

Faculdade de Engenharia da Universidade do Porto



Fundamentos de processamento de sinais visuais

Tomas Sousa Pereira & Joaquim Pedro Santos

Fevereiro 2019

Índice

1 Experiências com espaços de cor	1
1.1 Script converter para HSV	1
1.1.1 Importar imagem	1
1.1.2 Separar cada componente RGB	2
1.1.3 Converter imagem para o espaço de cores HSV	2
1.2 Converter para YCbCr	4
1.3 Converter para YUV	5
2 Variação das dimensões espaciais de imagem usando ou não filtros com imagem de teste “imzoneplate”	6
3 Experiências de filtragem	9
3.1 Utilização do fspecial e imfilter	9
3.2 Convolução explícita	10
3.3 Detetor de contornos	12

1. Experiências com espaços de cor

1.1 Script converter para HSV

1.1.1 Importar imagem

Vamos começar o trabalho por criar um script que nos permite importar uma imagem no formato bitmap e apresentar essa imagem. Com o código seguinte obtivemos a figura 1.

```
1 im = imread('..../images/praiabmp'); % importar imagem  
2 figure; imshow(im); % apresentar a imagem
```



Figura 1.1: Plot da imagem praia.bmp

1.1.2 Separar cada componente RGB

De seguida com o código que se segue vamos dividir a imagem em cada componente RGB.

Obtendo os resultados nas figuras 1.2 , 1.3, 1.4

```
1 r = im(:, :, 1);  
2 figure;imshow(r);  
3 g = im(:, :, 2);  
4 figure;imshow(g);  
5 b = im(:, :, 3);  
6 figure;imshow(b);
```



Figura 1.2: Componente R

Figura 1.3: Componente G

Figura 1.4: Componente B

1.1.3 Converter imagem para o espaço de cores HSV

Após termos separado as componentes RGB vamos converter a imagem para o espaço de cores HSV com o comando seguinte. Conseguindo assim vizualizar a imagem convertida na figura 1.5 e todas as suas componentes [H,S,V] nas figuras 1.6 , 1.7, 1.7 respetivamente.

```
1 [H,S,V] = rgb2HSV(im);
```



Figura 1.5: Plot da imagem praia.bmp no espaço de cores HSV



Figura 1.6: Componente H



Figura 1.7: Componente S



Figura 1.8: Componente V

De seguida vamos analisar a Figura 1.9 que representa 5 fotografias em 8 componentes de cor diferentes. Vamos então analisar em 2 espaços de cor difentes, RGB que reprensta a quantidade da cor vermelha (R), verde (G) e azul (B). A partir dos valores RGB normalizados e de $\text{MAX} = \max R, G, B$ e $\text{MIN} = \min R, G, B$, é possível chegar aos valores de cada componente. Com $R=G=B$, tratar-se-à de um estímulo de cor cinza, logo não existe H. Com $R=G=B=0$, trata-se-à de um estímulo de cor preta, logo S fica indefinido. Com $R=G=B=1$, trata-se-à de cor branca, logo S=0. O segundo espaço de cor a analisar é o HSV que representa tonalidade (Hue), saturação (Saturation) e intensidade (Value). As cinco fotografias escolhidas têm componentes de cor diferentes como podemos observar na primeira linha da Figura 1.9.

Na componente R (2^a linha) obser-

vamos que as folhasVerdes não tem

qualquer tipo de vermelho e deste modo a imagem parece escura. Por outro lado ambas as 4 outras foto-

grafias tem algum vermelho pelo que

se consegue vizualizar informação. Na

componente G (linha 3) observa-se

que a imagem das floresVermelhas

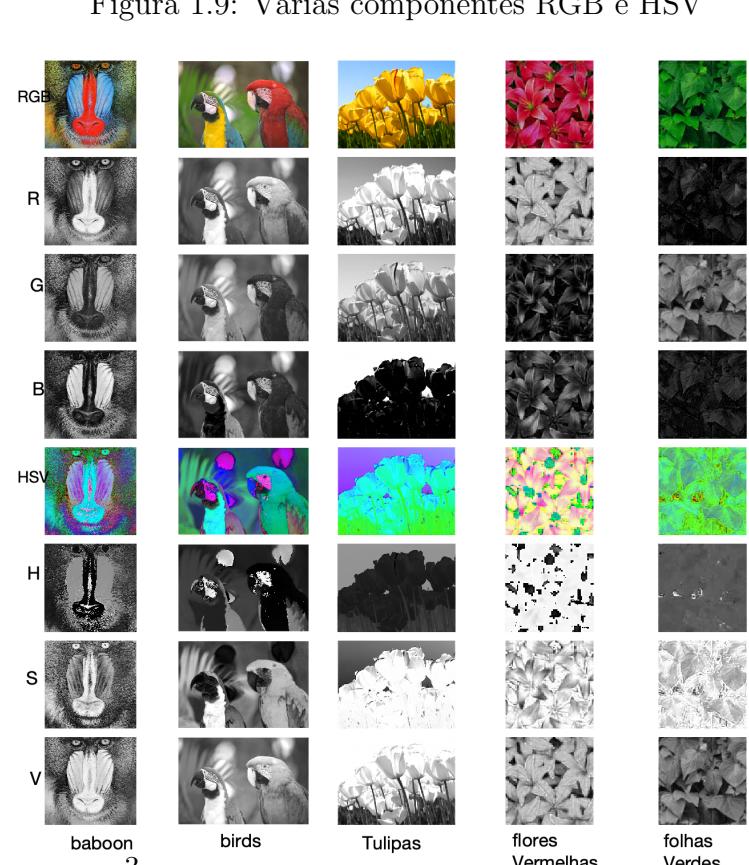
não tem qualquer tipo de vermelho.

Assim como na componente azul (li-

nha 4) as Tulipas têm o céu azul e

é exatamente essa região da imagem

que se observa a claridade que vem



existência da cor azul. Vamos analisar agora as imagens no espaço de cor HSV. Na ante-penúltima linha conseguimos observar a componente H das imagens correspondente à tonalidade da imagem. Na linha seguinte conseguimos observar a componente S correspondente à saturação da imagem . Por fim obtivemos a componente V, conseguindo-se identificar as regiões com cor mais intensidade, e.g. o céu na fotografia das tulipas.

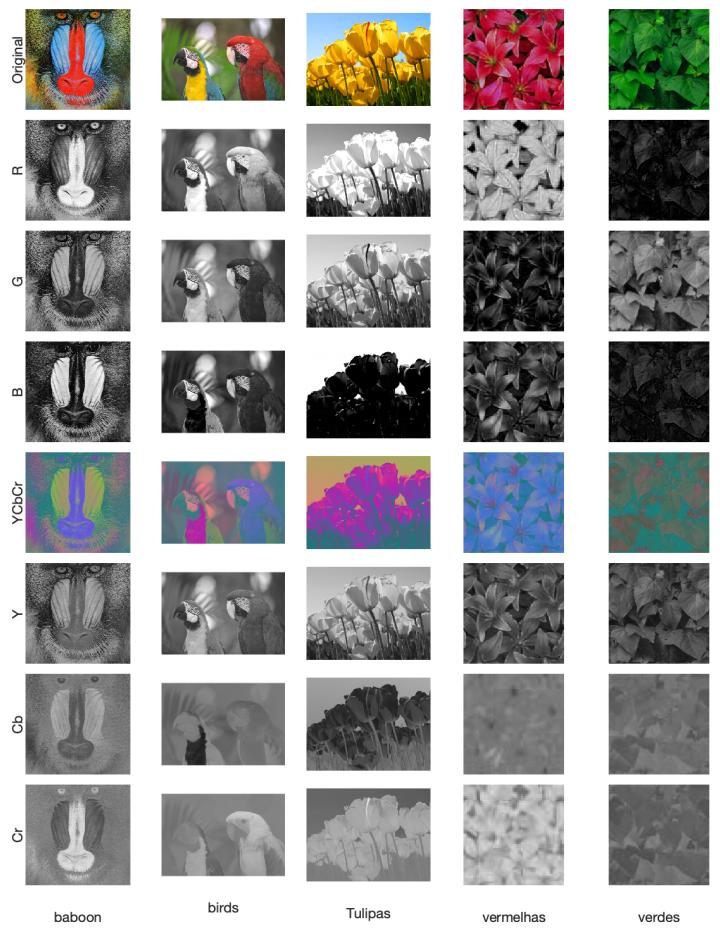
1.2 Converter para YCbCr

Nesta etapa vamos comparar o espaço de cores RGB com o espaço YCbCr.

Este espaço de cores pode ser semelhante ao RGB no entanto responde a algumas considerações do olho humano.

A característica mais importante para o olho humano é a intensidade e corresponde à componente Y. Na imagem baboon consegue-se identificar o animal pois todas as cores são intensas assim como na imagem das tulipas o céu também tem um valor alto de Y. A componente Cb identifica a componente azul da imagem mais uma vez o céu das imagens das tulipas e o rosto do baboon sobressaem devido à sua cor azul. Por outro lado a parte Cr representa a componente vermelha da imagens como observado na imagem birds onde o pássaro da direita é facilmente identificado devido à sua cor vermelha.

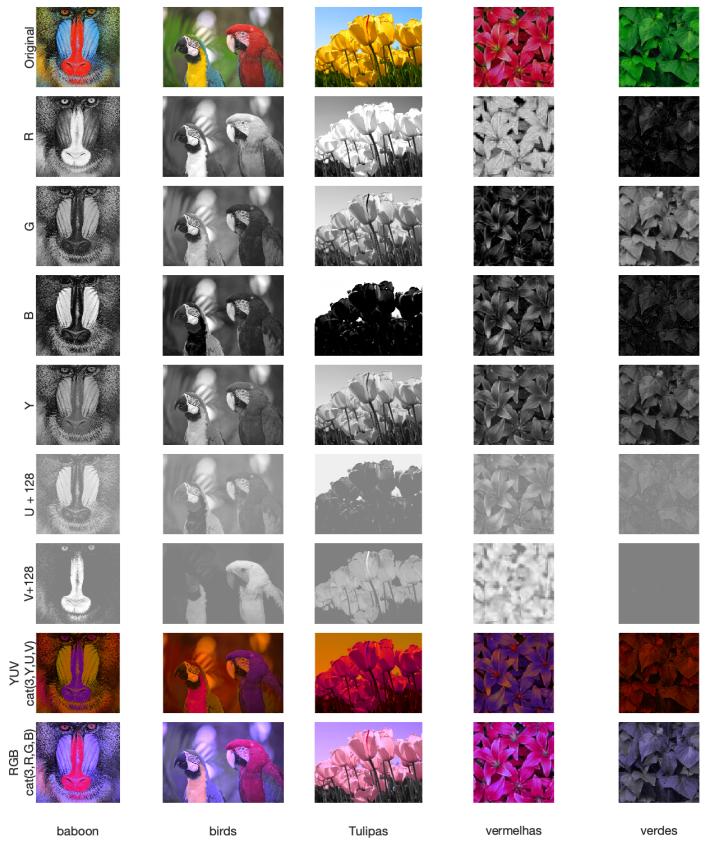
Figura 1.10: Várias componentes RGB e YCbCr



1.3 Converter para YUV

Para concluir o estudo do espaço de cores, vamos analisar as mesmas imagens no espaço de cores YUV. Para tal alteramos o script `rgb2yuv.m` para produzir a Figura 1.11. A componente Y é a mesma do que no espaço de cores YCbCr deste modo se compararmos as linhas da componente Y das Figuras 1.11 e 1.10 percebemos que são iguais. As componentes U e V são resultados das componentes da cor. Se voltarmos a comparar a Figura 1.11 e a Figura 1.10 notamos desde logo bastante parecências. Na realidade estes modelos são equivalentes YUV é usado no mundo analógico e YCbCr no mundo digital. O outro facto que identificamos é o facto das componentes destes espaços de cor serem bastante parecidas com as componentes RGB. Este facto acontece pois cada um destes valores é calculado com base nas componentes RGB, daí tanta semelhança entre as imagens.

Figura 1.11: Várias componentes RGB e YCbCr



2. Variação das dimensões espaciais de imagem usando ou não filtros com imagem de teste “imzoneplate”

Com o objetivo de estudarmos a variação das dimensões espaciais de imagens. O primeiro passo é observar a imagem base presente na Figura 2.1.

De seguida vamos estudar criar um script que nos permite vizualizar e modificar a imgem zoneplate de acordo com o pedido, (script ampliareduzpoweron.m). Com este script é nos permitido fazer o plot da Figura 2.1 Nesta Figura observamos 8 colunas. Na primeira coluna foi feita uma redução de escala usando a função imresize com um fator de 0,8. As duas colunas seguintes representam a mesma imagem mas com uso de um filtro gaussiano e de um filtro

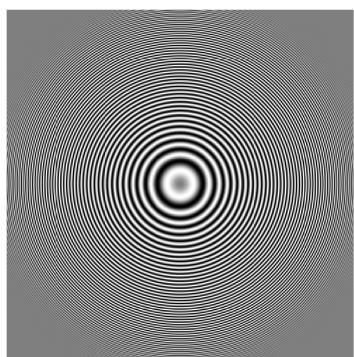
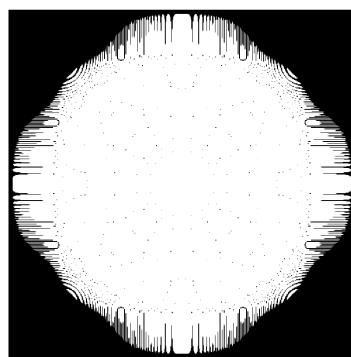


Imagen original



Dominio das frequencias

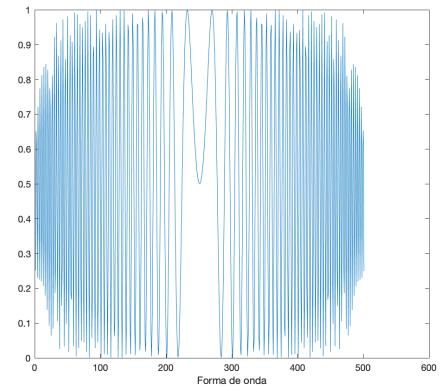


Figura 2.1: Estudo da imagem zone plate

médio.

O filtro gaussiano elimina informação nas extremidades da imagem, atenuando as altas frequências .

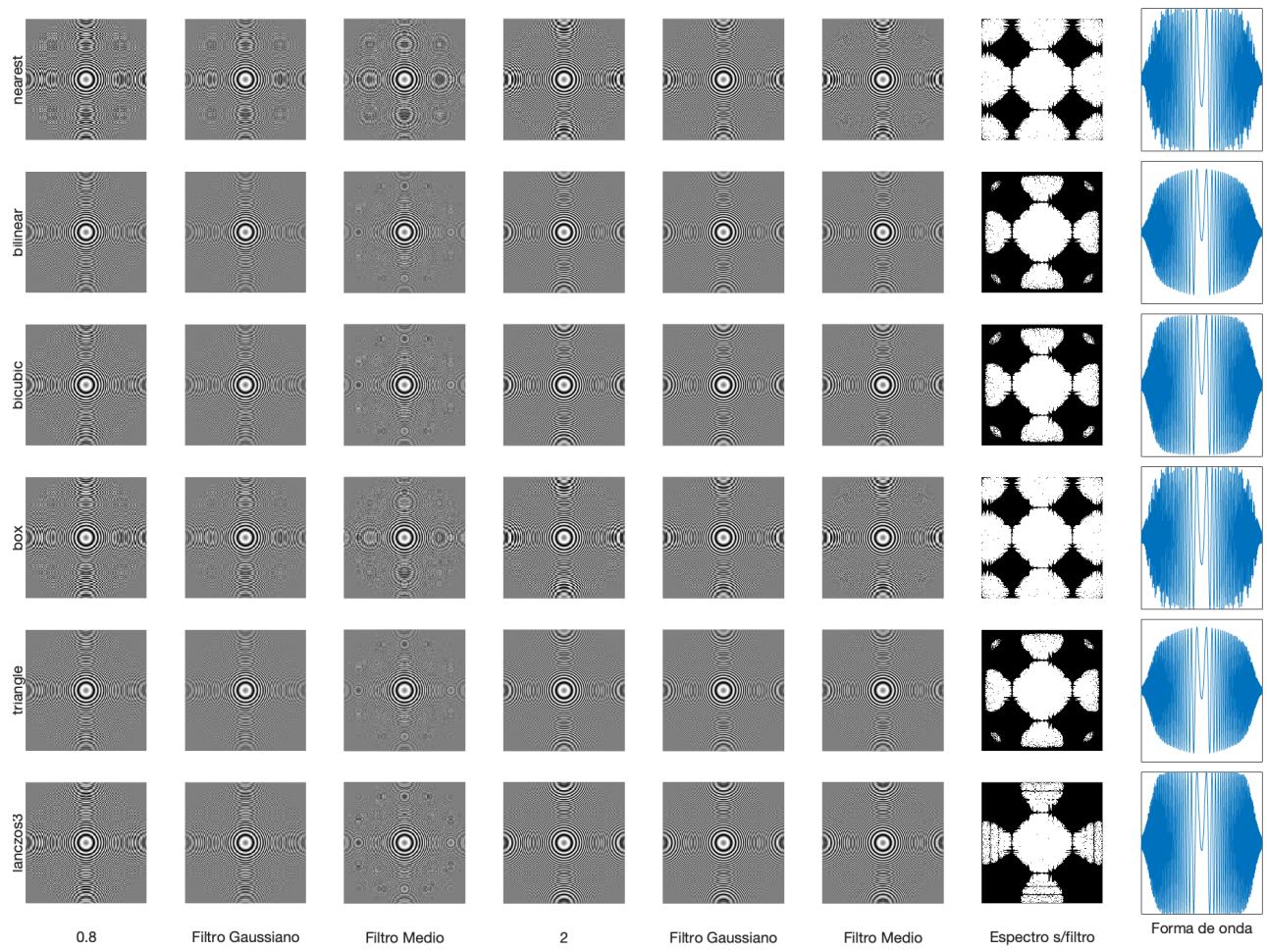
Por outro lado o filtro médio adiciona vários círculos concêntricos às imagens, o número destes artefactos varia conforme a escala desejada na redimensão. Quando ampliamos a imagem aparecem um número maior do que quando se reduz.

A partir da coluna 4 efetuamos uma redução da imagem pelo que conseguimos ver alguns efeitos desta operação. Na primeira linha conseguimos observar que os artefactos aumentaram, isto porque as técnicas de redução (zoom out) duplicam colunas e linhas. Conseguimos também reparar que utilizamos vários métodos / kernels de interpolação. O nearest determina o pixel de saída dependendo do valor do pixel que vai ficar.

Os métodos bilinear e bicubic dependem dos vizinhos, isto é o bilinear faz uma média do valor dos pixels numa região de 2 por 2, enquanto o bicubic usa os vizinhos numa região de 4 por 4. Para além de métodos também utilizamos matrizes kernel. O kernel box usa um kernel com a forma de uma caixa. Por outro lado o kernel triangle utiliza um kernel com a forma de um triângulo e é extremamente semelhante ao método bilinear. Por fim o kernel lanczos é construído com o recurso da função sinc o que vai reduzir o efeito de aliasing. Como observado no espetro da frequência.

Por último analisamos o espetro do aumento do tamanho x2, comparando com a Figura 2.1 reparamos que todos os métodos e kernels usados sofrem do efeito de Aliasing, no entanto há métodos piores do que outros. Os método nearest e o kernel box são os mais suscetíveis a aliasing.

Figura 2.2: Images com diferentes escalas e filtros



3. Experiências de filtragem

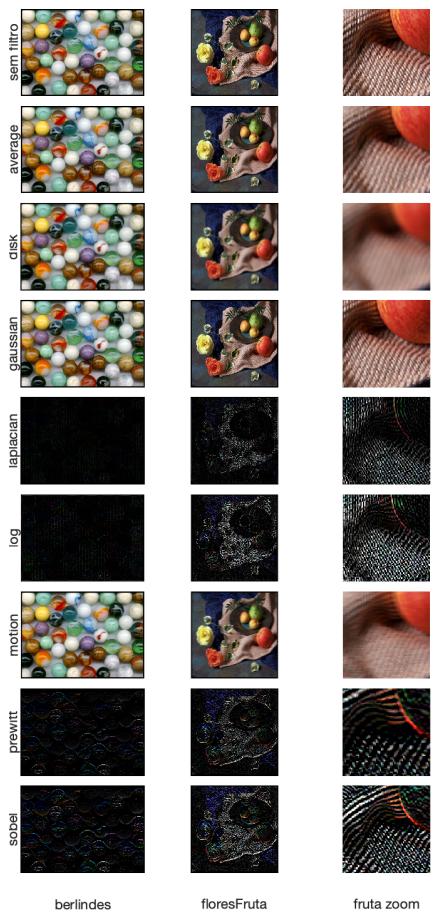
3.1 Utilização do fspecial e imfilter

Nesta experiência vamos estudar o efeito dos filtros nas imagens para isso vamos utilizar as funções `fspecial` e `imfilter`, para tal criamos o script `ex3_1.m` este cria a Figura 3.1, este script utiliza alguns dos tipos de filtro disponíveis. Na primeira linha observamos as imagens originais.

Na segunda linha contemplamos a utilização de um filtro médio, este filtro é muito usado para efetuar 'smoothing' às imagens. Este filtro reduz a variação de intensidade de píxeis vizinhos. Deste modo ao observar a coluna 3 (zoom da coluna 2) percebemos que a textura do pano é suavizada.

O filtro gaussiano foi usado na linha seguinte, este utiliza um filtro passa baixo filtrando e suavizando as imagens mas sem perder tanta informação como o average filter. O filtro gaussiano tem como base a percepção visual humana. Os filtros laplacian e laplacian of gaussian (LoG), apenas se diferenciam no facto de que o LoG é utilizado um filtro gaussiano a seguir a filtro laplaciano. Como os filtros seguintes são bastante usado

Figura 3.1: Vários Filtros



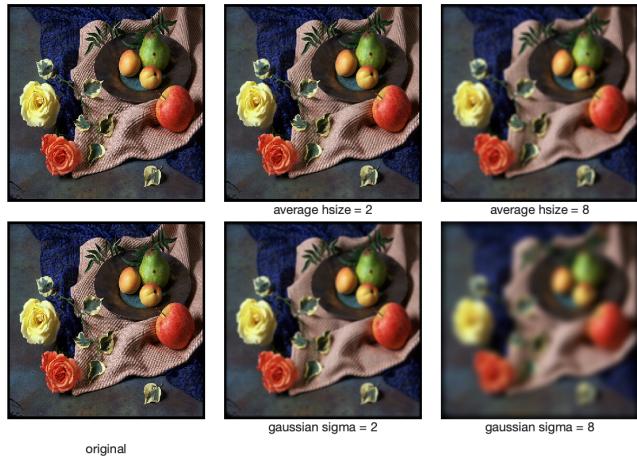


Figura 3.2: Filtros gaussian e average

em 'edge detection'. Nas últimas linha utilizamos o filtro de sobel e de prewitt ambos são muito usados em 'edge detection' como pode ser contemplado isto aconte pois estes filtros calculam o gradiente da intensidade da imagem.

Com o intuito de analisar as os filtros gaussiano e average obtivemos a Figura 3.2. Nesta podemos mais claramente observar aquilo dito no passo anterior. No filtro médio repara-se que claramente que o aumento da janela degrada a imagem tornando a desfocada. Como dito anteriormente este filtro tem em consideração os vizinhos, com o aumento da janela o número de vizinhos também aumenta pelo que o valor de cada ponto é calculado tendo em conta mais vizinhos por isso a imagem fica desfocada. No filtro gaussiano o aumento do desvio padrão desfocou a imagem isto deve-se ao facto de o filtro utilizar uma curva que gaussiana com o parâmetro sigma (desvio padrão). Como o sigma foi aumentado o a banda afetada pelo filtro também aumenta.

3.2 Convolução explícita

Vamos agora observar os efeitos dos filtros Média 3x3/5x5 e filtro mediana em duas imagens(Figura 3.3). Ambas as fotografias possuem ruído chamado 'salt pepper' e vamos identificar a resposta destes filtros a este ruído.

Na eliminação do ruído, entre os filtros média é esclarecedor que o kernel 5x5 é bastante melhor



Figura 3.3: Filtros gaussian e average

que o 3×3 . No entanto também há muito mais perda de informação. O filtro mediana é aquele que tem a melhor performance eliminando bastante ruído e mantendo muito detalhe.

O próximo passo é comparar o efeito dos operadores sobel e prewitt como mostrado na Figura 3.4. O resultado da utilização destes operadores são imagens a preto e branco compostas por linhas orientadas verticalmente ou horizontalmente em redor das zonas de contorno da imagem, sendo perceptíveis a maior parte dos detalhes da imagem original. Estas imagens resultantes dão enfase aos contornos das imagens originais. O resultado dos operadores é relativamente semelhante havendo diferenças na deteção de detalhes. Em imagens como a casa2 que os contornos da imagem tem orientação horizontal observamos que os operadores com zeros na horizontal tem um desempenho bastante melhor do que com zeros na vertical. Em comparação das imagens o operador sobel tem um desempenho melhor do que o operador prewitt.

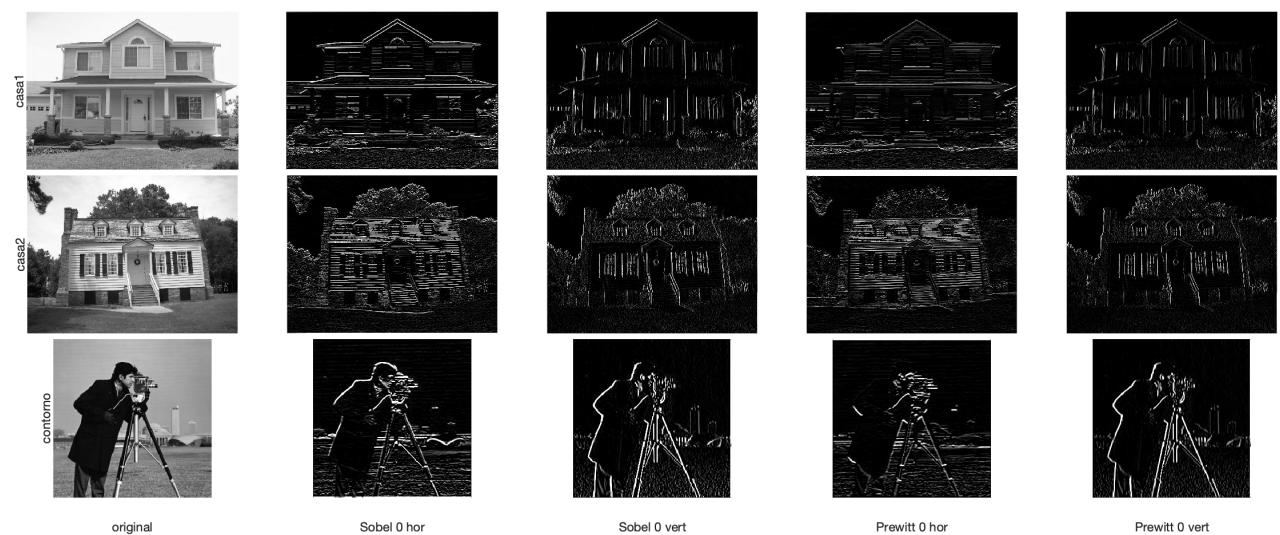


Figura 3.4: Filtros sobel e prewitt

3.3 Detetor de contornos

Para concluir o trabalho analizamos os contornos das imagem contorno.jpg com ajuda da função edge. Foram testados diferentes valores de “sigma” e de “threshold”. Começando pela análise das diferenças nas variações do último parâmetro “threshold”, foi possível verificar que à medida que se aumenta o valor do low threshold os contornos com gradientes de pixel mais baixos poderão ficar como contornos “fracos” ou mesmo ser removidos. O valor do high threshold determina quão elevado tem de ser o gradiente de pixel na zona para o contorno ficar assinalado com uma fronteira “forte”. Relativamente aos valores de “sigma”, é possível verificar que diferentes valores apresentam diferentes resultados finais. Estes valores afetam o filtro que é utilizado no detetor de contornos Canny. Pequenos valores de sigma realçam transições intensas, enquanto que grandes valores de sigma realçam transições graduais.

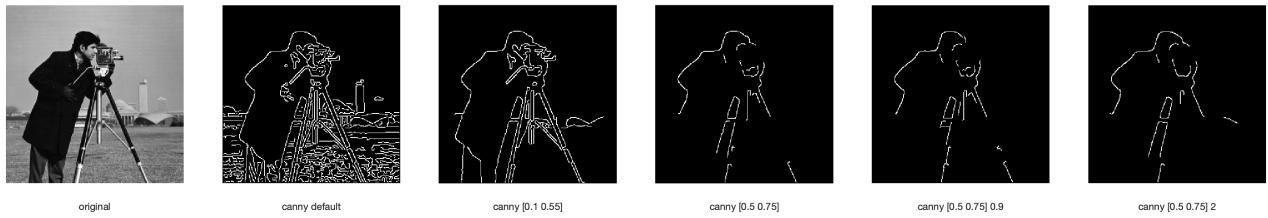


Figura 3.5: Vários tipos de contorno