VIT-AP
UNIVERSITY

Computer Vision

(Course Code: 4047)

Module-2:Lecture-11: Steerable Filters

Gundimeda Venugopal, Professor of Practice, SCOPE

Steerable filters

- “Steerable filters” [Freeman and Adelson 1991] a class of filters in which we can synthesize a filter for an arbitrary orientation through a linear combination of basis filters.
- if a class of filters is steerable we can have:

$$f^{\theta} = \sum_{i} k_i(\theta) f^{i}$$

where: ***f*** is our filter, the various ***f*** $_{i}$ are the filters in the basis and the factors $k_i(\theta)$ are what we can call “steering functions”.

- Not only we can easily create filters with arbitrary orientation, we can actually skip that and obtain directly their response if we already have a response for the basis filters. This is because when it comes to convolution:

$$f^{\theta} * I = \sum_{i} (k(\theta) f^{i}) * I = \sum_{i} k(\theta)(f^{i} * I)$$

It means that if we need outputs for lots of orientations, we don’t have to compute lots of expensive convolutions, but only as many as the number of filters in our basis.

What is a Steerable Filter

A Steerable Filter is able to provide an Image Gradient in any orientation/direction that we choose. We can steer the filter in this direction by taking a linear combination of filters that produce a gradient in a known direction.

How do we synthesize a Steerable Filter? **(steerable filters using Sobel Kernels)**

We have two kernels one oriented at 0 degrees and the other at 90 degrees, we denote them as: G^{o} and G^{9o}. We denote the responses of these kernels as: R^{o} and R^{9o}.

We can steer the gradient direction by taking a linear combination of their responses:

Where the responses (R^{o} and R^{9o}) are the result of convolving the respective Sobel Kernels with image *I(x,y)*

$$R^{\theta} = cos(\theta)R^{0} + sin(\theta)R^{90}$$

(The cosine captures the horizontal component) (The sine captures the vertical component.)

The resulting response R^{θ} is steered at the desired angle θ

$$R^{0} = I(x,y) * G^{0}$$
$$R^{90} = I(x,y) * G^{90}$$

This issue is we need to perform two separate filtering operations. Re-write the expression above to obtain a single filter kernel, first we re-write it by explicitly considering the convolution with image I(x,y) in the Sobel Filtering Operations, then simplify via the distributive property of Convolution.

$$\begin{aligned} R^{\theta} &= cos(\theta)R^{0} + sin(\theta)R^{90} \\ &= cos(\theta)(I(x,y) * G^{0}) + sin(\theta)(I(x,y) * G^{90}) \\ &= I(x,y) * (cos(\theta)G^{0} + sin(\theta)G^{90}) \\ &= I(x,y) * G^{\theta} \end{aligned}$$

Designing Custom Filters

Design our own filters by taking the first derivative of a Gaussian function

$$G = e^{-(x^2+y^2)}$$

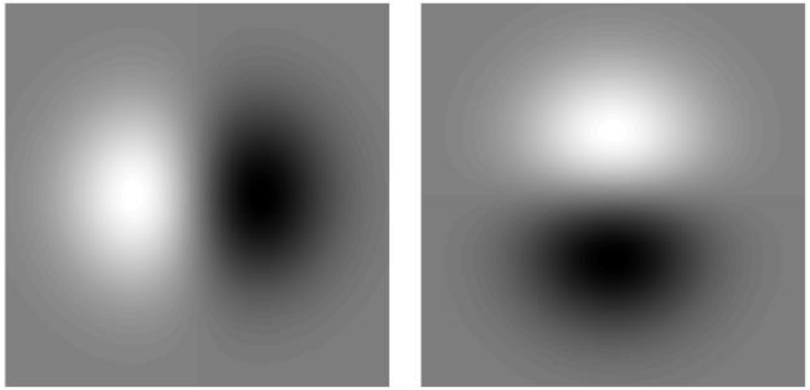
$$G_1^0 = \frac{\partial}{\partial x} e^{-(x^2+y^2)} = -2xe^{-(x^2+y^2)}$$

$$G_1^{90} = \frac{\partial}{\partial y} e^{-(x^2+y^2)} = -2ye^{-(x^2+y^2)}$$

Notice how G^{0} is the x derivative and G^{90} is the y derivative.

Derivative of Gaussian as Steerable filters

- Derivative of Gaussians are used a lot in vision/image processing as a way to detect edges (or generally features)
  - They are great to compute gradients and therefore to detect discontinuities.
    - With a large standard deviation we have poor localization (bad at finding the exact location containing the edge) but good detection (good at detecting proper edges).
    - With low standard deviation we have the opposite, good localization, but poor detection.
  - First order derivatives at different scales can be used to approximate a Laplacian of Gaussian (LoG)
  - The second order can be used to actually compute the Laplacian. LoG are extremely useful to detect blobs.



First order of Derivative of Gaussians in x and y directions

- These two Gaussians are one choice of basis filters to compute a Oriented Derivative of Gaussians of angle θ. We want to find the filter oriented along the direction $d = (\cos\theta, \sin\theta)$
- The steering functions are simply: $k_x(\theta) = \cos(\theta)$ $k_y(\theta) = \sin(\theta)$
- Therefore, using the first equation in the post, our oriented DoG is obtained with: $G_\theta = \cos(\theta)G_x + \sin(\theta)G_y$

https://flashypixels.wordpress.com/2017/02/17/derivative-of-gaussians-steerability/

Derivation of Steerability functions for Derivative of Gaussians

Let's first express the first order derivative of a gaussian in the x direction in polar coordinates as:

$$G_x(r, \phi) = -2re^{-r^2}cos(\phi)$$

If we then rotate ϕ by $\frac{\pi}{2}$ we will obviously obtain the derivative in the y direction:

$$G_x(r, \phi - \frac{\pi}{2}) = G_y(r, \phi) = -2re^{-r^2}cos(\phi - \frac{\pi}{2}) = -2re^{-r^2}sin(\phi)$$

Now, let's rotate the filter by an arbitrary angle θ, using trigonometric identities we obtain:

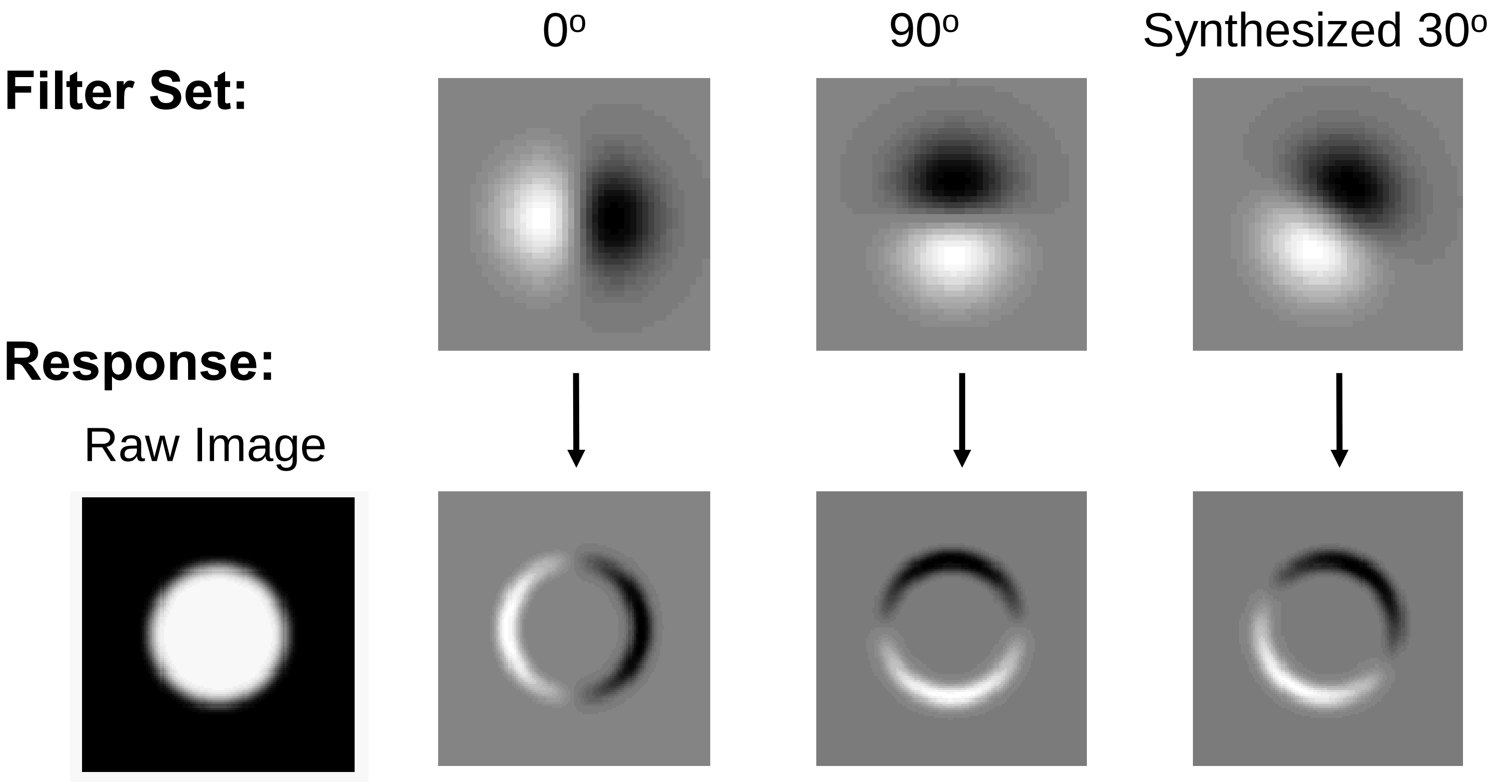
$$G_x(r, \phi - \theta) = -2re^{-r^2}\cos(\phi - \theta) = -2re^{-r^2}(\ \cos(\phi)\cos(\theta) + \sin(\phi)\sin(\theta)\ ) =$$

$$= \cos(\theta)G_x(r, \phi) + \sin(\theta)G_y(r, \phi)$$

Steerability example

“Steerability”-- the ability to synthesize a filter of any orientation from a linear combination of filters at fixed orientations.

$$G_{\theta}^{1} = \cos(\theta)G_{0}^{1} + \sin(\theta)G_{90}^{1}$$



0º

90º

Synthesized 30º

Filter Set:

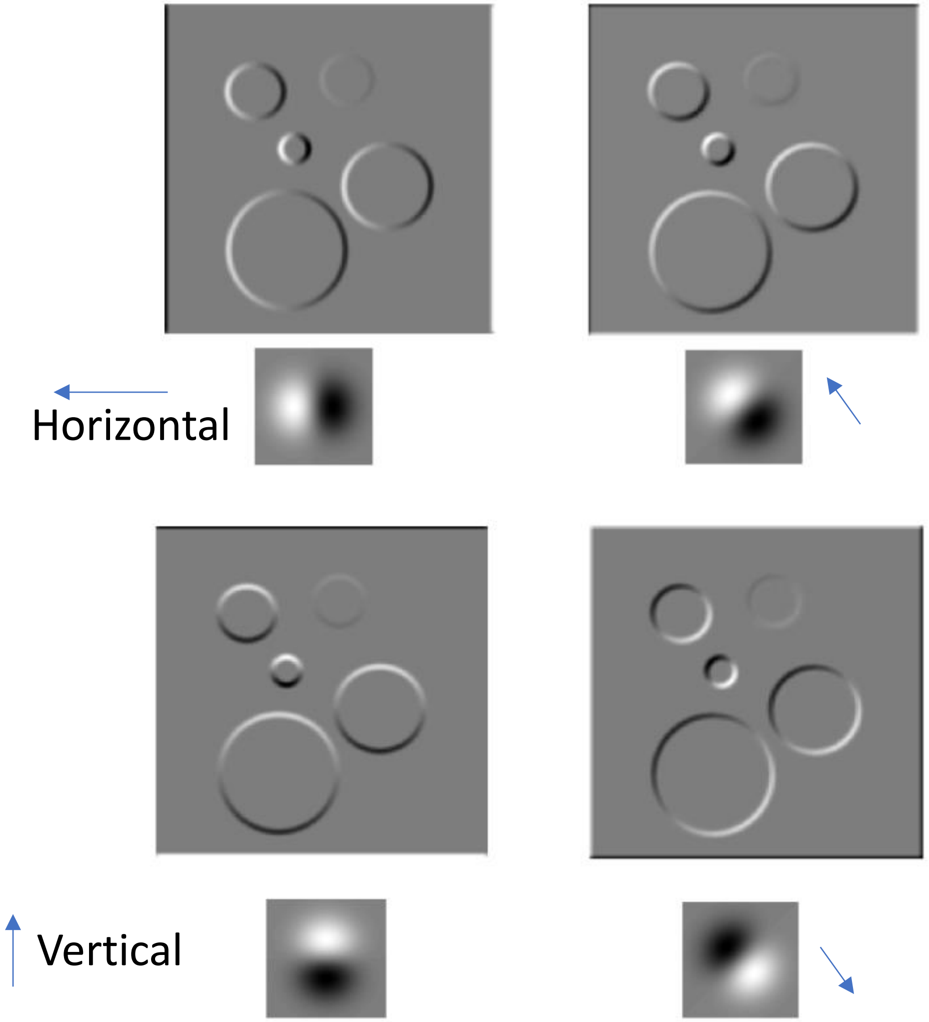
Response:

Raw Image

Taken from:
W. Freeman, T. Adelson, “The Design and Use of Sterrable Filters”, IEEE Trans. Patt, Anal. and Machine Intell., vol 13, #9, pp 891-900, Sept 1991

Derivative of Gaussian Steerability Example

Compute the gradients at a certain direction:



References

- https://medium.com/@itberrios6/steerable-pyramids-6bfd4d23c10d

- https://medium.com/@itberrios6/a-quick-introduction-to-steerable-filters-3fd8813f2e63

- https://rafat.github.io/sites/wavebook/advanced/steer.html

- The design and use of steerable filters: https://www.cs.jhu.edu/~misha/ReadingSeminar/Papers/Freeman91.pdf

- https://flashypixels.wordpress.com/2017/02/17/derivative-of-gaussians-steerability/