UNIVERSIDADE DE SÃO PAULO–USP
ESCOLA POLITÉCNICA

Carlos Fabbri Jr

**Recuperação de imagens
multiespectrais por sensoramento
compressivo com o uso de câmeras de
pixel único e algoritmos convexos de
programação linear**

São Paulo
2020

Carlos Fabbri Jr

**Recuperação de imagens
multiespectrais por sensoramento
compressivo com o uso de câmeras de
pixel único e algoritmos convexos de
programação linear**

Versão Revisada

Dissertação de mestrado apresentada à Escola Politécnica para a obtenção do título de Mestre em Ciências.

Área de concentração: Sistemas Elétricos de Potência

Orientador: Prof. Dr. Josemir Coelho Santos

São Paulo
2020

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, _____ de _____ de _____

Assinatura do autor: _____

Assinatura do orientador: _____

Catálogo-na-publicação

Fabbri Jr, Carlos

Recuperação de imagens multiespectrais por sensoriamento compressivo com o uso de câmeras de pixel único e algoritmos convexos de programação linear / C. Fabbri Jr -- versão corr. -- São Paulo, 2020.
62 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Energia e Automação Elétricas.

1.Amostragem 2.Recuperação de imagens 3.Processamento de imagens
4.Sensoriamento compressivo I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Energia e Automação Elétricas II.t.

A meu pai, Carlos Fabbri, que me ensinou pelo exemplo que: “querer é poder!”

Agradecimentos

Agradeço à dedicação, paciência e orientação de meu orientador, Dr. Josemir.

Agradeço também ao amigo Mestre Shigueru Nagao Jr, pela ajuda sincera e sempre presente.

Em especial sou grato à minha esposa, amiga e companheira Angela pelas palavras de ânimo, carinho e orientação nos momentos que pareciam ser mais difíceis!

“Só é digno da liberdade, como da vida, aquele que se empenha em conquistá-la.”
(Johann Goethe)

Resumo

Fabbri Jr, Carlos **Recuperação de imagens multiespectrais por sensoriamento compressivo com o uso de câmeras de pixel único e algoritmos convexos de programação linear.** 62 p. Dissertação de mestrado – Escola Politécnica, Universidade de São Paulo, 2020.

O presente trabalho mostra uma nova metodologia para a aquisição de imagens do espectro visível, IR e UV usando um número significativamente menor de amostras do que a teoria convencional de Shannon-Nyquist recomenda. Essa nova metodologia é baseada em uma teoria inovadora e revolucionária chamada de Compressed Sensing ou sensoriamento compressivo. Ela propõe um novo método de captura das informações essenciais da imagem ou objeto sendo amostrados, baseado no conhecimento de que essas informações são esparsas em uma determinada base de representação da informação. Duas características fundamentais para esse feito são a esparsidade da imagem e a incoerência entre a base de representação e a base de medida do objeto. Algoritmos de programação linear foram desenvolvidos para reconstruir a imagem original a partir das amostras obtidas, com alto grau de sucesso. O presente trabalho explica o funcionamento de um destes algoritmos etapa por etapa. Exemplos de imagens reconstruídas usando um dos algoritmos propostos fazem parte do presente trabalho.

Palavras-chave: Amostragem, processamento de imagens, recuperação de imagens, sensoriamento compressivo, amostragem compressiva, esparsidade, redução de dimensionalidade.

Abstract

Fabbri Jr, Carlos **Compressive Sensing multispectral images recovering by using single pixel camera and linear programming convex algorithms.** 62 p. Master Thesis – Polytechnic School, University of São Paulo, 2020.

The present work shows a new methodology for the capture of IR, UV and visible images using a significantly smaller number of samples as dictated by the main Shannon-Nyquist sampling theory. This new methodology is based on a revolutionary and innovative theory called compressed sensing or compressive sampling. This theory proposes a new method for capturing the main information of the image or the object being sampled, based on the knowledge that these informations are sparse on a specific information coding base. Two essential characteristics for this feature are sparsity of the image and the incoherence between the coding base and the measurement base. Linear programming algorithms were developed to reconstruct the original image from the samples obtained, with high level of success. The present work explains step by step the working principle of such algorithm. Examples of reconstructed images using the proposed algorithms are included in this work.

Keywords: Sampling, image processing, image recovery, compressed sensing, compressive sampling sparsity, dimensionality reduction.

Lista de ilustrações

Figura 1 – Tipos de problemas em unidades hidro-geradoras. Fonte:(CIGRE, 2003)	18
Figura 2 – Espectro de emissão do corona no ar. Fonte: adaptado de Santos et al(2009)	19
Figura 3 – Resultado da composição de imagem visível e UV na detecção do Corona. (Fonte: Santos et al, 2009)	20
Figura 4 – Exemplo de sinal esparsos, no caso uma imagem bidimensional. (Fonte: (WILLETT, 2011))	23
Figura 5 – Resultado do processo de compressão: Imagem Original (esquerda) e Recuperada (direita). (Fonte: adaptado de “What’s happening in the mathematical sciences”, 2007, (LAUWERS; KANTOR; HOLLIS, 2007))	24
Figura 6 – Processo de sensoriamento compressivo por uma matriz aleatória Gaussiana Θ e uma matriz DCT Ψ . Adaptado de Baraniuk(2007) pelo autor.	26
Figura 7 – Diagrama de funcionamento da câmera de pixel único. Adaptado de Baraniuk (2007) pelo autor.	27
Figura 8 – Uma visão do conjunto de espelhos do modulador espacial de luz, DMD. (Fonte: TI/DLP)	27
Figura 9 – Exemplo de padrões aleatórios gerados pelo DMD. (Fonte: Baraniuk (2007), adaptado pelo autor)	28
Figura 10 – Os scripts em Matlab disponibilizados por Candès et al(2004).	29
Figura 11 – Diagrama de blocos do kit DLP3000, base do protótipo do projeto. Fonte:(TI/DLP)	31
Figura 12 – Phantom recovery experiment. a: Padrão Phantom b: Frequências radiais domínio Fourier c: Recuperação por Mínimos quadrados d: Recuperação pelo método TV (Fonte: Candès e Romberg, 2004).	32
Figura 13 – Operação de multiplicação entre imagens original e matriz aleatória. (Fonte:autor)	33
Figura 14 – Primeira imagem recuperada pelo algoritmo modificado. (Fonte:autor)	33
Figura 15 – Imagem original “Lena” com 256 pixels. (Fonte:autor)	34

Figura 16 – Imagem recuperada pelo algoritmo tvphantom.(Fonte:autor)	34
Figura 17 – Imagem recuperada pelo algoritmo tvqc. (Fonte:autor)	34
Figura 18 – Imagem recuperada usando 500 leituras pelo DMD. (Fonte:autor) . . .	35
Figura 19 – Imagem recuperada usando 1000 leituras pelo DMD. (Fonte:autor) . .	35
Figura 20 – O algoritmo original - bloco comentários. (Fonte: Stuart Gibson (2013))	37
Figura 21 – O algoritmo original - bloco inicialização e matrizPhi. (Fonte: Stuart Gibson (2013))	37
Figura 22 – O algoritmo original - bloco compressão , matriz Theta e soluções. (Fonte: Stuart Gibson (2013))	39
Figura 23 – O algoritmo original - bloco reconstrução das imagens. (Fonte: Stuart Gibson (2013))	39
Figura 24 – O algoritmo original - Resultado da reconstrução da imagem. (Fonte: Stuart Gibson (2013))	40
Figura 25 – O algoritmo original - Diagrama de blocos elaborado. (Fonte: autor) .	40
Figura 26 – O algoritmo desenvolvido - Comentários. (Fonte: autor)	41
Figura 27 – O algoritmo desenvolvido - Inicialização. (Fonte: autor)	42
Figura 28 – O algoritmo desenvolvido - Criar matrizes. (Fonte: autor)	43
Figura 29 – O padrão aleatório criado pela linha 71 do código. (Fonte: autor) . . .	44
Figura 30 – O resultado da leitura da imagem original pelo padrão simulado do DMD. (Fonte: autor)	44
Figura 31 – Uma representação gráfica do processo da leitura e compressão da ima- gem. (Fonte: adaptado de (SCHULZ; VELHO; SILVA, 2009) pelo autor)	45
Figura 32 – O algoritmo desenvolvido - Obter y. (Fonte: autor)	45
Figura 33 – O algoritmo desenvolvido - Solução norma L2. (Fonte: autor)	45
Figura 34 – O algoritmo desenvolvido - Solução por CS. (Fonte: autor)	46
Figura 35 – O algoritmo desenvolvido - Reconstrução. (Fonte: autor)	47
Figura 36 – Imagens produzidas pelo algoritmo ao final de sua execução exibindo as soluções encontradas. (Fonte: autor)	47
Figura 37 – O algoritmo desenvolvido - Finalização do programa. (Fonte: autor) . .	48
Figura 38 – Imagem original, 128 pixels de lado. (Fonte: autor)	50
Figura 39 – Imagem recuperada em 23 segundos, com 500 leituras pelo DMD. (Fonte: autor)	50
Figura 40 – Imagem recuperada, matriz ortogonalizada, 1000 leituras. (Fonte: autor)	50
Figura 41 – Imagem recuperada, matriz não-ortogonalizada, 1000 leituras. (Fonte: autor)	51
Figura 42 – Imagem recuperada, 1500 medidas pelo DMD. (Fonte: autor)	51
Figura 43 – Imagem recuperada pelo algoritmo proposto, 250 medidas pelo DMD, 4,22 seg. (Fonte: autor)	52

Figura 44 – Imagem recuperada pelo algoritmo proposto, 500 medidas pelo DMD, 7,19 seg. (Fonte: autor)	52
Figura 45 – Imagem recuperada pelo algoritmo proposto, 1000 medidas pelo DMD, 19,18 seg. (Fonte: autor)	52
Figura 46 – Imagem recuperada pelo algoritmo proposto, 1500 medidas pelo DMD, 34,16 seg. (Fonte: autor)	53
Figura 47 – Imagem recuperada pelo algoritmo proposto, 2400 medidas pelo DMD, 78,92 seg. (Fonte: autor)	53
Figura 48 – Imagens recuperadas com diferentes tamanhos de amostras com o solver <i>l-1eqpd</i> (Fonte: autor)	53
Figura 49 – Imagens recuperadas com diferentes tamanhos de amostras com o solver <i>l-1eqpdvazio</i> (Fonte: autor)	54
Figura 50 – Imagens recuperadas com diferentes tamanhos de amostras com o solver <i>l-1qclogbarrier</i> . (Fonte: autor)	54
Figura 51 – Imagens recuperadas com diferentes tamanhos de amostras com o solver <i>l-1eqpd</i> . (Fonte: autor)	55
Figura 52 – Imagens recuperadas com diferentes tamanhos de amostras com o solver <i>l-1qclogbarrier</i> . (Fonte: autor)	55

Lista de tabelas

Tabela 1 – Tempos de recuperação das imagens. Fonte: Autor	56
--	----

Lista de siglas

ANEEL *Agência Nacional de Energia Elétrica*

Atom *Atom* (Átomo)

BP *Basis Pursuit* (Procura de Base)

DCT *Discrete Cosine Transform* (Transformada Discreta do Cosseno)

IR Infravermelho

SPC *Single Pixel Camera* (Câmera de Pixel Único)

TV *Total Variation* (Variação Total)

USP Universidade de São Paulo

UV Ultravioleta

Sumário

1	INTRODUÇÃO	15
2	OBJETIVOS E JUSTIFICATIVAS	18
2.1	Justificativas	18
2.2	Objetivos	20
3	SENSORIAMENTO COMPRESSIVO, CÂMERA DE PIXEL ÚNICO E RECUPERAÇÃO DE IMAGENS	22
3.1	Paradigma de amostragem de Shannon-Nyquist	22
3.2	O padrão JPEG/JPEG2000 e esparsidade	23
3.3	Sensoriamento Compressivo	24
3.4	Como funciona o processo de CS - A Câmera de Pixel Único .	26
3.5	Recuperação de imagens por programação linear	28
4	MATERIAIS E MÉTODO DO TRABALHO	30
4.1	Etapas do Trabalho	30
4.2	Detalhamento das etapas do trabalho	30
5	DESENVOLVIMENTO DO TRABALHO	36
5.1	Algoritmo desenvolvido - Comentários	41
5.2	Algoritmo desenvolvido - Inicialização	42
5.3	Algoritmo desenvolvido - Criar matrizes aleatórias e base es- parsidade	43
5.4	Algoritmo desenvolvido - Compressão da imagem, Obter y - o sinal comprimido	43
5.5	Algoritmo desenvolvido - Solução por Norma L2	45
5.6	Algoritmo desenvolvido - Solução por CS	46
5.7	Algoritmo desenvolvido - Reconstrução da imagem original . .	46

6	RESULTADO DAS SIMULAÇÕES	49
6.1	Recuperação com a lógica original	49
6.2	Recuperação com a lógica proposta no trabalho	51
7	CONSIDERAÇÕES FINAIS	57
8	DESENVOLVIMENTO FUTURO E SUGESTÕES	59
	REFERÊNCIAS	60

Introdução

Este trabalho tem por ponto de partida uma proposta de projeto de pesquisa e desenvolvimento junto a *Agência Nacional de Energia Elétrica* (ANEEL), para desenvolver um sistema de monitoramento do efeito Corona em barras estatóricas nas unidades geradoras nas usinas, usando elementos inovadores tais como câmeras de pixel único para a detecção da radiação Ultravioleta (UV), unidade de redirecionamento das radiações detectadas para uma unidade de captação, processamento e a interface de visualização e comunicação para uso do corpo operacional e gerencial da usina. Tal sistema tem como componente inovador o desenvolvimento da unidade de redirecionamento da imagem em UV para um módulo de captura de baixo custo e posterior processamento. A unidade de captura de baixo custo usará metodologia conhecida como CS em conjunto com a *Single Pixel Camera* (Câmera de Pixel Único) (SPC) como forma de detecção da radiação UV esperada.

Essa perspectiva de redução significativa do custo do sistema é um grande incentivador para sua consecução. Essa teoria, “*Compressive Sensing*”, se baseia em uma radical interpretação do paradigma de amostragem do sinal usando a taxa de Nyquist, onde o sinal é lido, amostrado e já comprimido em uma única operação e usando um número muito menor de medidas do que seria necessário segundo o paradigma atual. A carga para a recuperação da imagem é transferida para o processamento da imagem pós-captura, usando algoritmos computacionais de última geração e métodos de programação linear convexa para tal propósito. A presente dissertação concentra-se nos métodos para a captura da imagem e sua posterior reconstrução através de algoritmos computacionais implementados usando o programa Matlab.

Inicialmente o trabalho tinha como objetivo principal o desenvolvimento de um protótipo da unidade de captação das imagens em UV em conjunto com o módulo de processamento e recuperação das imagens capturadas. Uma dificuldade encontrada foi a ausência de um sistema de desenvolvimento associado ao principal componente da unidade de captação, um dispositivo chamado de *Digital Micromirror Device* (DMD). Além

dessa dificuldade também não havia um entendimento claro e completo de como o processo de recuperação das imagens funcionava, como usar o aparato proposto como prova de conceito do método com o DMD ou qual algoritmo usar para tanto.

Por esses motivos o escopo inicial do presente trabalho foi reduzido e modificado para o entendimento dos princípios de funcionamento dos algoritmos usados na recuperação das imagens. Esse novo objetivo seria executado com o uso de simulações do protótipo criadas no programa escolhido, usando os pressupostos adequados e assim podendo avaliar se os resultados obtidos seriam úteis e válidos dentro das hipóteses de trabalho iniciais e se motivariam a continuação dos esforços para futuros trabalhos de pesquisa e desenvolvimento.

O algoritmo usado conseguiria recuperar as imagens originais? Quais as dificuldades, exigências, restrições encontradas? Quanto tempo era necessário para recuperar uma imagem? Seria possível melhorar a qualidade da imagem recuperada obtida? Como melhorá-la? Que tipo de padrão usar no DMD para obter a imagem? Essas foram algumas das dúvidas iniciais que queríamos responder com o desenvolvimento deste trabalho. Encontramos respostas para algumas delas e elas são apresentadas ao longo deste texto.

O presente trabalho está dividido da seguinte maneira: no capítulo 2 definem-se os objetivos e justificativas do trabalho, a dimensão do problema causado pelo efeito corona e os desafios de sua detecção pelo uso de captura de imagens na região do visível e do UV/Infravermelho (IR). Introduz a teoria de Compressive Sensing responsável pela revolucionária e inovadora câmera de pixel único, dispositivo base para esse trabalho de mestrado. Expõe para que problemas a aplicação desenvolvida serve ou pode ser usada.

No capítulo 3 apresentam-se a teoria de sensoriamento compressivo, as propriedades matemáticas necessárias para o funcionamento da teoria e a implementação física da mesma na forma da camera de pixel único, desenvolvida pelo grupo do professor Richard Baraniuk da universidade de Rice, Texas nos EUA em 2004.

No capítulo 4 apresentam-se as etapas do trabalho realizado, uma rápida revisão da bibliografia consultada e uma explicação mais detalhada das dificuldades e desafios encontrados durante a realização do presente trabalho.

No capítulo 5 explicam-se em detalhes a lógica do algoritmo desenvolvido.

No capítulo 6 mostram-se os resultados obtidos com a recuperação de imagens usando o algoritmo no ambiente de simulação escolhido.

No capítulo 7 apresentam-se as considerações finais.

No capítulo 8 sugerem-se linhas de ação futuras para o desenvolvimento do presente trabalho.

Objetivos e Justificativas

2.1 Justificativas

Um problema que ocorre em usinas hidroelétricas de geração de energia relaciona-se a falhas em decorrência do efeito Corona nas barras estatóricas de unidades geradoras. Essas falhas causam a entrada da unidade em manutenção para a correção de falhas na isolação nos enrolamentos do estator e demais componentes relacionados. A detecção do efeito pode ajudar a identificar o momento mais adequado para uma intervenção na unidade geradora antes que os seus efeitos possam prejudicar a mesma. O estudo realizado pelo (CIGRE, 2003) identificou que mais da metade dos problemas de manutenção corretiva em unidades hidro-geradoras eram causados por problemas relacionados a isolação danificada, um dos principais efeitos causado pelo Corona, conforme figura 1.

Segundo (SANTOS et al., 2009), o efeito corona é um tipo de descarga elétrica localizada, resultante da ionização transitória de gases em um sistema de isolamento quando a

Figura 1 – Tipos de problemas em unidades hidro-geradoras. Fonte:(CIGRE, 2003)

tensão aplicada excede um valor crítico aplicado V_0 (kV/cm) dependente das condições atmosféricas do local bem como das dimensões dos condutores envolvidos, da distância entre as fases do sistema de transmissão entre outras.

Os elétrons e íons do gás ionizado, em suas colisões, ganham e liberam energia continuamente tendo como resultado a emissão de ondas de luz. O comprimento de onda dessa luz está relacionado aos elementos que compõem o gás, que no caso do ar é composto de 80% de gás nitrogênio, e então o espectro da luz emitida é em sua maior parte ultravioleta (UV), ou seja, fora da região visível do espectro, conforme mostrado na figura 2.

Figura 2 – Espectro de emissão do corona no ar. Fonte: adaptado de Santos et al(2009)

A detecção do efeito corona se faz por inspeção com câmeras que operam nessa região do espectro. Um exemplo de aplicação é o exemplo da concessionária Furnas apresentado no estudo de Santos et al (2009), que recomenda o uso de câmeras de duplo espectro - visível e UV, e as imagens combinadas ajudam a identificação do exato local de ocorrência do fenômeno, conforme figura 3. Esse estudo concentra seu foco nos efeitos do Corona em ambientes abertos e que sofrem interferência da luz solar na sua caracterização. A aplicação do presente estudo concentra-se em ambientes internos à unidade geradora e livres, portanto, da interferência da luz solar citada no estudo de (SANTOS et al., 2009).

Uma justificativa para o presente trabalho relaciona-se com o custo elevado dos equipamentos de inspeção do Corona por UV. Diferentemente das câmeras digitais presentes em qualquer produto de consumo final, onde os dispositivos CCD (Charged Couple Devices) embutidos usam sensores que operam dentro da faixa visível do espectro eletromagnético, as câmeras de detecção na faixa do UV usam sensores apropriados a essa faixa e não desfrutam das vantagens econômicas de produção em massa de seus equivalentes citados acima. Esse é o principal motivo de seu alto custo.

Uma opção que reduzisse o número de sensores especiais para a adequada operação da câmera de detecção por UV seria muito bem recebida, já que como resultado seu custo seria também reduzido.

A proposição de uma nova teoria de amostragem e captura de sinais, feita por (CANDÈS; ROMBERG; TAO, 2006) e intitulada de *Compressive Sensing* ou *Compressed Sensing* ou

Figura 3 – Resultado da composição de imagem visível e UV na detecção do Corona. (Fonte: Santos et al, 2009)

ainda *Compressive Sampling*, e posteriormente implementada fisicamente em um dispositivo chamado de “*Single Pixel Cam*” (SPC), ou câmera de pixel único pelo grupo dos professores Baraniuk e Kelly, da universidade de Rice no Texas, EUA em 2004, trouxe uma nova linha de ação e pesquisa para resolver o problema inicial do alto custo das soluções de detecção do UV com as câmeras tradicionais disponíveis no mercado. Segundo essa teoria, podemos capturar a informação presente em uma imagem efetuando um número muito menor de amostras da mesma, ou no caso das imagens do efeito Corona, usando um número muito menor de sensores para tal feito. Essa possibilidade de efetuar a captura do efeito Corona mas agora dependendo de um número menor de sensores e consequentemente com custo reduzido é o principal motivador do presente trabalho.

Como prova da capacidade de detecção das imagens usando um número muito menor de sensores, foi desenvolvida a SPC, que utiliza um único sensor ou pixel. Aliada a ele foi proposta a utilização de um novo dispositivo para efetuar as leituras, um modulador espacial de luz, conhecido como DMD e produzido pela empresa Texas Instruments dos EUA.

2.2 Objetivos

Com base na revisão da literatura disponível, já havia um entendimento inicial de que a metodologia de CS consome tempo na recuperação das imagens assim como os métodos tradicionais de captura de imagens naturais, mas os motivos de uso do tempo são diferentes em cada abordagem. Os métodos tradicionais de captura e processamento de imagens usam e consomem o tempo durante a compressão da imagem original captada pelos sensores usados, processo este posterior à captura inicial. Nosso método usa ou consome tempo durante o processo da captura inicial em função do dispositivo usado nele, o DMD e seu modo de operação e também pelo algoritmo de recuperação de imagens usado posteriormente à captura inicial.

Alguns efeitos físicos possuem características dinâmicas transientes ou temporárias, como o efeito Corona enquanto outros efeitos possuem características mais estáticas ou com menos dinamismo tais como os efeitos térmicos ou do IR. Um objetivo geral é entender

para quais destes efeitos o algoritmo proposto pode ser usado, em função dos tempos de recuperação obtidos.

O objetivo mais geral desse trabalho é entender e desenvolver o processo de recuperação das imagens criadas por uma câmera de pixel único adaptada à captura de imagens do efeito Corona, na faixa do ultravioleta (UV), ou adaptada para outras faixas de operação como o IR. Esse objetivo será obtido com o uso de simulação das condições reais, com um programa de simulação escolhido e com o uso de algoritmos existentes para o processo da recuperação, por conta das limitações encontradas já descritas na introdução do trabalho.

Os objetivos específicos são:

Desenvolver o software - algoritmo para recuperação de imagens produzidas pelo protótipo simulado.

Entender o funcionamento do algoritmo escolhido e que problemas podem ser resolvidos com o uso do mesmo.

Apresentar o funcionamento do algoritmo usado na recuperação das imagens.

Sensoriamento Compressivo, Câmera de pixel único e Recuperação de imagens

3.1 Paradigma de amostragem de Shannon-Nyquist

Um ponto de partida para compreender a teoria de sensoriamento compressivo é o entendimento que os processos usados nos sistemas digitais para a captura e imediata compressão dos sinais mostraram-se ao longo dos anos intrinsecamente ineficientes. O teorema de amostragem de Shannon-Nyquist, princípio fundamental que embasou o modelo *de facto* para a conversão analógico para digital, usado durante 50 anos não leva em conta a “estrutura” inerente ao sinal. Ele propõe, para sinais limitados em frequência, a amostragem pelo dobro da sua frequência máxima para que sejam completamente recuperados posteriormente. A recuperação do sinal usa como fundamento a interpolação linear de funções “sinc” adequadas ao sinal. Com o avanço da tecnologia digital, tornou-se cada vez mais desafiante acompanhar a complexidade e a quantidade dos dados disponíveis. A enorme quantidade de dados disponíveis não facilita a sua transmissão pelos canais digitais, que possuem limites quanto à sua capacidade de transporte de dados e, portanto, tornou-se prática comum a compressão dos dados digitais antes de seu envio pelas redes de comunicação.

A compressão dos dados torna clara a existência de redundância interna nos sinais, fato explorado pelos padrões de compressão desenvolvidos. Assim foi sendo compreendido pelos cientistas de que a grande maioria dos sinais produzidos e utilizados pelo homem são, na realidade, esparsos no sentido de possuir em sua composição interna, ou estrutura, poucos valores ou coeficientes que concentram a energia do mesmo e que podem ser identificados. (CANDÈS; ROMBERG; TAO, 2004) bem como (DONOHO, 2004) propuseram um nova teoria que revolucionou os processos de amostragem de sinais e criou um novo paradigma pelo qual obtém-se diretamente a estrutura do sinal com um número muito menor de medidas do mesmo. Esta teoria conduziu à técnica conhecida como Sensoriamento Compressivo (CS - do inglês Compressive Sensing).

3.2 O padrão JPEG/JPEG2000 e esparsidade

O exemplo mais citado para justificar os conceitos presentes em CS é o padrão de compressão JPEG, baseado no uso da *Discrete Cosine Transform* (Transformada Discreta do Cosseno) (DCT) como base para seu funcionamento. Esse padrão reconhece que a maioria das imagens associadas ao ambiente humano são redundantes do ponto de vista de seu “conteúdo de informação” ou seja, existem muitos coeficientes (ou pixels, no caso de imagens em duas dimensões) repetidos ao longo de sua estrutura espacial. Essa característica fundamental presente no sinal é definida como “esparsidade” e o sinal é considerado “esparso”, pois a energia presente no sinal está concentrada somente em uns poucos dos seus elementos internos. Um exemplo de sinal esparso encontra-se na figura 4.

Figura 4 – Exemplo de sinal esparso, no caso uma imagem bidimensional. (Fonte: (WILLETT, 2011))

Essa imagem de 400 pixels de lado é 344-pixels esparsa.

A maioria de seus pixels é de intensidade zero (preto)

Adaptado de R.Willett, aula 1, IAS (2011)

O entendimento dessa estrutura permite o cálculo de todos os coeficientes da DCT para cada pixel da imagem, a seleção dos maiores coeficientes da mesma e o descarte de todos os demais no processo de compressão. Podemos entender esse processo como uma redução de dimensionalidade do sinal original, esparso com relação ao seu conteúdo de informação, e o sinal comprimido resultante da transformação aplicada ao mesmo. O processo inverso de descompressão da imagem ou do sinal e sua recuperação a partir do sinal comprimido sofre pouca influência da compressão e a imagem ou sinal é praticamente idêntica ao original, considerada pelo leitor comum.

O padrão JPEG2000 muda da transformada usada do DCT para a transformada Wavelet (MALLAT, 1999), com um ganho significativo na compressão dos sinais e mantendo a percepção de fidelidade ao original. Um exemplo dessa fidelidade ao original ou da qualidade da compressão obtida é mostrado na figura 5 a seguir.

A imagem original está exibida na parte esquerda e a imagem recuperada está exibida na parte direita.

Figura 5 – Resultado do processo de compressão: Imagem Original (esquerda) e Recuperada (direita). (Fonte: adaptado de “What’s happening in the mathematical sciences”, 2007, (LAUWERS; KANTOR; HOLLIS, 2007))

3.3 Sensoriamento Compressivo

(CHEN; DONOHO; SAUNDERS, 1998) expõem um método para representar sinais esparsos ou compressíveis através da combinação linear do que eles chamam de *Atom* (Átomo) (Atom). Sua ideia é de que um sinal complexo em sua base de representação pode ser decomposto por uma outra representação esparsa do mesmo em outra base. O método descrito por eles para encontrar os coeficientes dessa combinação linear ficou conhecido como *Basis Pursuit* (Procura de Base) (BP) e usa um algoritmo de programação linear convexo minimizando a norma l_1 do sinal em estudo. Logo em seguida, (CANDÈS; ROMBERG; TAO, 2006) propõe condições estruturais para tal processo de transformação. Define-se uma propriedade do sinal, chamada de RIP (do inglês Restricted Isometry Property), como a condição de reversibilidade necessária para obter a recuperação da imagem.

Duas são as condições fundamentais para que a teoria de sensoriamento compressivo funcione: esparsidade do sinal e incoerência entre as bases de representação do mesmo e de sua amostragem. As duas condições são explicadas a seguir:

Esparsidade

Qualquer sinal em um espaço n -dimensional pode ser decomposto em termos da base adequada nesse espaço. Seja x um sinal unidimensional, de comprimento finito, com valores reais e que pode ser visualizado como um vetor-coluna $N \times 1$ em R^N com seus

elementos $x[n]$, com $n= 1,2,3,...,N$ sua representação na base $\{\psi_i\}_{i=1}^N$. O sinal pode ser representado na base ψ , composta por $\psi_1, \psi_2, \psi_3, \dots, \psi_N$ da seguinte maneira

$$x = \psi c \quad \text{ou} \quad x = \sum_{i=1}^N c_i \psi_i \quad (1)$$

onde c é o vetor-coluna dos coeficientes ponderados de x . O sinal x é K esparso se ele é uma combinação linear de apenas K vetores da base. Isso significa que somente K dos c_i coeficientes é diferente de zero e $(N - K)$ coeficientes são zero.

O sinal será considerado compressível se na representação da equação (1) ele possui apenas alguns coeficientes de valor alto e muitos coeficientes de valor próximos a zero. Apesar de procurar na literatura valores numéricos que pudessem quantificar a alegação anterior de definição da compressibilidade do sinal, não foi encontrado nenhum número associado à definição.

Um dos pontos fundamentais da teoria é o entendimento de que o sinal original pode ser representado por outro sinal, “esparso” em outro sistema de coordenadas ou base. Neste espaço, são necessários menos amostras do sinal para sua completa caracterização. Essa é uma vantagem do método e da teoria, obter a informação do sinal efetuando somente um número adequado de medidas do mesmo na base de esparsidade. Para identificar quais valores da imagem/sinal são importantes, usa-se da aleatoriedade da matriz de medidas para encontrar tais elementos.

Incoerência

Outro ponto fundamental da nova teoria é a compreensão de que essas duas bases, a da esparsidade do sinal e a da base de medida devem ser incoerentes entre si, ou seja, com o mínimo de possibilidade de se representar qualquer sinal de uma na outra base. Considere $M < N$ medidas lineares de x e um conjunto de funções de teste $\{\varphi_m\}_{m=1}^M$ tais que $y_m = \langle x, \varphi_m \rangle$. Empilhando as medidas y_m em um vetor-coluna y de dimensões $M \times 1$ e as funções de teste φ_m^T como linhas em uma matriz de medidas Θ de dimensões $M \times N$, podemos escrever:

$$y = \Phi x = \Phi \Psi c = \Theta c \quad (2)$$

A medida de coerência entre a base de medida Φ e a base de representação esparsa Ψ pode ser dada pela equação:

$$\mu(\Theta, \Psi) = \sqrt{N} \max_{k,j} |\langle \varphi_k, \psi_j \rangle| \quad (3)$$

Essa equação mede a correlação entre quaisquer dois elementos de Θ e de Ψ . Em sensoriamento compressivo estamos interessados em pares de bases que tenham baixos

valores de coerência. (BARANIUK, 2007) representou essa transformação muito apropriadamente em formato gráfico na figura 6:

Figura 6 – Processo de sensoriamento compressivo por uma matriz aleatória Gaussiana Θ e uma matriz DCT Ψ . Adaptado de Baraniuk(2007) pelo autor.

(CANDÈS; ROMBERG; TAO, 2006), (ROMBERG, 2008) e (DONOHO, 2004) afirmam que matrizes aleatórias são altamente incoerentes com qualquer base fixa Ψ . Essa característica impulsiona o uso de transformadas rápidas do tipo Walsh, Hadamard ou Noiselets.

3.4 Como funciona o processo de CS - A Câmera de Pixel Único

A fim de obter em uma única operação os coeficientes mais importantes na base de representação comprimida como prometem Candès, Tao e Romberg, o processo de amostragem é transformado em uma operação física entre o sinal que se pretende capturar e uma função de teste ou de medida apropriada para tanto. Esse procedimento foi implementado fisicamente no dispositivo conhecido como Single Pixel Cam (SPC) desenvolvido pelo grupo dos professores Kevin Kelly e Richard Baraniuk da universidade de Rice em 2006 (TAKHAR et al., 2006) e exemplificado na figura 7.

A SPC é um computador óptico composto de lentes, do modulador espacial de luz DMD, do fotodetector e de um conversor analógico/digital. A recuperação da imagem é feita por um computador externo a esse conjunto. Os principais componentes são o modulador de luz e o fotodetector.

O modulador espacial de luz, chamado de DMD foi desenvolvido pela empresa Texas Instruments a partir de 1987. Ele é composto por uma base de células de memória estática CMOS sobre as quais é adicionado um conjunto de espelho e barra de torção para cada uma das células de memória (figura 9). Cada célula opera de forma independente e permite a orientação do espelho acoplado para duas possíveis direções dependendo do

Figura 7 – Diagrama de funcionamento da câmera de pixel único. Adaptado de Baraniuk (2007) pelo autor.

valor do sinal aplicado. Se o sinal aplicado corresponde ao valor binário “0” o espelho orienta-se numa direção. Aplicando o valor oposto correspondendo ao valor binário “1” o espelho orienta-se em outra direção, oposta a anterior (figura 8).

Figura 8 – Uma visão do conjunto de espelhos do modulador espacial de luz, DMD. (Fonte: TI/DLP)

A SPC funciona da seguinte maneira: a imagem a ser capturada é exposta ao dispositivo DMD várias vezes enquanto a cada leitura o padrão de organização dos microscópicos espelhos do DMD é alterado com um padrão aleatório diferente, conforme figura 9.

Como o DMD possui os espelhos que podem ser dirigidos para um de dois lados de operação, os padrões aleatórios definem quanto da imagem sendo capturada será percebida e direcionada ao outro componente do conjunto: o fotodetector. O fotodetector recebe portanto, somente as partes da imagem que foram direcionadas pelo DMD naquela configuração. O resultado na saída do sensor é um nível de tensão que é proporcional a essa quantidade de informação capturada pelo DMD e repassada ao fotodetector. O

Figura 9 – Exemplo de padrões aleatórios gerados pelo DMD. (Fonte: Baraniuk (2007), adaptado pelo autor)

processo é repetido várias vezes até obter-se a quantidade necessária para a recuperação da imagem, feita posteriormente e fora do dispositivo SPC.

3.5 Recuperação de imagens por programação linear

A tentativa de recuperar o sinal x a partir de sua representação esparsa c mostrou-se muito interessante. Inicialmente foi usada a recuperação com base na minimização da quase-norma $\|x\|_0$ que é computacionalmente muito complexa para não dizer inexequível. Como dito anteriormente o processo de sensoriamento compressivo pode ser entendido como uma redução de dimensionalidade na passagem de x para y e, de modo geral, há perda de informação. O tradicional método dos mínimos quadrados falha com alta probabilidade na recuperação. Ao invés do método dos mínimos quadrados foi proposto por (CANDÈS; TAO, 2006) o uso da otimização da norma l_1 :

$$\hat{c} = \underset{c}{\operatorname{argmin}} \|c\|_{l_1} \quad \text{sujeito a} \quad \Theta c = y \quad (4)$$

Com isso os autores demonstraram a possibilidade de reconstrução de um sinal K esparsos com alta probabilidade usando $M \geq O(K \log(\frac{N}{K}))$ medidas aleatórias. Esse método de minimização da norma l_1 é conhecido como Basis Pursuit (BP) e pode ser resolvido por métodos baseados em *Total Variation* (Variação Total) (TV), gananciosa (greedy) e estocástica.

(CANDÈS; ROMBERG; TAO, 2004) publicaram um conjunto de algoritmos em Matlab que resolve diferentes tipos de problemas de minimização l_1 , chamado de pacote “ l_1 magic” e que está disponível na internet. (figura 10)

Atualmente mais de 50 programs do tipo “solver” estão disponíveis em vários locais na internet à disposição para serem utilizados em recuperação de imagens, como em (HAYASHI; NAGAHARA; TANAKA, 2013). (CARRON, 2011) mantém um site sobre o status atual das técnicas relacionadas ao tema do sensoriamento compressivo e que relaciona os “solvers” mais atualizados.

Figura 10 – Os scripts em Matlab disponibilizados por Candès et al(2004).

<code>cgsolve</code>	Solves $Ax = b$, where A is symmetric positive definite, using the Conjugate Gradient method.
<code>l1dantzig_pd</code>	Solves (P_D) (the Dantzig selector) using a primal-dual algorithm.
<code>l1decode_pd</code>	Solves the norm approximation problem (P_A) (for decoding via linear programming) using a primal-dual algorithm.
<code>l1eq_pd</code>	Solves the standard Basis Pursuit problem (P_1) using a primal-dual algorithm.
<code>l1qc_logbarrier</code>	Barrier (“outer”) iterations for solving quadratically constrained ℓ_1 minimization (P_2) .
<code>l1qc_newton</code>	Newton (“inner”) iterations for solving quadratically constrained ℓ_1 minimization (P_2) .
<code>tv_dantzig_logbarrier</code>	Barrier iterations for solving the TV Dantzig selector (TV_D) .
<code>tv_dantzig_newton</code>	Newton iterations for (TV_D) .
<code>tveq_logbarrier</code>	Barrier iterations for equality constrained TV minimization (TV_1) .
<code>tveq_newton</code>	Newton iterations for (TV_1) .
<code>tvqc_logbarrier</code>	Barrier iterations for quadratically constrained TV minimization (TV_2) .
<code>tvqc_newton</code>	Newton iterations for (TV_2) .

Materiais e Método do Trabalho

4.1 Etapas do Trabalho

Esse trabalho foi dividido em etapas abrangendo desde a pesquisa e estudo para a assimilação dos conceitos da teoria de CS e do funcionamento e operação do DMD até a pesquisa e desenvolvimento dos algoritmos para recuperação das imagens. A principal contribuição do presente trabalho de mestrado é a adaptação do algoritmo de simulação usado para recuperar as imagens à estrutura física matricial do dispositivo DMD, preparando seu uso para as implementações experimentais de recuperação de imagens a serem realizadas futuramente. Foram usados para as simulações o software Matlab, os pacotes de scripts baixados da internet, os demais scripts obtidos também da internet, rodando em um notebook com o sistema operacional Windows, 8GB de memória RAM e processador Core 2 Duo rodando com clock de 2.4 gigahertz.

As etapas do presente trabalho são:

Etapa 1: Pesquisa e estudo da teoria Sensoriamento Compressivo

Etapa 2: Especificações do projeto do protótipo da SPC UspIEAv

Etapa 3: Simulações de imagens recuperadas

Etapa 4: Desenvolvimento e adaptação de algoritmos para simulação com o DMD

4.2 Detalhamento das etapas do trabalho

Etapa 1

Foram pesquisadas as referências bibliográficas existentes na teoria de sensoriamento compressivo (revisão da literatura) e realizados os estudos para sua aplicação. A teoria de CS abriu um leque de possibilidades para o desenvolvimento de aplicações em diversos campos de estudo e pesquisa como em (WU et al., 2011), detecção de bordas em (RIZA; MARRACCINI; BAXLEY, 2012), processamento de imagens em (WANG et al., 2016), imagens em (VAZ et al., 2020), imagem em alta velocidade em (LIMA; MIOSSO; FARIAS, 2017), imagens em THz em (HU et al., 2020), aplicações multiespectrais em (CHEN et al.,

2011), em (GALVIS; ARGUELLO; ARCE, 2014) e em (BRORSSON, 2018), aplicações em criptografia em (ZHANG et al., 2019) e (XIAO; ZHOU; CHEN, 2019), processamento de sinais em (HANNAK et al., 2018) e (AHMED; ROMBERG, 2020), filtros ópticos em (RUEDA; ARGUELLO; ARCE, 2015), microscopia em (YU et al., 2015), avaliação de tamanho de partículas em amostras em (ZHOU et al., 2016) e holografia óptica em (AQIAN et al., 2017) entre outras.

O autor simultaneamente reforçou sua capacitação nas ferramentas matemáticas como programação linear convexa, otimização por algoritmos e processamento de imagens em Matlab.

Etapa 2

Não há disponível no mercado um módulo de desenvolvimento específico para a SPC usando o dispositivo DMD. Existe uma plataforma de desenvolvimento do fabricante focada em usos com objetivos distintos da detecção do corona chamada de Lightcrafter. Iniciou-se a definição dos diagramas de blocos da SPC e dos respectivos diagramas lógicos para o controle dos espelhos do DMD a partir do zero.

O módulo disponível Lightcrafter visa usos relacionados à projeção de imagens substituindo dispositivos com a tecnologia LCD (de Liquid Crystal Display) e uma boa parte do circuito destina-se aos leds e demais elementos da projeção da imagem, conforme figura 11.

Figura 11 – Diagrama de blocos do kit DLP3000, base do protótipo do projeto.
Fonte:(TI/DLP)

Para a nossa aplicação, onde o foco principal é o controle do dispositivo DMD na geração de padrões aleatórios, usar o módulo Lightcrafter significaria ter de desativar ou

desligar uma grande parte do circuito existente no módulo. Considerando essa informação definiu-se que seria mais adequado desenvolver um módulo próprio.

Avaliando as necessidades associadas a esse desenvolvimento com a disponibilidade de tempo e recursos para tanto, entendeu-se que esta parte desta etapa seria modificada e reorientada optando-se por reduzir o escopo do projeto inicial, focar no processamento da imagem pós-captura pelo DMD e na recuperação das imagens com o uso dos algoritmos de programação linear.

Etapa 3

Nas referências bibliográficas foi encontrado um pacote de scripts do software Matlab, chamado *l1magic* (CANDÈS; ROMBERG, 2004) contendo 7 programas para a experimentação com a recuperação de imagens pelo sensoriamento compressivo e um exemplo da eficiência destes programas é exibido na figura 12 que apresenta o caso conhecido como Phantom recovery experiment. Foram estudados os scripts e realizada a adaptação do mesmo para nosso propósito obtendo alguns bons resultados iniciais.

Figura 12 – Phantom recovery experiment. a: Padrão Phantom b: Frequências radiais domínio Fourier c: Recuperação por Mínimos quadrados d: Recuperação pelo método TV (Fonte: Candès e Romberg, 2004).

Com base no algoritmo obtido e no entendimento do processo a ser usado para capturar e recuperar a imagem, foi utilizada como imagem original uma figura de 128 x 128 pixels com as iniciais das duas instituições participantes do projeto, a Universidade de São Paulo (USP) e o IEAv. Num primeiro momento o autor usou uma operação de multiplicação entre a imagem original e o padrão aleatório que é produzido pelo DMD. A figura 13 mostra este processo.

Usando os scripts fornecidos no pacote *l1-magic* e alterando apenas a imagem inicial destes scripts pela imagem original criada para nosso projeto obtivemos a primeira imagem recuperada, exibida na figura 14.

Além do script usando o método de total variation (TV) efetuaram-se recuperações de outra imagem original exibida na figura 15 usando dois outros algoritmos do pacote *l1-magic*, “*tvphantom*” e “*tvqc*”, obtendo-se as imagens recuperadas nas figuras 16 e 17 abaixo:

Figura 13 – Operação de multiplicação entre imagens original e matriz aleatória.
(Fonte:autor)

Figura 14 – Primeira imagem recuperada pelo algoritmo modificado. (Fonte:autor)

Os estudos e experimentações concentraram-se nessa fase principalmente no tempo de recuperação total da imagem, objetivando os algoritmos com os menores tempos totais. A aplicação do algoritmo em campo pressupõe um tempo de resposta e de reconstrução adequado à dinâmica do local e ao contexto envolvido, que fica facilitado pelos menores tempos de processamento. Outros scripts mais avançados estão disponíveis para posterior análise, no site de (CARRON, 2011).

Figura 15 – Imagem original “Lena” com 256 pixels. (Fonte:autor)

Figura 16 – Imagem recuperada pelo algoritmo tvphantom.(Fonte:autor)

Figura 17 – Imagem recuperada pelo algoritmo tvqc. (Fonte:autor)

Etapas 4

Dos estudos e experimentações realizados na etapa anterior, escolheu-se um algoritmo de recuperação baseado no método Basis Pursuit que foi adaptado ao modelo da SPC. Foram obtidos os primeiros resultados satisfatórios de recuperação da imagem original UspIEAv. O algoritmo foi sendo modificado de maneira que seu tempo foi sendo reduzido

gradativamente.

Nessa etapa o algoritmo ainda estava adaptado para vetorizar a imagem original antes de entregar o vetor ao módulo de recuperação da imagem. Esse procedimento é adequado quando a imagem é gerada por dispositivos intrinsecamente seriais, por exemplo CCDs. No algoritmo proposto esse mecanismo de preparação da imagem foi ajustado e adaptado para a configuração física do dispositivo DMD, uma matriz de micro-espelhos.

Os primeiros resultados de recuperação de imagens usando o algoritmo modificado pelo autor são exibidos nas figura 18 e 19. As figuras tem 3 imagens cada sendo que, a imagem original é a mais à esquerda, a imagem recuperada pelo método dos mínimos quadrados (least squares) é a do centro e a imagem recuperada pelo método Basis Pursuit é a mais à direita.

Figura 18 – Imagem recuperada usando 500 leituras pelo DMD. (Fonte:autor)

Figura 19 – Imagem recuperada usando 1000 leituras pelo DMD. (Fonte:autor)

Desenvolvimento do trabalho

Buscando identificar, durante o processo de pesquisa, algoritmos que fossem mais eficientes e que apresentassem boas possibilidades de aplicação ao problema em estudo, encontrou-se um algoritmo simples e direto desenvolvido por (GIBSON, 2013) que implementa os passos necessários para a captura, processamento e recuperação da imagem original.

O autor modificou o script incorporando uma outra interpretação da função de geração de números aleatórios, “randi”, do Matlab e alterou-se a forma como o script calcula a transformada DCT na medição da imagem e também na sua recuperação. Nas figuras de 20 a 24, a seguir, são reproduzidas as transcrições do algoritmo original.

Figura 20 – O algoritmo original - bloco comentários. (Fonte: Stuart Gibson (2013))

```

% compressed sensing example
%
% __DESCRIPTION__
% MATLAB implementation of compressive sensing example as described in R.
% Baraniuk, Compressive Sensing, IEEE Signal Processing Magazine, [118],
% July 2007. The code acquires 250 averaged random measurements of a 2500
% pixel image. We assume that the image has a sparse representation in the
% DCT domain (not very sparse in practice). Hence the image can be
% recovered from its compressed form using basis pursuit.
%
% __DEPENDENCIES__
% Requires the MATLAB toolbox l_1-MAGIC: Recovery of Sparse Signals via
% Convex
% Programming v1.11 by J. Candes and J. Romberg, Caltech, 2005.
%
% __VARIABLES__
% x = original signal (nx1) y = compressed signal (mx1) Phi = measurement
% matrix (mxn) Psi = Basis functions (nxn) Theta = Phi * Psi (mxn) s =
% sparse coefficient vector (to be determined) (nx1)
%
% __PROBLEM__
% Invert the matrix equation y = Theta * s and therefore recover hat_x as
% k-sparse linear combination of basis functions contained in Psi. Note
% also that y = Phi * x.
%
% __SOLUTION__
% Let Phi be a matrix of i.i.d. Gaussian variables. Solve matrix inversion
% problem using basis pursuit (BP).
%
% __CREATED__
% o By S.Gibson, School of Physical Sciences, University of Kent.
% o 1st May, 2013.
% o version 1.0
% o NOTES: If the max number of iterations exceeds 25, error sometimes
% occurs in lleq_pd function call.
%
% __DISCLAIMER__
% The code below is my interpretation of Baraniuk's compressed sensing
% article. I don't claim to be an authority on the subject!

```

Figura 21 – O algoritmo original - bloco inicialização e matrizPhi. (Fonte: Stuart Gibson (2013))

```

% __INPUT IMAGE__
clear, close all clc
A = imread('cameraman.tif');
A = A([50:99],[50:99]);
x = double(A(:));
n = length(x);

% __MEASUREMENT MATRIX__
m = 250; % NOTE: small error still present after increasing m to 1500;
Phi = randn(m,n);
% __ALTERNATIVES TO THE ABOVE MEASUREMENT MATRIX__
% Phi = (sign(randn(m,n))+ones(m,n))/2; % micro mirror array (mma) e.g.
single
% pixel camera Phi = orth(Phi)'; % NOTE: See Candes & Romberg, 11
% magic, Caltech, 2005.

```

Na figura 25 apresenta-se o diagrama de blocos do algoritmo desenvolvido. Na parte superior apresentam-se os 8 blocos de código nos quais se divide o algoritmo, cada um responsável por efetuar uma etapa do processo da captura até a reconstrução. Esse diagrama mostra a ordem dos blocos de códigos, do início até o fim, passando pelos comentários, a inicialização, a criação da matriz Phi, a compressão e a obtenção de y (o sinal comprimido), a criação da matriz Theta do domínio da esparsidade, a obtenção da solução pela norma L2, a obtenção da solução pelo método CS - Basis Pursuit e finalmente a reconstrução da imagem original.

Na parte inferior da figura 25, encontra-se a representação gráfica das etapas do algoritmo e efetuou-se o uso de setas para mostrar qual elemento dela faz conexão com o respectivo bloco do algoritmo na parte superior. Assim há uma indicação do bloco que produz a variável y , uma indicação do bloco que cria a matriz Phi, uma indicação do bloco que cria a matriz Theta e uma indicação do bloco que obtém o sinal esparso recuperado, c .

Em seguida o algoritmo, desenvolvido e adaptado pelo autor, e sua lógica de funcionamento são explicados em detalhes. O processo de captura e recuperação das imagens pode ser dividido em 3 macro-etapas, a coleta das amostras, o posterior processamento das mesmas e a reconstrução da imagem obtida. O algoritmo contempla as macro-etapas e as divide em 8 “blocos” de processamento ilustrados na figura 25. O algoritmo original foi modificado progressivamente para ajustar o processo de captura da imagem à conformação física do dispositivo DMD, que é uma matriz de micro-espelhos. Definir uma matriz de valores “0” e “1” aleatórios como ponto de partida adapta-se perfeitamente com a matriz física dos micro-espelhos do dispositivo DMD.

Figura 22 – O algoritmo original - bloco compressão , matriz Theta e soluções. (Fonte: Stuart Gibson (2013))

```
% __COMPRESSION__
y = Phi*x;

% __THETA__
% NOTE: Avoid calculating Psi (nxn) directly to avoid memory issues.
Theta = zeros(m,n);
for ii = 1:n;
    ii;
    ek = zeros(1,n);
    ek(ii) = 1;
    psi = idct(ek)';
    Theta(:,ii) = Phi*psi;
end

% __L2 NORM SOLUTION__ s2 = Theta\y; %s2 = pinv(Theta)*y
s2 = pinv(Theta)*y;

% __BP SOLUTION__
s1 = lleq_pd(s2,Theta,Theta',y,5e-3,20); % L1-magic toolbox
%x = lleq_pd(y,A,A',b,5e-3,32);

% __DISPLAY SOLUTIONS__
plot(s2,'b'), hold
plot(s1,'r.-')
legend('least squares','basis pursuit')
title('solution to y = \Theta s')
```

Figura 23 – O algoritmo original - bloco reconstrução das imagens. (Fonte: Stuart Gibson (2013))

```
% __IMAGE RECONSTRUCTIONS__
x2 = zeros(n,1);
for ii = 1:n;
    ii;
    ek = zeros(1,n);
    ek(ii) = 1;
    psi = idct(ek)';
    x2 = x2+psi*s2(ii);
end

x1 = zeros(n,1);
for ii = 1:n;
    ii;
    ek = zeros(1,n);
    ek(ii) = 1;
    psi = idct(ek)';
    x1 = x1+psi*s1(ii);
end

figure('name','Compressive sensing image reconstructions')
subplot(1,3,1), imagesc(reshape(x,50,50)), xlabel('original'), axis image
subplot(1,3,2), imagesc(reshape(x2,50,50)), xlabel('least squares'), axis image
subplot(1,3,3), imagesc(reshape(x1,50,50)), xlabel('basis pursuit'), axis image
colormap gray
```


Figura 24 – O algoritmo original - Resultado da reconstrução da imagem. (Fonte: Stuart Gibson (2013))

Figura 25 – O algoritmo original - Diagrama de blocos elaborado. (Fonte: autor)

5.1 Algoritmo desenvolvido - Comentários

O primeiro passo no desenvolvimento do algoritmo foi a introdução do bloco de comentários, informando ao usuário o propósito do código, as variáveis utilizadas, suas dependências e especificidades como dimensões, bem como o contexto de seu uso. Informa também a necessidade do código do pacote de scripts *l1-magic*, figura 26.

Figura 26 – O algoritmo desenvolvido - Comentários. (Fonte: autor)

```

1 % Código para Recuperação de Imagens por Sensoriamento Compressivo usando a
2 % configuração física do dispositivo de DMD.
3 %
4 % _DESCRIÇÃO_
5 % Adaptação do código usado por Stuart Gibson para implementar o exemplo de
6 % sensoriamento compressivo proposto por R. Baraniuk em Compressive Sensing
7 % em IEEE Signal Processing Magazine, Julho 2007.
8 % A operação física entre os padrões aleatórios e a imagem original foi
9 % usada para "ler" o sinal original implementando o uso do dispositivo DMD
10 % na prática.
11 %
12 % _DEPENDÊNCIAS_
13 % Requer o conjunto de scripts l1-magic: Recovery of Sparse Signals via
14 % convex programming, criado por E.Candès e J. Romberg, Caltech , 2005.
15 % Usa a estrutura de pastas propostas pelos autores do pacote l1-magic.
16 %
17 % _VARIÁVEIS_
18 % A = imagem original(a x b);
19 % n = tamanho de A vetorizado (vetor coluna);
20 % y = sinal comprimido (mx1);
21 % matrizPhi(m x n) = matriz de medidas aleatórias;
22 % mdmd = padrão aleatório individual, extraído de matrizPhi;
23 % Theta(m x n) = Matriz base de representação Esparsa;
24 % s(n x 1) = vetor de coeficientes esparsos a ser recuperado;
25 % x1(n x 1) = vetor de coeficientes do sinal recuperado pelo algoritmo.
26 %
27 % criado em 28/08/2019
28 % por Carlos Fabbri Jr, Escola Politécnica da USP, Depto PEA
29 % Modificado em 06/01/2020

```

5.2 Algoritmo desenvolvido - Inicialização

Prosseguindo no desenvolvimento do algoritmo, define-se nesse bloco o procedimento de inicialização do ambiente global do mesmo. Utiliza-se a mesma estrutura de pastas com os códigos dos “solvers”, assim como o local para armazenar os dados e demais ajustes existentes no pacote *l1-magic* citado em (CANDÈS; ROMBERG, 2004), otimização, medidas e dados. Nesse bloco do código apagam-se as antigas variáveis e a memória pré-existente e demais variáveis do programa de modo a inicializar o processamento a partir de um ponto neutro. Define-se também a imagem original e formata-se a mesma para uso posterior. Prealoca-se espaço de memória de execução para algumas variáveis que serão usadas mais adiante no código, cuja transcrição é mostrada na figura 27.

Figura 27 – O algoritmo desenvolvido - Inicialização. (Fonte: autor)

```

33
34 %% ===== Inicialização =====%%
35 path(path, './Optimization');           % pasta Otimização
36 path(path, './Measurement');           % pasta Medidas
37 path(path, './Data');                   % pasta Dados
38 % mesma estrutura de pastas usadas pelo pacote l1-magic
39 %
40 tic                                     % inicializar o contador tempo
41 % Para analisar o desempenho de cada algoritmo/solver
42
43 clear, close all; clc;                  % apagar variáveis ambiente
44
45 A = imread('original128peb.bmp');       % Ler a imagem original
46 A = A(40:79,20:79);                    % selecionar área da imagem
47 [a, b]=size(A);                         % obter as medidas da imagem
48 n = length(double(A(:)));               % vetorizar a imagem
49
50 y=[];                                   % inicializar a matriz de leituras y
51 m=1000;                                % número de amostras, leituras da imagem
52 Theta = zeros(m,n);                     % inicializa a matriz Theta
53 x1 = zeros(n,1);                        % inicializa o vetor x1
54
55 fprintf('Inicialização Efetuada... \n')
56

```

5.3 Algoritmo desenvolvido - Criar matrizes aleatórias e base esparsidade

Continuando o desenvolvimento do algoritmo cria-se nesse bloco a matriz de padrões aleatórios, chamada MatrizPhi, que vai ser fornecida aos micro-espelhos do DMD. Essa matriz será usada posteriormente na etapa de recuperação da imagem original. Esse bloco também prepara a matriz Theta, a matriz do domínio esparsa, que vai ser aplicada à imagem original. Durante muito tempo o autor esforçou-se para entender como obter os valores no domínio esparsa a partir da aplicação dos padrões aleatórios do DMD, sem conseguir implementar esse método. O mérito do algoritmo original de Gibson é precisamente implementar esse método em duas partes, a primeira com a criação da matrizPhi e a segunda implementando esse bloco, que cria a matriz Theta. A adaptação feita por este autor cria diretamente a matriz Theta em uma única linha de código, diferentemente do código original que implementava essa matriz em um loop “for” que consumia um tempo considerável de processamento, como pode ser visto na figura 28.

Figura 28 – O algoritmo desenvolvido - Criar matrizes. (Fonte: autor)

```

57 %%=====Gerar Matrizes - Aleatória e Representação =====%%
58
59 matrizPhi = randi([0 1],m,n);      % Criar a matriz de padrões aleatórios;
60 ... ela será usada depois
61 %Phi = orth(matrizPhi)';
62
63
64 Theta = matrizPhi*dct(eye(n));      % popula a matriz com os valores da dct
65 ... dos padrões aleatórios
66 % No código original, era criada a matriz Psi = dct(eye(n)) em um loop for;
67

```

5.4 Algoritmo desenvolvido - Compressão da imagem, Obter y - o sinal comprimido

Esta parte do algoritmo e do código desenvolvido é a mais importante do trabalho. Dando seguimento no desenvolvimento do algoritmo, esse bloco efetua a compressão do sinal original, capturada pelas leituras feitas com os padrões do DMD aplicados à imagem original.

Na linha 71 do código é criado o padrão aleatório que vai “ler” a imagem original. Esse padrão foi selecionado da matrizPhi criada em um passo anterior e formatado com o comando reshape do programa Matlab para ajustar-se à configuração física do dispositivo DMD. Esse padrão encontra-se pronto para ser fornecido ao dispositivo DMD quando de uma implementação experimental do algoritmo aqui desenvolvido.

Um exemplo de padrão criado pela linha 71 é exibido na figura 29.

Figura 29 – O padrão aleatório criado pela linha 71 do código. (Fonte: autor)

A leitura e compressão simultâneas são executadas nas linhas 72 e 73 do código. Cada padrão aleatório aplicado à imagem resulta em um valor, calculado pelos componentes da mesma, capturados pelos micro-espelhos que foram adequadamente orientados para o fotodetector.

Um exemplo do resultado obtido pela linha 72 é exibido na figura 30.

Figura 30 – O resultado da leitura da imagem original pelo padrão simulado do DMD. (Fonte: autor)

Cada novo padrão repete esse processo e produz uma nova leitura comprimida da imagem original.

A composição de todas as leituras torna-se o vetor y , usado no código que ao final é normalizado pelas dimensões da imagem original, conforme linha 77 do código. Este processo de composição das leituras é exibido na figura 31.

Esta parte do algoritmo é exibido na figura 32 a seguir.

Figura 31 – Uma representação gráfica do processo da leitura e compressão da imagem.
(Fonte: adaptado de (SCHULZ; VELHO; SILVA, 2009) pelo autor)

Figura 32 – O algoritmo desenvolvido - Obter y. (Fonte: autor)

```

68 %%===== Leituras da imagem via exposição ao DMD =====%%
69
70 for ii =1:m
71     mdmd=reshape(matrizPhi(ii,:),a,b);           % usar o padrão como matriz a x b
72     leitura=immultiply(mdmd,A);                  % "ler" a imagem pelo padrão
73     amostra=sum(sum(leitura));                    % Totalizar pixels da imagem
74     y(ii)=[amostra];                             % armazenar em y
75 end
76
77 y=(y/(a*b))';                                   % normalização das leituras da imagem
78
79 fprintf('Compressão Efetuada... \n')
80

```

5.5 Algoritmo desenvolvido - Solução por Norma L2

O desenvolvimento do algoritmo prossegue com a implementação, nesse bloco, do procedimento de programação linear que resolve o problema inverso de encontrar o sinal s , pelo método da norma L2. Utiliza-se essa solução como ponto de partida para encontrar a solução esparsa no próximo passo. Esta parte do algoritmo é transcrita na figura 33.

Figura 33 – O algoritmo desenvolvido - Solução norma L2. (Fonte: autor)

```

81 %%=====Solução pela Norma L2=====%%
82 %s2 = Theta\y; %s2 = pinv(Theta)*y
83 s2 = pinv(Theta)*y;           % solução da inversão por norma L2
84
85 fprintf('Recuperando a imagem original usando o pacote l_1-magic... \n')
86

```

5.6 Algoritmo desenvolvido - Solução por CS

Continuando com o desenvolvimento do algoritmo, nesse bloco de código é implementado o procedimento de programação linear convexa no ambiente do programa Matlab, chamando a função “linprog”, existente dentro do código do pacote *l1-magic*. Os valores do sinal original, da matriz Theta e do sinal amostrado, criados nos blocos anteriores, são usados para esse fim. O algoritmo devolve o sinal original, esparso, recuperado pelo procedimento linear convexo. Esse sinal esparso será usado como base para a recuperação da imagem original no próximo bloco. A figura 34 mostra esta parte do algoritmo.

Figura 34 – O algoritmo desenvolvido - Solução por CS. (Fonte: autor)

```
87 %=====Solução por Basis Pursuit=====%%
88
89 s1 = l1eq_pd(s2,Theta,Theta',y,5e-3,20);% L1-magic toolbox do pacote l1_magic
90
91 fprintf('Solução encontrada... \n')
92
```

5.7 Algoritmo desenvolvido - Reconstrução da imagem original

Finalizando o desenvolvimento do algoritmo esse bloco usa a solução encontrada no bloco anterior e reconstrói a imagem original, pixel por pixel, usando um procedimento inverso ao usado no início do código. A imagem vai sendo reconstruída pixel a pixel e armazenada na variável *x1*. O código termina formatando a imagem para o aspecto original, com o uso da função “reshape” e dos parâmetros de tamanho da imagem principal guardados nas variáveis “a” e “b”, conforme figura 35.

Uma figura contendo a imagem original e a imagem recuperada é exibida em seguida pelo algoritmo e mostrada na figura 36 a seguir.

O tempo de processamento é calculado e exibido no console do programa matlab e o programa termina a execução, situação essa exibida na figura 37.

Figura 35 – O algoritmo desenvolvido - Reconstrução. (Fonte: autor)

```

93 %=====Reconstrução das Imagens=====%%
94
95 psiinversodireto = idct(eye(n));    % criar matriz de valores dct inversos
96
97 for ii = 1:n                        % para cada pixel da imagem
98     x1 = x1+psiinversodireto(:,ii)*s1(ii);    % recalcula o coeficiente
99 end
100
101 fprintf('Reconstrução Efetuada... \n')
102
103 fprintf('Efetuando a plotagem das imagens... \n')
104
105 figure('name','Reconstrução de Imagem por Compressive Sensing')
106 subplot(1,2,1), imagesc(A), xlabel('original'), axis image % imagem original
107 subplot(1,2,2), imagesc(reshape(x1,a,b)), xlabel('Imagem recuperada'), axis image
108 colormap gray
109
110 toc                                % Interrompe o contador tempo

```

Figura 36 – Imagens produzidas pelo algoritmo ao final de sua execução exibindo as soluções encontradas. (Fonte: autor)

Figura 37 – O algoritmo desenvolvido - Finalização do programa. (Fonte: autor)

The screenshot displays the MATLAB environment. On the left, the 'Current Folder' pane lists several files, including 'lenna256.jpg'. The 'Command Window' on the right shows the execution of a script. The script's output includes iteration details from 9 to 13, showing values for tau, Primal, PDGap, and Dual res. It concludes with 'Solução encontrada...' and 'Reconstrução Efetuada...'. A warning message states: 'Warning: MATLAB has disabled some advanced graphics rendering features by switching to software Op more information, click here.' The elapsed time is 24.122472 seconds. The script code is visible in the background, showing lines 101 to 107.

```

101 - fprintf('Reconstrução Efetuada...\n')
102
103 - fprintf('Efetuando a plotagem das imagens...\n')
104
105 - figure('name','Reconstrução de Imagem por Compressive Sensing')
106 - subplot(1,2,1), imagesc(A), xlabel('original'), axis image % imagem original
107 - subplot(1,2,2), imagesc(reshape(xl,a,b)), xlabel('Imagem recuperada'), axis image

```

Command Window

```

Iteration = 9, tau = 1.838e+06, Primal = 1.332e-01, PDGap = 2.612e-02, Dual res = 3.793e+02, Prima
Hllp condition number = 8.650e-06
Iteration = 10, tau = 3.204e+06, Primal = 1.256e-01, PDGap = 1.498e-02, Dual res = 1.996e+02, Prim
Hllp condition number = 4.454e-06
Iteration = 11, tau = 5.634e+06, Primal = 1.212e-01, PDGap = 8.519e-03, Dual res = 1.039e+02, Prim
Hllp condition number = 2.409e-06
Iteration = 12, tau = 9.491e+06, Primal = 1.188e-01, PDGap = 5.057e-03, Dual res = 5.700e+01, Prim
Hllp condition number = 1.297e-06
Iteration = 13, tau = 1.791e+07, Primal = 1.171e-01, PDGap = 2.679e-03, Dual res = 2.722e+01, Prim
Hllp condition number = 7.587e-07

Solução encontrada...
Reconstrução Efetuada...
Efetuando a plotagem das imagens...
Warning: MATLAB has disabled some advanced graphics rendering features by switching to software Op
more information, click here.
Elapsed time is 24.122472 seconds.
fx >>

```

name	Value
a	40
A	40x60 logical
amostra	391
b	60
ii	2400
leitura	40x60 double
m	1000
matrizPhi	1000x2400 double
mdmd	40x60 double
n	2400

Resultado das simulações

Os resultados obtidos até o momento neste trabalho encontram-se a seguir. A literatura pesquisada cita o método de recuperação Basis Pursuit (BP), total variation minimization (TV) e outros, mas sem fornecer, na maioria das vezes, o algoritmo utilizado ou o script responsável pelo resultado prático exibido. O presente trabalho, por outro lado, apresenta todos os algoritmos desenvolvidos e, a seguir, exemplos de resultados obtidos com a aplicação desses algoritmos.

6.1 Recuperação com a lógica original

Os resultados iniciais, em termos de tempo de processamento, usando a lógica original foram de 80 segundos para a recuperação completa da imagem. Após os ajustes feitos pelo autor, nas linhas do programa que calculavam a matriz Theta, os tempos obtidos caíram para 24 segundos. Nessa etapa o algoritmo ainda usa o método de vetorizar a imagem original antes da entrega ao processo de recuperação da imagem e não usa o conhecimento da configuração do dispositivo DMD para tanto.

A imagem inicial criada para esta fase é exibida na figura 38, com as iniciais das instituições parceiras, USP e IEAv.

Nas figuras 39, 40, 41 e 42, a seguir, observam-se as imagens recuperadas, variando o número de leituras do DMD ou a opção de ortogonalizar as colunas da matriz aleatória criada no Matlab. Em todos os exemplos foram usadas a transformada DCT como base de esparsidade do sinal/imagem e uma matriz aleatória criada pela função interna “randi” como base de medida.

A análise dos códigos é clara quanto à melhoria progressiva da qualidade da imagem recuperada com o aumento do número de amostras da imagem original porém causando o aumento no tempo de recuperação da mesma. Uma solução de compromisso entre a qualidade da imagem desejada e o tempo necessário para obtê-la é necessária nas aplicações

Figura 38 – Imagem original, 128 pixels de lado. (Fonte: autor)

práticas da técnica proposta.

Figura 39 – Imagem recuperada em 23 segundos, com 500 leituras pelo DMD. (Fonte: autor)

Figura 40 – Imagem recuperada, matriz ortogonalizada, 1000 leituras. (Fonte: autor)

Figura 41 – Imagem recuperada, matriz não-ortogonalizada, 1000 leituras. (Fonte: autor)

Figura 42 – Imagem recuperada, 1500 medidas pelo DMD. (Fonte: autor)

6.2 Recuperação com a lógica proposta no trabalho

Com o algoritmo desenvolvido e modificado foram obtidas as seguintes imagens recuperadas, ilustradas nas figuras 43, 44, 45, 46 e 47, usando como imagens originais a figura Usp-IEAv, mais simples contendo áreas de intensidade constante e a figura padrão de processamento de imagens, “Lena”, mais complexa contendo vários tons de cinza, nesse exemplo. Esses exemplos usam o algoritmo com o método de captura ajustado à configuração física do dispositivo DMD com uma matriz de micro-espelhos. As imagens recuperadas melhoram progressivamente com o aumento das amostras feitas e o tempo de recuperação aumenta também, como acontece com os algoritmos originais. A vantagem dessa abordagem é que o código está pronto para a etapa de implementação física com o dispositivo DMD, a ser executada por outra equipe de desenvolvimento posteriormente.

A seguir, apresenta-se a sequência de imagens recuperadas da figura exemplo Usp-IEAv, usando o solver baseado no algoritmo l_1 eqpd, com 250, 500, 1000, 1500 e 2400 amostras.

Os algoritmos usam diferentes “solvers” para obter a imagem recuperada. Alguns são mais eficientes com poucas amostras enquanto outros foram mais eficientes com um maior número de amostras, considerando o fator tempo como critério de julgamento.

A figura 48 apresenta as imagens obtidas pelo solver l_1 eqpd:

A figura 49 apresenta as imagens obtidas pelo solver l_1 eqpdvazio:

Figura 43 – Imagem recuperada pelo algoritmo proposto, 250 medidas pelo DMD, 4,22 seg. (Fonte: autor)

Figura 44 – Imagem recuperada pelo algoritmo proposto, 500 medidas pelo DMD, 7,19 seg. (Fonte: autor)

Figura 45 – Imagem recuperada pelo algoritmo proposto, 1000 medidas pelo DMD, 19,18 seg. (Fonte: autor)

A figura 50 apresenta as imagens obtidas pelo solver $l-1$ qlogbarrier:

A figura 51 apresenta as imagens obtidas pelo solver $l-1$ eqpd com a imagem mais complexa Lena:

Finalmente, a figura 52 apresenta as imagens obtidas pelo solver $l-1$ qlogbarrier com a imagem mais complexa Lena:

Os tempos de recuperação das imagens obtidos com cada algoritmo foram computados e exibidos na tabela 1, a seguir. Alguns dos algoritmos apresentaram melhor performance com poucas amostras, por exemplo o DMd11eqpd, enquanto outros desempenharam com

Figura 46 – Imagem recuperada pelo algoritmo proposto, 1500 medidas pelo DMD, 34,16 seg. (Fonte: autor)

Figura 47 – Imagem recuperada pelo algoritmo proposto, 2400 medidas pelo DMD, 78,92 seg. (Fonte: autor)

Figura 48 – Imagens recuperadas com diferentes tamanhos de amostras com o solver l_1 eqpd (Fonte: autor)

Figura 49 – Imagens recuperadas com diferentes tamanhos de amostras com o solver l_1 eqpdvazio (Fonte: autor)

Figura 50 – Imagens recuperadas com diferentes tamanhos de amostras com o solver l_1 qclogbarrier. (Fonte: autor)

Figura 51 – Imagens recuperadas com diferentes tamanhos de amostras com o solver l_1 eqpd. (Fonte: autor)

Figura 52 – Imagens recuperadas com diferentes tamanhos de amostras com o solver l_1 qclgbarrier. (Fonte: autor)

mais amostras, no caso o dmdl1qclgbarrier.

Tabela 1 – Tempos de recuperação das imagens. Fonte: Autor

Tempo de recuperação das Imagens (segundos)							
Algoritmo	Número de Amostras						
	250	300	500	1000	1500	2400	3000
DmdL1eqpd	4,22		7,19	19,18	34,16	78,92	
DmdL1eqpdvazio	4,74		8,28	21,09	40,81	79,00	
DmdL1qclogbarrier	14,46		17,47	21,23	30,70	57,01	
DmdLenaL1eqpd		10,64	18,52	40,04	67,35		240,70
DmdLenaL1qclogbarrier		47,94	54,18	57,77	75,24		133,34

Considerações finais

Os resultados encontrados até a presente data podem ser considerados satisfatórios e permitem responder a algumas das dúvidas que tínhamos no início do trabalho.

Quanto à dúvida se o algoritmo usado conseguiria recuperar as imagens originais conclui-se que o algoritmo consegue recuperar as imagens originais.

Quanto às dificuldades, exigências e restrições que tínhamos notou-se que a falta de uma plataforma de desenvolvimento adequada à operação do DMD em processos de captura de imagens foi uma dificuldade encontrada que mudou o escopo original do presente trabalho. Focamos na simulação do processo de captura usando o algoritmo desenvolvido para validar os pressupostos iniciais e ajudar no entendimento das variáveis gerais do processo usado.

Já quanto aos tempos para se obter a recuperação de imagens eles são da ordem de segundos a centenas de segundos e dependem fundamentalmente das dimensões da imagem original e do solver usado no processo de recuperação. Os tempos obtidos são associados apenas ao processamento do algoritmo de recuperação usado e não levam em consideração o tempo para enviar os padrões aleatórios ao DMD e o tempo de resposta do mesmo em cada leitura. Os tempos de operação do DMD deverão ser somados aos obtidos aqui em futuros experimentos práticos deste trabalho.

Também sobre a possibilidade de melhorar a qualidade da imagem recuperada nota-se que é possível melhorar a qualidade da imagem recuperada aumentando o número de amostras ou leituras da imagem usadas pelo algoritmo, modificando uma variável interna ao mesmo, a variável “m”. Verificamos com o algoritmo que aumentando “m” melhoramos a qualidade da imagem recuperada mas também aumentamos o tempo necessário para que o algoritmo reconstrua a imagem.

Conforme nosso entendimento inicial de que a técnica empregada demandaria um tempo significativo na recuperação da imagem, os resultados das simulações confirmaram esse entendimento, com valores totais para recuperar a imagem variando de 4 e 240 segundos dentro das condições aqui usadas. Esses tempos obtidos nos levam a considerar que a presente técnica pode ser usada na mensuração e captura de imagens dos fenômenos mais

estáticos e menos dinâmicos como os efeitos da radiação térmica (ou IR) que apresentam pequenas taxas de variação no tempo.

Consideramos que a técnica é promissora para ser usada nos processos de monitoramento do efeito corona nas barras de geradores de forma contínua, como um processamento de um vídeo contínuo, visto que ao efeito Corona é mais dinâmico e transitório na sua ocorrência.

As simulações realizadas com os algoritmos mostram que há espaço e motivos para a melhoria das imagens recuperadas e dos tempos obtidos e que devem ser levados em consideração em futuros trabalhos na área.

Desenvolvimento futuro e sugestões

O uso do algoritmo Basis Pursuit ainda está no início e, neste trabalho, foi associado apenas com a base de esparsidade à DCT.

Com o desenvolvimento e a aplicação das wavelets no final dos anos 90 (MALLAT, 1999), sinais mais complexos e com diferentes taxas de variação puderam ser melhor analisados em comparação com a análise Fourier usada até então. Na técnica de sensoriamento compressivo as wavelets também foram empregadas nos processos de recuperação das imagens, com uma variante chamada de “noiselets”, como no artigo de (PASTUSZCZAK et al., 2016).

O algoritmo aqui proposto pode ser modificado para usar com base de esparsidade não a DCT mas as Wavelets e suas variantes com potencial de redução de tempo de processamento e melhoria das imagens obtidas. O presente trabalho e o citado acima (PASTUSZCZAK et al., 2016) podem ser um ponto de partida para tanto.

Há ainda a expectativa de avançar nos trabalhos experimentais, construindo um reconstrutor de imagens e uma câmera de 1 pixel com DMD, que permitirá testar na prática a eficiência dos métodos e algoritmos desenvolvidos nas simulações aqui apresentadas.

Referências

- A-QIAN, S. et al. Optical scanning holography based on compressive sensing using a digital micro-mirror device. *Optics Communications*, Elsevier B.V., v. 385, p. 19–24, feb 2017. ISSN 00304018.
- AHMED, A.; ROMBERG, J. Compressive sampling of ensembles of correlated signals. **IEEE Transactions on Information Theory**, Institute of Electrical and Electronics Engineers Inc., v. 66, n. 2, p. 1078–1098, feb 2020. ISSN 15579654.
- BARANIUK, R. Compressive sensing [lecture notes]. **IEEE Signal Process. Mag.**, v. 24, p. 118–121, 2007.
- BRORSSON, A. **Compressive Sensing: Single Pixel SWIR Imaging of Natural Scenes**. [S.l.], 2018.
- CANDES, E. J.; TAO, T. Near-optimal signal recovery from random projections: Universal encoding strategies? **IEEE Transactions on Information Theory**, v. 52, n. 12, p. 5406–5425, 2006. ISSN 00189448.
- CANDÈS, E.; ROMBERG, J. **l1magic**. 2004. Disponível em: <<https://statweb.stanford.edu/candes/l1magic/>>.
- CANDÈS, E.; ROMBERG, J.; TAO, T. **Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information**. [S.l.], 2004. 489–509 p.
- CANDÈS, E. J.; ROMBERG, J. K.; TAO, T. Stable signal recovery from incomplete and inaccurate measurements. **Communications on Pure and Applied Mathematics**, v. 59, n. 8, p. 1207–1223, 2006. ISSN 00103640.
- CARRON, I. **Compressive Sensing, the Big Picture**. 2011. Disponível em: <<https://sites.google.com/site/igorcarron2/cs#reconstruction>>.
- CHEN, H. et al. Single pixel infrared camera using a carbon nanotube photodetector. In: **Proceedings of IEEE Sensors**. [S.l.: s.n.], 2011. p. 1362–1366. ISBN 9781424492886.
- CHEN, S. S.; DONOHO, D. L.; SAUNDERS, M. A. Atomic decomposition by basis pursuit. **SIAM Journal of Scientific Computing**, v. 20, n. 1, p. 33–61, 1998.
- CIGRE. Survey of hydrogenerator failures. **Study Committee SC11**, n. EG11.02, 2003.

DONOHU, D. L. Compressed sensing. **IEEE Transactions on Information Theory**, v. 52, p. 1289–1306, 2004.

GALVIS, L.; ARGUELLO, H.; ARCE, G. R. Synthetic coded apertures in compressive spectral imaging. In: **ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2014. p. 3181–3185. ISSN 15206149.

GIBSON, S. **Simple compressed sensing example**. 2013. Disponível em: <<https://www.mathworks.com/matlabcentral/fileexchange/41792-simple-compressed-sensing-example?focused=3783702&tab=function>>.

HANNAK, G. et al. Performance analysis of approximate message passing for distributed compressed sensing. **IEEE Journal on Selected Topics in Signal Processing**, Institute of Electrical and Electronics Engineers Inc., v. 12, n. 5, p. 857–870, oct 2018. ISSN 19324553.

HAYASHI, K.; NAGAHARA, M.; TANAKA, T. A user's guide to compressed sensing for communications systems. **IEICE Transactions on Communications**, E96-B, n. 3, p. 685–712, 2013. ISSN 17451345.

HU, M. et al. Terahertz meta-holograms reconstruction based on compressed sensing. **IEEE Photonics Journal**, v. 12, n. 4, p. 1–9, aug 2020. ISSN 1943-0655. Disponível em: <<https://ieeexplore.ieee.org/document/9130058/>>.

LAUWERS, T.; KANTOR, G.; HOLLIS, R. One is enough! **Springer Tracts in Advanced Robotics**, v. 28, n. July, p. 114–127, 2007.

LIMA, J. A.; MIOSSO, C. J.; FARIAS, M. C. Per-pixel mirror-based method for high-speed video acquisition. **Journal of Visual Communication and Image Representation**, Academic Press Inc., v. 47, p. 23–35, aug 2017. ISSN 10959076.

MALLAT, S. **A Wavelet Tour of Signal Processing**. 2.ed.. ed. [S.l.: s.n.], 1999.

PASTUSZCZAK, A. et al. Modified noiselet transform and its application to compressive sensing with optical single-pixel detectors. 2016.

RIZA, N. A.; MARRACCINI, P. J.; BAXLEY, C. R. Data efficient digital micromirror device-based image edge detection sensor using space-time processing. **IEEE Sensors Journal**, v. 12, n. 5, p. 1043–1047, 2012. ISSN 1530437X.

ROMBERG, J. Imaging via compressive sampling. **IEEE Signal Processing Magazine**, v. 25, n. 2, p. 14–20, 2008. ISSN 1053-5888.

RUEDA, H.; ARGUELLO, H.; ARCE, G. R. Dmd-based implementation of patterned optical filter arrays for compressive spectral imaging. **Journal of the Optical Society of America A**, The Optical Society, v. 32, n. 1, p. 80, jan 2015. ISSN 1084-7529.

SANTOS, L. et al. **Tecnologias de infravermelho e ultravioleta aplicadas a sistemas elétricos de alta tensão**: em xx snptee seminário nacional de produção e transmissão de energia elétrica, recife, 2009. 2009.

Disponível em: <<http://www.cgti.org.br/publicacoes/wp-content/uploads/2016/03/TECNOLOGIAS-DE-INFRAVERMELHO-E-ULTRAVIOLETA-APLICADAS-A-SISTEMAS-EL%C3%89TRICOS-DE-ALTA-TENS%C3%83O.pdf>>.

SCHULZ, A.; VELHO, L.; SILVA, E. Cs applications in graphics vision [lecture 5 notes]. In: **Compressive Sensing**. [S.l.: s.n.], 2009. ISBN 9788524402944.

TAKHAR, D. et al. A new compressive imaging camera architecture using optical-domain compression. **Computational Imaging IV**, v. 6065, p. 606509–606509–10, 2006. ISSN 0277786X.

VAZ, P. G. et al. Image quality of compressive single-pixel imaging using different hadamard orderings. **Optics Express**, The Optical Society, v. 28, n. 8, p. 11666, apr 2020. ISSN 1094-4087.

WANG, Y. et al. Dct inspired feature transform for image retrieval and reconstruction. **IEEE Transactions on Image Processing**, Institute of Electrical and Electronics Engineers Inc., v. 25, n. 9, p. 4406–4420, sep 2016. ISSN 10577149.

WILLETT, R. **Methods for sparse analysis of high dimensional data, aulas 1,2 e 3**. 2011. Disponível em: <<https://www.math.ias.edu/wam/2011/schedule>>.

WU, Y. et al. **Demonstration of a DMD-based Compressive Sensing (CS) Spectral Imaging System**. [S.l.], 2011. Disponível em: <http://earthobservatory.nasa.gov/Features/EO1/eo1_2.php>

XIAO, Y.; ZHOU, L.; CHEN, W. Single-pixel imaging authentication using sparse hadamard spectrum coefficients. **IEEE Photonics Technology Letters**, Institute of Electrical and Electronics Engineers Inc., v. 31, n. 24, p. 1975–1978, dec 2019. ISSN 19410174.

YU, W.-K. et al. Single-pixel ghost microscopy based on compressed sensing and complementary modulation. jan 2015. Disponível em: <<http://arxiv.org/abs/1501.06002><http://dx.doi.org/10.1016/j.optcom.2016.03.067>>.

ZHANG, C. et al. A novel compressive optical encryption via single-pixel imaging. **IEEE Photonics Journal**, Institute of Electrical and Electronics Engineers Inc., v. 11, n. 4, aug 2019. ISSN 19430655.

ZHOU, J. et al. Compressive sensing for particle size retrieval by using a digital micro-mirror device-based detector. **Powder Technology**, Elsevier B.V., v. 304, p. 27–31, dec 2016. ISSN 1873328X.