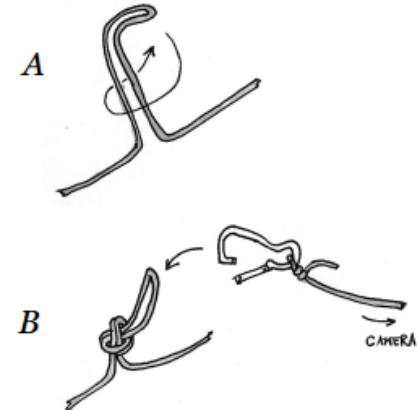
### Be safe and responsible

Check that you are five miles or more away from the nearest airport. Otherwise, speak with the airport about sending a "Notice to Airmen". Print out satellite imagery of where you'll be mapping (Google, USGS, etc) to help in planning.

Do a test flight first, without a camera: reel out, then reel in, with about a hundred feet of string.

### Attach your camera

Make a temporary loop by tying an overhand knot on a loop on the string below the balloon:



A. Slacken the line below the balloon & gather a loop in your hand

B. Loop it around & through itself and pull taut; attach your camera!

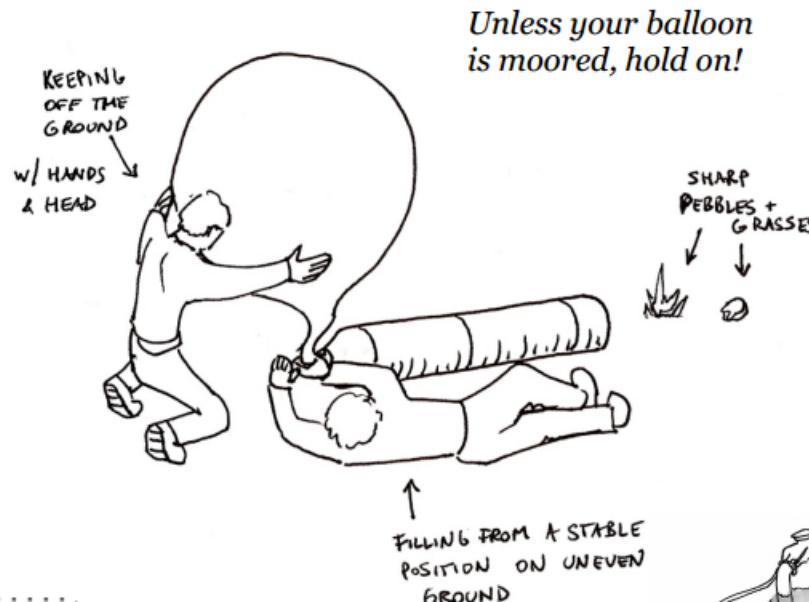
# Balloon Mapping Quick Start Guide

## Filling, closing, and mooring your balloon

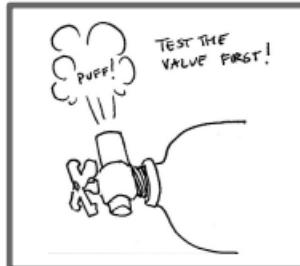
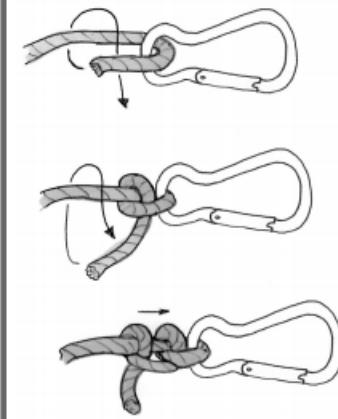
1) Tie string to a carabiner with double overhand knot (see upper right box).

2) Tie the other end (5ft or so) to something heavy like a 1 gallon jug full of water -- so your balloon won't fly away as you're working.

3) Tie the clip swivel to the reel of kite string with the same knot.



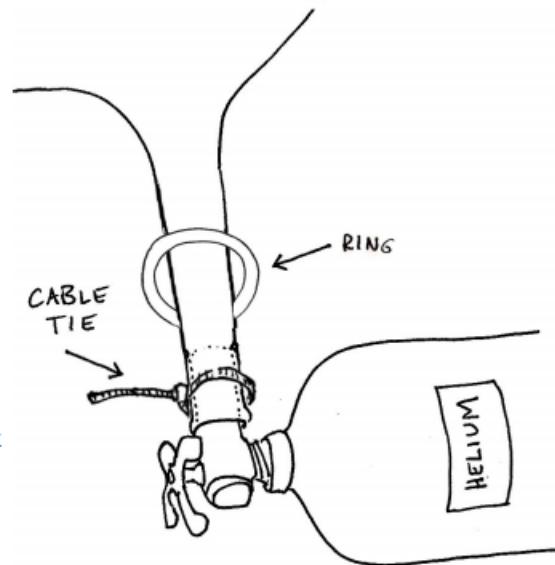
Unless your balloon  
is moored, hold on!



Test the valve, and lay the tank on the ground .

1) Pull the balloon neck through the ring.

2) Cable tie the balloon to the helium tank. The balloon neck may need to be folded and squeezed tight. Continue holding on to your balloon.

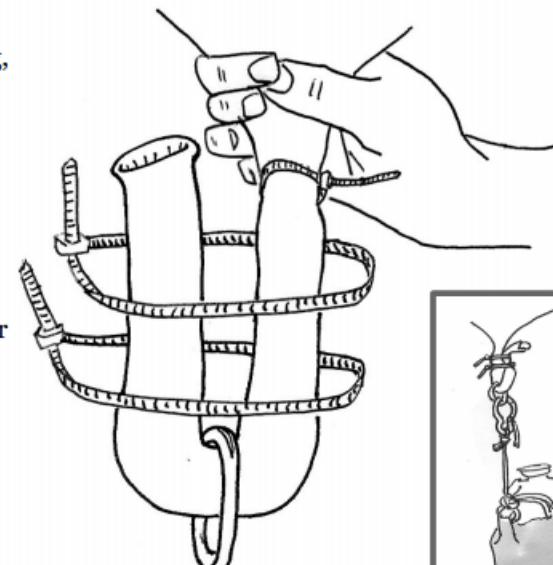


3) When done filling,  
push helium out of  
the neck and close  
with a cable tie just  
below the balloon.

4) Release nozzle  
cable tie.

5) Fold the neck over  
onto itself and  
around the ring.  
Attach two more  
cable ties and pull  
tight.

6) Attach ring to the  
mooring point.





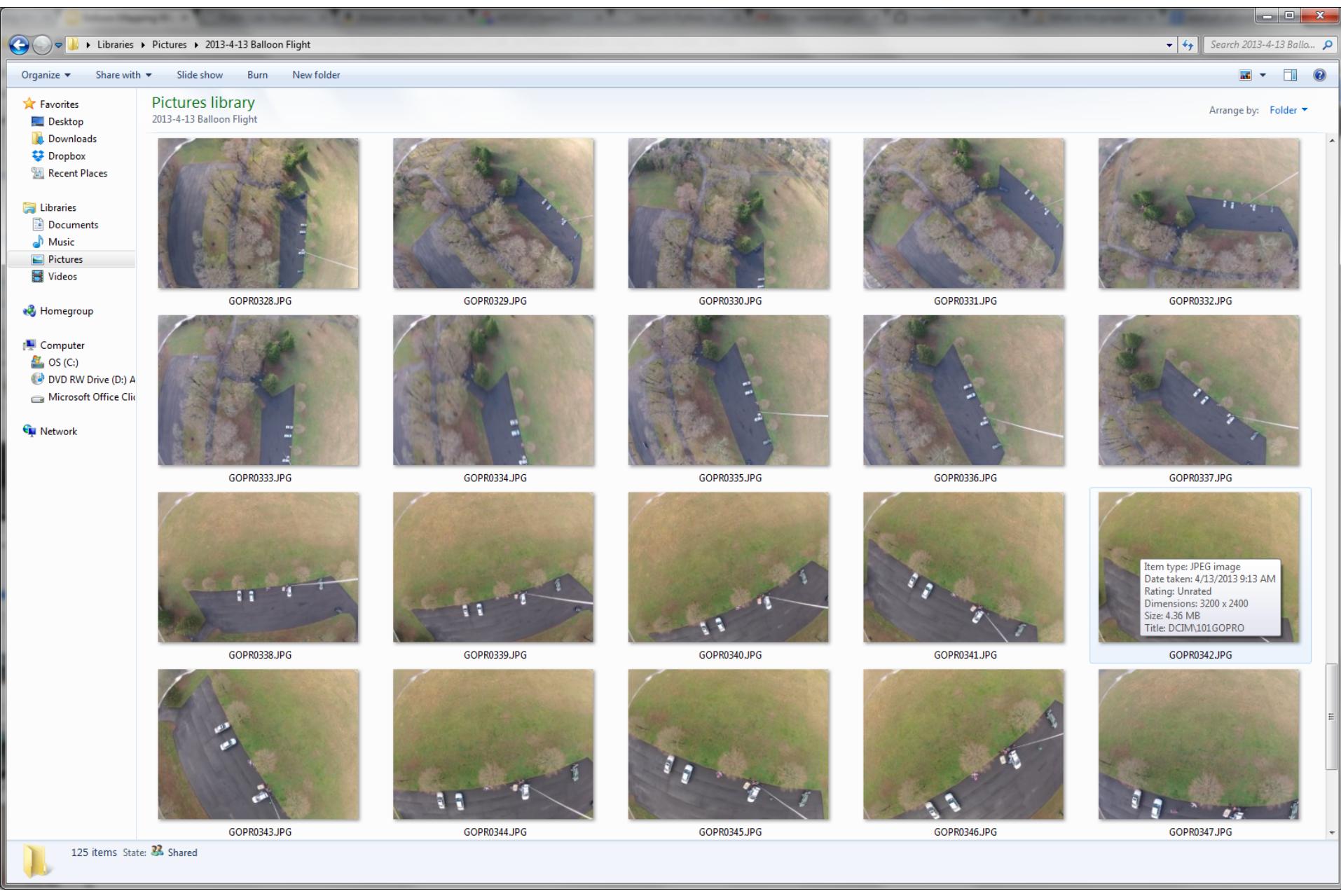












# Now What?

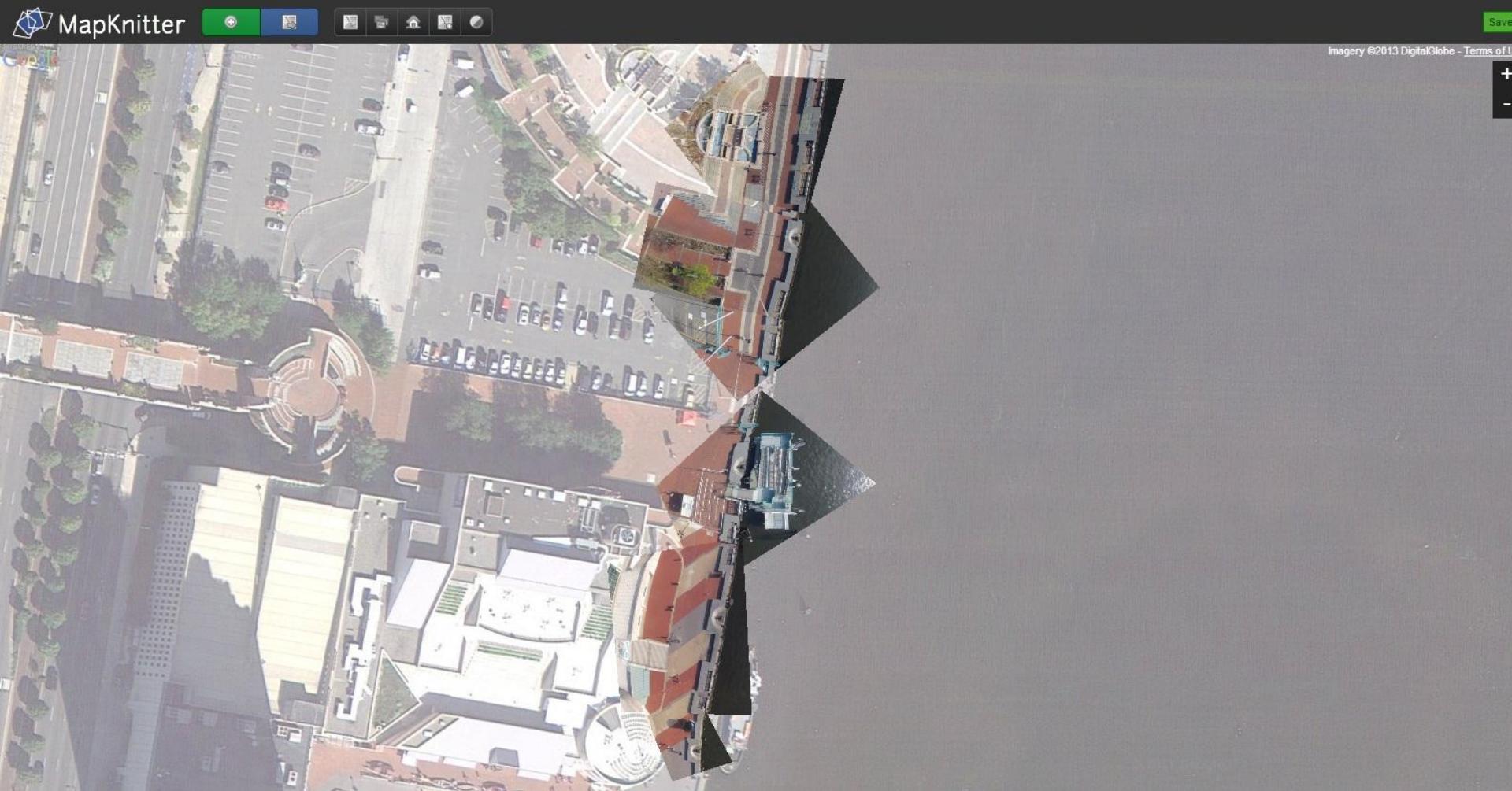
## Open Source Options

- MapKnitter.org
- Quantum GIS (QGIS)

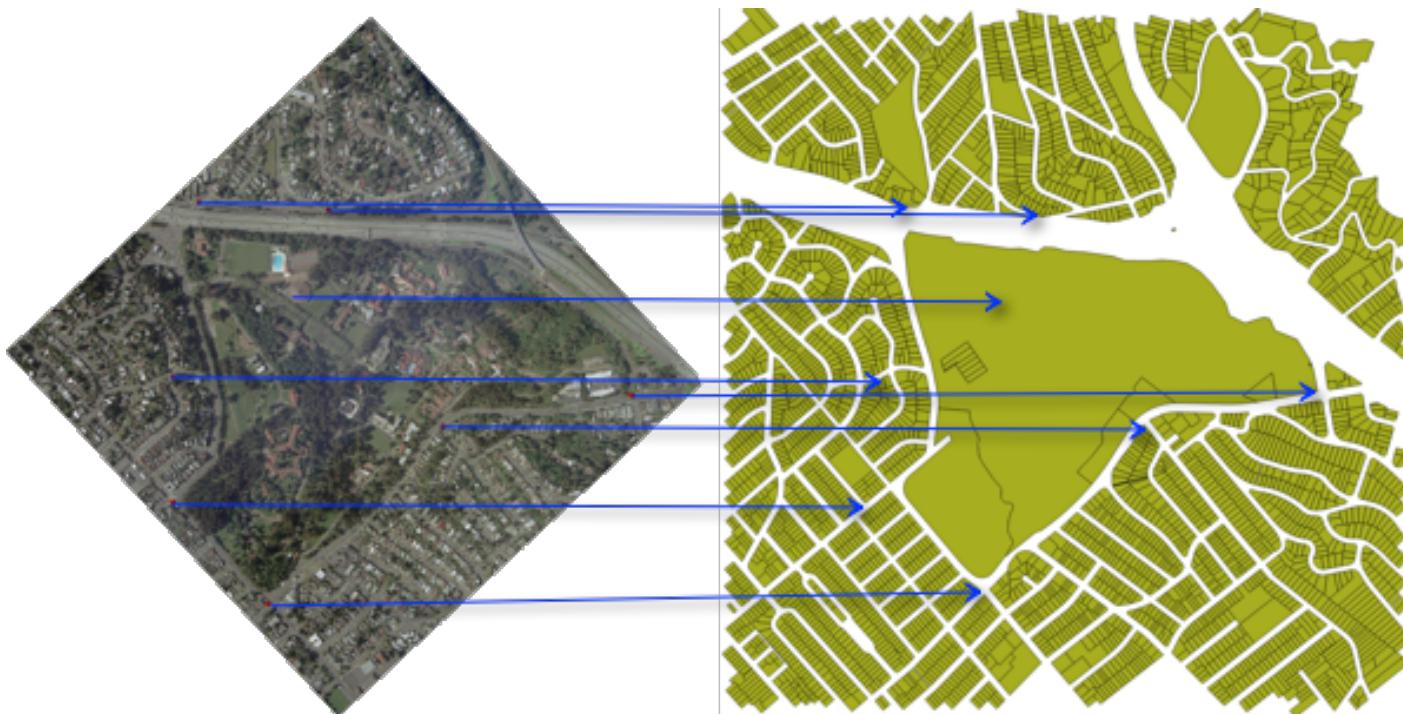
## Proprietary Options

- Desktop GIS (Esri, MapInfo)
- Adobe Photoshop
- Microsoft Photosynth

# MapKnitter.org



# Quantum GIS (QGIS)



# Microsoft PhotoSynth

Microsoft Photosynth | Tech Preview | Explore | About | My Synths | Search | New Account | Sign In | Create

Photosynth Tips X

Click on the white boxes to see different photos.

Use the arrows to see more of the scene.

Use the buttons or mouse scroll wheel to zoom in & out.

Don't show again [More Info](#)



Aerial Photo in Clarksville, TN By: surf89

Add to Favorites Embed Facebook Report Abuse Share

Description	Related Photosynths	Comments (0)
Aerial Photo in Clarksville.		<a href="#">Sign in to add a comment</a> New to Photosynth? <a href="#">Sign Up</a> for a free account.
Stats		
Synthy 100%		
Views 1		
Favorites 0		
Photos 13		
Date Created 9/15/2014		



*thank you*

Sean McGinnis

e: seankmcginnia@gmail.com

t: @seankmcginnis

w: [www.georamblings.com](http://www.georamblings.com)