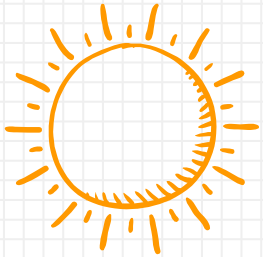


EE16A Imaging 2

Why?

- Imaging 1:
  - Finding a link between physical quantities and voltage is powerful
  - If you can digitize it, you can do anything (IOT devices, internet, code, processing)
- Imaging 2:
  - What measurements are good measurements?
    - Remember Kody and Nara?

Kody and Nara

2. Finding The Bright Cave

Nara the one-handed druid and Kody the one-handed ranger find themselves in dire straits. Before them is a cliff with four cave entrances arranged in a square: two upper caves and two lower caves. Each entrance emits a certain amount of light, and the two wish to find exactly the amount of light coming from each cave. Here's the catch: after contracting a particularly potent strain of ghoul fever, our intrepid heroes are only able to see the total intensity of light before them (so their eyes operate like a single-pixel camera). Kody and Nara are capable adventurers, but they don't know any linear algebra – and they need your help.

Kody proposes an imaging strategy where he uses his hand to completely block the light from two caves at a time. He is able to take measurements using the following four masks (black means the light is blocked from that cave):

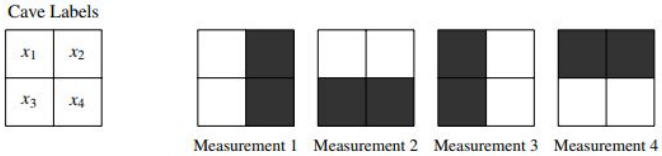


Figure 1: Four image masks.

(a) Let $\vec{x}$ be the four-element vector that represents the magnitude of light emanating from the four cave entrances. Write a matrix $\mathbf{K}$ that performs the masking process in Figure 1 on the vector $\vec{x}$, such that $\mathbf{K}\vec{x}$ is the result of the four measurements.

(b) Does Kody's set of masks give us a unique solution for all four caves' light intensities? Why or why not?

(c) Nara, in her infinite wisdom, places her one hand diagonally across the entrances, covering two of the cave entrances. However, her hand is not wide enough, letting in 50% of the light from the caves covered and 100% of the light from the caves not covered. The following diagram shows the percentage of light let through from each cave:

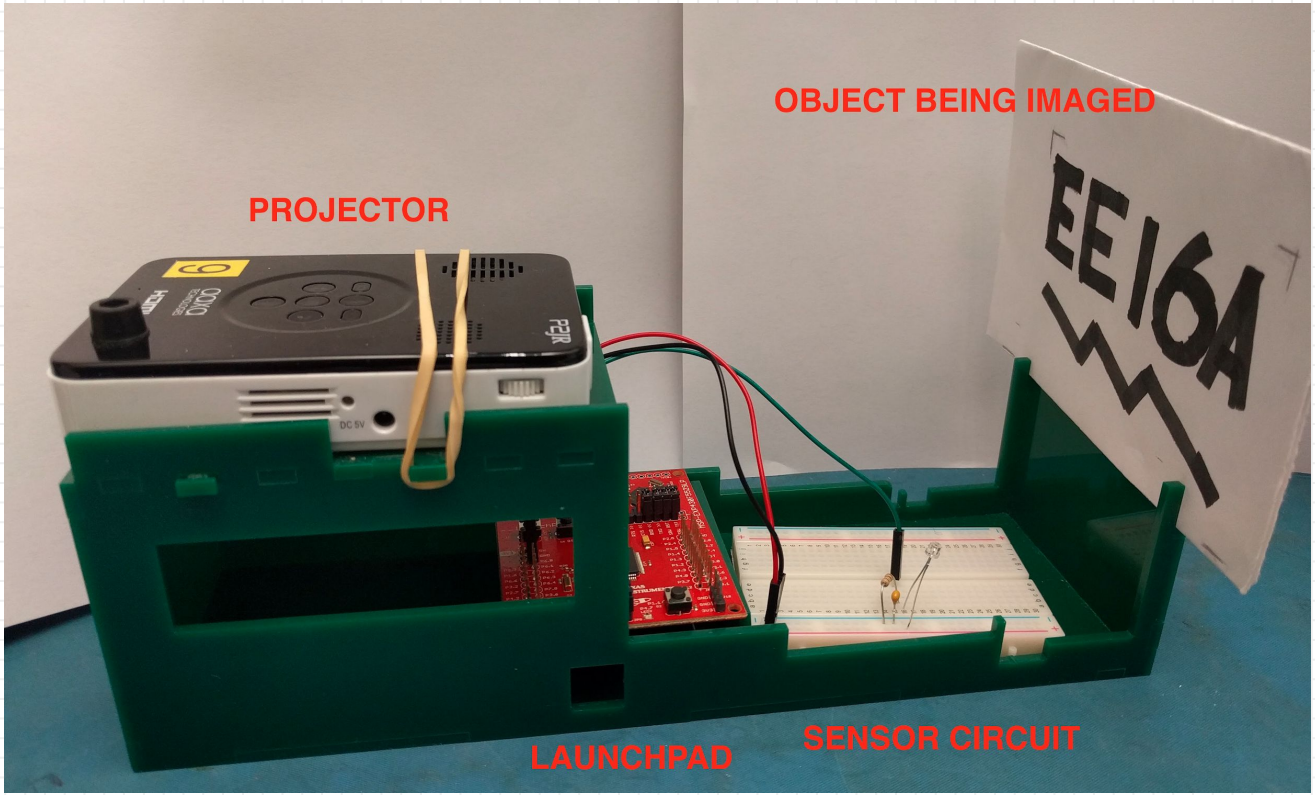
Illuminating the Big Picture

- ✘ Linear dependance
  - ✗ When can you even recover your image?
  - ✗ Does it matter what mask matrix you pick?
  - ✗ Does it matter how you cover the pixels?
- ✘ Invertibility
  - ✗ When can you solve Ax = b
  - ✗ Why do we care? How does this affect the way we pick our imaging matrix?

Today's Lab: Single Pixel Scanning

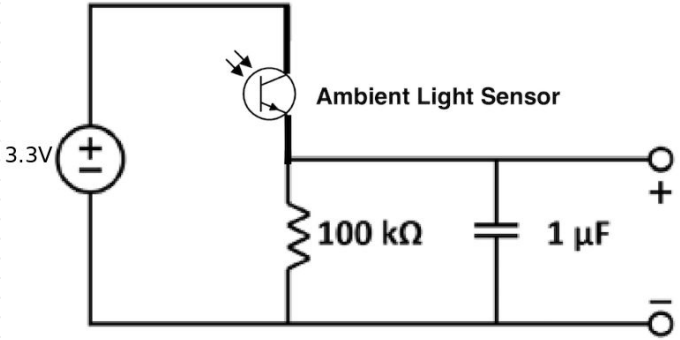
- ✘ Circuit from last week measures **light** intensity
- ✘ Projector illuminates index card in a controlled way
- ✘ Python programming to reconstruct image

Setup:

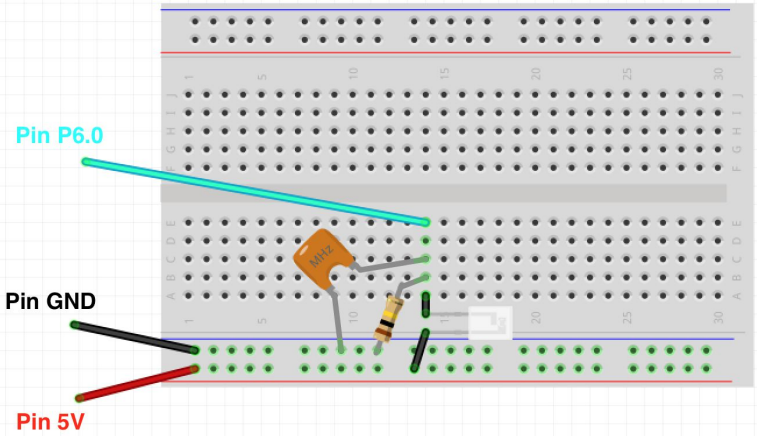

OBJECT BEING IMAGED
PROJECTOR
EE 16A
LAUNCHPAD
SENSOR CIRCUIT

Circuit From Last Week

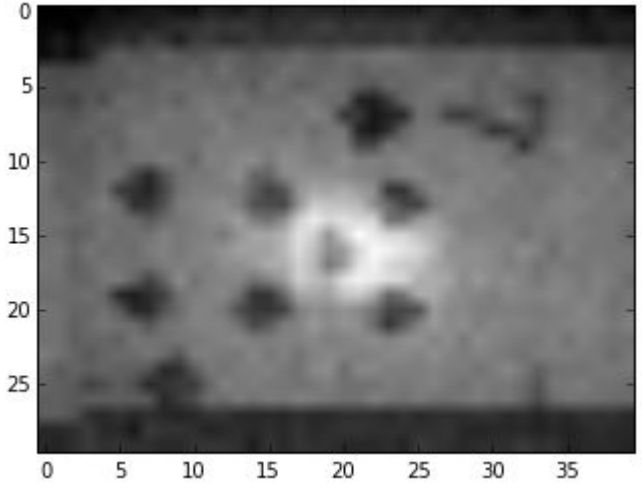
Circuit Diagram



Breadboard Diagram

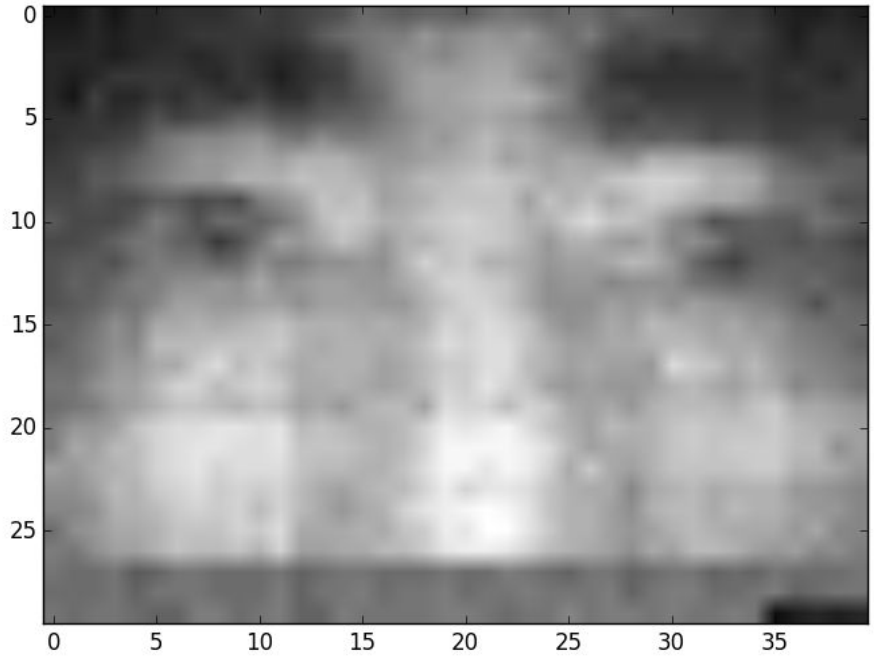


Sample Images



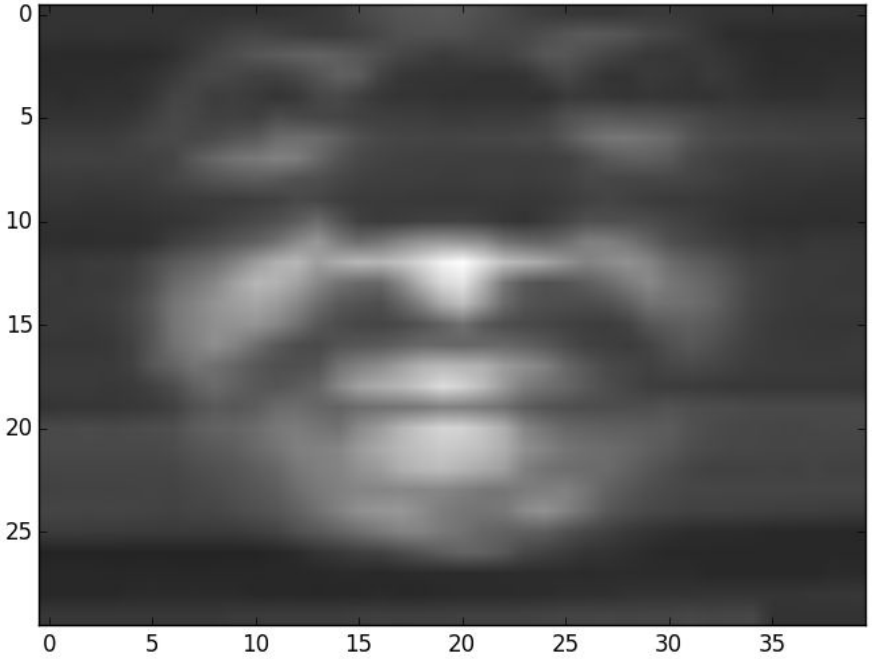
Sample Images

Seiya:



Sample Images

Nick:



Images, Matrices, Vectors

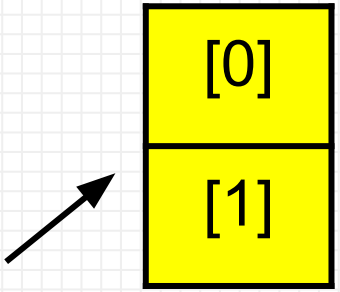
- ✘ What are the unknowns in our system?
- ✘ Want to do an experiment to get information about these unknowns.
- ✘ We can do a lot of interesting processing on vectors, but we need to convert the image into one first
  - ✗ In lecture and discussion, you have seen how to turn an image into a vector. How?

Images, Matrices, Vectors

[0]	[1]
[2]	[3]
[4]	[5]

Images, Matrices, Vectors

[0]	[1]
[2]	[3]
[4]	[5]



Images, Matrices, Vectors

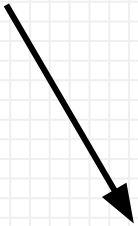
[0]	[1]
[2]	[3]
[4]	[5]

→

[0]
[1]
[2]
[3]

Images, Matrices, Vectors

[0]	[1]
[2]	[3]
[4]	[5]



[0]
[1]
[2]
[3]
[4]
[5]

Images, Matrices, Vectors

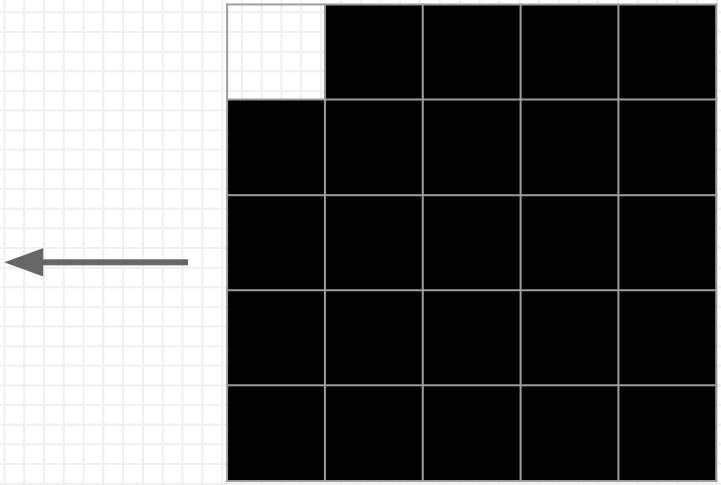
[0]	[1]
[2]	[3]
[4]	[5]

→

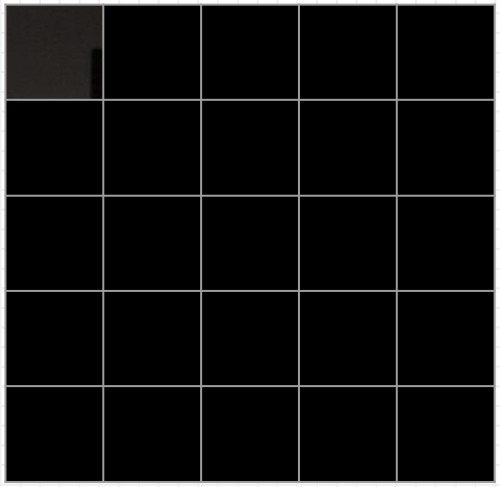
[0]
[1]
[2]
[3]
[4]
[5]

Pixel-by-Pixel Scan of an Image

Pixel-by-Pixel Scan of an Image



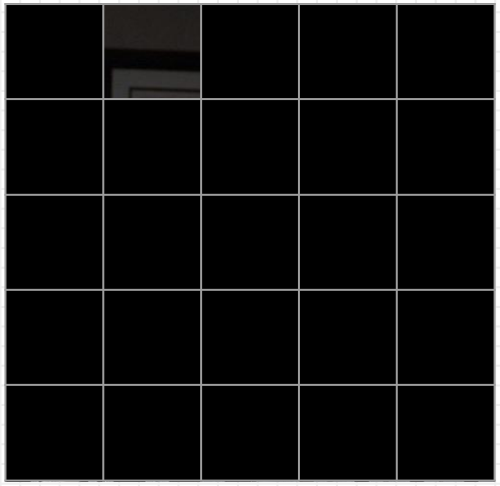
Pixel-by-Pixel Scan of an Image



Masked image

Image

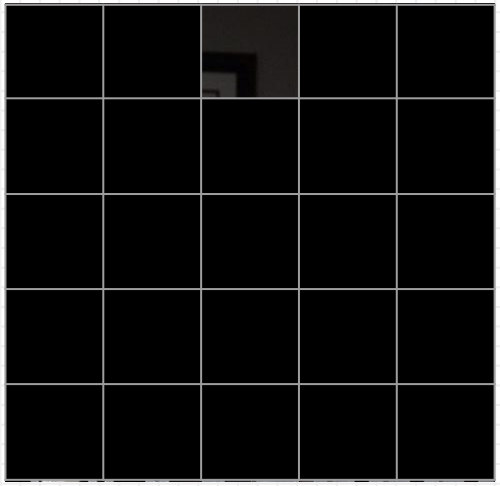
Pixel-by-Pixel Scan of an Image



Masked image

Image

Pixel-by-Pixel Scan of an Image



Masked image

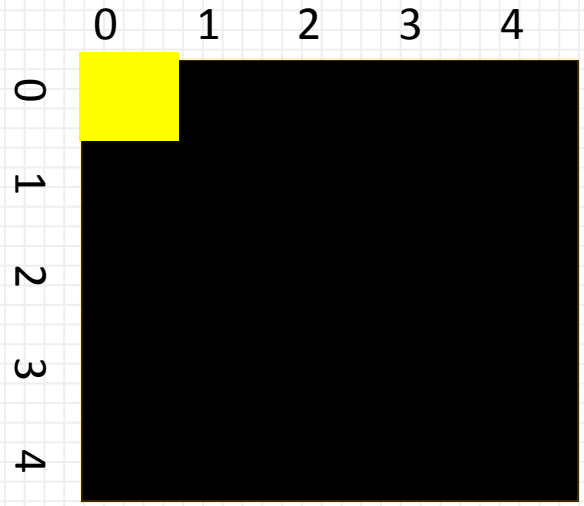
Image

Question Time

✘ To read all the pixels of a 4x4 image, how many pixel-by-pixel scans do we need to do?

Representing our Masks in Python

Imaging Mask 0



mask0 =

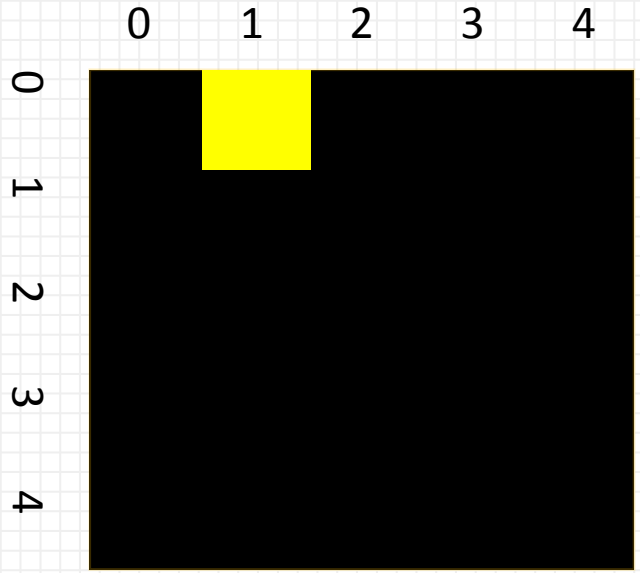
np.array([

[	**1,**	**0,**	**0,**	**0,**	**0**]
[	**0,**	**0,**	**0,**	**0,**	**0**]
[	**0,**	**0,**	**0,**	**0,**	**0**]
[	**0,**	**0,**	**0,**	**0,**	**0**]
[	**0,**	**0,**	**0,**	**0,**	**0**]]

)

Representing our Masks in Python

Imaging Mask 1



mask1 =

np.array([

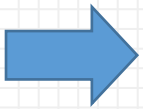
[	0,	1,	0,	0,	0] ,
[	0,	0,	0,	0,	0] ,
[	0,	0,	0,	0,	0] ,
[	0,	0,	0,	0,	0] ,
[	0,	0,	0,	0,	0]]

)

Turning the Masks Into Vectors

mask0 =

[[	1,	0,	0,	0,	0],
[	0,	0,	0,	0,	0],
[	0,	0,	0,	0,	0],
[	0,	0,	0,	0,	0],
[	0,	0,	0,	0,	0]]



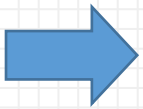
T

Row 0	1
Row 1	0
Row 2	0
Row 3	0
Row 4	0
⋮	
Row 24	0

Turning the Masks Into Vectors

mask1 =

[[	0,	1,	0,	0,	0],
[	0,	0,	0,	0,	0],
[	0,	0,	0,	0,	0],
[	0,	0,	0,	0,	0],
[	0,	0,	0,	0,	0]]



T

Row 0	0
Row 1	1
Row 2	0
Row 3	0
Row 4	0
⋮	
Row 24	0

Generating the Masking Matrix from the Masks

mask0

1,	0,	0,	0,	0
0,	0,	0,	0,	0
0,	0,	0,	0,	0
0,	0,	0,	0,	0
0,	0,	0,	0,	0

Generating the Masking Matrix from the Masks

mask0

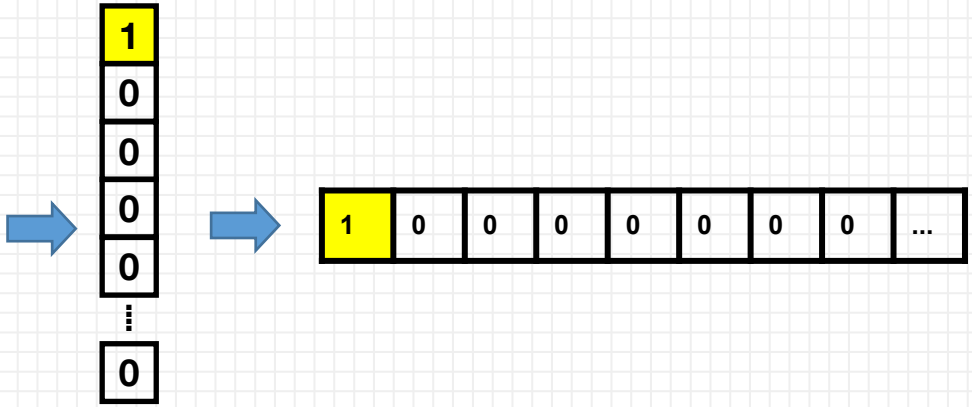
1,	0,	0,	0,	0
0,	0,	0,	0,	0
0,	0,	0,	0,	0
0,	0,	0,	0,	0
0,	0,	0,	0,	0

1
0
0
0
0
⋮
0

Generating the Masking Matrix from the Masks

mask0

1,	0,	0,	0,	0
0,	0,	0,	0,	0
0,	0,	0,	0,	0
0,	0,	0,	0,	0
0,	0,	0,	0,	0



Generating the Masking Matrix from the Masks

mask1

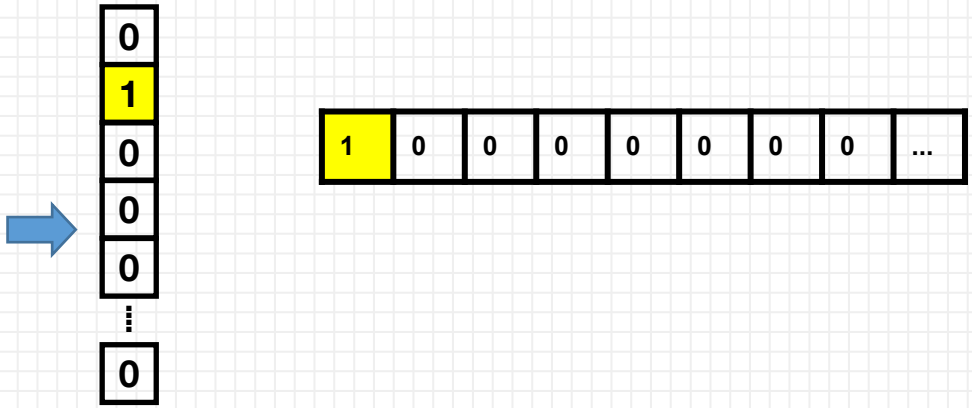
0,	1,	0,	0,	0
0,	0,	0,	0,	0
0,	0,	0,	0,	0
0,	0,	0,	0,	0
0,	0,	0,	0,	0

1	0	0	0	0	0	0	0	...

Generating the Masking Matrix from the Masks

mask1

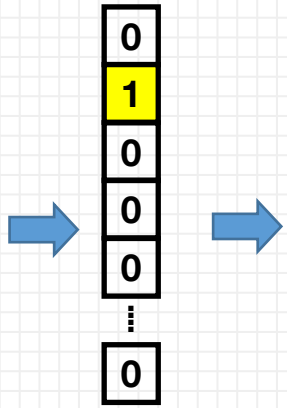
0,	1,	0,	0,	0
0,	0,	0,	0,	0
0,	0,	0,	0,	0
0,	0,	0,	0,	0
0,	0,	0,	0,	0



Generating the Masking Matrix from the Masks

mask1

0,	1,	0,	0,	0
0,	0,	0,	0,	0
0,	0,	0,	0,	0
0,	0,	0,	0,	0
0,	0,	0,	0,	0



1	0	0	0	0	0	0	0	...
0	1	0	0	0	0	0	0	...

Generating the Masking Matrix from the Masks

1	0	0	0	0	0	0	0	...
0	1	0	0	0	0	0	0	...
0	0	1	0	0	0	0	0	...

Generating the Masking Matrix from the Masks

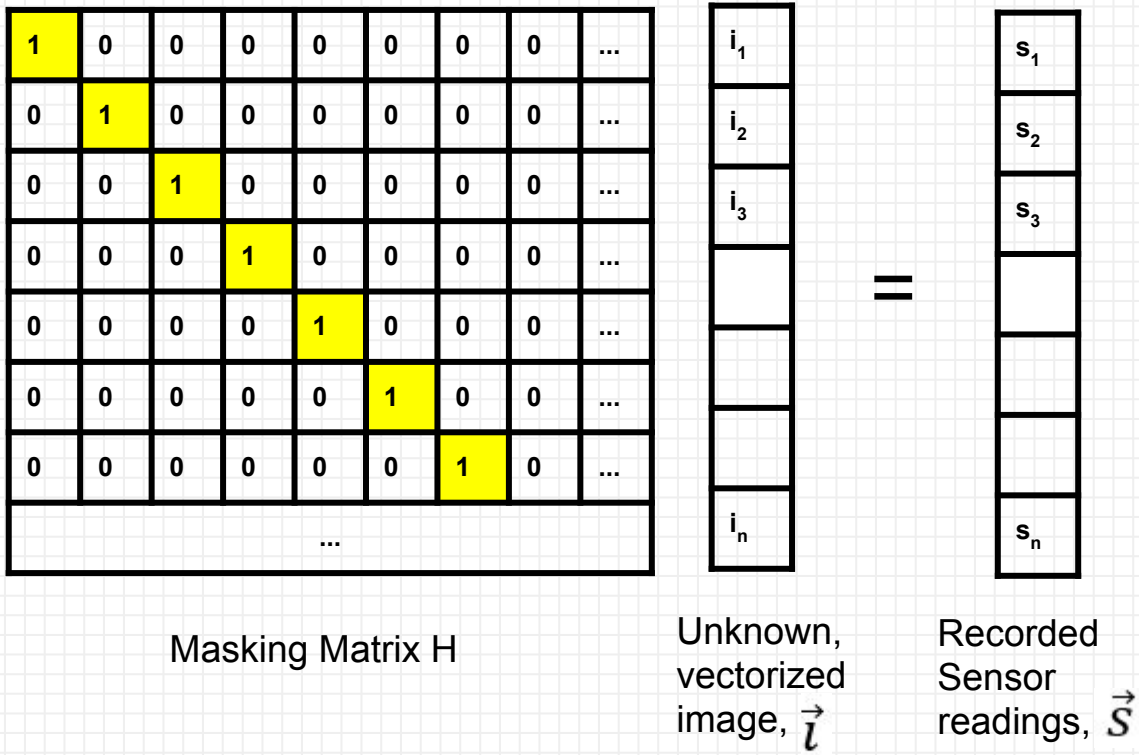
1	0	0	0	0	0	0	0	...
0	1	0	0	0	0	0	0	...
0	0	1	0	0	0	0	0	...
0	0	0	1	0	0	0	0	...

Each Row is a Different Experiment

$H =$

1	0	0	0	0	0	0	0	...
0	1	0	0	0	0	0	0	...
0	0	1	0	0	0	0	0	...
0	0	0	1	0	0	0	0	...
0	0	0	0	1	0	0	0	...
0	0	0	0	0	1	0	0	...
0	0	0	0	0	0	1	0	...
...								

Measuring a Pixel is Matrix-Vector Multiplication



Masking Matrix H

Unknown, vectorized image, $\vec{i}$

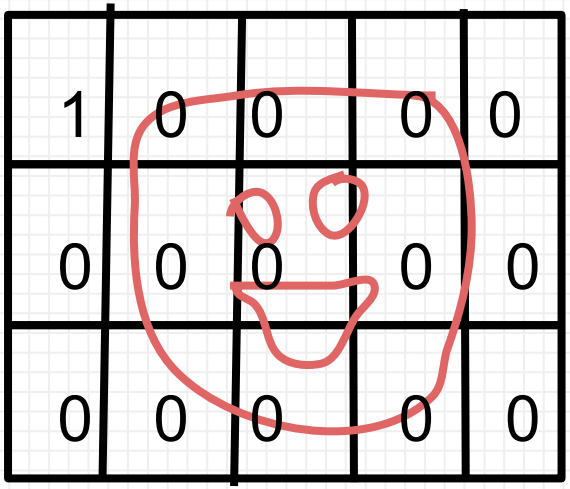
Recorded Sensor readings, $\vec{s}$

Measuring a Pixel is Matrix-Vector Multiplication

$$\vec{s} = H\vec{\imath}$$

- ✘ We know H and we have the sensor readings, how do we get the image?
- ✘ How do we solve this?
- ✘ When can we solve this?
  - ✗ Conditions on H

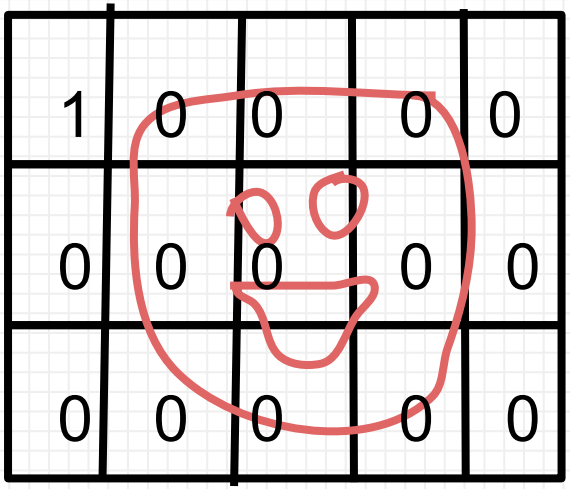
Example



What will the scanning matrix's dimensions be?

How many pixels?

Example



What will the scanning matrix's dimensions be?

15x15

How many pixels?

15

How Scanning Works: iPython

$H =$

1	0	0	0	0	0	0	0	...
0	1	0	0	0	0	0	0	...
0	0	1	0	0	0	0	0	...
0	0	0	1	0	0	0	0	...
0	0	0	0	1	0	0	0	...
0	0	0	0	0	1	0	0	...
0	0	0	0	0	0	1	0	...
...								

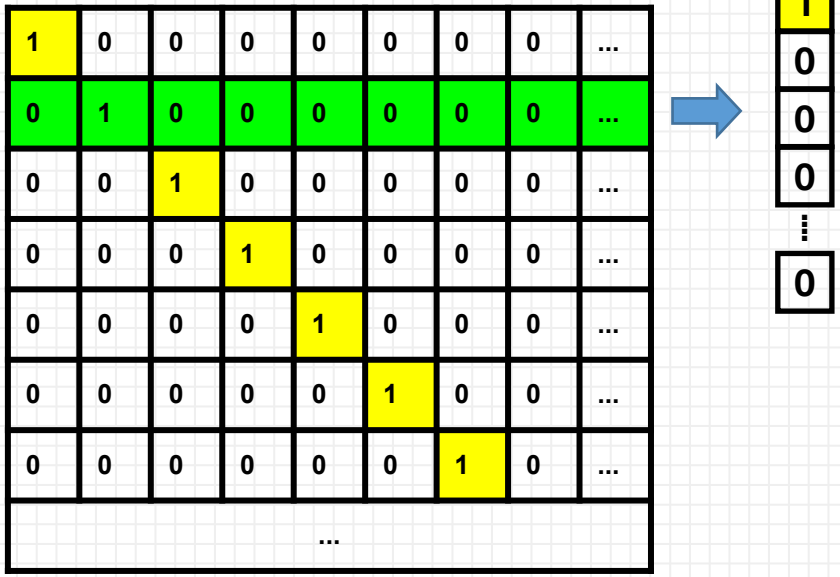
How Scanning Works: iPython

$H =$

1	0	0	0	0	0	0	0	...
0	1	0	0	0	0	0	0	...
0	0	1	0	0	0	0	0	...
0	0	0	1	0	0	0	0	...
0	0	0	0	1	0	0	0	...
0	0	0	0	0	1	0	0	...
0	0	0	0	0	0	1	0	...
...								

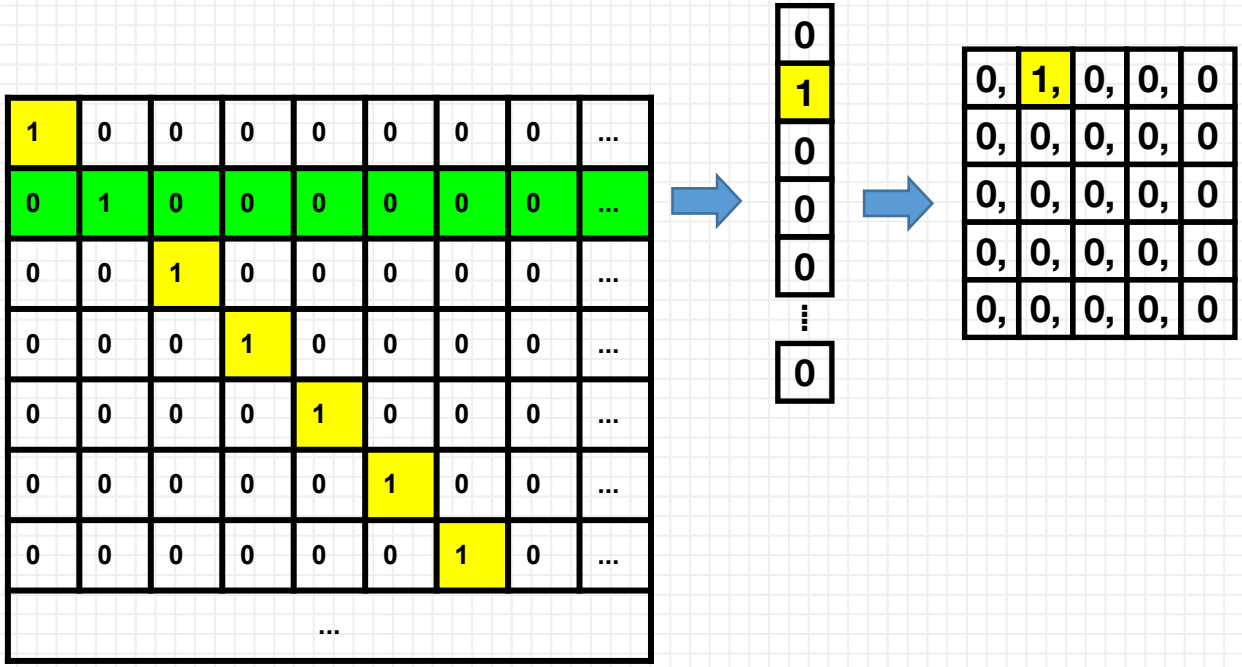
How Scanning Works: iPython

$H =$



How Scanning Works: iPython

$H =$



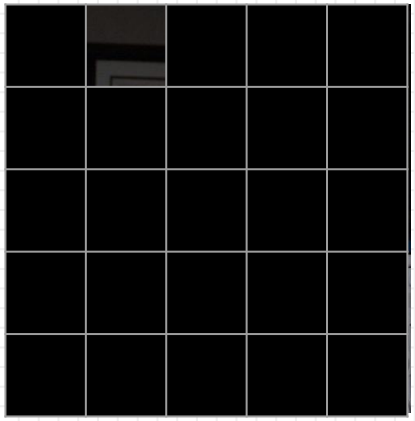
How Scanning Works: iPython

$H =$

1	0	0	0	0	0	0	0	...
0	1	0	0	0	0	0	0	...
0	0	1	0	0	0	0	0	...
0	0	0	1	0	0	0	0	...
0	0	0	0	1	0	0	0	...
0	0	0	0	0	1	0	0	...
0	0	0	0	0	0	1	0	...
...								

0
1
0
0
0
⋮
0

0,	1,	0,	0,	0
0,	0,	0,	0,	0
0,	0,	0,	0,	0
0,	0,	0,	0,	0
0,	0,	0,	0,	0



What Makes A Mask Good?

- ✘ Linearly independent columns -> Invertible
  - ✗ Can't get a solution without this
  - ✗ There is a unique solution
- ✘ What would be a bad mask?
- ✘ Food for thought: Are all invertible matrices equally as good?
  - ✗ Find out next week...

Tips for a Good Image

- ✘ READ CLOSELY. There are many small directions that help you get a good setup
- ✘ Focus projector using dial on the side
- ✘ Close the box firmly & scan under dark conditions

Notes

- ✘ No signal when testing the oscope on their previous circuit
  - ✘ **Unplug P6.0 from MSP and debug if necessary**
- ✘ UART Application Com Port not showing up as an option when scanning
  - ✘ **Close serial monitor!**
- ✘ Do not take sharpies and tape from the desk
- ✘ If something isn't working, close everything and turn it back on (works 9/10 times)

Important Notes

- ✘ You should have your kit from last week
- ✘ Equipment in cardboard box:
  - ✗ Don't break the plastic stand!
  - ✗ Put everything back before you leave!
    - Including Projector's Power
- ✘ Make sure you are using the right com port at all times
  - ✗ Not COM1, and not the debugger