

# Assignment 4

Paolo Deidda, Raffaele Perri

March 25, 2025

## Problem 1 [5 points]

A synthetic color image 1 was created where:

- The red channel had a strong transition from bright to dark on the left side.
- The green channel had the opposite transition from dark to bright.
- The blue channel was kept constant.

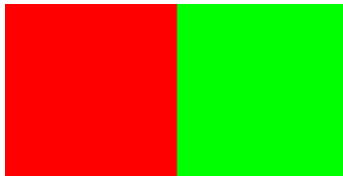


Figure 1: Synthetic image

Applying the Sobel operator to each channel individually and summing the results led to the cancellation of gradients, demonstrating the issue.

## Implementation

The Sobel operator was applied separately to the R, G, and B channels, and their gradients were summed.

```
20 gx = np.array([[ -1,  0,  1],
21                [ -2,  0,  2],
22                [ -1,  0,  1]], dtype=np.float32)
23 gy = gx.T # swaps rows and columns of gx
24
25 # Function to compute the x and y gradients using cv2.filter2D
26 def compute_gradients(channel):
27     grad_x = cv2.filter2D(channel.astype(np.float32), -1, gx)
28     grad_y = cv2.filter2D(channel.astype(np.float32), -1, gy)
29     return grad_x, grad_y
```

```

30
31 # Split the image into B, G, R channels (note: OpenCV uses BGR
    order)
32 B, G, R = cv2.split(img)
33
34 # Compute gradients for each channel
35 Rx, Ry = compute_gradients(R)
36 Gx, Gy = compute_gradients(G)
37 Bx, By = compute_gradients(B) # This will be zero everywhere
38
39 # Sum the gradients of the three channels as described in the
    problem
40 grad_x_total = Rx + Gx + Bx
41 grad_y_total = Ry + Gy + By

```

## Results

Figure 2 shows the computed gradients for the synthetic image. The first image represents the gradient in the x-direction, the second image shows the gradient in the y-direction, and the third image depicts the overall gradient magnitude.

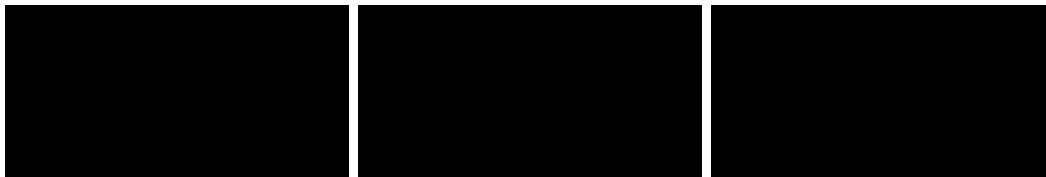


Figure 2: Computed gradients: (left) Gradient X, (middle) Gradient Y, (right) Gradient Magnitude

## Conclusion

This happens because when computing the gradient across channels, the strong increase in one channel can be canceled by a strong decrease in another. In this case, the red channel transitions from strong to zero, while the green channel transitions from zero to strong. When summed, these opposing gradients effectively nullify each other, resulting in a misleading overall gradient calculation.

This demonstrates the issue with summing individual color gradients. The results show that edges can be present in a color image while the summed gradient remains zero. Therefore we need an alternative gradient computation methods in color image processing.

## Problem 2 [5 points]