

Técnicas de Realce de Imagens Digitais e suas Aplicações

Angélica Alves Viana*

Setembro de 2019

Resumo

Aplicações de realce em imagens vêm sendo cada vez mais requisitadas em diversas áreas com destaque para numerosas aplicações na área da medicina. Por esse motivo, muitas técnicas foram e vem sendo desenvolvidas com o objetivo de suprir essa demanda. Esse trabalho tem como objetivo apresentar, implementar, testar e analisar algumas dessas técnicas para compreender plenamente o seu funcionamento e verificar o comportamento das mesmas sob diferentes condições de aplicação.

Palavras-chaves: realce. *kernels*. técnicas.

Introdução

Uma imagem digital pode ser definida como uma função $f(x, y)$ em que cada ponto (x, y) de coordenadas espaciais da mesma apresenta um nível de cinza e essas coordenadas e intensidade de cores são finitas (Gonzalez, 2007).

Imagens Digitais são amplamente utilizadas em inúmeras aplicações desde o seu surgimento. Algumas delas são na obtenção de imagens médicas, meteorologia, restauração de imagens e imagens de satélite. No entanto, durante o processo de aquisição dessas imagens devido a fatores externos como condições de luminosidade ou qualidade do sensor de aquisição, as mesmas podem ser corrompidas por ruídos ou apresentarem uma baixa qualidade visual para a extração de alguma informação de interesse.

Devido a essa gama de aplicações e a necessidade de em muitos casos melhorar a qualidade dessas imagens, muitas técnicas de realce de imagens digitais vêm sendo desenvolvidas com essa finalidade, como técnicas de suavização de imagens para a remoção de ruídos, técnicas de aguçamento de imagens com o objetivo de realçar bordas, limiarização para destacar objetos de interesse e equalização de histogramas para aumentar o contraste e outras.

Diante da importância dessas técnicas, esse trabalho tem como objetivo realizar um estudo sobre algumas delas, destacando aspectos teóricos relacionados ao funcionamento e essas técnicas serão comparadas e analisadas a partir de testes realizados.

*angelicaalvesviana@gmail.com

1 Filtragem Espacial

A filtragem espacial é um processamento digital realizado diretamente na imagem utilizando *kernels*. Ela pode ser utilizada para realizar filtragens lineares ou não lineares.

De acordo com Aldalur e Santamaría (2002) os pixels adjacentes ao pixel central fornecem informação importante sobre a tendência de brilho na região. Os autores descrevem a filtragem envolvendo esses pixels como operações de processamento por grupo de pixels ou vizinhança.

O *kernel* consiste em uma vizinhança centrada sobre o pixel a ser modificado da imagem cuja operação a ser realizada sobre os vizinhos já está predefinida. Essa operação definirá um novo valor para o pixel da imagem cuja posição coincide com a posição central do *kernel*. A medida que o centro do *kernel* vai percorrendo os pixels da imagem - procedimento denominado convolução - uma nova imagem filtrada vai sendo gerada como resultado desse processamento. Se a operação utilizada na filtragem for linear, o *kernel* é denominado *kernel* linear, caso contrário, o *kernel* é não linear.

1.1 Filtros de Suavização

Os filtros de suavização também são conhecidos como filtros passa-baixas pois permitem a passagem de baixas frequências e são amplamente utilizados na remoção de ruídos que geralmente são introduzidos na conversão analógico-digital.

As técnicas de suavização podem ser categorizadas em conservativas e não conservativas. As conservativas são as que removem os ruídos mas mantêm os detalhes de alta frequência como as bordas, em geral, essas técnicas são boas na remoção de ruídos de *salt and pepper*, como exemplo temos o filtro da mediana. Já as não conservativas removem detalhes e curvas acentuadas, deixando a imagem visualmente borrada, essas são muito utilizadas na remoção de ruídos aditivos como o ruído Gaussiano, são exemplos dessas técnicas o filtro da média e o filtro gaussiano.

- Filtro da Média

O filtro da média é um *kernel* linear e segundo Sanches (2015) sua aplicação é responsável por reduzir a variação de intensidade entre um pixel e seus vizinhos, eliminando ruídos. A filtragem utilizando o *kernel* da média consiste em simplesmente substituir o pixel da imagem centrado no pixel central do *kernel* pela média de seus pixels vizinhos, incluindo ele próprio. O resultado visual desse procedimento de filtragem é uma imagem embaçada.

Em Gonzalez (2007) é definida a expressão que representa a implementação geral para a filtragem de uma imagem $M \times N$ com um filtro de média ponderada de tamanho $m \times n$ (m e n ímpares). A mesma é apresentada abaixo, através da equação 1.

$$g(x, y) = \frac{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)}{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t)} \quad (1)$$

Em que w representa o *kernel* aplicado, f a imagem original e g a imagem filtrada.

Para o filtro da média aritmética, Gupta (2011) apresenta uma equação reduzida da expressão mais geral em (1), dada por

$$g(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} f(s, t)$$

Em que S_{xy} representa um conjunto de coordenadas em uma janela retangular de tamanho $m \times n$ na imagem original centrada no ponto (x, y) . Esse filtro pode ser implementado com um *kernel* contendo todos os valores iguais a $\frac{1}{mn}$. A imagem abaixo exemplifica um *kernel* de tamanho 3×3 .

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

Tabela 1 – *kernel* 3×3 da média.

- Filtro Gaussiano: O filtro Gaussiano assim como o filtro da média é linear e é denominado assim, pois os valores do *kernel* são obtidos a partir da função Gaussiana, descrita em Sanches (2015) pela equação abaixo

$$w(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2)$$

Sendo portanto, o grau de suavização proporcionado por esse *kernel* associado ao parâmetro σ . Que corresponde ao desvio padrão da distribuição gerada. Quanto maior esse valor, mais suavizada será a imagem filtrada. Outro aspecto importante desse filtro é o fato de ele ser simétrico, isso garante que a suavização será de igual intensidade em ambas as direções da imagem.

Por apresentar características bem particulares e resultados satisfatórios na remoção de alguns tipos de ruídos, esse *kernel* é bastante utilizado em testes para escolha da técnica mais adequada de remoção do ruído com o qual se está trabalhando. Como no caso do trabalho de dissertação de Faria (2005), em que são realizados testes com os filtros da média, da mediana e o gaussiano na remoção de ruídos uniforme e *Salt and Pepper*. Nesse trabalho o autor obteve melhores resultados de aplicação do filtro gaussiano na remoção de ruído uniforme.

Por ser considerado um filtro de média ponderada, a equação que representa o processo de convolução desse *kernel* com a imagem também é representado pela equação geral (1). A imagem abaixo representa um *kernel* gaussiano de tamanho 3×3 retirado do livro do Gonzalez (2007).

1	2	1
2	4	1
1	2	1

Tabela 2 – *kernel* 3×3 gaussiano.

- Filtro da Mediana: O filtro da mediana é um filtro de estatística de ordem (não linear). A convolução com esse *kernel* substitui os pixels centrados na imagem pela

mediana dos pixels vizinhos incluindo a si próprio. Como exemplo, se for realizada uma convolução com um *kernel* de tamanho 3 x 3, o quinto elemento do vetor ordenado de elementos do *kernel* será a mediana que tomará a posição daquele pixel na imagem de saída. A equação que representa a implementação para a filtragem de uma imagem usando o *kernel* da mediana também está descrita em Gupta (2015)

$$g(x, y) = \text{median}_{(s,t) \in S_{xy}} f(x, y)$$

Além de ser útil na remoção do ruído *Salt and Pepper* como já mencionado, esse *kernel* fornece excelente performance na remoção de ruídos randômicos e uniformes, como pode ser visto também nos testes de Faria (2005). Resultados de aplicações do filtro da mediana para remoção de ruídos datam de muito antes, como pode ser visto no trabalho de Yong e Huang(1981), em que o mesmo é utilizado para a remoção de ruído de canal simétrico binário e esse resultado impacta diretamente no posterior processo de detecção de bordas de rampa.

1.2 Filtros de Aguçamento

Na seção anterior foram apresentadas algumas técnicas de suavização de imagens, que consistem em pré-processamentos aplicados a imagens digitais com o objetivo de torná-las mais adequadas para aplicações futuras. Nesta seção serão apresentadas outras técnicas de pré-processamento mas que têm como finalidade realçar os contornos de imagens digitais.

Essas técnicas são úteis em diversas aplicações, com destaque para inúmeras na área da medicina, como o trabalho de Silva (2008) em que são aplicadas técnicas de pré-processamento digital em imagens de rins direito de ultrassom, com o objetivo final de encontrar o contorno perfeito do órgão em questão, que auxiliaria no diagnóstico de patologias. Outro exemplo de aplicação também no campo médico é dado no trabalho de Sharma, Diwakar e Choudhary (2012), em que a técnica é utilizada juntamente a outras de pré-processamento para salientar as regiões de maiores intensidades em imagens de cérebros com tumores, com o objetivo posterior de identificar a localização exata do mesmos, bem como os seus tamanhos.

Em geral, essas técnicas de realce de bordas são *kernels* baseados no conceito de derivada primeira (Prewitt e Sobel) ou derivada segunda (Laplaciano). As subseções seguintes trarão maiores detalhes a respeito de cada uma dessas técnicas.

- Filtro Laplaciano: O filtro Laplaciano é um *kernel* baseado em derivadas bidimensionais de segunda ordem e é um dos mais simples com essa abordagem, pois sua implementação necessita apenas de uma máscara de tamanho 3 x 3 para detectar tanto as bordas verticais como as horizontais. O resultado da aplicação desse *kernel* consiste em uma imagem de fundo preto com detalhes de bordas da imagem original demarcados por linhas brancas. A equação que define a implementação desse *kernel* pode ser descrita pela equação abaixo retirada do livro do Gonzalez (2007)

$$\nabla^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y) \quad (3)$$

Uma desvantagem desse filtro é que ele é bastante sensível a ruídos e em geral antes de sua aplicação a imagem é submetida a uma filtragem de suavização para a remoção

0	1	0	1	1	1
1	-4	1	1	-8	1
0	1	0	1	1	1

Tabela 3 – *kernels* não-estendido e estendido.

de ruídos. As imagens abaixo representam o *kernel* laplaciano para implementar a equação (3) e um *kernel* estendido para detectar as linhas nas diagonais além das horizontais e verticais.

- Filtro Prewitt: O operador de Prewitt é um *kernel* de tamanho 3 x 3 utilizado para a detecção de bordas em uma imagem. apresentando duas máscaras, uma para a detecção de bordas horizontais e outra para as verticais que foram geradas com base em conceitos matemáticos de derivada primeira, conforme pode ser visto na fórmula abaixo que define esse operador

$$M(x, y) \approx |(z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)| + |(z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)| \quad (4)$$

Onde z_i correspondem as intensidades de cinza da imagem cobertos pela janela da máscara.

A Figura 3 mostram os operadores horizontais e verticais do Prewitt conforme pode ser visto em Silva (2008).

-1	-1	-1	-1	0	1
0	0	0	-1	0	1
1	1	1	-1	0	1

Tabela 4 – Operadores de Prewitt (Horizontal e Vertical).

De um modo geral, o procedimento de convolução com esse *kernel* resulta em somas ponderadas iguais a zero para vizinhanças com intensidade uniforme e valores diferentes de zero em regiões referentes as bordas da imagem.

- Filtro Sobel:

O filtro sobel é um *kernel* com embasamento matemático semelhante ao do Prewitt, utilizando da mesma maneira duas máscaras de tamanho 3 x 3, uma para a detecção de bordas horizontais e a outra para a detecção de bordas verticais. O que o diferencia do Prewitt é que os pixels da máscara que estão mais próximos do pixel central apresentam peso 2, pois considera-se que os mesmos são mais importantes que os demais quando se trata da detecção de bordas. O resultado dessa sutil diferença entre esses filtros são imagens com bordas mais bem demarcadas e com mais detalhes para o operador de Sobel.

A equação que representa a implementação desse operador é semelhante a equação (4) do *kernel* Prewitt e é dada por

$$M(x, y) \approx |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)| \quad (5)$$

A imagens abaixo representam os *kernels* horizontal e vertical do Sobel, respectivamente.

-1	-2	-1	-1	0	1
0	0	0	-2	0	2
1	2	1	-1	0	1

Tabela 5 – Operadores de Sobel (Horizontal e Vertical).

2 Equalização de Histogramas

Nas seções anteriores foram apresentadas algumas técnicas de realce de imagens incluindo filtragens para remoção de ruídos e filtragens para detecção de bordas para melhorar a nitidez de imagens digitais. Nessa seção será apresentado um outro método de realce, principalmente para imagens com escala de intensidade concentrada, baseado em um processamento de histograma da imagem.

Um histograma nada mais é do que uma função discreta $h(r_k) = n_k$ em que r_k é o k -ésimo valor de intensidade de uma imagem e n_k é a quantidade de pixels da imagem que apresentam esse valor de intensidade. Se uma imagem possui um histograma estreito significa que ela tem pouca variação em seus tons de cinza e esse fator pode dificultar a análise e extração de alguma informação da imagem. Com o objetivo de aumentar o contraste desse tipo de imagem, ou seja, esticar o histograma da mesma de modo que ela passe a ser composta por todos os tons de cinzas possíveis na faixa de $[0, 255]$ é realizado um processamento do histograma da mesma através da definição de uma transformação conhecida como Equalização de Histogramas.

Existem várias formas de se definir uma função de transformação para a equalização de uma imagem. De acordo com Gonzalez (2007) os únicos requisitos para a sua viabilidade são descritos em duas condições

- (a) Deve ser uma função estritamente monotonicamente crescente no intervalo $0 \leq r \leq 255$;
- (b) $0 \leq T(r) \leq 255$ para $0 \leq r \leq 255$.

Uma das maneiras de realizar essa equalização de histogramas e a que foi escolhida para ser implementada nesse trabalho foi uma equalização baseada em uma proporção realizada entre os valores mínimo (Min) e máximo (Max) de intensidade da imagem e o mínimo e máximo aceitável como tom de cinza $[0, 255]$. A equação utilizada para mapear os pixels da imagem através da equalização descrita acima é apresentada abaixo

$$g(x, y) = 255 \frac{f(x, y) - Min}{Max - Min} \quad (6)$$

Onde $g(x, y)$ corresponde a imagem equalizada e o resultado desse processo é uma imagem de maior contraste cubrindo todos os tons de cinza possíveis.

3 Limiarização e Multilimiarização

Em muitos casos deseja-se analisar características de objetos que apresentam um *background* uniforme, ou seja, o objeto de interesse apresenta variações de tons de cinzas diferentes do fundo da imagem. Se o objeto tiver uma cor mais clara e o fundo mais escura, o mesmo pode ser segmentado do fundo simplesmente aplicando-se um limiar T , tornando no histograma da imagem tudo que está abaixo desse limiar de uma intensidade e tudo que está acima de outra intensidade. Trata-se nesse caso de um processo de limiarização simples, que pode ser definido como

$$g(x, y) = \begin{cases} 1, & \text{se } f(x, y) > T \\ 0, & \text{se } f(x, y) \leq T \end{cases} \quad (7)$$

Onde $g(x, y)$ é a imagem limiarizada, resultando em uma imagem em preto e branco. Já se o interesse é em manter um dos grupos de pixels com seus valores originais e isolar o outro grupo de pixels, deve-se alternativamente utilizar a seguinte definição

$$g(x, y) = \begin{cases} 1, & \text{se } f(x, y) > T \\ f(x, y), & \text{se } f(x, y) \leq T \end{cases} \quad (8)$$

Outra aplicação da limiarização é para o procedimento de truncagem, em que deseja limitar o intervalo de tons de cinza para $[0, T]$, para isso uma outra definição de $g(x, y)$ é dada

$$g(x, y) = \begin{cases} T, & \text{se } f(x, y) > T \\ f(x, y), & \text{se } f(x, y) \leq T \end{cases} \quad (9)$$

Acima foram apresentadas algumas maneiras de aplicar a limiarização utilizando apenas um limiar T e é importante ressaltar que essas funções foram definidas conforme o funcionamento da função *threshold* do Opencv que serviu de embasamento para a implementação da função *threshold* desse trabalho, no entanto, em muitos casos um único limiar pode não ser útil para segmentar os objetos corretamente, como no caso de haver mais de um objeto em uma mesma imagem, nesse caso o ideal é trabalhar com a multilimiarização em que os limiares T_1, T_2, \dots, T_n são os limiares pré-definidos e R_1, R_2, \dots, R_{n+1} são as intensidades que serão aplicadas a cada intervalo, ou seja

$$g(x, y) = \begin{cases} R_{n+1}, & \text{se } f(x, y) > T_n \\ R_k, & \text{se } T_k > f(x, y) > T_{k+1}, \text{ onde } 1 < k < n - 1 \\ R_1, & \text{se } f(x, y) < T_1 \end{cases}$$

A função acima define a implementação de uma multilimiarização para uma quantidade de limiares generalizada. O resultado do processo de multilimiarização é uma imagem segmentada em mais de duas regiões com intensidades distintas da escala de cinzas.

4 Metodologia

Neste trabalho, todas as técnicas de realce vistas nas seções anteriores foram implementadas na linguagem Python. Foi feita a implementação de um filtro padrão 2D que recebe a imagem e um *kernel* e o resultado é uma imagem filtrada. Esse filtro padrão

foi utilizado tanto para as aplicações dos *kernels* de suavização como os de aguçamento. Além disso, foram implementadas funções para calcular os histogramas das imagens e apresentá-las bem como a técnica de realce de equalização de histogramas utilizando a função matemática descrita em (4)

Foram realizados alguns testes para cada uma dessas técnicas apresentadas com o objetivo de analisar o comportamento dos mesmos sob diferentes condições de aplicação.

Os primeiros testes realizados foram feitos com os filtros de suavização. Para os *kernels* da média, da mediana e o gaussiano, foram aplicados consecutivamente os *kernels* de tamanho 3x3, 5x5, 7x7 e 9x9 e os resultados foram analisados. Em seguida, foi realizado um segundo teste no qual o *kernels* 3x3 da média, da mediana e do gaussiano foram aplicados 2, 3, 4 e 5 vezes.

Já os filtros de aguçamento foram aplicados para a detecção de bordas e comparados entre si, em especial com o Laplaciano foram utilizadas tanto a máscara para as detecções de bordas horizontais e verticais como a versão estendida para a detecção de bordas nas diagonais também.

Por fim, foram utilizadas algumas imagens com diferentes contrastes no processo de equalização de histogramas e esses resultados novamente foram analisados. Além disso, foram feitos testes com diferentes valores de *thresholds* para a limiarização e multilimiarização de algumas imagens digitais.

5 Análise e Interpretação dos Dados

Os resultados dos testes realizados com as técnicas implementadas: filtragem, equalização de histogramas e limiarização são apresentados a seguir.

As imagens abaixo representam os resultados da aplicação de *kernels* da média quadrados de tamanho $m = 3, 5, 9, 15$ e 23 em uma imagem de 200×300 pixels.

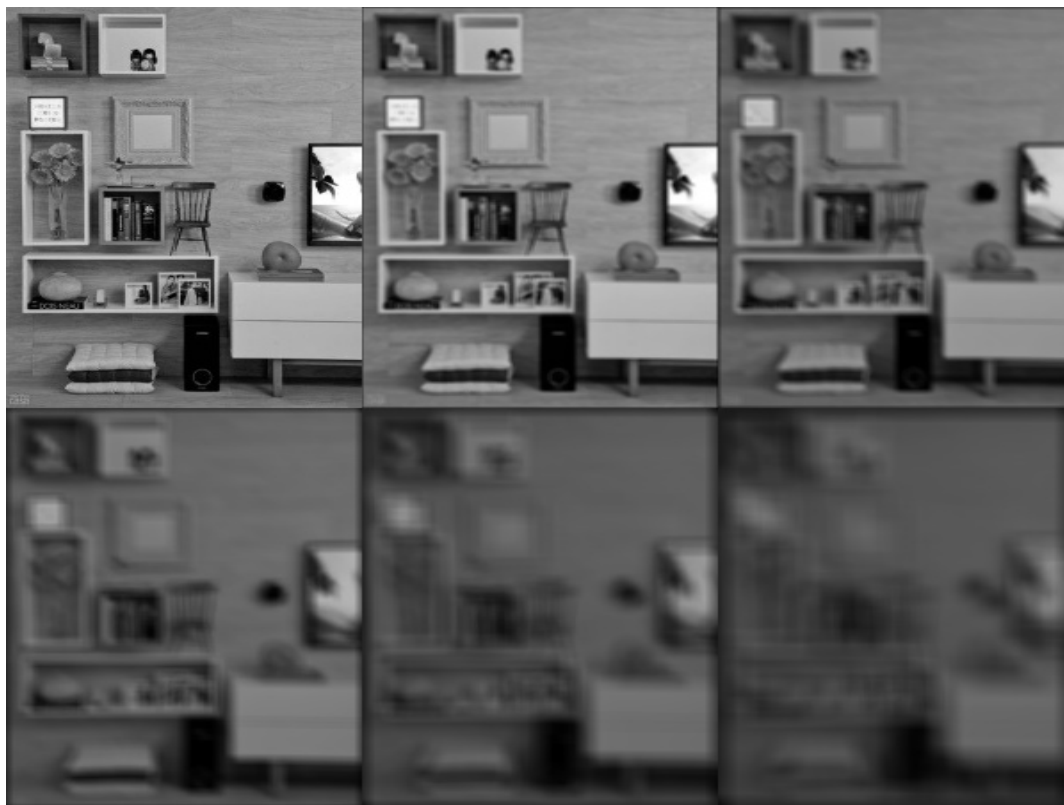


Figura 1 – Imagem original à esquerda e resultados de suavização com o *kernel* da média de tamanhos 3, 5, 9, 15 e 23.

É possível observar que da imagem original para a filtragem realizada com o *kernel* de tamanho 3x3 houve uma pequena suavização da mesma, já com o *kernel* 5x5 a imagem já perdeu completamente o brilho e os contornos dos objetos estão ainda mais suaves, além disso o objeto que estava sobre a estante de livros já quase não aparece e some completamente com a filtragem com o *kernel* de tamanho 9x9. Já com o *kernel* de tamanho 15x15 o objeto que estava sobre a mesinha abaixo da televisão perdeu completamente seus detalhes, resultando em uma mancha acinzentada. E por fim, com um *kernel* de tamanho 23x23 todos os objetos da imagem perderam a sua forma inicial, a imagem se encontra bem mais escura do que antes do processo de filtragem, devido a uniformização dos tons de cinza e redução de contraste provocado pela aplicação da técnica e uma borda preta pode ser vista devido ao preenchimento inicial das bordas da imagem com 0's.

O desaparecimento de alguns objetos com a aplicação de um *kernel* da média ocorre sempre que o objeto da imagem é menor do que o tamanho do

Um segundo teste realizado com o *kernel* da média foi a aplicação do *kernel* de tamanho 3 dez vezes seguidas nessa mesma imagem e depois vinte vezes na imagem original. Os resultados são apresentados abaixo

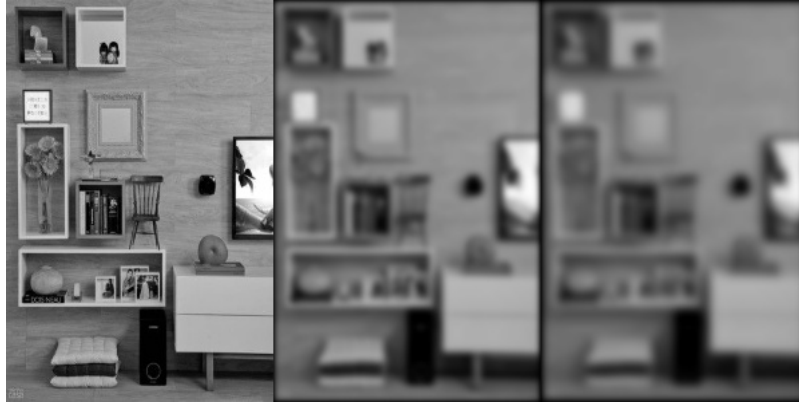


Figura 2 – Imagem Original; Aplicação de *kernel* 3x3 dez vezes; Aplicação de *kernel* 3x3 vinte vezes. (*kernel da média*)

Observa-se que um *kernel* de tamanho 3 aplicado várias vezes produz como resultado uma imagem bastante suavizada semelhante a aplicação de um único *kernel* de tamanho maior, no entanto por ser uma máscara pequena, quase todos os objetos da imagem se mantêm, mas com muito menos detalhes que a imagem original e as bordas pretas na imagem ficam ainda mais evidentes.

Já com o *kernel* da mediana, os mesmos testes aplicados ao *kernel* da média foram repetidos. As imagens abaixo representam a aplicação dos *kernels* de tamanho $m = 3, 5, 9, 15$ e 23 da mediana.

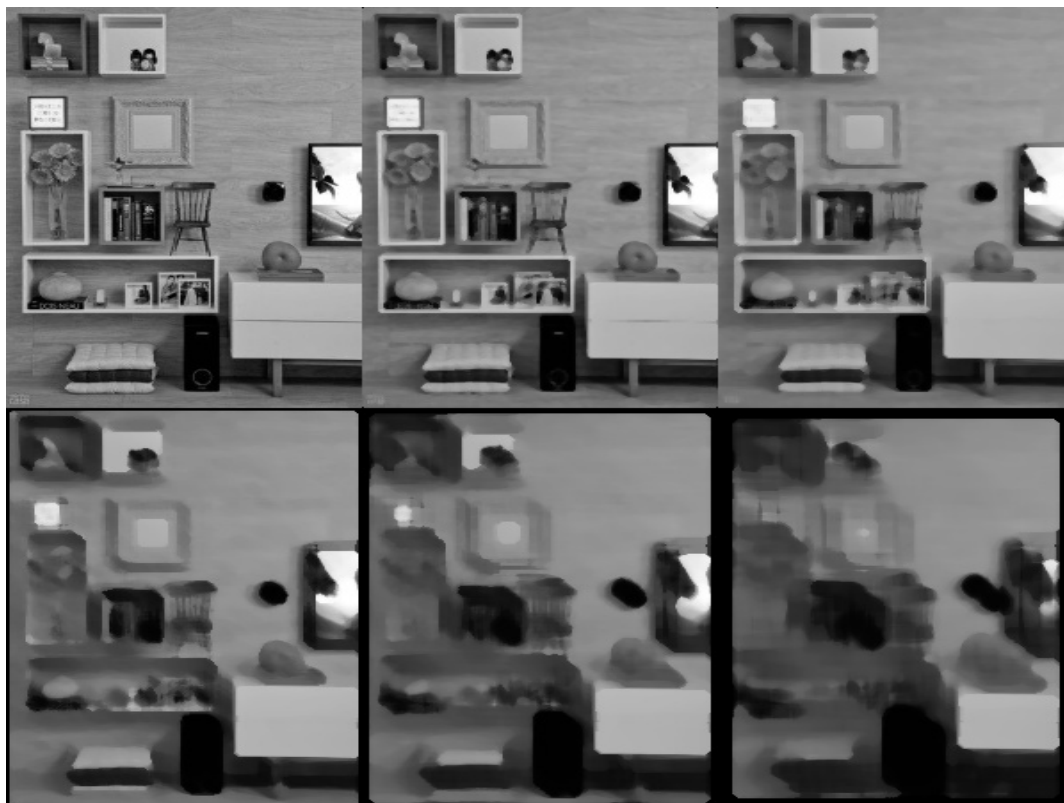


Figura 3 – Imagem original à esquerda e resultados de suavização com o *kernel* da mediana de tamanhos 3, 5, 9, 15 e 23.

Pode-se observar que a aplicação do *kernel* da mediana torna as pontas dos objetos retangulares da imagem mais arredondadas e esse efeito é leve com um *kernel* de tamanho 3x3. Já com um *kernel* de tamanho 5x5 o objeto que se encontra ao lado da TV está completamente circular e os detalhes da caixa de som ao lado da almofada desapareceram, bem como o objeto de cima da prateleira de livros. Como constatado a partir de estudos como os de Faria (2005) e pelos resultados obtidos acima, dependendo da aplicação, o *kernel* da mediana pode ser melhor do que o da média, pois suaviza preservando as bordas.

O segundo teste aplicado ao *kernel* da media foi reproduzido agora com o *kernel* da mediana, na qual o mesmo foi aplicado 10 vezes consecutivas a imagem original e em seguida 20 vezes. O resultado é apresentado abaixo



Figura 4 – Imagem Original; Aplicação de *kernel* 3x3 dez vezes; Aplicação de *kernel* 3x3 vinte vezes. (*kernel da mediana*)

Esses resultados indicam que existe uma certa estabilidade após consecutivas aplicações do *kernel* da mediana, de modo que os resultados da aplicação do *kernel* de tamanho 3x3 dez vezes é igual ao resultado de sua aplicação vinte vezes. E mesmo após sucessivas aplicações, o filtro ainda mantém a maior parte da informação da imagem original, sendo possível identificar a forma geral da maioria dos objetos nela presentes.

Novamente o mesmo esquema de aplicações para os filtros da média e da mediana foi repetido para o filtro da gaussiana e o resultado das aplicações desse *kernel* com tamanhos $m = 3, 5, 9, 15$ e 23 podem ser vistos nas imagens abaixo.



Figura 5 – Imagem original à esquerda e resultados de suavização com o *kernel* gaussiano de tamanhos 3, 5, 9, 15 e 23.

É importante ressaltar que as máscaras utilizadas para essa filtragem foram construídas a partir da função da Gaussiana definida na seção 1.2 e que o σ utilizado foi $\sigma = 3$. A partir de uma breve análise é possível ver que o *kernel* gaussiano proporciona uma suavização mais regularizada, independente da região, pois como foi visto esse *kernel* é simétrico e o grau de suavização do mesmo depende unicamente do parâmetro σ .

Serão apresentados agora os testes que foram realizados com as técnicas de realce para detecção de bordas. Primeiramente, foram aplicados os *kernels* laplacianos para detecção de bordas nas horizontais e verticais e a versão extendida para a detecção de bordas nas diagonais respectivamente e ambos foram comparados entre si. Os *kernels* aplicados aqui foram apresentados na seção 1.2 e podem ser vistos na figura 6. Abaixo tem-se os resultados dessas aplicações

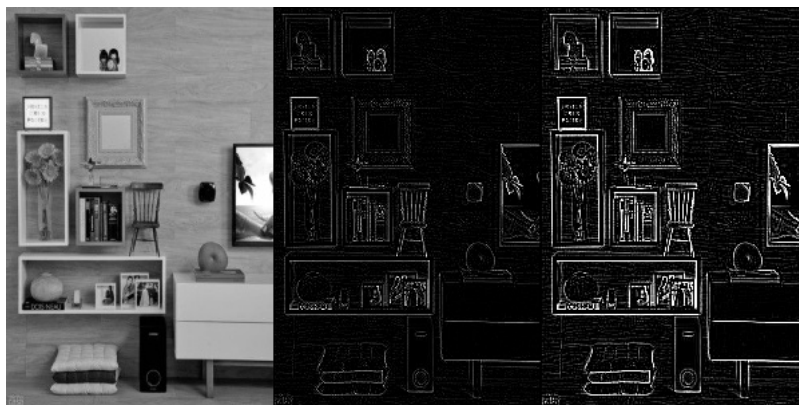


Figura 6 – Imagem Original; Resultados *kernels* não-estendido e estendido respectivamente.

A filtragem com o *kernel* estendido trás uma imagem com bordas muito mais acentuadas que a versão para a detecção de bordas nas horizontais e verticais, tendo também muito mais detalhes. Além disso na versão não estendida objetos com bordas arredondadas tendem a ter uma detecção com linhas bem suaves. Outro aspecto importante a se observar é que apesar de ter bordas muito mais bem demarcadas, a versão estendida do laplaciano também é mais sensível ao ruído e isso pode ser visto através das linhas nas horizontais no fundo da imagem.

Continuando a análise dos filtros de aguçamento, o segundo teste consistiu na aplicação do *kernel* do Prewitt, cuja teoria é explorada com detalhes na seção 1.2, para a detecção de bordas horizontais e verticais.



Figura 7 – Imagem Original; Aplicação *kernel* Prewitt.

A imagem acima demonstra que o *kernel* Prewitt é muito mais eficiente que o *kernel* Laplaciano, pois as bordas dos objetos na imagem realmente são mais bem demarcadas e apesar de o mesmo resultar em uma imagem ainda com linhas nas horizontais por efeito de ruídos, elas são bastante suaves, quase imperceptíveis.

Para finalizar a análise dos filtros de aguçamento, por fim foi aplicado o *kernel* Sobel na mesma imagem e o resultado é apresentado abaixo



Figura 8 – Imagem Original; Aplicação *kernel* Sobel.

A propriedade do operador de Sobel de ponderar os pixels centrais com peso dois faz com que o resultado de sua aplicação sejam imagens com bordas mais grossas e com maiores detalhes que o *kernel* Prewitt.

Agora, para finalizar os testes com os filtros de realce serão utilizadas algumas técnicas baseadas em histogramas. Algumas imagens apresentam uma concentração de tons de cina de modo que os objetos contidos nela são difíceis de serem visualizados ou analisados. Para solucionar esse problema, uma técnica bastante utilizada é a equalização de histogramas que foi apresentada na seção 2. Abaixo, tem-se uma imagem originalmente escura e ao lado a sua versão equalizada de acordo com a equação (6).

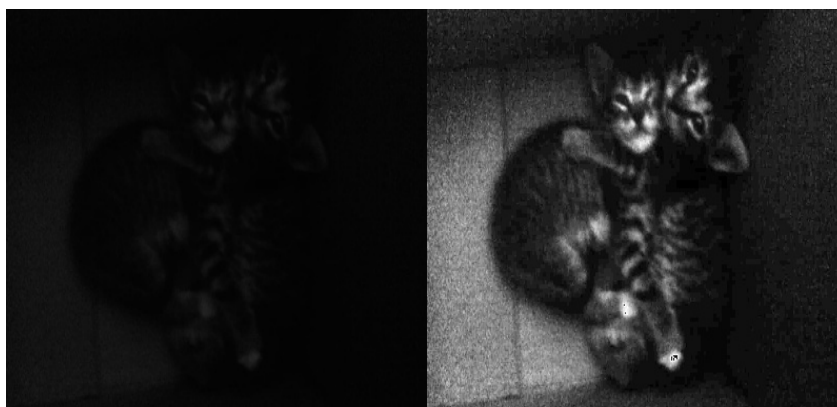


Figura 9 – Imagem Original; Imagem Equalizada.

As imagens a seguir representam os histogramas das imagens original e equalizada respectivamente

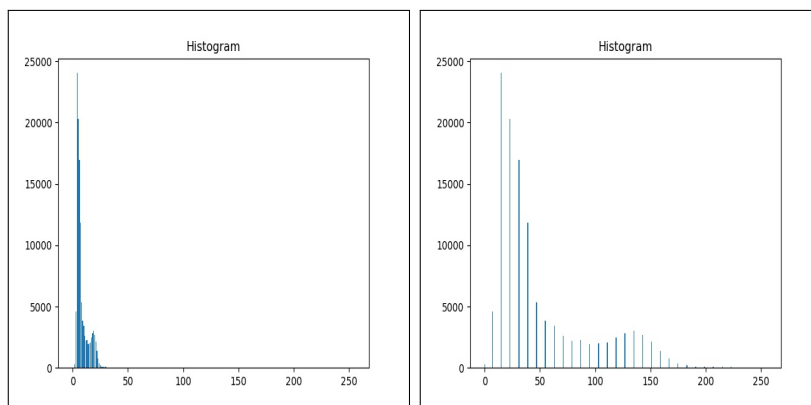


Figura 10 – Histogramas das imagens original e equalizada respectivamente.

Os histogramas das imagens apresentados acima indicam que a imagem original tinha uma alta concentração de pixels escuros, com um histograma bastante estreito e que a equalização do histograma fez com que o mesmo se tornasse mais espalhado, abrangendo uma maior quantidade de tons de cinza.

É importante ressaltar que para a implementação da equalização de histogramas de acordo com a equação dada na seção 2, para a obtenção dos valores máximos e mínimos foi considerado um valor mínimo de ocorrência dos mesmos padronizado como sendo no mínimo 10 ocorrências. Essa metodologia adotada, evita que os resultados não sejam tão bons quanto o esperado por se considerar pixels pouco frequentes.

Outro tipo de aplicação em que a análise de histograma é importante e nesse caso consiste em uma etapa primordial antes de submeter a técnica final é na limiarização, que como explorada na seção 3 pode ser aplicada considerando-se apenas um limiar T , nesse caso trata-se de uma limiarização simples que pode ter outras extensões além da forma binária (7), como a forma saturada vista na equação (8) ou a forma truncada vista na equação (9). A imagem abaixo mostra a imagem original seguida de sua limiarização binária, saturada e truncada.

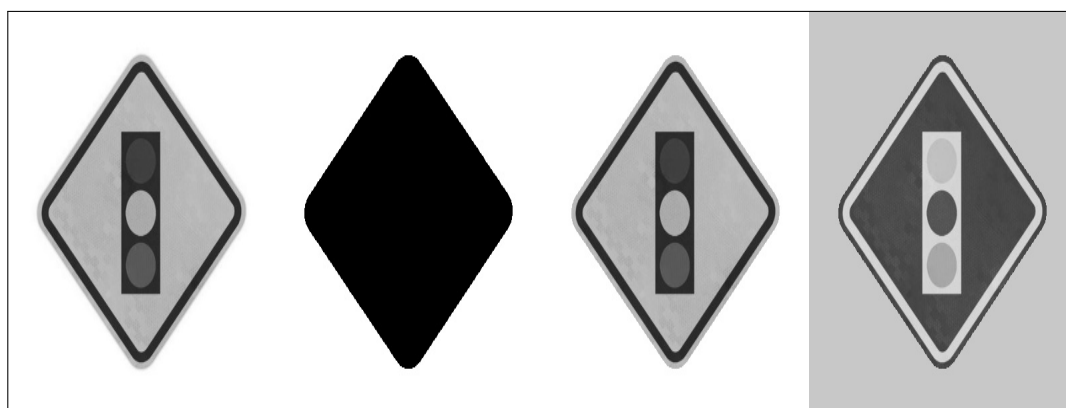


Figura 11 – Imagem Original; Resultados de Limiarização Binária, Saturada e Truncada respectivamente

O limiar T escolhido para as segmentações acima foram baseados na análise do

histograma, conforme pode ser visto no histograma da imagem apresentado abaixo o limiar $T = 200$ é uma boa escolha para segmentar a placa de seu fundo.

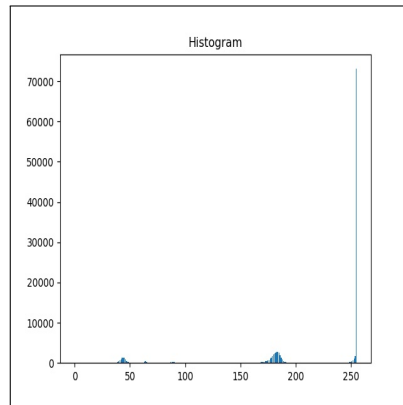


Figura 12 – Histograma da imagem original da placa.

Observando com cuidado o histograma dessa imagem, é possível observar que o mesmo apresenta um pico antes do 50, outro pico entre o 100 e o 200 e outro no 255. Nesse caso, é possível utilizar a multilimiarização para separar os objetos por agrupamento de tons de cinza. A imagem abaixo representa a multilimiarização aplicada sobre a imagem da placa com dois limiares $T_1 = 100$ e $T_2 = 200$ e com intensidade de cores aplicadas nos intervalos de 0, 127 e 255 respectivamente.

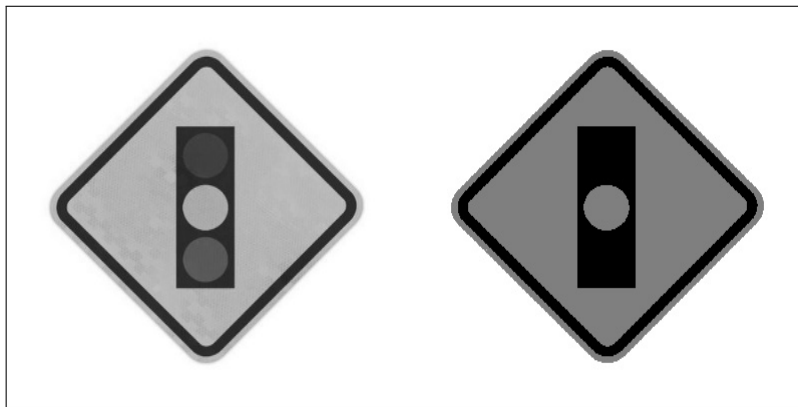


Figura 13 – Imagem Original; Resultado da multilimiarização.

6 Conclusão

Após a análise dos resultados foi possível concluir que as técnicas de realce apresentadas e testadas nesse trabalho variam conforme as condições de aplicação, dependendo não apenas da finalidade para qual se deseja utilizá-las como de fatores inerentes a própria imagem, como luminosidade, intensidade dos pixels, se apresentam algum tipo de ruído ou não e muitos outros aspectos. No entanto, sob as condições adequadas de aplicação de cada uma delas, pode-se obter resultados muito bons ou até excelentes. Esse estudo, portanto, demonstrou ser de fundamental importância para a compreensão plena do funcionamento de cada uma dessas técnicas e para aprender em quais circunstâncias se deve utilizá-las,

pois em inúmeras áreas estão sendo cada vez mais requisitadas aplicações de realce em imagens, não somente para melhorar o aspecto visual das mesmas, como também para reduzir a quantidade de informação para facilitar na extração do que realmente interessa.

7 Referências

ALDALUR. B.; SANTAMARIA. M., Realce de Imágenes: filtrado espacial, Revista de teledetección, vol. XVII, pp. 31-42, 2002.

FARIA, D. R.. Reconhecimento de impressões digitais com baixo custo computacional para um sistema de controle de acesso. Dissertação (Mestrado), Universidade Federal do Paraná, 2005.

GONZALEZ, R. C.; WOODS, R. E. 3rd. Edition. Digital Image Processing, Prentice Hall, 2007. 976p.

GUPTA, G.. Algorithm for image processing using improved median filter and comparison of mean, median and improved median filter. Int J Soft Comput Eng (IJSCE) 1(5):304–311, 2011.

SHARMA, P.; DIWAKAR, M.; CHOUDHARY, S.. Application of Edge Detection for Brain Tumor Detection, International Journal of Computer Applications, vol.58, no.16, pp 21-25, 2012.

SANCHES, Carlos H. et al. Técnicas de Suavização de Imagens e Eliminação de Ruídos. Anais do EATI-Encontro Anual de Tecnologia da Informação, p. 21-30. 2015.

SILVA, F.; ALVES, F.. Aplicação de Técnicas de Processamento de Imagens Digitais em Imagens Geradas por Ultra-Som, em 8º Encontro Regional de Matemática Aplicada e Computacional, Universidade Federal do Rio Grande do Norte, 2008.

YONG, G. J.; HUANG, T. S.. The effect of median filtering in edge location estimation, Computer Vision Graphics and Image Processing, vol. 15, pp. 224-245, 1981.