SciComp with Py

Edge Detection as Derivative of Luminosity

Vladimir Kulyukin
Department of Computer Science
Utah State University

Outline

- Gradients
- Edge Detection with Gradients

Gradients

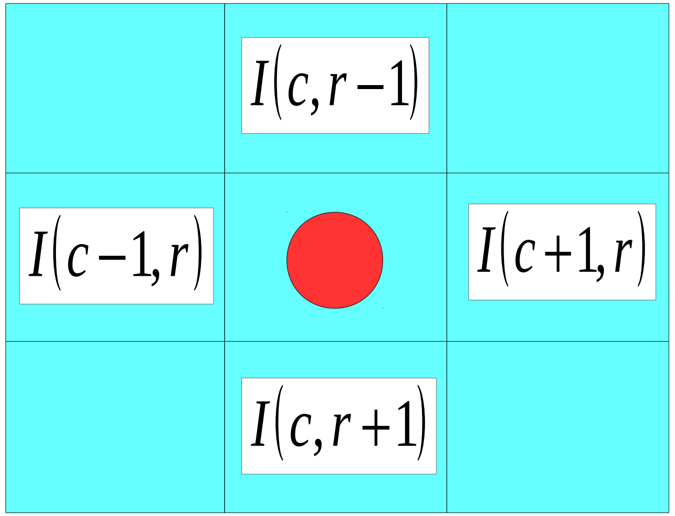
Gradients

- Gradients are changes in image intensity/color
- Gradients, if viewed as vectors, have directions and magnitudes
- Gradients can be computed for each pixels or for image regions

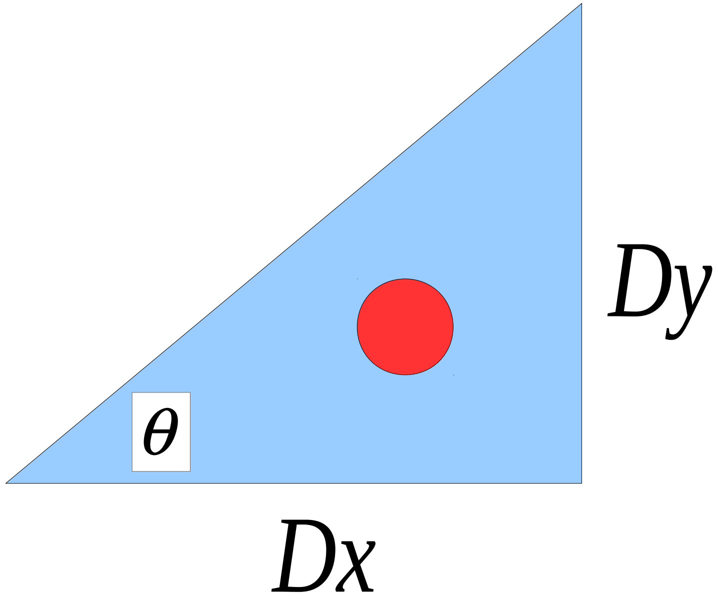
Vertical & Horizontal Changes: *Dy* & *Dx*

$$Dy = I(c, r-1) - I(c, r+1)$$

$$Dx = I(c+1, r) - I(c-1, r)$$



	$I(c, r-1)$	
$I(c-1, r)$	●	$I(c+1, r)$
	$I(c, r+1)$	



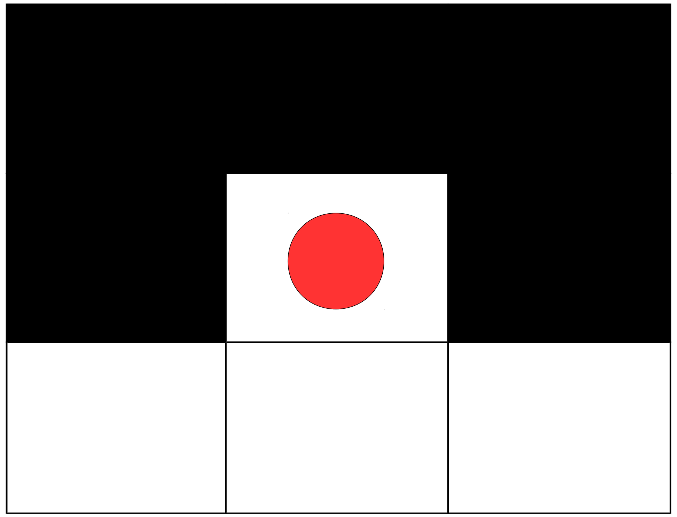
$\|G\| = \sqrt{Dy^2 + Dx^2}$ is the gradient's magnitude at $I(c, r)$

$\theta = \tan^{-1}\left(\frac{Dy}{Dx}\right)$ is the gradient's orientation

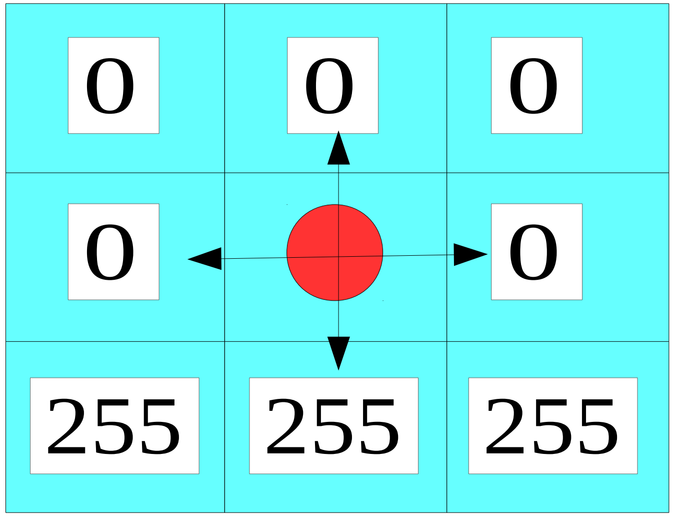
What if $Dx = 0$? In this case, we can set Dx to some small default value, e.g., 1.

Example

Image



Pixel Values



0	0	0
0		0
255	255	255

$$dy = 0 - 255 = -255; dx = 0 - 0 \approx 1$$

$$\| G \| = \sqrt{(-255)^2 + 1^2} = 255.00196078 \approx 255$$

$$\theta = \left(\tan^{-1}\left(\frac{-255}{1} \right) \right) \frac{180}{\pi} = -89.775311^{\circ} \approx -90^{\circ}$$

Edge Detection with Gradients

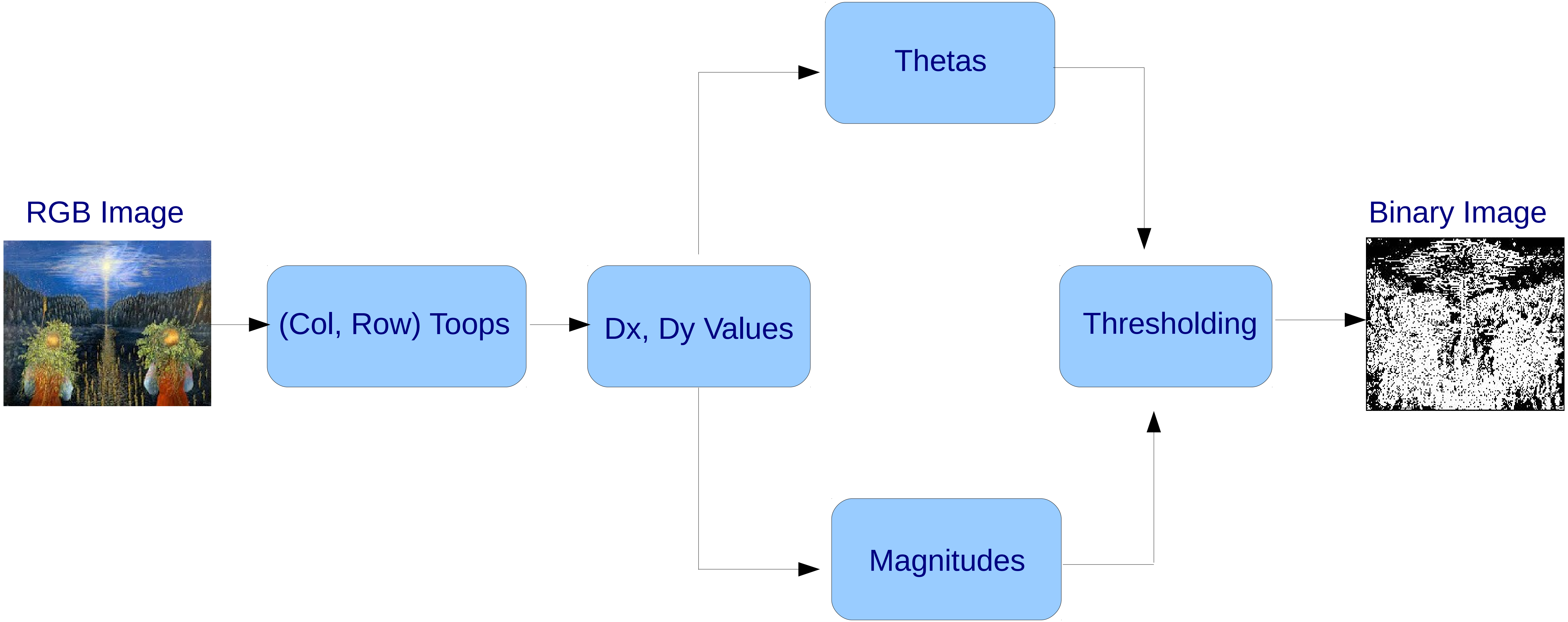
Implementation Steps

- Grayscale RGB pixels with relative luminosity
- Compute ***Dy*** and ***Dx*** at each pixel
- Compute gradient's magnitude and orientation for each pixel
- Implement an edge detection pipeline

PIL vs. OpenCV

- We can use PIL (Python Image Library) in this implementation
- PIL is for Python 2
- PILLOW is PIL for Python 3
- PIL/PILLOW is a simpler version of OpenCV
- It is great for rapid prototyping and easier to install than OpenCV

Edge Detection Pipeline



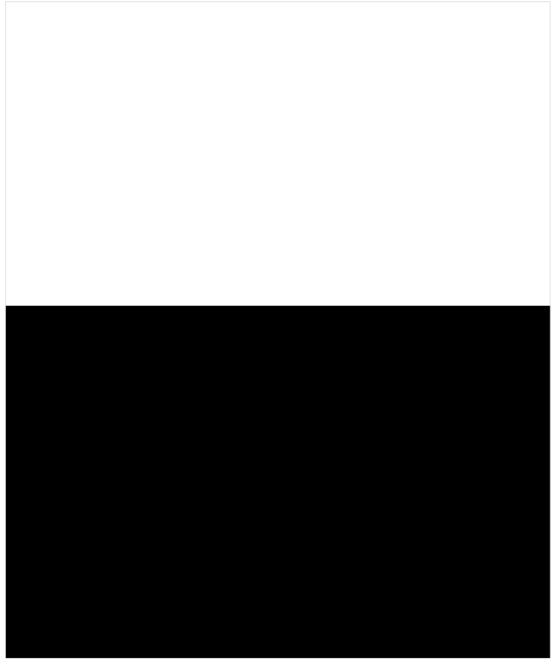
Command Line

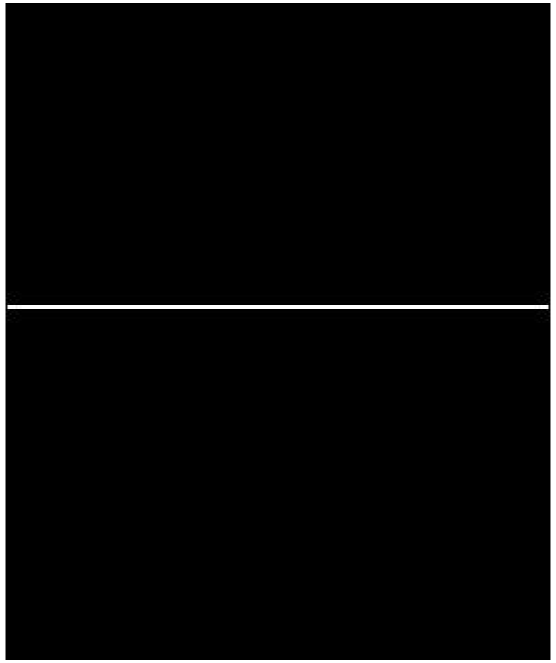
```
if __name__ == '__main__':
    input_image = Image.open(args['input_path'])
    output_image = detect_edges(input_image)
    output_image.save(args['output_path'])
    del input_image
    del output_image
```

Py source in gd_detect_edges.py

Test Run 1

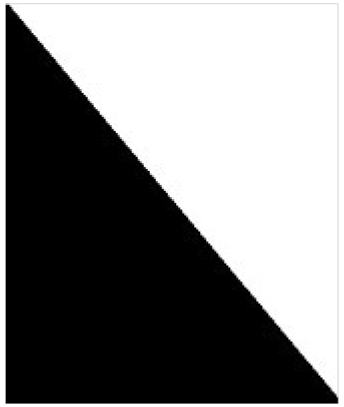
```
$python   gd_detect_edges.py -ip EdgeImage_01.jpg -op EdgeImage_01_gd_ed.jpg
```

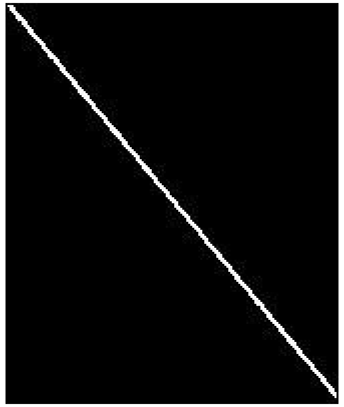




Test Run 2

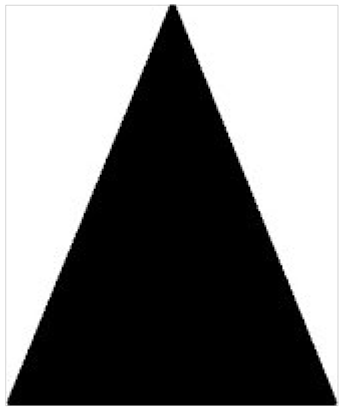
```
$python   gd_detect_edges.py  -ip EdgeImage_02.jpg -op EdgeImage_02_gd_ed.jpg
```

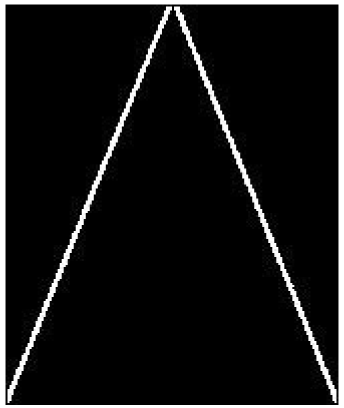




Test Run 3

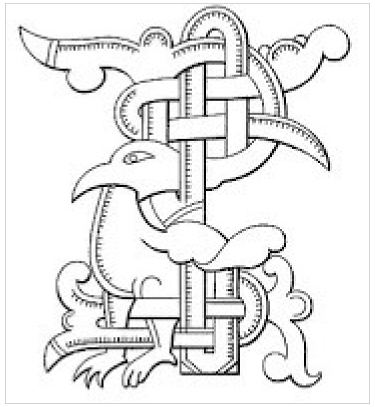
```
$python   gd_detect_edges.py -ip EdgeImage_02.jpg  -op EdgeImage_02_gd_ed.jpg
```

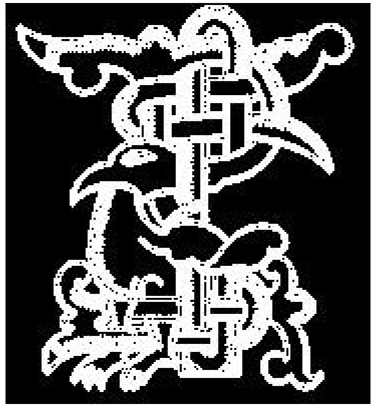




Test Run 4

```
$ python gd_detect_edges.py  -ip BirdOrnament.jpg -op BirdOrnament_gd_ed.jpg
```





Test Run 5

```
$ python gd_detect_edges.py -ip june.jpg -op june_gd_ed.jpg
```