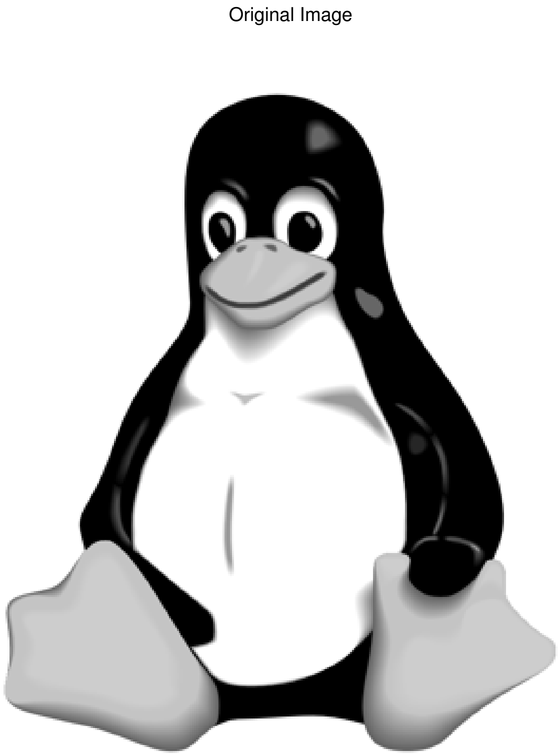
Contents

image preprocessing...

```
clear all, clc, close all;
addpath ../commonFunctions;
%code for my custom functions can be found on
%https://github.com/curtismuntz/machine_vision/tree/master/commonFunctions
original = getIMG('Tux2.png');
I = imresize(original, [400,400]);
I = rgb2gray(im2double(I));
finger= getIMG('HW5finger.jpg');
F = im2bw(rgb2gray(finger));
rmpath ../commonFunctions;
figure('name','original img'), imshow(I), title('Original Image');
```

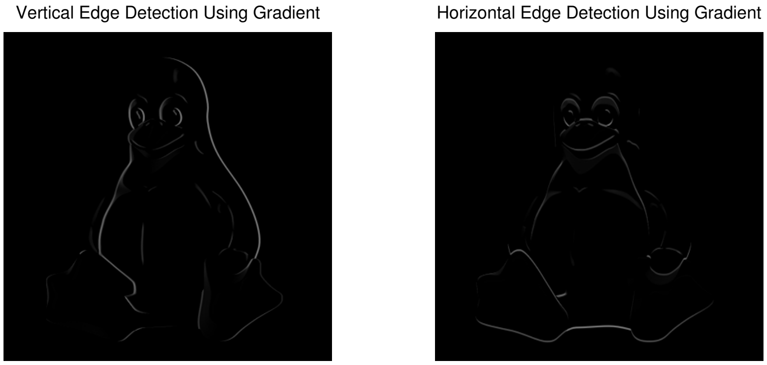


Original Image

Problem 1: Edge detection approximations of the gradient magnitude

Part 1

```
%Perform horizontal and vertical edge detection and display them seperately.
[Ix,Iy]=gradient(I);
figure('name','edges');
subplot(121), imshow(Ix), title('Vertical Edge Detection Using Gradient');
subplot(122), imshow(Iy), title('Horizontal Edge Detection Using Gradient');
```



Vertical Edge Detection Using Gradient

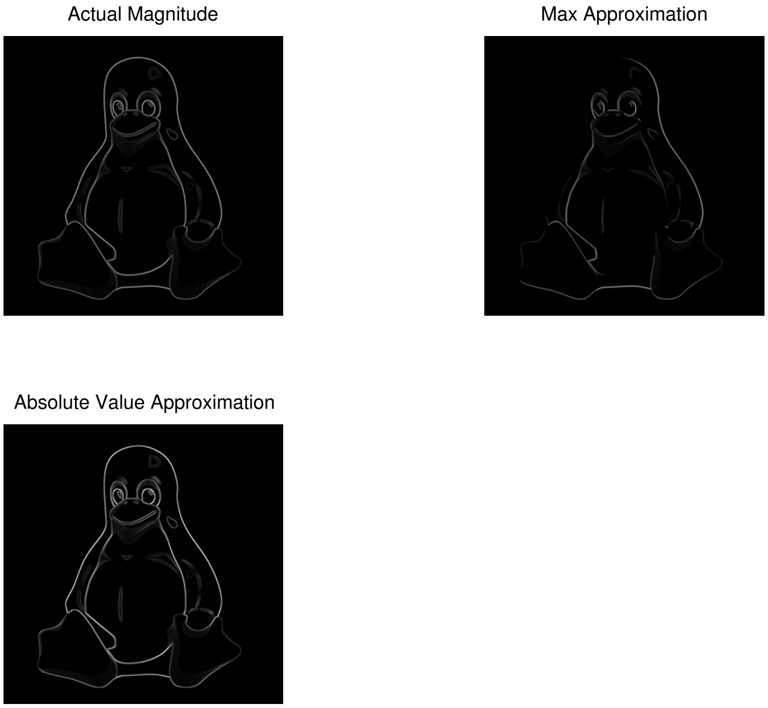
Horizontal Edge Detection Using Gradient

Part 2

```
%Now we want to combine the horizontal and vertical edges. We suggest to use different appro
%a) Magnitude of the gradient: $\nabla f = \sqrt(Ix^2+Iy^2)$
actualMag = sqrt((Ix.^2)+(Iy.^2));
%b) approximation of gradient: $\nabla f = max(Ix,Iy)$
approx1 = max(Ix,Iy);
%c) approximation of gradient: $\abs{\nabla x} + abs{\nabla y}$
approx2 = (abs(Ix) + abs(Iy));
```

Comparing the three, the actual magnitude and the approximation are very similar, meaning the approximations are excellent and have the benefit of being less computationally expensive.

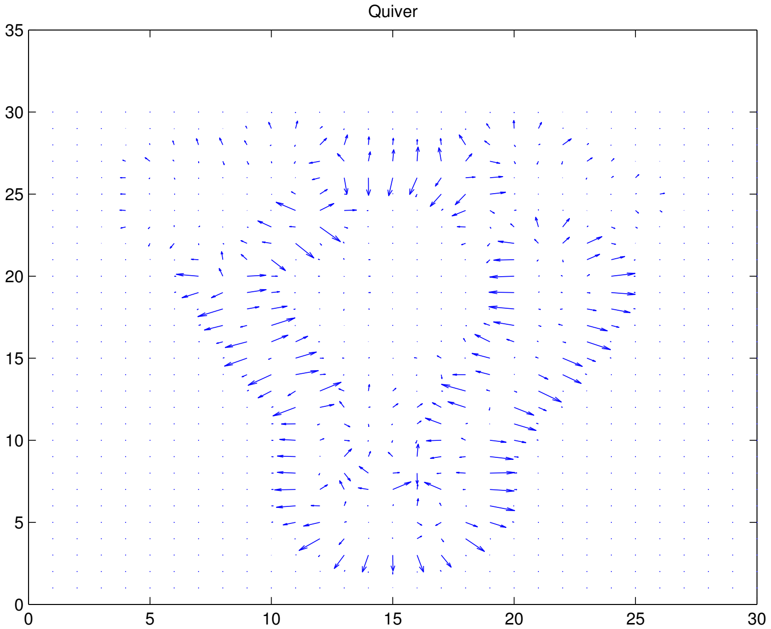
```
figure('name', 'gradient stuff');
subplot(221), imshow(actualMag), title('Actual Magnitude');
subplot(222), imshow(approx1), title('Max Approximation');
subplot(223), imshow(approx2), title('Absolute Value Approximation');
```

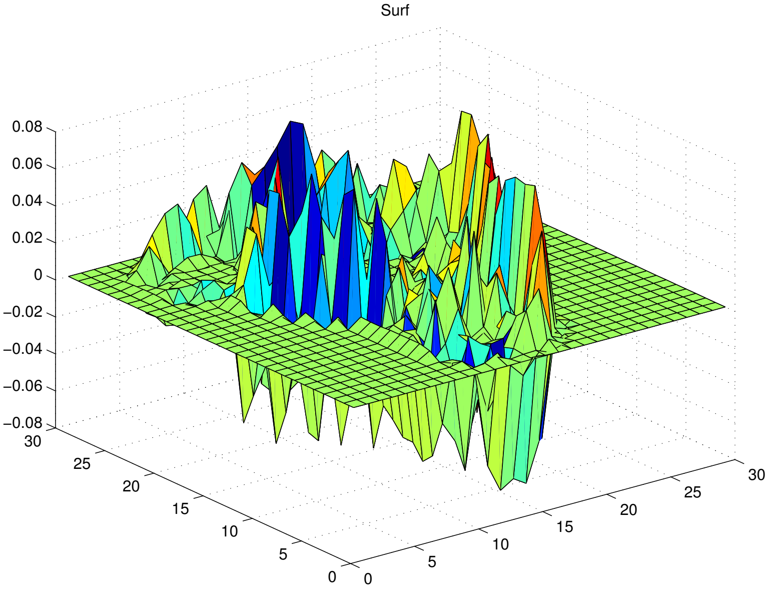


Problem 2: Properties of the gradient

There are other ways of visualizing the gradient such as the built-in 'quiver' and 'surf' plots. These can be effective in seeing the strength of edges, and where the image is flat. The quiver plot shows the strength and relative direction of the gradient, thus combining the ∇f_x and ∇f_y gradients into various arrows corresponding with the direction of the edge. The surf plot shows the same information but in three dimensions. The height of the surf plot corresponds to how fast the image is changing, and it has the highest magnitude on the edges of my chosen image.

```
figure('name', 'Quiver plot')
Ix = imresize(Ix, [30, 30]);
Iy = imresize(Iy, [30, 30]);
quiver(Ix,Iy), title('Quiver');
figure('name', 'Surf plot')
surf(Ix), hold on, surf(Iy), title('Surf');
```

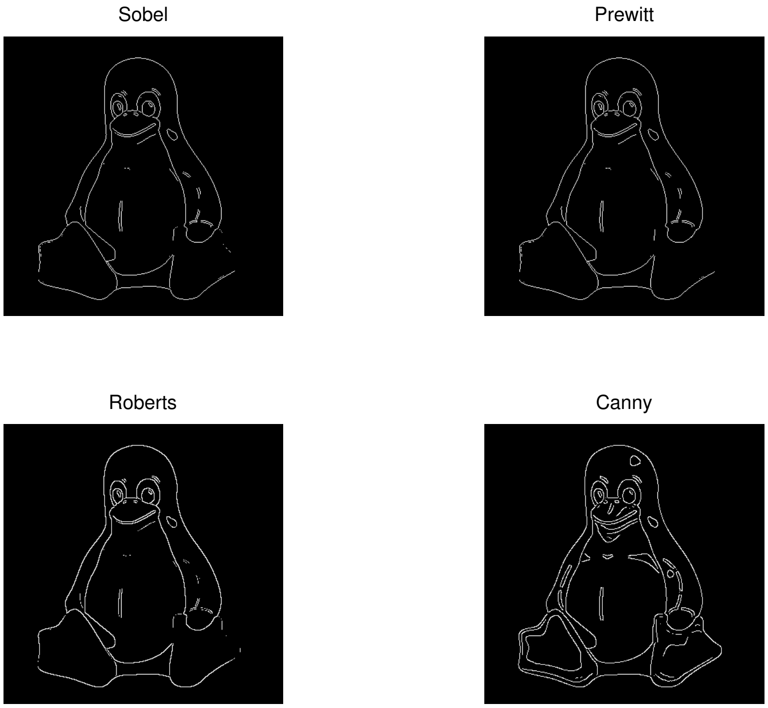




Problem 3: Comparing differnet edge detection approaches

Comparing between the various edge detectors, we can see that for the most part they are very similar. The Canny edge detector has some additional detections, and it found some instances where localized intensity changes create edges. I believe this is due to the additional thresholding process that is built inside of the Canny detection method.

```
%reprocess original image:
I = im2double(rgb2gray(imresize(original, [400, 400])));
figure(’name’, ’Comparing Edge Detection Masks’)
BW = edge(I, ’Sobel’);
subplot(221), imshow(BW), title(’Sobel’);
BW = edge(I, ’Prewitt’);
subplot(222), imshow(BW), title(’Prewitt’);
BW = edge(I, ’Roberts’);
subplot(223), imshow(BW), title(’Roberts’);
BW = edge(I, ’Canny’);
subplot(224), imshow(BW), title(’Canny’);
```



Sobel

Prewitt

Roberts

Canny

Problem 4: Applying morphological filters to remove noise

After applying various elementary morphological filters with different structuring elements, it seems an effective noise removal combination was the disk structuring element being dilated with the original image. Also shown in the remaining plots are dilation with various structuring elements to compare their effectiveness.

```
figure('name', 'fingerprint noise reduction')
subplot(221), imshow(F), title('Original Image with Noise');
S= strel('disk', 2);
BW = imdilate(F, S);
subplot(222), imshow(BW), title('strel Disk Filtered');
S= strel('square', 3);
BW = imdilate(F, S);
subplot(223), imshow(BW), title('strel Square Filtered');
S= strel('line', 3, 45);
BW = imdilate(F, S);
subplot(224), imshow(BW), title('strel Line filtered');
```