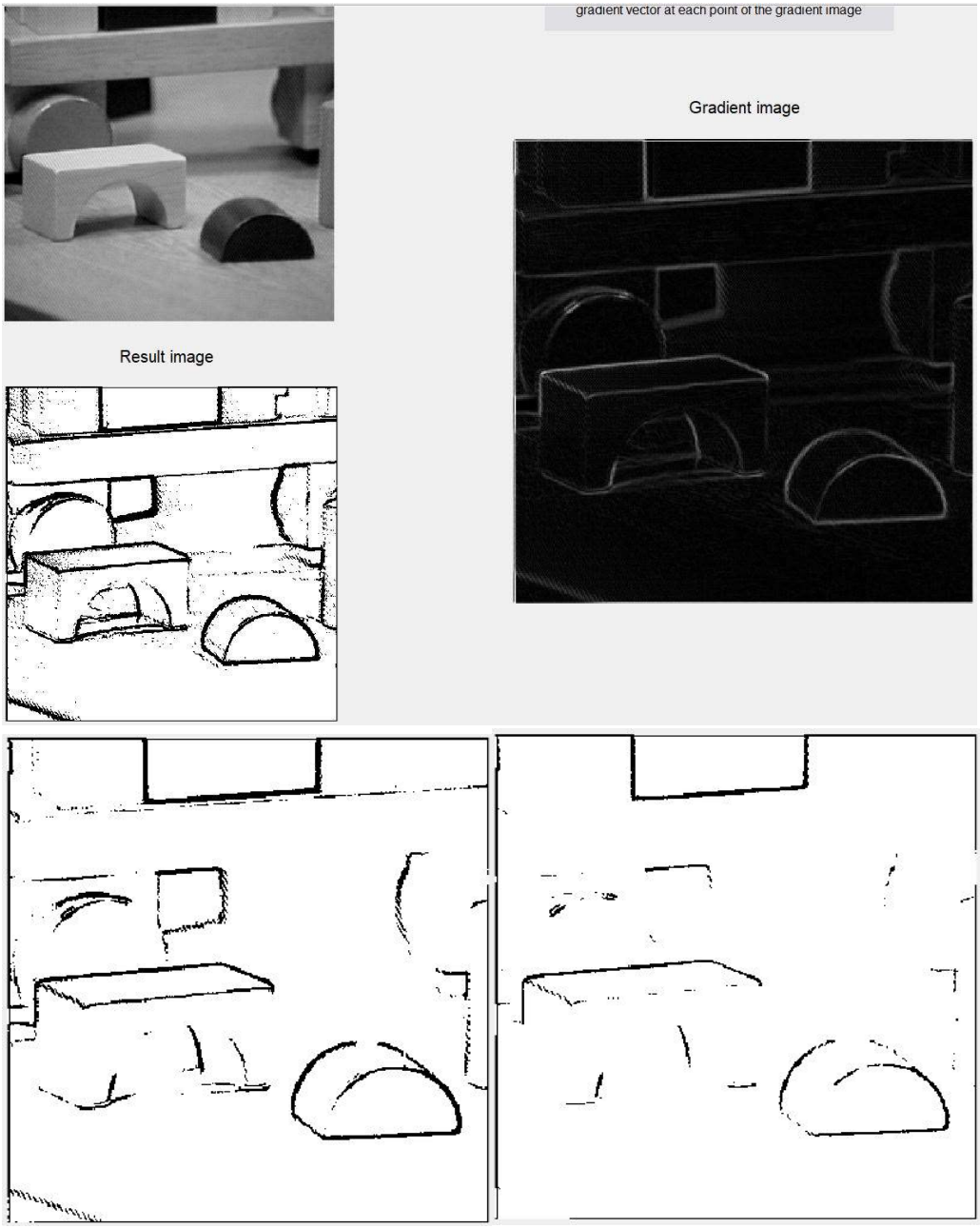
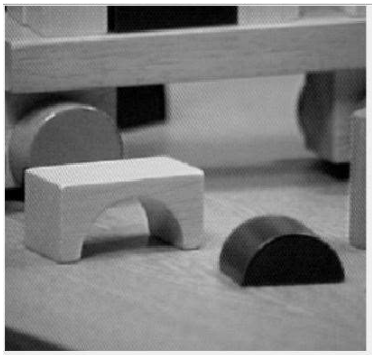
Exercise 2a: Edges Detection with mVision

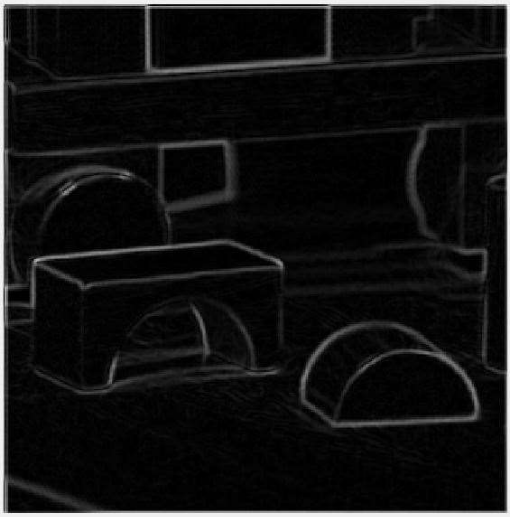


With the increment of the threshold for sobel3 we eliminate a lot of multiple response, the compromise is that we will lose many edges of small scale (above images are for 20,40 and 60 of threshold).

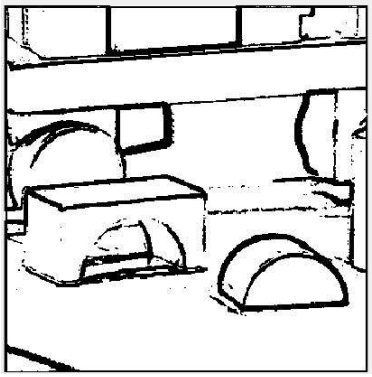


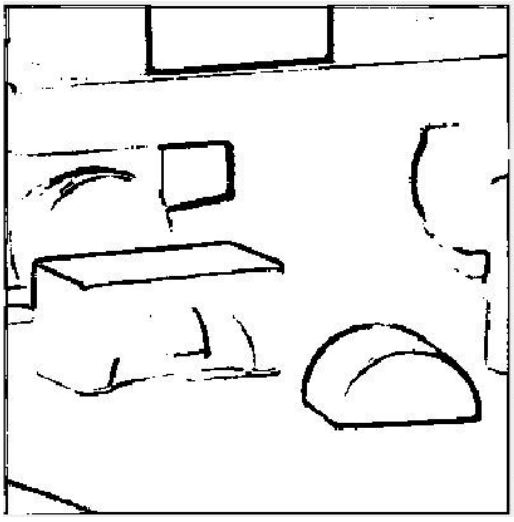
gradient vector at each point of the gradient image

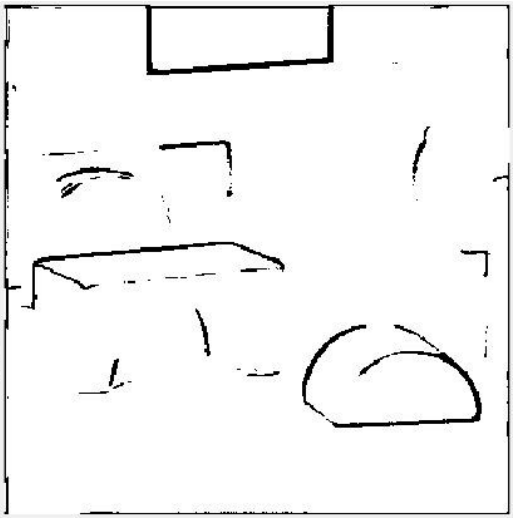
Gradient image



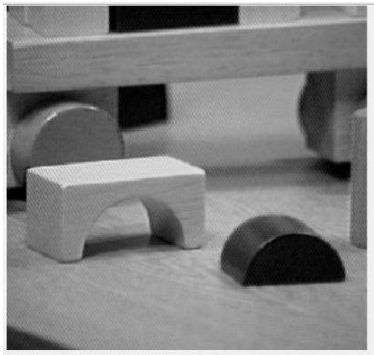
Result image







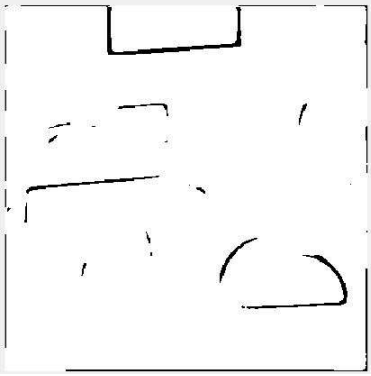
With sobel5 we improve get less multiple response but now the small scale edge detection is even worse (once again images for 20,40 and 60 of threshold)

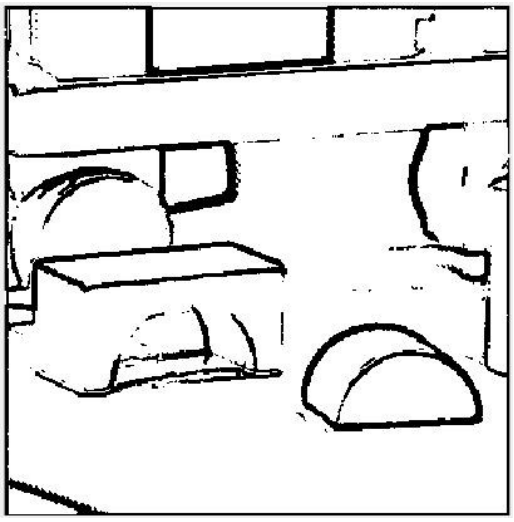


gradient vector at each point of the gradient image

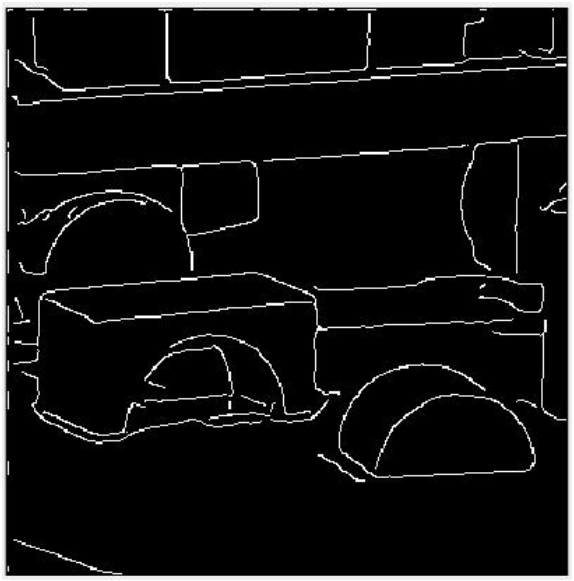
Gradient image

Result image





The DroG's strength is capturing the edges at small and big scales at the same time, although it has a severe case of multiple response, fidgeting with the controls I've discovered that it worksbetter for me with a threshold of 10 a mask size of 5 and a sigma of 1 for this image.



I am getting this image of Canny with a threshold of 0.1 and a Sigma of 2.In my opinion the best edge detector is Canny for its single response and overall precission.

Exercise 2b: Implementing edges detection:

My matlab code for the function:

```
function angles = supersobel(imagen, threshold)
% SUPPERSOBEL Gets an input image and detects the borders by
% means of sobel.

%   The function gets an image and the border detection threshold as the
%   parameters and shows the result for horizontal and vertical sobel and
%   an image of the combination of the original image overlayed with the
%   edges detected shown in green colour, Then returns a matrix with a
%   value of angle of the derivative for each pixel, the normalization
%   of the Sobel matrix makes the threshold useless so Inf is recomended.

    if size(imagen,3)==3
        imagen = rgb2gray(imagen);
    end
    subplot(2,2,1);imshow(imagen);title('Imagen original');

    %Horizontal sobel calculation
    mask = fspecial('sobel');
    horsob = conv2(imagen, mask, 'same');
    horsob = abs(horsob);
    subplot(2,2,2);imshow(uint8(abs(horsob)));
    title('Imagen Sobel horizontal');

    %Vertical sobel calculation
    mask = mask';
    versob = conv2(imagen, mask, 'same');
    versob = abs(versob);
    subplot(2,2,3);imshow(uint8(abs(versob)));
    title('Imagen Sobel vertical');

    %Sobel is created from the combination of both convolutions
    sobel = abs(imadd(horsob,versob));
    sobel(sobel > threshold) = 0;
    %Sobel image is normalized
    aux = 255 * sobel ./ max(max(sobel));
```

```
    %We select what we want in colour
    mask = (aux > 50);
    %Three identical images for the three components of rgb
    im_modulo_r = imagen;
    im_modulo_g = imagen;
    im_modulo_b = imagen;
    % Pixels marked by mask are given 255 in green component = [0 255 0]
    im_modulo_r(mask) = 0;
    im_modulo_g(mask) = 255;
    im_modulo_b(mask) = 0;
    % Bands are created
    im_modulo_rgb = cat(3,im_modulo_r,im_modulo_g,im_modulo_b);

    subplot(2,2,4); imshow(uint8(abs(im_modulo_rgb))); title('Sobel');

    angles = atan(versob/horsob);
end
```

and the resulting image using rice is: