

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

// Function to compute the gradient magnitude from horizontal and vertical
// differential images
void computeGradient(unsigned char *horizontal, unsigned char *vertical, unsigned
char *gradient, int width, int height) {
    for (int i = 0; i < height; ++i) {
        for (int j = 0; j < width; ++j) {
            int horizontalValue = horizontal[i * width + j];
            int verticalValue = vertical[i * width + j];

            // Calculate the gradient magnitude using the Euclidean distance
            formula
            int magnitude = (int)sqrt(horizontalValue * horizontalValue +
            verticalValue * verticalValue);

            // Store the result in the gradient image
            gradient[i * width + j] = (unsigned char)magnitude;
        }
    }
}

// Function to detect edge points using a threshold
void detectEdges(unsigned char *gradient, unsigned char *edge, int width, int
height, unsigned char threshold) {
    for (int i = 0; i < height; ++i) {
        for (int j = 0; j < width; ++j) {
            if (gradient[i * width + j] >= threshold) {
                edge[i * width + j] = 200; // Mark edge points with a bright
            intensity
            } else {
                edge[i * width + j] = 0; // Set non-edge points to zero
            }
        }
    }
}

unsigned char *loadImage(const char *filename, int width, int height) {
    FILE *file = fopen(filename, "rb");
    if (!file) {
        printf("Error opening file %s\n", filename);
        return NULL;
    }

    unsigned char *image = (unsigned char *)malloc(width * height * sizeof(unsigned
char));
    fread(image, sizeof(unsigned char), width * height, file);
    fclose(file);

    return image;
}

int main() {
    int width = 512; // Adjusting to image dimensions
    int height = 512;

    // Load the 3x3 horizontal and vertical differential images

```

```

    unsigned char *horizontal3x3 =
loadImage("/Users/akshatboudh/Desktop/MULTIMEDIA/ladyoutputImage3x3Horizontal.raw",
width, height);
    unsigned char *vertical3x3 =
loadImage("/Users/akshatboudh/Desktop/MULTIMEDIA/ladyoutputImage3x3Vertical.raw",
width, height);

    // Load the 5x5 horizontal and vertical differential images
    unsigned char *horizontal5x5 =
loadImage("/Users/akshatboudh/Desktop/MULTIMEDIA/ladyoutputImage5x5Horizontal.raw",
width, height);
    unsigned char *vertical5x5 =
loadImage("/Users/akshatboudh/Desktop/MULTIMEDIA/ladyoutputImage5x5Vertical.raw",
width, height);

    // Create images for the gradient magnitude and edge detection
    unsigned char *gradientImage = (unsigned char *)malloc(width * height *
sizeof(unsigned char));
    unsigned char *edgeImage = (unsigned char *)malloc(width * height *
sizeof(unsigned char));

    // Compute the gradient magnitude for 3x3 horizontal and vertical images
computeGradient(horizontal3x3, vertical3x3, gradientImage, width, height);

    // Apply a threshold to detect edges for 3x3 images
unsigned char threshold3x3 = 50; // Adjust the threshold based on your criteria
detectEdges(gradientImage, edgeImage, width, height, threshold3x3);

    // Save the 3x3 gradient magnitude and edge images to files
FILE *gradientFile3x3 =
fopen("/Users/akshatboudh/Desktop/MULTIMEDIA/ladyoutputImage3x3Vertical.rawgradient
_image_3x3.raw", "wb");
    fwrite(gradientImage, sizeof(unsigned char), width * height, gradientFile3x3);
    fclose(gradientFile3x3);

    FILE *edgeFile3x3 =
fopen("/Users/akshatboudh/Desktop/MULTIMEDIA/ladyoutputImage3x3Vertical.rawedge_ima
ge_3x3.raw", "wb");
    fwrite(edgeImage, sizeof(unsigned char), width * height, edgeFile3x3);
    fclose(edgeFile3x3);

    // Compute the gradient magnitude for 5x5 horizontal and vertical images
computeGradient(horizontal5x5, vertical5x5, gradientImage, width, height);

    // Apply a threshold to detect edges for 5x5 images
unsigned char threshold5x5 = 100; // Adjust the threshold based on your
criteria
detectEdges(gradientImage, edgeImage, width, height, threshold5x5);

    // Save the 5x5 gradient magnitude and edge images to files
FILE *gradientFile5x5 =
fopen("/Users/akshatboudh/Desktop/MULTIMEDIA/ladyoutputImage5x5Vertical.rawgradient
_image_5x5.raw", "wb");
    fwrite(gradientImage, sizeof(unsigned char), width * height, gradientFile5x5);
    fclose(gradientFile5x5);

    FILE *edgeFile5x5 =
fopen("/Users/akshatboudh/Desktop/MULTIMEDIA/ladyoutputImage5x5Vertical.rawedge_ima
ge_5x5.raw", "wb");

```

```
fwrite(edgeImage, sizeof(unsigned char), width * height, edgeFile5x5);
fclose(edgeFile5x5);

// Free allocated memory
free(gradientImage);
free(edgeImage);
free(horizontal3x3);
free(vertical3x3);
free(horizontal5x5);
free(vertical5x5);

return 0;
}
```