

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

**OTIMIZAÇÃO DO ALGORITMO NON LOCAL MEANS
MEDIANTE AGRUPAMENTO POR SIMILARIDADE NO
DOMÍNIO DA FREQUÊNCIA**

DOUGLAS ANDRADE DE MENESES

JOÃO PESSOA-PB
AGOSTO-2017

DOUGLAS ANDRADE DE MENESES

**OTIMIZAÇÃO DO ALGORITMO NON LOCAL MEANS
MEDIANTE AGRUPAMENTO POR SIMILARIDADE NO
DOMÍNIO DA FREQUÊNCIA**

DISSERTAÇÃO APRESENTADA AO CENTRO DE INFORMÁTICA DA
UNIVERSIDADE FEDERAL DA PARAÍBA, COMO REQUISITO PARCIAL
PARA OBTENÇÃO DO TÍTULO DE MESTRE EM INFORMÁTICA (SISTEMAS
DE COMPUTAÇÃO).

Orientador: Prof. Dr. Leonardo Vidal Batista

JOÃO PESSOA-PB
Agosto-2017

Catálogo na publicação
Setor de Catalogação e Classificação

M543o Meneses, Douglas Andrade de.
Otimização do algoritmo Non local Means mediante agrupamento por similaridade no domínio da frequência / Douglas Andrade de Meneses. – João Pessoa, 2017.
53 f. : il.

Orientador: Leonardo Vidal Batista.
Dissertação (Mestrado) – UFPB/CI

1. Informática. 2. Remoção de ruído. 3. Non local means. 4. K-Means. 5. Transformada discreta do cosseno. I. Título.

UFPB/BC

CDU - 004(043)



UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA



Ata da Sessão Pública de Defesa de Dissertação de Mestrado de Douglas Andrade de Meneses, candidato ao título de Mestre em Informática na Área de Sistemas de Computação, realizada em 31 de agosto de 2017.

1 Aos trinta e um dias do mês de agosto, do ano de dois mil e dezessete, às quatorze horas,
2 no Centro de Informática da Universidade Federal da Paraíba, em Mangabeira, reuniram-se
3 os membros da Banca Examinadora constituída para julgar o Trabalho Final do Sr. Douglas
4 Andrade de Meneses, vinculado a esta Universidade sob a matrícula nº 2015104041,
5 candidato ao grau de Mestre em Informática, na área de "Sistemas de Computação", na linha
6 de pesquisa "Sinais, sistemas digitais e gráficos", do Programa de Pós-Graduação em
7 Informática, da Universidade Federal da Paraíba. A comissão examinadora foi composta
8 pelos professores: Leonardo Vidal Batista (PPGI-UFPB), Orientador e Presidente da Banca,
9 Thais Gaudêncio do Rego (PPGI-UFPB), Examinador interno, José Antonio Gomes de Lima
10 (UFPB), Examinador externo ao programa, e Suzete Élide Nóbrega Correia (IFPB),
11 Examinador externo à instituição. Dando início aos trabalhos, o Presidente da Banca,
12 cumprimentou os presentes, comunicou aos mesmos a finalidade da reunião e passou a
13 palavra ao candidato para que o mesmo fizesse a exposição oral do trabalho de dissertação
14 intitulado "Otimização do algoritmo Non Local Means mediante agrupamento por similaridade
15 no domínio da frequência". Concluída a exposição, o candidato foi arguido pela Banca
16 Examinadora que emitiu o seguinte parecer: "aprovado". Do ocorrido, eu, Claurton de
17 Albuquerque Siebra, Coordenador do Programa de Pós-Graduação em Informática, lavrei a
18 presente ata que vai assinada por mim e pelos membros da banca examinadora. João
19 Pessoa, 31 de agosto de 2017.


Prof. Dr. Claurton de Albuquerque Siebra

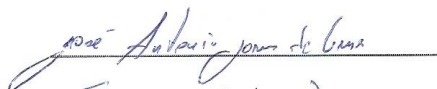
Prof. Dr. Leonardo Vidal Batista
Orientador (PPGI-UFPB)



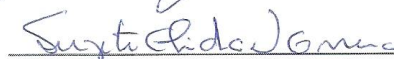
Prof. Dr. Thais Gaudêncio do Rego
Examinador interno (PPGI-UFPB)



Prof. Dr. José Antonio Gomes De Lima
Examinador externo ao programa (UFPB)



Prof. Dr. Suzete Élide Nóbrega Correia
Examinador externo à instituição (IFPB)



Agradecimentos

Ao Divino Amigo, que vive em mim, por ser o Tudo em meu nada.

À minha família, por ser sempre meu porto seguro.

À minha namorada e amigos, por me amarem pelo que sou e sempre me apoiarem nas minhas decisões.

Ao meu orientador, o professor Dr. Leonardo Vidal Batista, por toda paciência, confiança e aprendizado durante todo este tempo.

“Enquanto houver vontade de lutar haverá esperança de vencer.”

Santo Agostinho de Hipona

RESUMO

Dentre os algoritmos de remoção de ruído de imagens no estado da arte, o Non Local Means (NLM) ganha destaque devido a sua eficiência. No entanto, sua complexidade computacional torna-se um obstáculo para a maioria das aplicações atuais. Neste trabalho propõe-se uma nova abordagem que visa reduzir o tempo de processamento, sem recorrer a janelas de busca, preservando assim os cálculos não locais que caracterizam o NLM. O método proposto usa o algoritmo de agrupamento K-means para agrupar pixels com vizinhanças semelhantes no domínio da frequência através da Transformada discreta do cosseno. Para evitar transições severas nas bordas dos agrupamentos, um pixel pode ser atribuído a diferentes clusters. Após esse passo inicial, o algoritmo NLM executa uma pesquisa baseada em cluster. Os resultados experimentais testificam uma redução em torno de 19 vezes no tempo computacional e, em alguns casos, melhorias nos valores de Erro Quadrático Médio, quando comparados ao algoritmo original

Palavras-chave: Remoção de ruído, Non local means, K-Means, Transformada discreta do cosseno.

ABSTRACT

Among the image denoising methods in state-of-the-art the Non Local Means (NLM) is highlighted due to its efficiency. However, its computational complexity becomes an obstacle to most applications. In this paper, we propose a new approach which aims to reduce processing time, without resorting to search windows, thus preserving the non-local calculations that characterize NLM. The proposed method uses K-means clustering to group pixels with similar neighborhood in the frequency domain by Discrete Cosine Transform. In order to avoid harsh transitions at cluster borders, one pixel can be assigned to different clusters. After this initial step, the NLM algorithm performs a cluster-based search. Experimental results testify a reduction in the computational time around 19 times and, in some cases, improvements in the Mean Squared Error values, when compared to the original algorithm.

Keywords: Denoising, Non local means, K-Means, Discrete cosine transform

LISTA DE ILUSTRAÇÕES

Figura 1 - Imagem limpa acrescida de ruído gaussiano resultando na terceira imagem	21
Figura 2 - Relação do valor do MSE com o nível de ruído ou semelhança com a imagem original. a) Imagem original, b) MSE = 19.40, c) MSE = 293.32, d) MSE = 749.19.....	22
Figura 3 - Imagens resultantes do ruído de método com alguns algoritmos de filtragem. Da esquerda para direita e de cima para baixo: imagem original, filtro Gaussiano, filtro Anisotrópico, minimização da Variação Total, filtragem por vizinhanças e NLM	23
Figura 4 - Análise de pixels e vizinhanças processadas para a filtragem do pixel i	24
Figura 5 - Kernel Gaussiano bidimensional utilizado para definir o peso de cada pixel em uma vizinhança.....	25
Figura 6 - Resultado da filtragem com variação do fator de decaimento h. No topo à esquerda está a imagem original; abaixo dela, o resultado da filtragem com $h^2 = 100$; no topo à direita com $h^2 = 200$; e abaixo, $h^2 = 2000$	26
Figura 7 - Iterações do algoritmo K-Médias com $K = 2$	28
Figura 8 - Resultado visual da Transformada Cosseno Discreta de uma imagem real	30
Figura 9 - Diagrama do funcionamento do método desenvolvido	32
Figura 10 - Representação de agrupamentos com problemas nas fronteiras.....	34
Figura 11 - Representação de agrupamentos com pixels participando de mais de um <i>cluster</i>	35
Figura 12 - Coeficientes da janela DCT 5x5	39

Figura 13 - Ruído de método do NLM original (à esquerda) e do método proposto (à direita) para imagem Lena com nível de ruído 0,008..... 47

Figura 14 - Resultado visual da filtragem da Lena. Acima, a imagem ruidosa; à esquerda o resultado do NLM original e à direita, do método proposto..... 48

LISTA DE EQUAÇÕES

Equação 1 - Ruído em uma imagem digital	20
Equação 2 – MSE	21
Equação 3 – NLM	24
Equação 4 - Peso de um pixel em relação a outro no NLM	24
Equação 5 - Fator de normalização	25
Equação 6 - DCT em imagens	29

LISTA DE GRÁFICOS

Gráfico 1 - Quantidade de coeficientes DCT x Tempo (em minutos).....	38
Gráfico 2 - Quantidade de coeficientes DCT x MSE.....	38
Gráfico 3 - Comparação dos valores do MSE obtidos entre o NLM original e o algoritmo proposto para as três imagens utilizadas com variância de ruído 0,002.....	45
Gráfico 4 - Comparação dos valores do MSE obtidos entre o NLM original e o algoritmo proposto para as três imagens utilizadas com variância de ruído 0,008.....	45

LISTA DE TABELAS

Tabela 1 - MSE e tempo de execução obtidos através da variação da tolerância para Lena com variância do ruído Gaussiano 0,008.....	40
Tabela 2 - MSE e tempo de execução obtidos através da variação da tolerância para Lena com variância do ruído Gaussiano 0,020	40
Tabela 3 - Valores de MSE e tempo de execução para imagem <i>Boats</i> com variância de ruído 0,002, alternando-se a quantidade de rodadas do <i>K-Médias</i>	42
Tabela 4 - Valores do MSE e tempo de execução para imagem Barbara com variância de ruído 0,002 alternando-se a quantidade de clusters.....	43
Tabela 5 - Valores do MSE e tempo de execução para a imagem Barbara com variância de ruído 0,008 alternando-se a quantidade de clusters	43
Tabela 6 - Comparação dos tempos de processamento (em minutos) do NLM Original e o algoritmo proposto para imagens com variância de ruído 0,002 e 0,008.....	46

LISTA DE SIGLAS

DCT Discrete Cosine Transform (Transformada Cosseno Discreta)

NLM Non-local Means (Médias não-locais)

MSE Mean squared error (Erro quadrático médio)

SUMÁRIO

1. INTRODUÇÃO	16
1.1 CONTEXTUALIZAÇÃO OU DEFINIÇÃO DO PROBLEMA.....	17
1.2 OBJETIVOS	18
1.3 ESTRUTURA DA DISSERTAÇÃO.....	19
2. FUNDAMENTAÇÃO TEÓRICA.....	20
2.1 IMAGENS DIGITAIS E RUÍDO	20
2.2 ALGORITMO NLM.....	23
2.3 MÉTODO DE AGRUPAMENTO <i>K-MÉDIAS</i>	27
2.4 TRANSFORMADA DISCRETA DO COSSENO	29
3. MATERIAIS E MÉTODOS	31
3.1 MÉTODO DESENVOLVIDO	31
3.2 CONVERTENDO OS PIXELS DA IMAGEM PARA O DOMÍNIO DA FREQUÊNCIA	32
3.3 AGRUPANDO OS PIXELS SEMELHANTES EM <i>CLUSTERS</i>	33
3.4 FILTRAGEM ATRAVÉS DO NLM	35
3.5 METODOLOGIA DA AVALIAÇÃO EXPERIMENTAL.....	36
4. RESULTADOS E DISCUSSÕES	37
4.1 QUANTIDADE DE COEFICIENTES DA JANELA DCT	37
4.2 TOLERÂNCIA PARA RESOLUÇÃO DE FRONTEIRAS NO <i>K-MÉDIAS</i>	39
4.3 QUANTIDADE DE RODADAS DO <i>K-MÉDIAS</i>	41
4.4 QUANTIDADE DE <i>CLUSTERS</i> IDEAL.....	43
4.5 COMPARAÇÃO COM O NLM ORIGINAL.....	44
5. CONSIDERAÇÕES FINAIS.....	49
REFERÊNCIAS	51

1 INTRODUÇÃO

Apesar do crescente avanço tecnológico nos processos de aquisição e processamento de imagens digitais, o ruído continua sendo um obstáculo a ser superado. Suas fontes são diversas: provenientes de circuitos eletrônicos; ruído de sensor causado por temperatura ou iluminação fraca; fotodiodos mortos; flutuações quânticas estatísticas; ruído de quantização, etc. Além disso, o ruído pode ser amplificado por correções digitais das câmeras ou por softwares de processamento de imagens – ao se acrescentar contraste ou remover regiões desfocadas (*blur*), por exemplo, o ruído torna-se ainda mais realçado (BUADES; COLL; MOREL, 2011).

Como solução prática, diversos algoritmos foram criados na tentativa de, eliminando o ruído indesejado, recuperar a imagem original. Tais algoritmos receberam a denominação de *denoising methods*, ou métodos de remoção de ruído. Muitos foram os métodos desenvolvidos dentro da grande área de Processamento Digital de Imagens na busca por, removendo as perturbações indesejadas, obter imagens filtradas limpas - essenciais para outras aplicações como compressão de dados, reconhecimento de padrões, análise de imagens médicas, entre outras. Esses mesmos métodos buscam preservar ao máximo as bordas e outras características essenciais da imagem enquanto eliminam as degradações existentes (JAIN; TYAGI, 2014).

Durante muito tempo, as abordagens criadas para implementar esses métodos envolviam procedimentos locais, isto é: para filtrar um determinado pixel (abreviação de *picture element*) da imagem, utilizava-se a informação da sua vizinhança, ou seja, os pixels vizinhos a ele. Entre as mais famosas, destacam-se a filtragem gaussiana (LINDENBAUM; FISCHER; BRUCKSTEIN, 1994), anisotrópica (PERONA; MALIK, 1990) e bilateral (TOMASI; MANDUCHI, 1998). Porém, a limitação à vizinhanças não permite que se explore a redundância de informação ao longo de toda a imagem, que é uma característica muito recorrente, por exemplo, nos cantos ou bordas da imagem.

Compreendendo que existem padrões que se repetem por toda imagem, como pixels que compõem um céu ou mar, no artigo de Buades, Coll e Morel (2004) notou-se a necessidade de criar um método não-local que filtrasse cada pixel utilizando a informação contida nos demais ao longo de toda a imagem, atribuindo pesos de acordo com o grau de semelhança entre eles. Desse modo, propôs-se o algoritmo *Non Local Means* (NLM), que foi

o precursor, segundo Salmon e Strozeck (2010), de tantos outros algoritmos não-locais como pode-se encontrar em Kevrann e Boulanger (2006), Dabov, Katkovnik e Egiazarian (2007) e Mairal *et. Al* (2009).

1.1 CONTEXTUALIZAÇÃO OU DEFINIÇÃO DO PROBLEMA

Embora apresente resultados muito relevantes, tanto visualmente quanto matematicamente, em comparação com os demais métodos tradicionais de remoção de ruído, o NLM possui uma complexidade computacional bastante elevada, uma vez que precisa processar todos os pixels da imagem para a filtragem de cada pixel alvo. Por essa razão, torna-se inapropriado para aplicações tecnológicas atuais que necessitam de resultados velozes ou quase instantâneos. Sabendo disso, no artigo de Buades, Coll e Morel (2005), os autores sugerem uma adaptação ao método, limitando o processo de filtragem a uma janela de busca relativamente grande (21 x 21), na tentativa de reduzir a complexidade do algoritmo original.

Seguindo os mesmos passos, muitos pesquisadores passaram a investigar e desenvolver diversas outras otimizações para o NLM, visando reduzir ainda mais a sua complexidade computacional. Entre elas, destacam-se algumas que foram apresentadas utilizando, por exemplo: limitação de vizinhanças semelhantes para o cálculo da média (MAHMOUDI; SAPIRO, 2005), árvores de *clusters* (BROX; KLEINSCHMIDT; CREMERS, 2008), média e variância baseando-se em blocos (COUPÉ et al., 2010), agrupamentos com K-Médias (PETER; RAMYA, 2012), dicionário de imagens (BHUJLE; CHAUDHURI, 2012), Transformada discreta do Cosseno (DCT) (HU et al., 2012), entre tantas outras.

As diversas melhorias encontradas na literatura conseguiram resultados significantes, a ponto de reduzir o tempo computacional e obter resultados visuais semelhantes, ou até melhores, que o método original. Porém, todas essas abordagens baseiam-se na adaptação citada anteriormente, na qual utiliza-se uma janela de busca e possui complexidade inferior. Desse modo, o algoritmo original continua conhecido apenas na teoria, uma vez que não se encontrou no estado da arte algum aprimoramento visando a redução de sua alta complexidade computacional.

Contudo, no artigo mais recente de Buades, Coll e Morel (2011), dois pontos relevantes foram levantados: o uso da janela de busca reduz a ‘não-localidade’ do algoritmo, sendo mais coerente a utilização do termo ‘semi-local’ – isso implica na redução da eficiência do algoritmo ao não mais explorar a redundância de informação ao longo de toda a imagem; e a confirmação de que, quanto mais intenso o ruído, maior deve ser a janela de busca, ou seja, para imagens muito ruidosas a filtragem com o NLM original (que realiza a busca em toda a imagem) deve ser mais eficiente. Isso foi confirmado na dissertação de mestrado de Figueiredo (2014), onde se prova que, para imagens com ruídos mais intensos, o NLM original obtém melhores resultados visuais e matemáticos, porém com tempo de execução muito mais elevado.

Em virtude disso, uma proposta de otimização para o NLM original é apresentada neste trabalho, a qual contribui para a declinação da complexidade computacional e, conseqüentemente, do tempo de processamento, possibilitando a utilização do algoritmo para os diversos fins atuais, assim como ocorre com a adaptação do algoritmo utilizando janela de busca.

O método utiliza o algoritmo de agrupamento K-Médias (do inglês *K-Means*) e a Transformada discreta do Cosseno (DCT), aplicando-se a imagens em escala de cinza, como o fez Buades, Coll e Morel (2005).

1.2 OBJETIVOS

O objetivo geral do presente trabalho é pesquisar e desenvolver uma otimização para o algoritmo NLM, proposto no artigo de Buades, Coll e Morel (2004), utilizando o método de agrupamento K-Médias e o domínio da frequência.

Os objetivos específicos são:

1. Reduzir a complexidade computacional do algoritmo NLM, limitando sua ação para cada agrupamento, que contém os pixels mais semelhantes entre si.
2. Reduzir o tempo de execução do algoritmo, mantendo, ou até melhorando, a qualidade matemática e visual das imagens filtradas resultantes.
3. Realizar testes comparativos entre o método proposto e o algoritmo original

1.3 ESTRUTURA DA DISSERTAÇÃO

Este trabalho está organizado em cinco capítulos. No Capítulo 2, são apresentados conceitos sobre remoção de ruído, NLM e outros algoritmos e heurísticas utilizadas na implementação do método proposto. Em seguida, no Capítulo 3, detalha-se a metodologia e desenvolvimento do método, o qual tem seus resultados apresentados e discutidos no capítulo subsequente.

Por fim, no Capítulo 5, expressa-se algumas considerações finais e conclusões acerca do trabalho, bem como propostas para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os principais conceitos teóricos empregados na realização deste trabalho. Primeiramente, são apresentados conceitos acerca de imagens digitais e ruído. Logo após, definições a respeito do algoritmo NLM são apresentadas. Por fim, noções acerca do método de agrupamento K-Médias e da DCT são dissertadas.

2.1 IMAGENS DIGITAIS E RUÍDO

A obtenção de imagens para o uso em um computador só é possível através de um processo chamado digitalização, composto de duas etapas - amostragem e quantização - que as transformam em sinais digitais (BATISTA, 2005). Desse modo, uma imagem monocromática pode ser entendida como um sinal $f(x,y)$ ou uma matriz $R \times C$, onde cada elemento indica um pixel e suas características: x e y , a sua posição espacial; e o valor de $f(x,y)$, o seu brilho ou nível de cinza.

Como citado anteriormente, o ruído é algo presente no processo de obtenção e digitalização das imagens, gerando alterações nos valores dos seus pixels. Assim, no artigo de Buades, Coll e Morel (2005), define-se:

$$v(i) = u(i) + n(i) \quad (1)$$

em que $v(i)$ é o valor observado do pixel, $u(i)$ é o valor "verdadeiro" e $n(i)$ é a perturbação do ruído sobre ele. Portanto, o valor que se observa em um pixel é, na verdade, a soma do seu valor original mais o ruído ali instaurado.

Em um pensamento mais abrangente, os valores visuais e matemáticos observados em uma imagem são os valores dos seus pixels somados à degradação que percorre toda a imagem. Em razão disso, os diversos métodos de remoção de ruído buscam, prioritariamente, remover toda a degradação a fim de que se tenha apenas os valores originais dos pixels, isto é, apenas a imagem "original".

Como ilustração, a Figura 1 apresenta uma imagem original limpa (à esquerda), sem ruído - representando os valores $u(i)$ da Equação 1, acrescida de ruído gaussiano (segunda

imagem) que representa os valores $n(i)$, resultando na terceira imagem, que são os valores observados $v(i)$.

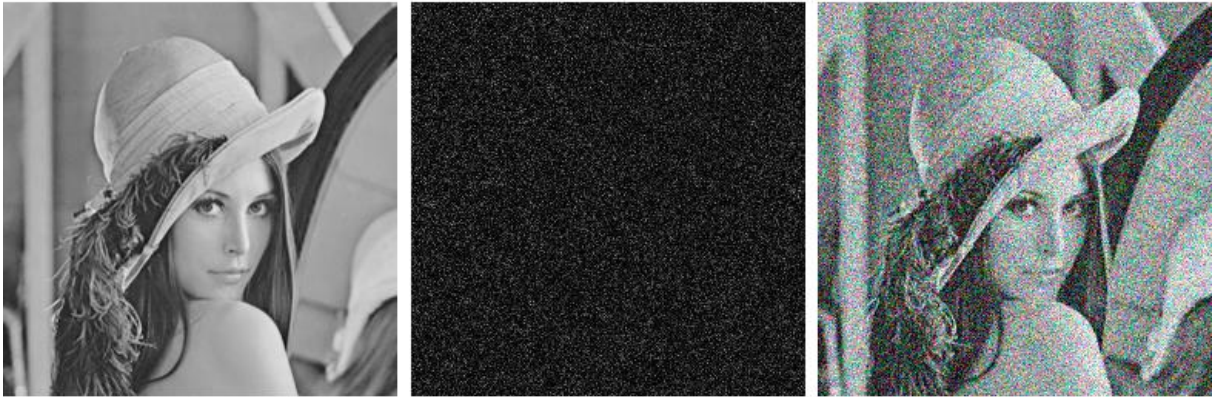


Figura 1. Imagem limpa acrescida de ruído gaussiano resultando na terceira imagem.

Bons algoritmos de remoção de ruído devem obter uma imagem filtrada mais próxima da imagem original quanto for possível e, para isso, é necessário retirar o ruído evitando a perda de características relevantes como as bordas, texturas e formas importantes. Para se avaliar quão boa foi determinada filtragem, no artigo de Buades, Coll e Morel (2004) são sugeridos dois métodos: o erro quadrático médio (ou MSE, sigla para *Mean squared error*) e o ruído de método (do inglês *Method noise*), além, obviamente, da avaliação visual do resultado da filtragem em comparação com a imagem original.

O erro quadrático médio é um método matemático que avalia a diferença entre um estimador e o verdadeiro valor da quantidade estimada. Na área de processamento de imagens, consiste em encontrar a distância Euclidiana entre a imagem filtrada e a original (BUADES; COLL; MOREL, 2005). Isso se dá através do cálculo pixel a pixel em um somatório, dividido pelo número de pixels da imagem, conforme a Equação 2:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2 \quad (2)$$

em que N é a quantidade de pixels das imagens e x_i e y_i são os valores dos pixels das imagens.

O valor do MSE de uma imagem filtrada demonstra o grau de semelhança que ela possui em relação à original de modo que, quanto menor for esse valor, mais restaurada está a

imagem - ou menos ruidosa. Para clarificar esse comportamento, a Figura 2 expressa a correlação entre o MSE e o nível de ruído ou semelhança com a imagem original.

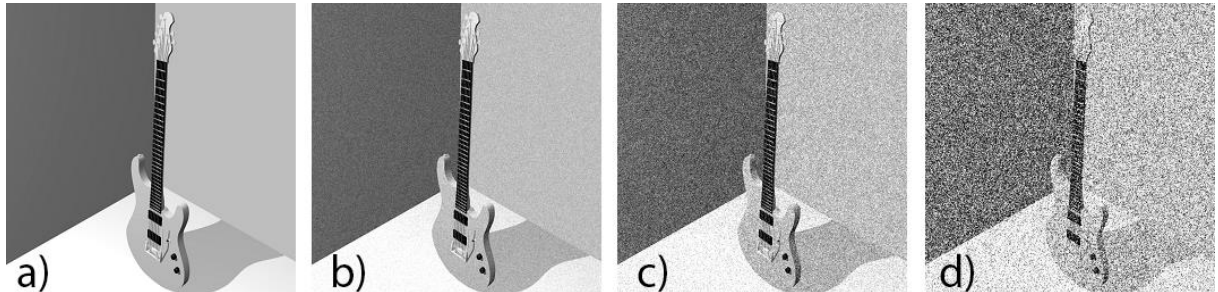


Figura 2. Relação do valor do MSE com o nível de ruído ou semelhança com a imagem original. a) Imagem original, b) MSE = 19,40, c) MSE = 293,32, d) MSE = 749,19

O ruído de método, por sua vez, possui uma aplicação mais visual e pode ser entendido como a diferença absoluta entre a imagem ruidosa e sua versão filtrada. A imagem resultante desse processo apresenta elementos que indicam a eficiência da filtragem submetida: quanto mais elementos da imagem (como bordas, traços e texturas) são vistos, pior é o algoritmo de redução de ruído, pois indica que, além do ruído em si, características importantes da imagem foram removidas.

Seguindo o raciocínio, quanto mais semelhante a um ruído branco for um resultado, melhor é o algoritmo utilizado, como observa-se na Figura 3, retirada do artigo de Buades, Coll e Morel (2005). No exemplo em questão, enquanto o resultado do filtro Gaussiano apresenta características da imagem, o NLM apresenta apenas o ruído branco, afirmando sua eficiência na filtragem ante outros algoritmos.

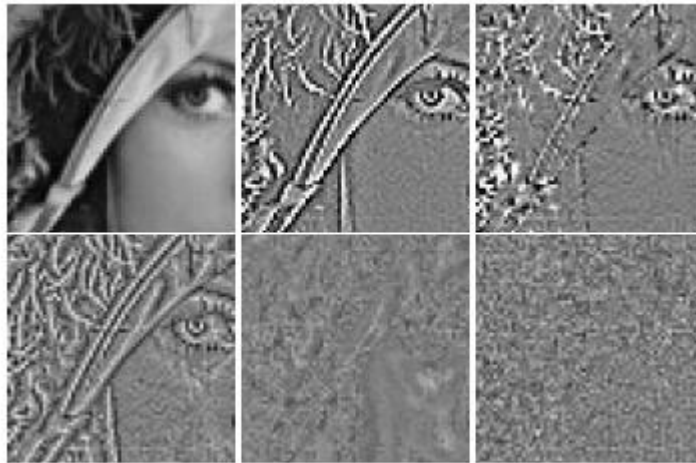


Figura 3. Imagens resultantes do ruído de método com alguns algoritmos de filtragem. Da esquerda para direita e de cima para baixo: imagem original, filtro Gaussiano, filtro Anisotrópico, minimização da Variação Total, filtragem por vizinhanças e NLM.

2.2 ALGORITMO NLM

O NLM é um algoritmo de remoção de ruído não-local - isto é, que utiliza a informação global da imagem para filtrar cada pixel individualmente - que foi desenvolvido e demonstrado no artigo de Buades, Coll e Morel (2004), na tentativa de, diferente dos demais métodos, explorar padrões ou características que se repetem ao longo de toda imagem. Desde então, encontra-se no estado da arte na área de remoção de ruídos e diversos pesquisadores tem estudado e aprimorado seu funcionamento, como citado no capítulo anterior.

No seu processamento, ao filtrar um determinado pixel, atribui pesos aos demais pixels da imagem de acordo com o grau de semelhança entre as suas vizinhanças. Desse modo, como se pode ver na Figura 4, os pixels J_2 e J_3 estão bem próximos do pixel alvo i , mas não possuem o mesmo peso, uma vez que a vizinhança de J_3 é bem diferente da vizinhança de i e, portanto, tem um peso irrelevante. Por conseguinte, o pixel J_1 está bem distante de i , mas possui um alto peso por possuir uma vizinhança bem semelhante.

Conclui-se, então, que pixels distantes podem ser extremamente importantes no processo de filtragem e que as características da imagem precisam ser exploradas nesse

mesmo processo - na figura em questão, todos os elementos encontrados na fronteira do braço da guitarra com a parede cinza-claro terão vizinhanças semelhantes à do pixel i em análise

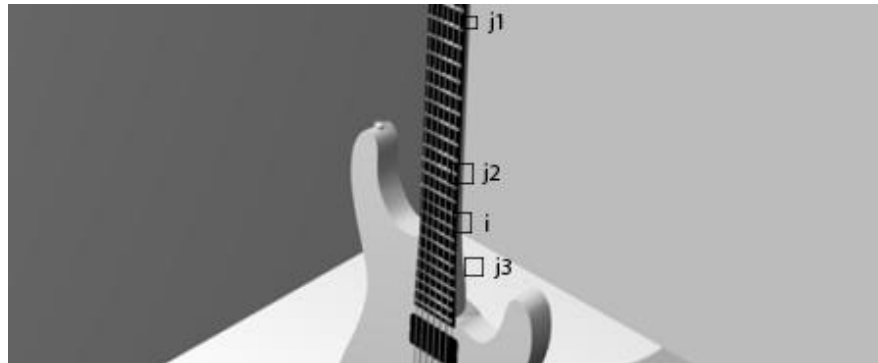


Figura 4. Análise de pixels e vizinhanças processadas para a filtragem do pixel i .

Para uma imagem ruidosa $v = \{v(i) \mid i \in I\}$, a equação que define o funcionamento do algoritmo sobre um pixel i pode ser descrita como:

$$NL(v)(i) = \sum_{j \in I} w(i, j)v(j) \quad (3)$$

onde $NL(v)(i)$ é o valor a ser encontrado da filtragem do pixel i ; $w(i, j)$ é o peso do pixel j em relação a i ; e $v(j)$ o valor de j . Faz-se, portanto, uma média ponderada de todos os pixels da imagem com relação ao pixel alvo para obter-se o valor final de sua filtragem.

Para realizar o cálculo do peso de um pixel com relação a outro, a fórmula abaixo é utilizada:

$$w(i, j) = \frac{1}{Z(i)} e^{(-1) \frac{\|v(N_i) - v(N_j)\|_{2,a}^2}{h^2}} \quad (4)$$

em que $w(i, j)$ é o peso do pixel j em relação a i ; $Z(i)$ é o fator de normalização; h é o fator de decaimento; e $\|v(N_i) - v(N_j)\|_{2,a}^2$ a Distância Euclidiana Quadrática Ponderada, que determina o peso de cada vizinhança por vetores de intensidade de nível de cinza, de acordo

com um *kernel* Gaussiano bidimensional pré-definido (Figura 5). Pode-se notar que os pixels mais próximos do pixel principal têm maior peso (FIGUEIREDO, 2014).

$$\frac{1}{273}$$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

Figura 5. Kernel Gaussiano bidimensional utilizado para definir o peso de cada pixel em uma vizinhança.

O fator de normalização pode ser definido como sendo:

$$Z(i) = \sum_j e^{(-1) \frac{\|v(Ni) - v(Nj)\|_{2,a}^2}{h^2}} \quad (5)$$

e o fator de decaimento h como sendo o responsável por afirmar os pesos em função da distância Euclidiana, isto é, determinar o grau da filtragem. Exemplificando: para baixos valores deste fator, a distância a ser processada será muito curta, e pouco ruído será removido; mas para altos valores, a distância será longa e detalhes importantes da imagem podem ser removidos juntamente com todo o ruído. Para melhor compreensão, a Figura 6 demonstra o resultado da variação do valor de h para uma mesma imagem - neste caso o valor ótimo para h^2 seria 200, uma vez que removeu bem o ruído sem perder as características da imagem.



Figura 6. Resultado da filtragem com variação do fator de decaimento h . No topo à esquerda está a imagem original; abaixo dela, o resultado da filtragem com $h^2 = 100$; no topo à direita com $h^2 = 200$; e abaixo, $h^2 = 2000$.

Todos esses cálculos matemáticos necessários para se filtrar um único pixel já mostram que a complexidade do algoritmo é bastante elevada, ainda mais porque esse processo será repetido para cada pixel da imagem. Assim, a complexidade do NLM pode ser determinada como $O(N^2u)$, tendo N como o número de pixels da imagem e u como o tamanho da janela de vizinhança.

Em virtude disso, no artigo de Buades, Coll e Morel (2005), os autores propõem uma adaptação ao método utilizando uma janela de busca ao invés de realizar a pesquisa por

vizinhanças semelhantes em toda a imagem. Utilizando tal abordagem, a complexidade reduz-se para $O(Nsu)$, em que s é o tamanho da janela de busca (FIGUEIREDO, 2014).

2.3 MÉTODO DE AGRUPAMENTO K-MÉDIAS

Proposto no trabalho de Macqueen (1967), o algoritmo de agrupamento *K-means* é largamente utilizado na área de mineração de dados devido a sua fácil implementação e complexidade linear $O(n)$, apesar de produzir agrupamentos de qualidade inferior a outros métodos. Consiste em um método não-supervisionado que utiliza o conceito de centroide, uma representação do centro de um grupo, obtido pela média de todos os objetos pertencentes a este mesmo grupo.

De acordo com o trabalho de Pimentel, França e Omar (2003), o objetivo do K-Médias é encontrar a melhor divisão de X dados em K grupos ou *clusters* (esse termo técnico será utilizado ao longo do texto como sinônimo de agrupamento), de modo que a distância entre os objetos de um grupo e seu centroide possa ser minimizada. Para entender melhor o seu procedimento, a Figura 7 é utilizada como referência.

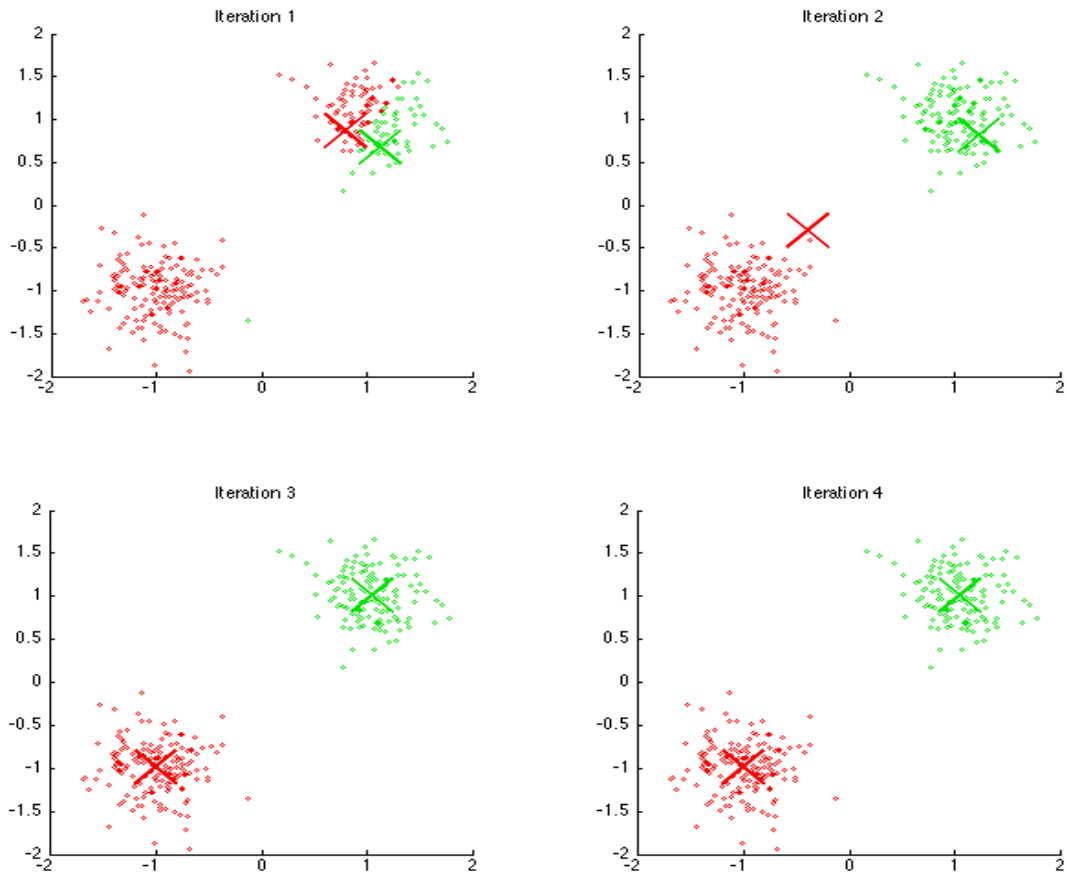


Figura 7. Iterações do algoritmo K-Médias com $K = 2$.

Na primeira iteração do algoritmo, utilizando apenas dois *clusters* ($K = 2$), dois pontos são escolhidos aleatoriamente para serem centroides. Então, a distância de cada objeto para cada centroide é calculada, e aqueles que possuem maior similaridade ou estão mais próximos de um determinado centroide, são agrupados em um mesmo *cluster*.

Em seguida, na segunda iteração, recalcula-se o valor do centroide de cada grupo através da média dos objetos que atualmente fazem parte deste. Repete-se o cálculo das distâncias de cada objeto para cada novo centroide e formam-se os novos *clusters*. Repete-se o processo até que não haja mais deslocamento dos centroides, isto é, até que se encontre um equilíbrio, como observa-se na iteração 4.

2.4 TRANSFORMADA DISCRETA DO COSSENO

Converter imagens do domínio do espaço para o domínio da frequência pode ser uma ótima estratégia na tentativa de explorar características não tão visíveis no primeiro domínio. Para realizar tal conversão, utiliza-se uma operação matemática denominada transformada, que possibilita o processamento do sinal em frequências e, posteriormente, utilizando a sua transformada inversa, pode-se retornar ao domínio original.

Entre as transformadas mais utilizadas, a Transformada Discreta do Cosseno (*Discrete Cosine Transform*, ou DCT), desenvolvida no artigo de Ahmed, Natarajan e Rao (1974), ganhou destaque pela sua praticidade e amplo uso na área de compressão de dados.

A DCT decompõe um sinal discreto em um somatório de N funções cosseno, isto é, em um conjunto de coeficientes onde cada um expressa a importância de uma onda cosseno para construir o sinal original (KHAYAM, 2003).

A equação que define a DCT para uso em imagens, isto é, sinais bidimensionais, é dada por:

$$X[k, l] = \frac{1}{\sqrt{2N}} c_k c_l \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x[m, n] \cos\left[\frac{(2m+1)k\pi}{2N}\right] \cos\left[\frac{(2n+1)l\pi}{2N}\right] \quad (6)$$

onde $X[k, l]$ é a seqüência produzida; $x[m, n]$ é o sinal bidimensional que representa uma imagem de tamanho $N \times N$; $0 \leq k \leq N - 1$ e $0 \leq l \leq N - 1$.

Dos coeficientes gerados, o que representa a componente de frequência zero ($X[0]$) é conhecido como coeficiente DC (*direct current*), pois possui relação direta com o valor médio do sinal - da imagem, ou até mesmo do pixel; enquanto que os demais coeficientes são chamados de AC (*alternating current*) (BATISTA, 2005). Dessa maneira, os coeficientes mais próximos do DC estão relacionados com as baixas frequências, nas quais, em imagens, a maior parte da energia está concentrada, isto é, onde estão as informações mais importantes.

A Figura 8 ilustra o resultado visual da DCT de uma imagem real e a concentração de energia que sempre há no canto superior esquerdo, o local onde se aglomeram as baixas frequências do sinal.



Figura 8. Resultado visual da Transformada Cosseno Discreta de uma imagem real.

3 MATERIAIS E MÉTODOS

Para implementação do método desenvolvido, utilizou-se a linguagem de programação Java, em um computador com processador Intel Core i5, 6GB de memória RAM DDR3 e sistema operacional Ubuntu 14.04.

A entrada do sistema é uma imagem monocromática ou em escala de cinza, assim como foi feito no artigo de Buades, Coll e Morel (2004), na intenção de facilitar os testes, uma vez que para processar imagens coloridas é preciso apenas repetir os cálculos nas outras duas bandas de cor. Para essa fase inicial de testes e implementação utilizou-se apenas a imagem *Lena*, extremamente utilizada em testes na área de remoção de ruído, obtida na webpage de Portilla (2003).

3.1 MÉTODO DESENVOLVIDO

Reiterando o que foi exposto na Seção 1.1, a proposta deste trabalho é otimizar o algoritmo original do NLM, diferente da grande maioria dos trabalhos da literatura que buscam aprimorar o NLM em sua adaptação com uma janela de busca.

Como explicado na dissertação de mestrado de Figueiredo (2014), essa adaptação acaba tornando o método efetivamente local ou ‘semi-local’, pois reduz o campo de procura de similaridade à vizinhanças próximas. Além disso, em regiões onde as diferenças de nível de cinza não são suaves - como nas texturas, ou em imagens com altos níveis de ruído - e as semelhanças locais tendem a diminuir, as metodologias globais tendem a obter um melhor desempenho.

Como inspiração para a metodologia aqui adotada, os trabalhos de Peter e Ramya (2012), utilizando o algoritmo K-Médias para agrupar pixels com nível de cinza semelhante, e de Hu *et al* (2012), com a Transformada Cosseno Discreta como artefato para calcular os pesos das vizinhanças, foram de grande importância para se abstrair conceitos importantes e gerar idéias que originaram o método aqui proposto.

A Figura 9 expressa um diagrama que resume o funcionamento do método desenvolvido. Primeiramente, recebe-se uma imagem em escala de cinza com ruído e, através da DCT, obtém-se os valores das vizinhanças de cada pixel no domínio da frequência. Em

seguida, realiza-se um processo de agrupamento de todos os pixels da imagem utilizando o K-Médias, de acordo com os valores obtidos na etapa anterior. Obtendo-se tais agrupamentos de pixels com vizinhanças semelhantes no domínio da frequência, executa-se o NLM para cada um dos K (valor escolhido na entrada do sistema) *clusters*. Por fim, os novos valores de cada pixel são obtidos e uma nova imagem filtrada é gerada como saída.

O processo apresentado pode ser subdividido em três etapas principais: transformação para o domínio da frequência, agrupamento e aplicação do NLM. Sendo assim, nas seções seguintes cada uma dessas etapas será melhor descrita e esclarecida quanto aos seus detalhes.

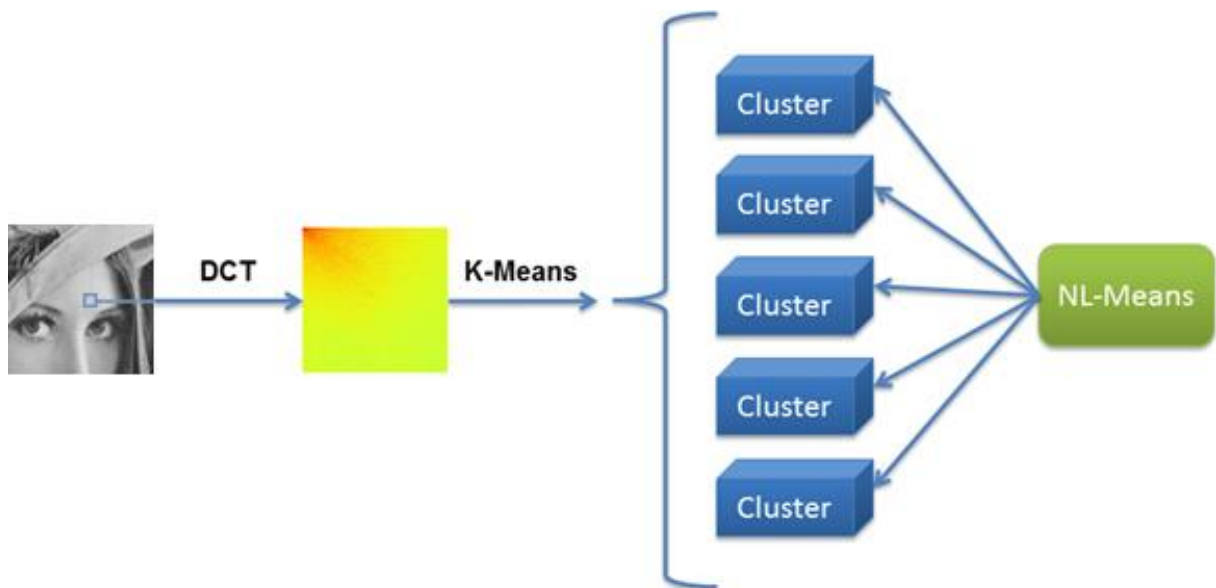


Figura 9. Diagrama do funcionamento do método desenvolvido

3.2 CONVERTENDO OS PIXELS DA IMAGEM PARA O DOMÍNIO DA FREQUÊNCIA

Após receber a imagem como entrada, aplica-se a DCT a cada pixel alvo e as suas respectivas vizinhanças, guardando os valores obtidos em cada sequência. Como tais valores são armazenados em forma de matriz, pode-se entender o tamanho dessa sequência como sendo o tamanho de uma janela $N \times N$. Desse modo, o tamanho da janela DCT escolhido predefine a quantidade de coeficientes que serão decompostos. E, visando padronizar com o

tamanho da janela gaussiana do NLM, optou-se pelo tamanho 5×5 , obtendo-se então 25 elementos de frequência para cada processamento em cada pixel alvo - os quais indicam as características deste pixel, como a intensidade de nível de cinza, texturas, bordas e ruído agregado. Essas informações são armazenadas e enviadas para a etapa de agrupamento.

3.3 AGRUPANDO OS PIXELS SEMELHANTES EM *CLUSTERS*

A partir dos dados obtidos no passo anterior, o algoritmo K-Médias inicia o processo de agrupamento enxergando cada pixel no domínio da frequência através dos seus coeficientes. Inicialmente escolhe-se de forma aleatória K centroides e computam-se as distâncias, que são medidas pelo grau de semelhança dos coeficientes do centroide com relação aos coeficientes dos demais pixels.

Formam-se os primeiros *clusters* e, logo em seguida, através da média de todos os elementos de cada *cluster*, recalcula-se o valor dos atuais centroides, obtendo-se novos, a partir desse resultado. Repete-se o processo n vezes até alcançar-se certo equilíbrio – quando não houver mais deslocamento dos elementos entre os agrupamentos. Têm-se então os K *clusters* contendo os pixels mais semelhantes entre si. O valor de n - isto é, a quantidade de rodadas do algoritmo - pode ser definido na entrada do método ou pode-se deixar que o próprio algoritmo encontre o valor adequado para se chegar num estado em que não há mais alteração de centroides e, assim, atinge-se tal equilíbrio citado anteriormente.

Foi observado que há um problema recorrente no processo de agrupamento: a existência de elementos localizados nas fronteiras dos *clusters*, que gera uma deficiência na imagem final, uma vez que pode-se notar visualmente a diferenciação dos *clusters* em algumas regiões da imagem.

A Figura 10 ilustra um caso típico, onde um elemento quadrado encontra-se na fronteira entre dois *clusters*. Por ser semelhante tanto a um centroide quanto a outro, tal elemento assumiu uma posição intermediária entre os dois. Para este caso, o ideal seria que ele participasse dos dois *clusters* ao mesmo tempo.

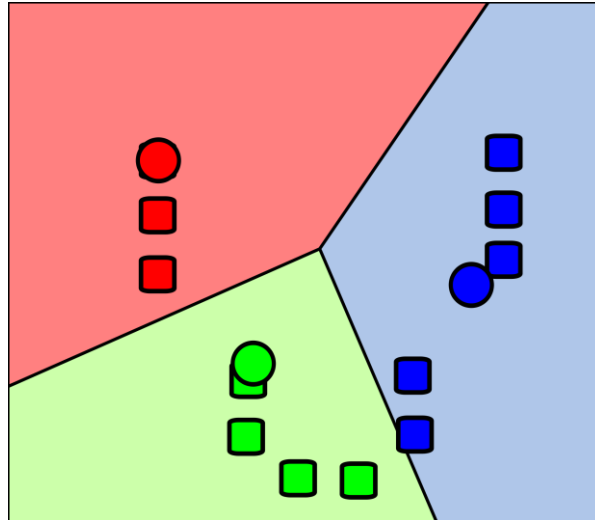


Figura 10. Representação de agrupamento com problemas nas fronteiras

No processo de filtragem, a abordagem mais interessante seria que um pixel localizado na fronteira fosse filtrado por aqueles que mais se assemelham a ele, pois, assim, as características próprias à sua vizinhança seriam preservadas e o resultado final seria mais fiel à realidade da imagem original (FIGUEIREDO, 2014).

Dessa maneira, criou-se um percentual de tolerância para a participação em mais de um *cluster*, funcionando da seguinte maneira: após atingir-se a estabilidade no agrupamento, isto é, na última iteração, calcula-se a distância de cada pixel para os outros $K-1$ centroides mais distantes (define-se D como sendo a distância dele para o centroide mais próximo); se a distância para algum deles for menor ou igual a $(D + x\%D)$, o pixel participará também desse outro *cluster*.

Dessa forma, se um pixel P possui uma distância de 100 para o centroide do seu *cluster* C e a tolerância pré-definida é de 20%, então, se a distância D' para um *cluster* C' é de 120 ou menos, P irá participar também do *cluster* C' no processo de filtragem. A configuração final do agrupamento ficará semelhante à Figura 11.

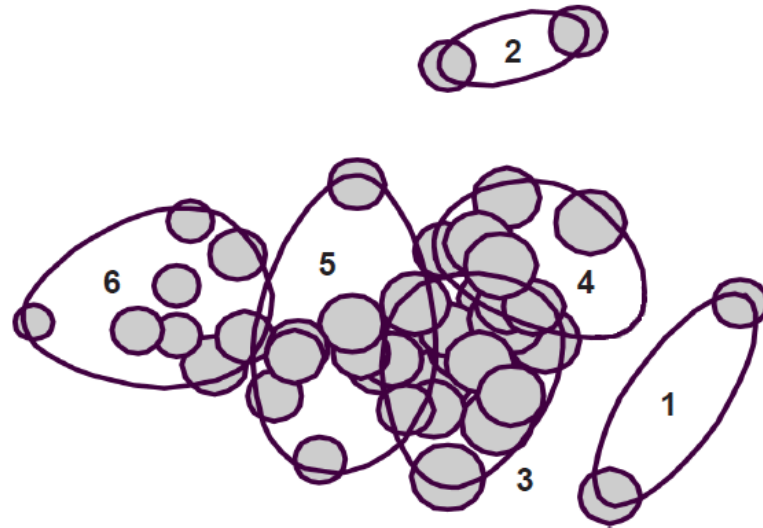


Figura 11. Representação de agrupamento com pixels participando de mais de um *cluster*.

3.4 FILTRAGEM ATRAVÉS DO NLM

Possuindo os K *clusters* contendo os pixels mais semelhantes entre si no domínio da frequência, o próximo passo então é a filtragem para remoção do ruído. Para esta etapa manteve-se a janela Gaussiana com tamanho 5×5 e o valor do fator de decaimento ao quadrado, empiricamente, em 200, assim como foi sugerido pelos criadores do NLM.

Aqui o NLM analisa cada pixel da imagem, identificando os *clusters* (ou o *cluster*) que ele faz parte, e utilizando as vizinhanças participantes para realizar o cálculo dos pesos - que são todos relevantes, uma vez que há semelhança entre todos os elementos do *cluster*. Assim, realiza-se a filtragem do pixel alvo na tentativa de eliminar o ruído indesejado, e logo em seguida, armazena-se o seu novo valor para transferir à imagem resultante que será gerada.

Através dessa abordagem a complexidade do NLM é reduzida, uma vez que limita-se o campo de procura de vizinhanças semelhantes aos *clusters* que o pixel pertence - ao invés de fazê-lo na imagem inteira - e, ainda, torna a busca mais eficiente pois dentro desses agrupamentos não se encontra vizinhanças distintas ou tão diferentes entre si. Por último, a busca por semelhanças no domínio da frequência é muito mais eficaz do que no domínio do espaço, uma vez que explora as características essenciais dos pixels de forma mais robusta (KHAYAM, 2003).

3.5 METODOLOGIA DA AVALIAÇÃO EXPERIMENTAL

Para os resultados obtidos e apresentados no Capítulo 4, o mesmo computador e a mesma imagem citada neste capítulo (além das imagens *Boats* e *Barbara*) foram utilizados – imagens com 256×256 pixels. Ainda, utilizou-se o *MATLAB* 2015 para inserção de ruído nas imagens - fixando-se o ruído gaussiano branco como padrão -, para o cálculo do MSE e para a geração do ruído de método (esses foram os parâmetros de medição de qualidade utilizados, como citado anteriormente, bem como o tempo de execução do processamento e a qualidade visual subjetiva observada).

4 RESULTADOS E DISCUSSÕES

Neste capítulo serão apresentados os resultados referentes aos testes de quantidade de coeficientes da DCT, bem como da tolerância para resolução de fronteiras e quantidade de rodadas no K-Médias e da quantidade de *clusters* ideal para um processamento eficaz do NLM. Por fim, serão apresentadas as comparações realizadas entre o algoritmo proposto e o algoritmo original.

4.1 QUANTIDADE DE COEFICIENTES DA JANELA DCT

Realizando-se os primeiros experimentos, notou-se que o resultado visual da filtragem estava bem inferior ao encontrado no algoritmo original e, além disso, o valor do MSE encontrado era bem maior. Decidiu-se então realizar testes com relação a quantidade de coeficientes da DCT, na tentativa de minimizar o MSE e melhorar a qualidade visual. Dessa maneira, filtrou-se a imagem *Lena* utilizando apenas o elemento DC ($[0,0]$), depois o DC mais um elemento AC de baixa frequência $[0,1]$, depois o DC mais o $[0,1]$ e o $[1,0]$ e assim sucessivamente, até chegar às altas frequências. Os gráficos 1 e 2 foram obtidos, os quais representam a quantidade de coeficientes da DCT *versus* tempo e a quantidade de coeficientes da DCT *versus* MSE, respectivamente.

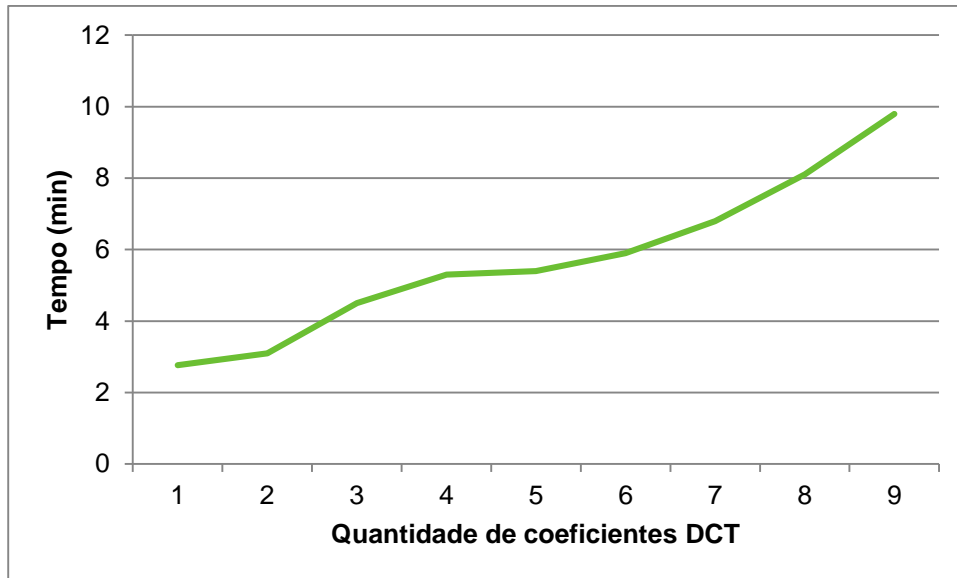


Gráfico 1. Quantidade de coeficientes DCT x Tempo (em minutos)

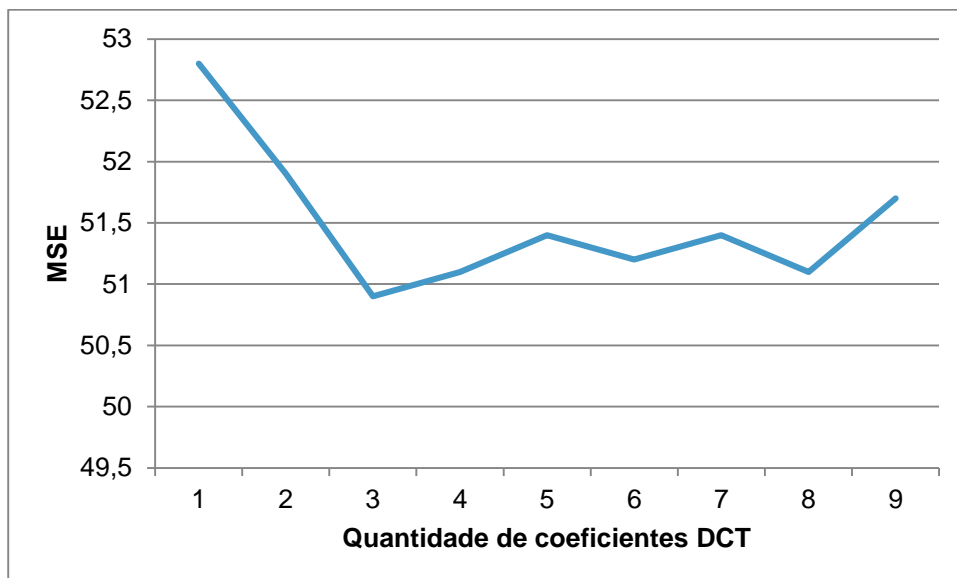


Gráfico 2. Quantidade de coeficientes DCT x MSE

No Gráfico 1, vê-se que quanto maior a quantidade de elementos maior o tempo de execução, como já era esperado, uma vez que mais cálculos são realizados. Mas no Gráfico 2, o valor do MSE não demonstra uma variação uniforme, pois ele sofre pelo menos duas quedas e subidas.

Unindo as informações obtidas de ambos gráficos e sabendo que a DCT armazena energia nas frequências mais baixas e, assim, algumas altas frequências podem ser descartadas sem causar uma degradação significativa na qualidade da imagem (KHAYAM,

2003), optou-se por armazenar apenas o elemento DC [0,0] e os elementos AC de baixa frequência [0,1] e [1,0]. Além disso, com apenas esses três coeficientes obteve-se o menor valor de MSE, com um tempo de execução não tão elevado.

Como se pode visualizar na Figura 12, esses três coeficientes representam as mais baixas frequências e, como o ruído tende a se concentrar nas altas frequências (GONZALEZ; WOODS, 2002), a informação armazenada de cada pixel para a próxima etapa já sofre uma menor influência do ruído, realizando-se uma “pré-filtragem”, garantindo uma otimização no processo de filtragem, bem como no resultado final do processamento.

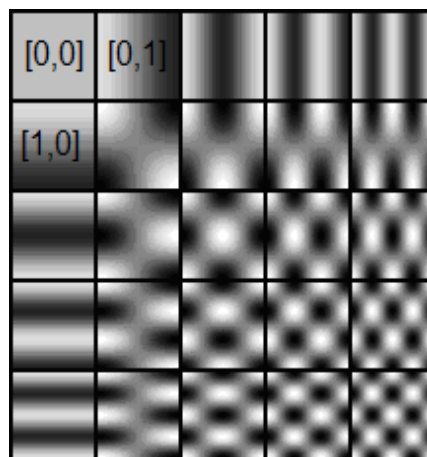


Figura 12. Coeficientes da janela DCT 5x5

4.2 TOLERÂNCIA PARA RESOLUÇÃO DE FRONTEIRAS NO K-MÉDIAS

Definido o escopo a ser utilizado quanto à DCT, o próximo passo foi a definição de qual a tolerância padrão a ser utilizada para resolver o problema das fronteiras nos *clusters*.

Após diversos testes variando as porcentagens para a tolerância, três valores principais foram escolhidos: 30, 50 e 70 por cento. Quando a tolerância é menor do que 30%, poucos são os pixels que passam a participar de outro *cluster*. Já quando é maior do que 70%, praticamente todos os pixels participam de outro *cluster*, tornando os agrupamentos demasiadamente grandes e o processamento muito lento.

A título de ilustração, as Tabelas 1 e 2 mostram os valores do MSE e do tempo de execução obtidos através da variação da tolerância do K-Médias para a imagem da *Lena* com

variância do ruído Gaussiano 0,0008 e 0,002. Nota-se que para uma imagem com pouco ruído, há uma alteração maior no valor do MSE quando se varia a tolerância, como pode-se observar na Tabela 1. Por outro lado, se o ruído é intenso, a alteração no valor do MSE é irrelevante (Tabela 2). No entanto, a discrepância entre os tempos de execução permanece a mesma para ambos os casos: uma variação de cerca de 0,75 minutos.

Tolerância	MSE	Tempo de execução (min.)
30%	20,55	4,96
50%	20,20	5,66
70%	19,88	6,40

Tabela 1. MSE e tempo de execução obtidos através da variação da tolerância para *Lena* com variância do ruído Gaussiano 0,0008.

Tolerância	MSE	Tempo de execução (min.)
30%	50,97	4,62
50%	50,87	5,28
70%	50,84	6,05

Tabela 2. MSE e tempo de execução obtidos através da variação da tolerância para *Lena* com variância do ruído Gaussiano 0,002.

Como este trabalho busca aproveitar ao máximo as situações em que o NLM original tem maior eficiência, isto é, em imagens com ruído mais intenso, a escolha da tolerância de 30% é adequada uma vez que gera *clusters* menores e, conseqüentemente, menor tempo de execução sem alteração considerável no valor do MSE.

4.3 QUANTIDADE DE RODADAS DO K-MÉDIAS

Como visto no capítulo anterior, o algoritmo de agrupamento escolhido pode receber como entrada um valor n que indicará a quantidade de iterações do método. Caso esse valor não seja informado, o K-Médias irá rodar tantas vezes quantas forem necessárias até encontrar um equilíbrio e assim finalizar seu processamento.

Sabendo que um dos objetivos deste trabalho é reduzir o tempo de processamento do algoritmo original, subentendeu-se que uma quantidade menor de iterações implicaria em um menor processamento e, assim, um menor tempo de execução final. Contudo, uma quantidade menor de iterações também poderia implicar numa menor eficiência do método de agrupamento, formando *clusters* com pixels que não fossem tão semelhantes entre si.

Para se obter respostas acerca desse questionamento, testes foram realizados com as três imagens escolhidas, variando-se o nível de ruído e a quantidade de *clusters*. Para ilustrar um padrão encontrado nos valores do MSE e no tempo de execução, a Tabela 3 apresenta os resultados encontrados para a imagem *Boats*, com variância de ruído 0,002, em que K é a quantidade de *clusters* e a quantidade de rodadas ‘todas’ seria o caso em que não se informa um valor fixo para elas, permitindo o K-Médias rodar até encontrar o seu equilíbrio.

K	Rodadas	MSE	Tempo total (min)
10	2	75,44	4,24
	5	75,08	3,22
	10	74,99	3,18
	todas	74,95	4,19
100	2	75,15	2,48
	5	74,74	1,30
	10	74,40	0,99
	todas	74,16	2,31
300	2	75,02	2,33
	5	74,89	2,07
	10	74,65	1,99
	todas	74,64	6,22

Tabela 3. Valores de MSE e tempo de execução para imagem *Boats* com variância de ruído 0,002, alternando-se a quantidade de rodadas do K-Médias

O padrão encontrado tem fundamento naquilo que foi citado anteriormente: como esperado, a diminuição na quantidade de iterações gera um aumento, não muito elevado, do valor do MSE – uma vez que o processo de agrupamento se dá de forma menos eficiente - e, ao mesmo tempo, obtém-se uma redução no tempo de execução do algoritmo (em alguns casos) – com duas rodadas, por os *clusters* ficarem extremamente mal organizados e bem superlotados, o tempo gasto no processamento da filtragem do K-Médias acaba por tornar-se mais elevado, gerando tal crescimento no tempo total do método.

Conclui-se, a partir do observado, que ao se fixar o número de rodadas em 10, é possível obter-se um valor de MSE bem semelhante mas com um tempo de execução bem inferior se comparado à utilização de todas as rodadas necessárias para se alcançar um equilíbrio no processamento do K-Médias.

4.4 QUANTIDADE DE *CLUSTERS* IDEAL

Estabelecidos então os parâmetros necessários para a DCT – janela 5x5 e apenas os três primeiros coeficientes - e o K-Médias – tolerância de 30% e apenas 10 rodadas -, a última etapa foi descobrir se há uma quantidade de *clusters* ideal para que o algoritmo desenvolvido atue da forma mais eficaz possível.

Repetiu-se, então, o mesmo procedimento de testes para as três imagens, mantendo os parâmetros citados anteriormente, variando-se apenas o nível de ruído e as quantidades de *clusters* entre 10, 50, 100, 200 e 300. Empiricamente achou-se menos do que 10 um valor muito baixo (e quase próximo ao processamento do NLM original) e acima de 300 muito pesado computacionalmente. Dos resultados obtidos, como forma de ilustração, as Tabelas 4 e 5 apresentam os valores de MSE e tempo de execução encontrados para a imagem Barbara com variância 0,002 e 0,008, respectivamente.

K	MSE	Tempo total (min)
10	76,84	2,87
50	77,23	1,89
100	76,61	0,76
200	76,90	1,13
300	77,41	1,81

Tabela 4. Valores do MSE e tempo de execução para imagem *Barbara* com variância de ruído 0,002 alternando-se a quantidade de *clusters*

K	MSE	Tempo total (min)
10	485,83	2,75
50	488,47	1,87
100	491,3	0,68
200	493,22	0,97
300	495,71	1,69

Tabela 5. Valores do MSE e tempo de execução para a imagem *Barbara* com variância de ruído 0,008 alternando-se a quantidade de *clusters*.

Como pode-se concluir, em imagens com ruído não tão intenso, com variância de 0,002 (Tabela 4) por exemplo, o aumento da quantidade de *clusters* gera pequenas variações nos valores do MSE - porém não tão bruscas, enquanto que em imagens com ruído mais intensos, com variância de 0,008 (Tabela 5) por exemplo, há um aumento progressivo nos valores do MSE.

Com relação ao tempo de processamento, percebe-se que a utilização de uma quantidade maior de *clusters* implica numa redução desse valor, encontrando um limiar entre 100 e 200 *clusters*, quando o método volta a demorar mais para finalizar seu processamento.

Por fim, para as três imagens, como observado nas Tabelas 4 e 5 para esses casos específicos, percebeu-se claramente que 100 *clusters* seria a quantidade ideal para a filtragem nas mais diversas intensidades de ruído.

4.5 COMPARAÇÃO COM O NLM ORIGINAL

A última etapa então consistiu em comparar o algoritmo desenvolvido com o NLM original, a fim de confirmar se houve otimização no tempo de execução e se foi possível manter, ou até melhorar, os demais parâmetros matemáticos e visuais.

Selecionando a configuração ideal encontrada (detalhada nos pontos passados deste capítulo) testou-se então, para as três imagens, com as mesmas variâncias de ruído utilizadas anteriormente, os valores de MSE encontrados. Esses valores foram, então, comparados com aqueles encontrados pela execução do algoritmo original, e os resultados encontram-se nos Gráficos 3 e 4 que foram produzidos utilizando-se variâncias 0,002 e 0,008, respectivamente:

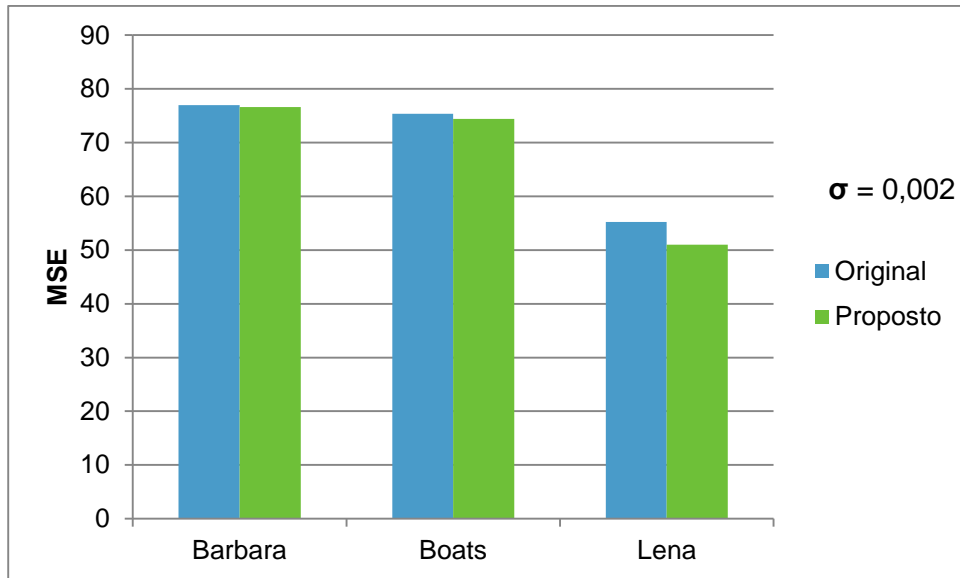


Gráfico 3. Comparação dos valores do MSE obtidos entre o NLM original e o algoritmo proposto para as três imagens utilizadas com variância de ruído 0,002

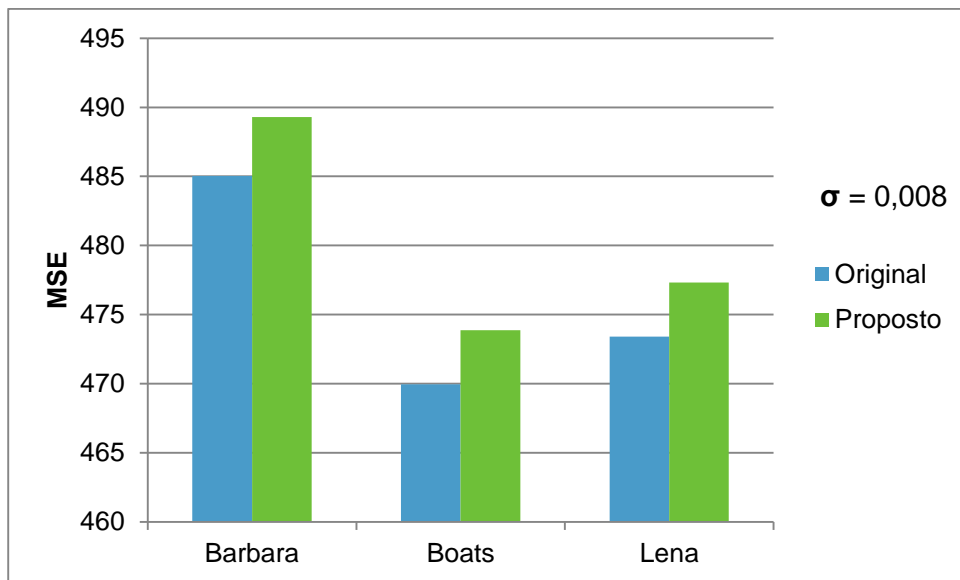


Gráfico 4. Comparação dos valores do MSE obtidos entre o NLM original e o algoritmo proposto para as três imagens utilizadas com variância de ruído 0,008

Ao se observar os Gráficos 3 e 4, conclui-se claramente que, para níveis de ruído não tão intensos (0,002), o método proposto consegue valores de MSE um pouco mais baixos – porém mais relevantes já que o valor absoluto (50-80) não é tão alto – do que o algoritmo original. Já para níveis de ruído mais intensos (0,008), o inverso ocorre, porém tendo o valor absoluto do MSE muito mais elevado (475-490), o que implica numa menor relevância da

diferença entre os valores encontrados para cada algoritmo. Portanto pode-se concluir aqui que, para os diversos níveis de ruído, os valores encontrados para o MSE para as três imagens testadas foram bem semelhantes, com um desvio padrão máximo de 5.

Com relação ao tempo de processamento, a Tabela 6 expressa os resultados obtidos em minutos para os mesmos casos ilustrados. Pode-se observar que há uma redução considerável de, em média, 19 vezes, atingindo-se então um dos principais objetivos deste trabalho.

Tempo (min)				
	$\sigma = 0,002$		$\sigma = 0,008$	
	Original	Proposto	Original	Proposto
Barbara	15,71	0,76	16,57	0,69
Boats	17,36	0,99	18,17	1
Lena	18,23	1	17,85	0,93

Tabela 6. Comparação dos tempos de processamento (em minutos) do NLM Original e o algoritmo proposto para imagens com variância de ruído 0,002 e 0,008.

Referente à qualidade visual, pode-se observar na Figura 13 que os ruídos de método, para imagem Lena com variância de ruído 0,008, são extremamente parecidos. Nos dois casos é possível identificar que o ruído foi filtrado eficientemente, uma vez que não é possível notar de forma clara vestígios das texturas ou bordas da imagem.

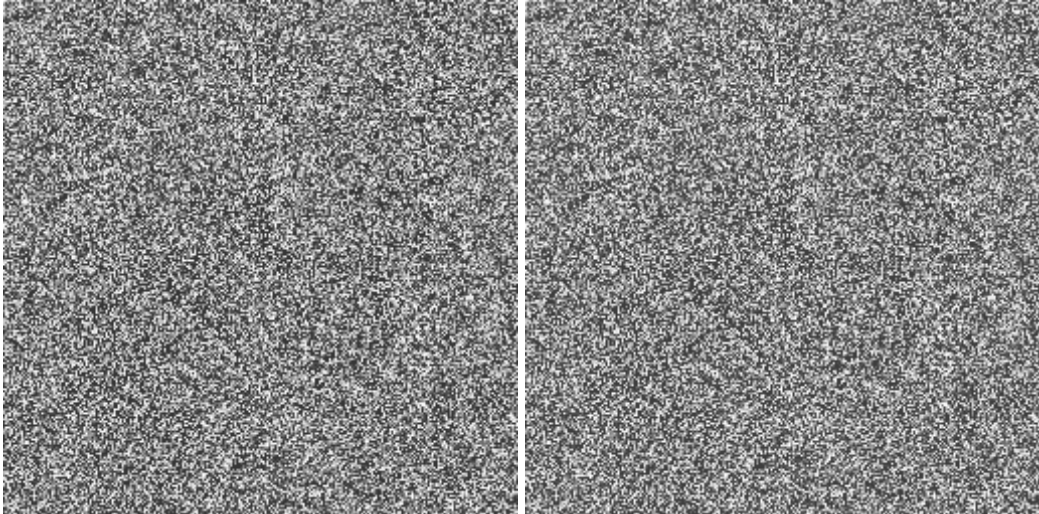


Figura 13. Ruído de método do NLM original (à esquerda) e do método proposto (à direita) para imagem *Lena* com nível de ruído 0,008.

Analisados o MSE, o tempo de execução e o ruído de método, o último passo então é a análise visual subjetiva acerca do resultado da filtragem. A Figura 14 expõe a imagem ruidosa (acima), o resultado da filtragem do NLM original (à esquerda) e do método proposto (à direita). É possível encontrar algumas pequenas diferenças em alguns trechos da imagem - onde o método proposto parece ter filtrado melhor do que o original - mas, de forma geral, o resultado visual de ambos os algoritmos é bem semelhante.



Figura 14. Resultado visual da filtragem da *Lena*. Acima, a imagem ruidosa; à esquerda o resultado do NLM original e à direita, do método proposto

5 CONSIDERAÇÕES FINAIS

Neste trabalho, uma otimização para o algoritmo de remoção de ruído NLM foi apresentado. Por estar no estado da arte na área de Processamento Digital de Imagens, esse algoritmo tem sido estudado por muitos pesquisadores e diversas propostas de otimização tem sido desenvolvidas e apresentadas na literatura. Porém, tais propostas visam aprimorar uma adaptação do algoritmo sugerida pelos próprios autores utilizando uma janela de busca, que possui complexidade computacional menor do que o algoritmo original e reduz o seu princípio de ‘não-localidade’.

Para imagens com ruído intenso, o NLM original possui maior eficiência na filtragem do que sua adaptação com a janela de busca, apesar de possuir um tempo de execução extremamente maior. Por tal motivo, este trabalho propôs uma otimização utilizando domínio da frequência e agrupamentos com K-Médias para reduzir a complexidade do algoritmo e, conseqüentemente, seu tempo de processamento, buscando manter ou até melhorar a qualidade visual e matemática.

O método consiste em transformar cada *pixel* da imagem para o domínio da frequência através da DCT e agrupar esses pixels através do K-Médias, de acordo com a semelhança das baixas frequências de suas vizinhanças. Após isso, direciona-se a ação do NLM para cada *cluster*, obtendo-se os valores dos novos pixels para gerar a imagem filtrada.

Os resultados obtidos testificam uma aceleração no tempo de execução de cerca de 19 vezes, obtendo valores do MSE semelhantes e, em alguns casos, melhores do que ao NLM original. Além disso, os resultados obtidos através do ruído de método aparentam ser equivalentes. E, por fim, os resultados visuais das imagens filtradas por ambos os métodos aparentam ser similares a ponto de serem confundidas.

Para trabalhos próximos, pretende-se: realizar todos os testes com outras imagens monocromáticas que são amplamente usadas na literatura; estudar a eficiência do algoritmo em alguma área específica de imagens digitais, como as mamográficas; estudar os algoritmos *Fast K-Médias* e *Fast DCT* e substituir as versões originais por estes, visando uma maior redução no tempo de processamento; realizar novos testes alterando o fator de decaimento, quantidade de elementos e tamanho da janela DCT e a tolerância na etapa de agrupamento; aprimorar a eficiência do algoritmo utilizando *multithread*; e, por fim, criar uma heurística

para descobrir o nível de ruído da imagem de entrada e realizar, automaticamente, a escolha dos valores dos parâmetros do algoritmo de forma a realizar uma filtragem ótima.

REFERÊNCIAS

- AHMED, N.; NATARAJAN, T.; RAO, K. R. On image processing and a discrete cosine transform. *IEEE Transactions on Computers* C-23(1): 90-93, 1974.
- BATISTA, L. *Notas de aula da disciplina Introdução ao Processamento Digital de Imagens*. Universidade Federal da Paraíba, 2005.
- BHUJLE, H.; CHAUDHURI, S. Accelerating non-local denoising with a patch based dictionary. In *Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing*, 2012.
- BROX, T.; KLEINSCHMIDT, O.; CREMERS, D. Efficient Nonlocal Means for Denoising of Textural Patterns, *Image Processing, IEEE Transactions on* , vol.17, no.7, pp.1083,1092, 2008.
- BUADES, A.; COLL, B.; MOREL, J. On image denoising methods , *Technical Note v. 5*, CMLA, p.40, 2004.
- BUADES, A.; COLL, B.; MOREL, J. A Non-Local Algorithm for Image Denoising. *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol.2, pp.60-65, 2005.
- BUADES, A.; COLL, B.; MOREL, J. Non-Local Means Denoising, *Image Processing On Line*, pp. 208–212, 2011.
- COUPÉ, P.; YGER, P.; PRIMA, S.; HELLIER, P.; KERVRANN, C.; BARILLOT, C. An optimized blockwise nonlocal means denoising filter for 3-D magnetic resonance images, *IEEE Transactions on Medical Imaging*, 27(4), 425–441. , 2008.
- DABOV, K.; KATKOVNIK, V; EGIAZARIAN, K. Image denoising by sparse 3-D transform-domain collaborative filtering, *IEEE Trans. Image Process.*, vol. 16, no. 8, 2007.
- FIGUEIREDO, L. *Filtragem de Imagens Baseado no Algoritmo Non-Local-Means em Ambientes Multiprocessados*, Tese (Mestrado) - Universidade Federal da Paraíba, 2014.

GONZALEZ, R.C; WOODS, R.E. Digital Image Processing, 2nd ed., p. 160, Prentice Hall, 2002.

HU, J.; PU, Y.; WU, X.; ZHANG, Y.; ZHOU, J. Improved DCT-Based Nonlocal Means Filter for MR Images Denoising, *Computational and Mathematical Methods in Medicine*, vol. 2012, 2012 .

JAIN, P.; TYAGI, V. A survey of edge-preserving image denoising methods, *Information Systems Frontiers*, p.1, 2014.

KEVRANN, C.; BOULANGER, J. Optimal spatial adaptation for patch-based image denoising, *IEEE Trans. Image Process.*, vol. 15, no. 10, 2006.

KHAYAM, S. A. *The Discrete Cosine Transform (DCT): Theory and Application*. Michigan State University, 2003.

LINDENBAUM, M.; FISCHER, M.; BRUCKSTEIN, A. On gabor contribution to image enhancement. *Pattern Recognition*, 27:1–8, 1994.

MACQUEEN, J. B. Some Methods for classification and Analysis of Multivariate Observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*. 281–297, University of California Press, 1967.

MAHMOUDI, M.; SAPIRO, G. Fast image and video denoising via nonlocal means of similar neighborhoods, *Signal Processing Letters, IEEE* , vol.12, no.12, pp.839,842, 2005.

MAIRAL, J.; BACH, F.; PONCE, J.; SAPIRO, G.; ZISSERMAN, A. Non-local sparse models for image restoration, *ICCV*, 2009.

PERONA, P.; MALIK, J.. Scale space and edge detection using anisotropic diffusion. *IEEE Trans. Patt. Anal. Mach. Intell.*, 12:629–639, 1990.

PETER, J.; RAMYA, R. A novel adaptive Non Local Means for image de-noising, *Procedia Engineering*, v. 38, p. 3278-3282, 2012.

PIMENTEL, E.; FRANÇA, V. F.; OMAR, N. A identificação de grupos de aprendizes no ensino presencial utilizando técnicas de clusterização. *Anais do Simpósio Brasileiro de Informática na Educação*, Rio de Janeiro, RJ, 2003.

PORTILLA, J. *BLS-GSM Image Denoising*, 2003. Disponível em: <http://decsai.ugr.es/~javier/denoise/test_images/>. Acesso em Fevereiro 06, 2015.

SALMON, J.; STROZECKI, Y. From patches to pixels in Non-Local methods: Weighted-average reprojection, *Image Processing (ICIP), 2010 17th IEEE International Conference on* , vol., no., pp.1929,1932, 2010

TOMASI, C.; MANDUCHI, R. Bilateral filtering for gray and color images. *In Proceedings of the Sixth International Conference on Computer Vision*, p. 839–846, 1998.