Computational Project 1

Numerical Analysis

Lado Turmanidze

April 28, 2024

1 Digital Images, Edges, and Derivatives

1.1 Define edges and provide examples to illustrate the concept.

A segment(region) of space, that undergoes sudden change for a function under inspection is called an **edge**. As we know from calculus, derivatives tell us about rate of change, so for $f : \mathbb{R}^n \to \mathbb{R}^n$, $\|\nabla f(x)\|$ is its first derivative.

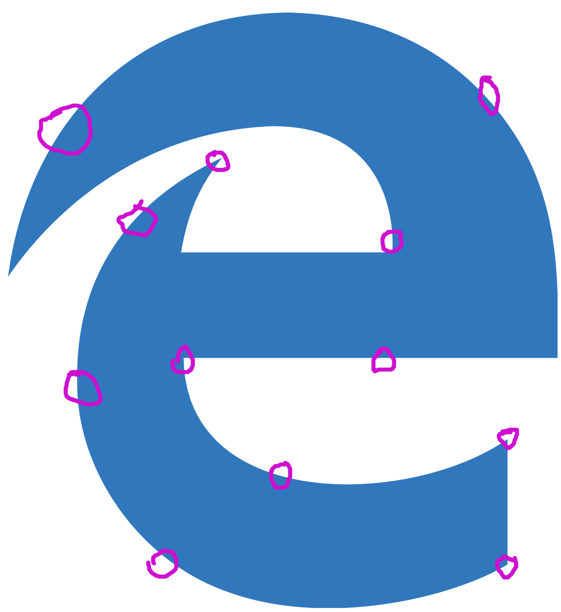

Figure 1: Microsoft Edge Logo with some edges circled in purple

1.2 Explain how derivatives are utilized for edge detection in one and two dimensions. Describe edge indicators and supplement your explanation with visual examples.

1. *One-Dimensional Edge Detection*: In the one-dimensional case, edge detection is typically performed on signal data(audio, time-series, financial data, ...). The derivative is used to detect abrupt changes or discontinuities in the signal.

2. *Two-Dimensional Edge Detection*: In the two-dimensional case, edge detection is applied to images, where edges correspond to boundaries between different regions, objects, entities in the image.

An **edge indicator** is a mathematical function or operator(example: Sobel operator) that detects edges in the image.

If the absolute value of the first derivative exceeds a certain threshold, it indicates the presence of an edge. The higher the absolute value of the derivative, the sharper the edge.For one-dimensional signal $f(x)$, The first derivative $\nabla f(x) = f'(x)$ can be approximated using finite differences:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

If $\|\nabla f(x)\| = \|f'(x)\|$ is greater than a predefined threshold, an edge is detected at x.

In the following example the photo(the first one in a row) gets Sobel operator applied horizontally(the second image) and vertically(the third image), and then these two are combined to get the final image, where we can clearly see edges of the original photo.

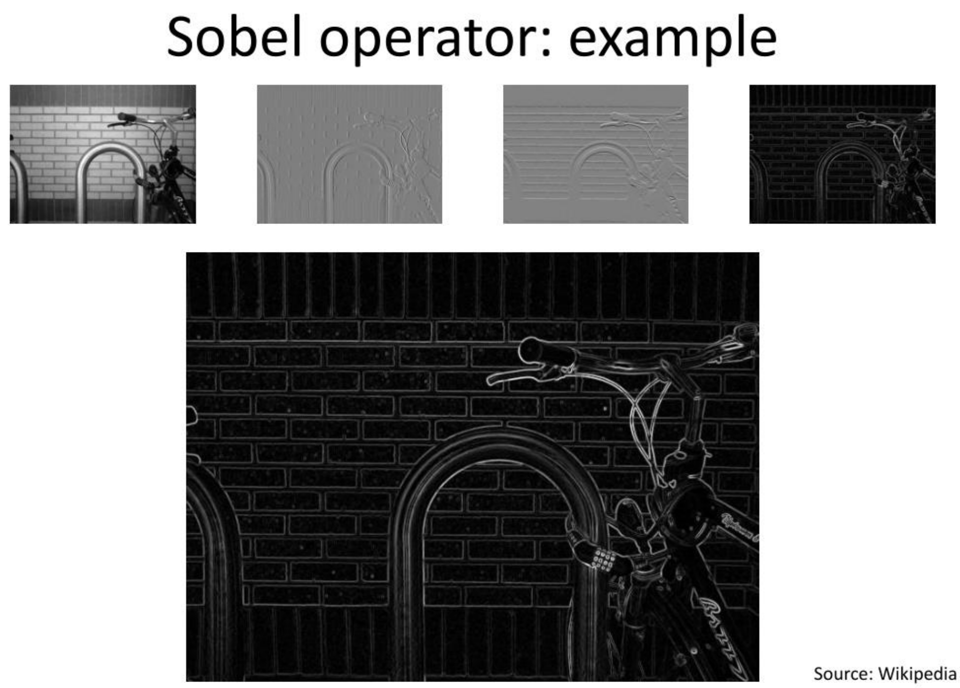

Figure 2: Example of Sobel Operator Application

One of the most widely used edge detection methods is the gradient-based approach, which utilizes first-order derivatives to estimate the gradient magnitude and direction at each pixel. The gradient of an image $F(x, y)$ is given by:

$$\nabla F = \left[\frac{\partial F}{\partial x}, \frac{\partial F}{\partial y}\right]$$

The gradient magnitude $|\nabla F|$ provides an indication of the edge strength, while the gradient direction $\theta = \tan^{-1}\left(\frac{\partial F}{\partial y} / \frac{\partial F}{\partial x}\right)$ gives the orientation of the edge.

The partial derivatives $\frac{\partial F}{\partial x}$ and $\frac{\partial F}{\partial y}$ are typically approximated using finite differences or gradient operators, such as the Sobel operator or the Prewitt operator.

For example, the Sobel operator already described above approximates the partial derivatives using the following kernels:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

The gradient magnitude $|\nabla F|$ and direction θ are then computed as:

$$|\nabla F| = \sqrt{(G_x * F)^2 + (G_y * F)^2}$$

$$\theta = \tan^{-1}\left(\frac{G_y * F}{G_x * F}\right)$$

where $*$ denotes the convolution operation.

1.3 Investigate the impact of truncation error in finite difference formulas on edge detection. Support your findings with visual evidence.

I will consider finite difference with respect to the first variable. Finite differences form that is accurate to the second degree is

$$\nabla_x f(x,y) = \frac{f(x+h,y) - f(x-h,y)}{2h} + O(h^2)$$

even though Centered Finite Differences is generally better approximation than Forward Finite Differences

$$\nabla_x f(x) = \frac{f(x+h,y) - f(x,y)}{h} + O(h)$$

since Taylor expansion(which derives second degree accuracy form) gets truncated at a finite length, there will also be error in second degree approximation.

As an example, let's look at

$$f(x) = \begin{cases} x^2, & \text{if } x < 0 \\ x^3, & \text{otherwise} \end{cases}$$

With $h = 0.2$, we get the following result:

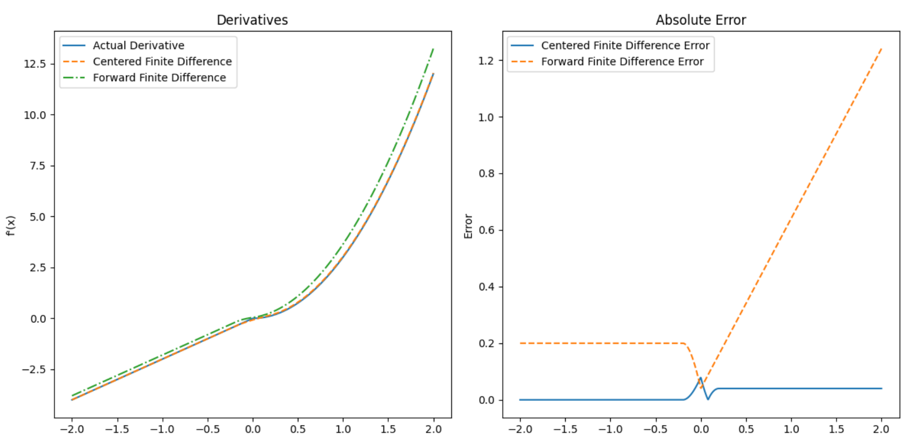

Figure 3: Comparison of Actual Derivative and Finite Differences

2 Digital Images, Features, and Higher Order Derivatives

2.1 Investigate whether higher order derivatives can be employed for edge detection.

An example of sharpening filter is the Laplacian, which can be expressed as

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

where

$$\frac{\partial^2 f}{\partial x^2} = f(x+h,y) + f(x-h,y) - 2f(x,y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x,y+h) + f(x,y-h) - 2f(x,y)$$

so

$$\nabla^2 f = f(x+h,y) + f(x-h,y) + f(x,y+h) + f(x,y-h) - 4f(x,y)$$

The filter mask for the Laplacian is: $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

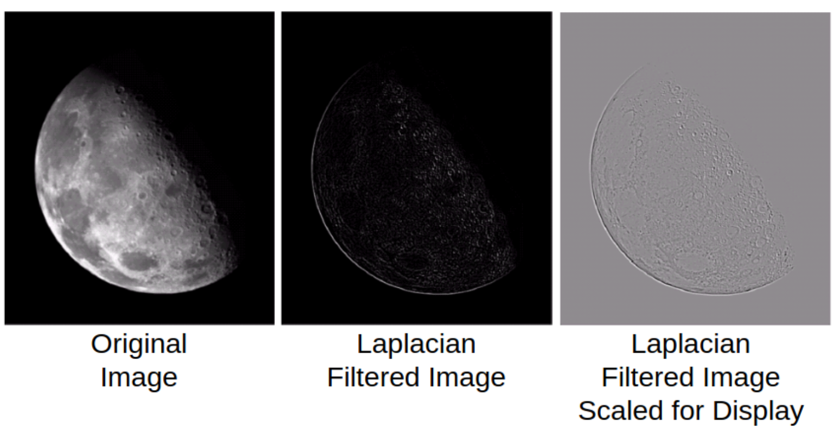

Figure 4: Highlighting edges using Laplacian

2.2 Explore the potential of higher order derivatives in extracting other features of digital images.

Higher order derivatives are used to enhance and refine the edge detection process. The second order derivative especially is used in edge detection for the following reasons:

1. Edge sharpening: The second order derivative can help sharpen and localize edges by identifying regions where the gradient changes rapidly, Since second order derivative is zero at points of constant or linear intensity changes and nonzero at points of high curvature or edges.

2. Noise suppression: The second order derivative can help suppress noise(unwanted or random disturbances that degrade the quality of an image) in images by identifying and removing regions with low curvature or gradual intensity changes, which are often associated with noise.

3. Edge thinning: Thinning is a process that aims to reduce the thickness of edges to a single pixel wide. This process is particularly useful for applications that require precise localization of object boundaries like object recognition, shape analysis, etc.

The result of the Laplacian filtering is not an enhanced image. To enhance image, we first define $g(x, y)$ for as follows:

$$\begin{bmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \\ p_7 & p_8 & p_9 \end{bmatrix} \qquad g(x,y) = \begin{cases} f(x,y) - \nabla^2 f, & p_5 < 0 \\ f(x,y) + \nabla^2 f, & p_5 > 0 \end{cases}$$

Just by simplifying, we can get: $g(x, y) = 5f(x, y) \pm f(x+h, y) - f(x-h, y) - f(x, y+h) - f(x, y-h)$, or:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

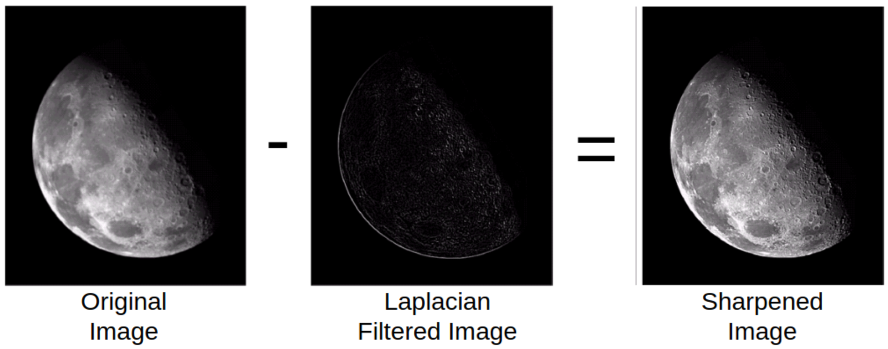

Figure 5: Image Enhancement using Laplacian

Figure 6: Same as above, just closer

3 Exploring the Applications of Derivatives

Provide two illustrations of how derivatives are applied in real-world scenarios. Ensure one of the examples is connected with the concept of edge detection. Present a visual representation for at least one of the examples. Clarify why this application works and mention the tools that can be utilized for this purpose.

In technical analysis of stock market, a company's stock is a function, which is constantly changing. Professional traders, financial analysts and quants(people who apply higher echelon of mathematics to the world of finance) use derivatives(actually finite differences, since graph can have discontinuities) to predict how the value of the stock might change. Of course, world of finance is far too complicated for derivative to solve everything, but at least they form basis for many things, using famous Black-Scholes formula, which gives a theoretical estimate of the price of European-style options and shows that the option has a unique price given the risk of the security and its expected return:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0$$

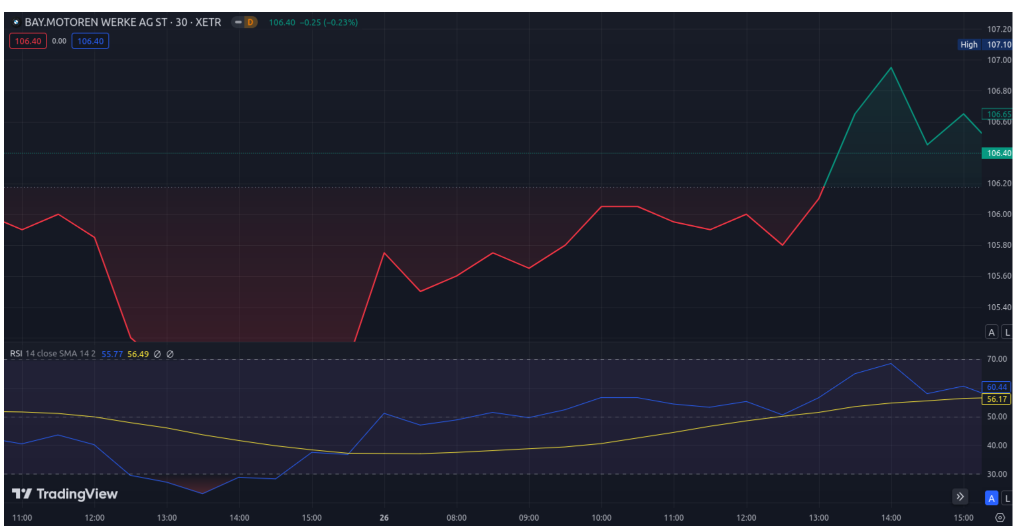

Figure 7: BMW German Stock Exchange Chart, April 28 2024

As mentioned before, edges can be seen as abrupt changes in intensity, which correspond to high values of the gradient. An example is shown below:

(a) Grayscale image.

(b) Horizontal derivative f_x.

(c) Vertical derivative f_y.

(d) Gradient module image $M = \sqrt{f_x^2 + f_y^2}$.

Figure 8: Wind Turbines with Sobel

Then, once the respective operators are applied, an approximation of the gradient (stored in two matrices) is obtained, containing the approximation of the partial derivatives f_x and f_y. The gradient magnitude image M is calculated using $\sqrt{f_x^2 + f_y^2}$. Finally, the edges image is obtained by thresholding the gradient magnitude image, i.e. for a pixel at position i, j in the image the algorithm computes for γ threshold:

$$\text{edges}[i,j] = \begin{cases} 1, & \text{if } M[i,j] \geq \gamma \\ 0, & \text{if } M[i,j] < \gamma \end{cases}$$

A black and white image is obtained with white edges. Different threshold values lead to different results: thicker or more edges for smaller thresholds, fewer and sparser edges for larger thresholds. This γ, and the others mentioned below, are defined as a percentage of the maximum value of the gradient image. The advantage of this approach is to adapt the threshold to the dynamic range of the image. Therefore, the parameter threshold in the algorithms takes values between 0 and 1. Pseudo-code for the algorithm is given below:

Algorithm 1 First derivative edge detection algorithm

Require: input image, threshold γ.

```
Define operator_x and operator_y
f_x ← convolution(image, operator_x)
f_y ← convolution(image, operator_y)
max_M ← 0
for all pixel i in image do
    M[i] ← sqrt(f_x^2 + f_y^2)
    if M[i] > max_M then
        max_M ← M[i]
    end if
end for
for all pixel i in image do
    if M[i] ≥ γ × max_M then
        edges[i] ← 1
    else
        edges[i] ← 0
    end if
end for
return edges
```

For this example, edge detection can be used to compute speed of the windmill turbine, or if camera has very high quality, "health" of the turbines can be checked by shape of the edge.

4 Digital Images, Features, and Linear Combinations of Derivatives

4.1 Investigate whether linear combinations of derivatives can be employed for edge detection. Describe the method used to select coefficients in the linear combination of derivatives and justify your approach.

Based on the analysis of the paper "The Design and Use of Steerable Filters" by Freeman and Adelson, linear combinations of derivatives can indeed be employed for edge detection. Steerable filter refers to a class of filters in which a filter of arbitrary orientation is synthesized as a linear combination of a set of "basis filters". The key idea is to construct an interpolation function $G(x, y)$ from the image intensities, and then find linear combinations of derivatives $of G(x, y)$ that approximate certain "template" edge models concentrated around the origin.

The method used to select the coefficients in the linear combination is as follows:

1. Define a set of basis functions, which are derivatives of $G(x, y)$ up to some maximum order. For example, the paper uses the basis functions $1, G_x, G_y, G_{xx}, G_{xy}, G_{yy}$.

2. Define template edge models $\phi(x, y)$, which are idealized edge profiles that we want to detect in the image. For example, a step edge model with a vertical orientation would be $\phi(x, y) = 1$ for $x < 0$ and $\phi(x, y) = 0$ for $x \geq 0$.

3. Find the linear combination of basis functions that best approximates each template edge model $\phi(x, y)$ in a least-squares sense over a small window around the origin. This determines the coefficients in the linear combination.

4. Convolve the image with these linear combinations of derivatives to obtain edge detector responses that are maximal when the local image pattern matches the corresponding template edge model.

The justification for this approach is that edges in real images can be locally modeled by the idealized template edge profiles, and the linear combinations provide optimal approximations to these templates using the available derivative basis functions over a small window. By convolving with different linear combinations tuned to different edge models, one can detect edges of various orientations and profiles in the image.

Consider the 2-dimensional, circularly symmetric Gaussian function, G, written in Cartesian coordinates, x and y:

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

where scaling and normalization constants have been set to 1 for convenience. The directional derivative operator is steerable as is well-known [8, 12, 16, 18, 21, 22, 23, 24, 27, 34]. Let us write the nth derivative of a Gaussian in the a direction as G_n. Let $(\cdots)^\theta$ represent the rotation operator, such that, for any $f(x, y)$, $f^\theta(x, y)$ is $f(x, y)$ rotated through an angle θ about the origin. The first x derivative of a Gaussian, $G_1^{0^\circ}$, is

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

That same function, rotated by 90°:

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

It is straightforward to show that a G_1 filter at an arbitrary orientation θ can be synthesized by taking a linear combination of $G_1^{0^\circ}$ and $G_1^{90^\circ}$:

$$G_1^{\theta^\circ} = \cos(\theta) G_1^{0^\circ} + \sin(\theta) G_1^{90^\circ}$$

Because convolution is a linear operation, we can synthesize an image filtered at an arbitrary orientation by taking linear combination of the images filtered with $G_1^{0^\circ}$ and $G_1^{90^\circ}$. Letting $*$ denote convolution, if

$$R_1^{0^\circ} = G_1^0 * I \text{ and } R_1^{90^\circ} = G_1^{90} * I \implies R_1^\theta = \cos(\theta) R_1^{0^\circ} + \sin(\theta) R_1^{90^\circ}$$

The derivative of Gaussian filters offer a simple illustration of steerability.

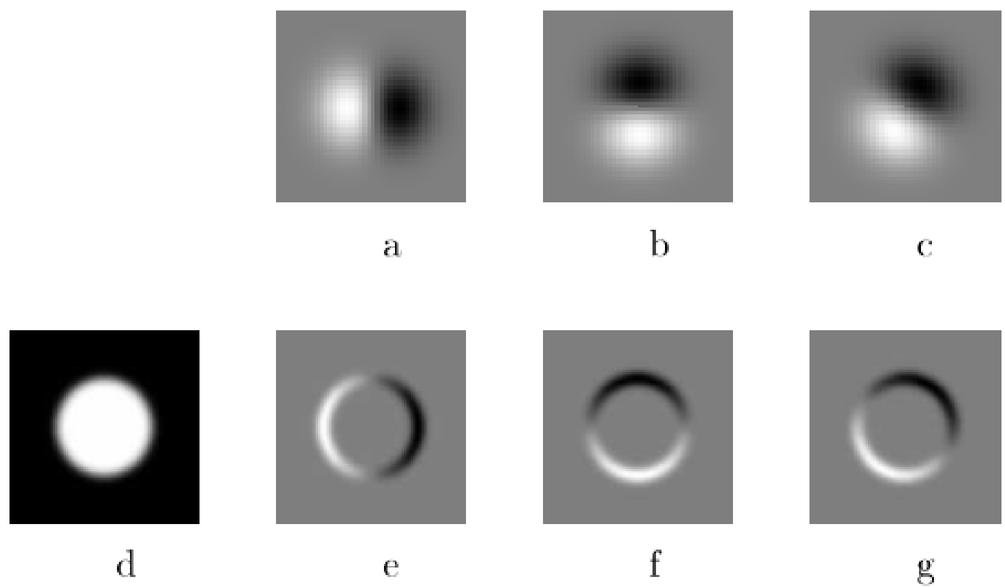

Figure 9: (a) $G_1^{0^\circ}$, first derivative with respect to x (horizontal) of a Gaussian.
(b) $G_1^{90^\circ}$, which is $G_1^{0^\circ}$, rotated by 90°. From a linear combination of these two filters, one can create G_1^θ, an arbitrary rotation of the first derivative of a Gaussian.
(c) $G_1^{30^\circ}$, formed by $\frac{1}{2}G_1^{0^\circ} + \frac{\sqrt{3}}{2}G_1^{90^\circ}$. The same linear combination is used to synthesize G_θ^1 from the basis filters will also synthesize the response of an image to G_1^θ from the responses of the image to the basis filters.
(d) Image of circular disk.
(e) $G_1^{0^\circ}$ (at a smaller scale than pictured above) convolved with the disk, (d).
(f) $G_1^{90^\circ}$ convolved with (d). (g) $G_1^{30^\circ}$ convolved with (d), obtained from $\frac{1}{2}$ [image (e)] + $\frac{\sqrt{3}}{2}$[image (f)].

4.2 Explore the potential of linear combinations of derivatives in extracting other features of digital images. Describe the method used to select coefficients in the linear combination of derivatives and justify your approach.

The same paper segment discusses applications of steerable filters in computer vision and image processing tasks. Some of which are:

1. Image enhancement and noise removal: Steerable filters can be used to enhance features like edges, lines, and ridges in images while simultaneously removing noise.

2. Feature detection: The filters can detect local features like edges, lines, corners etc. in images by analyzing the filter responses at different orientations.

3. Image reconstruction: By combining the oriented filter responses, the original image can be reconstructed from its components, enabling applications like image compression and coding.

4. Motion analysis: The oriented filter responses can be used to estimate local image motion and optical flow between frames in a video sequence.

5. Shape analysis: Steerable filters tuned to different scales can enable multi-scale shape representation and description of object boundaries and shapes in images.

The key advantage highlighted is that steerable filters provide analytical steerability to adapt their orientations without having to re-compute filter coefficients, making them computationally efficient for orientation-dependent imaging tasks compared to non-steerable filters.