

## Visão e Processamento de Imagens - Avaliação única - Parte II

### Preste atenção para as regras da prova

- 1- O fonte latex (.tex) da prova será disponibilizado para facilitar que você não tenha que copiar o enunciado das questões. Todas as questões devem ser respondidas no mesmo arquivo.
- 2- A prova é **individual**. É permitido a consulta a livros, apontamentos ou Internet, desde que devidamente referenciada. Não é permitida a consulta a colegas, amigos, família, cachorro, papagaio e etc.
- 3- A prova deve ser entregue diretamente no Paca, assim como todos os códigos e imagens devem ser entregues no mesmo arquivo comprimido. **Duração da prova: 14 dias.**
- 4- Cada questão vale 20 pontos (pois são apenas 3 questões) para a graduação e 15 pontos para a pós-graduação (pois são 4 questões).

**Q1.** Para fazer esta questão, leia primeiro o artigo abaixo:

- <http://www.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-85.pdf>
- Faça um resumo do artigo de acordo com as indicações que deixei no paca (artigos sobre como fazer um resumo).

---

### Resumo: *Digital Image Manipulation Forensic*

*Rodrigo Augusto Dias Faria*

*Instituto de Matemática e Estatística – Universidade de São Paulo (IME-USP)*  
*Departamento de Ciência da Computação*

rofaria@ime.usp.br, rodrigoadfaria@gmail.com

#### 1. Resumo

##### 1.1. Motivação

O artigo contextualiza o cenário atual no que diz respeito à manipulação de imagens por ferramentas de edição, ressaltando a ampla gama de técnicas que essas ferramentas possuem e que permitem aos seus usuários a manipulação de imagens com extrema facilidade, o que implica em uma análise detalhista por um profissional especializado para descobrir qualquer tipo de fraude.

Aliado a este fator, pode-se destacar ainda que pessoas podem coletar evidências de crimes ou eventos quaisquer de uma forma trivial, considerando a difusão de dispositivos móveis equipados com câmeras de boa qualidade em qualquer lugar do mundo.

Podemos afirmar que ainda não há crise na área pesquisada, porém, tendências de mercado norteiam para que processos historicamente feitos sob arquivos impressos e com a presença dos envolvidos, sejam totalmente

realizados por plataformas online, o que pode, futuramente, causar grandes transtornos às empresas e ao estado de um modo geral pelo risco de desvio de conduta e possibilidades de fraudes em processos sigilosos, de grande valor agregado, avaliações contratuais, etc.

Não é demasiado lembrar que processos judiciais já são interpretados com ajuda de imagens digitais, o que nos remete a necessidade de garantir a autenticidade das mesmas, assim como também no jornalismo, cuja integridade pode ser colocada em xeque em casos de adulteração de fotos em quaisquer publicações.

Sendo assim, a motivação principal do trabalho proposto é de criar uma ferramenta capaz de avaliar uma dada imagem, de forma que qualquer pessoa comum, entenda-se por aquela que não tem conhecimento específico na área de processamento de imagens, possa identificar potenciais alterações, sem a necessidade de auxílio de um analista forense, por exemplo.

Tal ferramenta deve ser uma plataforma web, disponível em larga escala, sendo que uma das principais técnicas abordadas pelos autores para compor a paleta de funcionalidades do serviço é o Error Level Analysis (ELA).

É sabido que imagens JPEG perdem informação durante a compactação e uma certa quantidade de erro é introduzida cada vez que uma imagem JPEG é comprimida / salva novamente [4]. Logo, cada pixel contém um certo nível de erro, e a execução de uma nova compressão irá alterar o nível de erro do mesmo. No entanto, se uma imagem é manipulada, é muito provável que as diferentes partes da imagem terão diferentes níveis de erro, e o ELA é uma técnica que se concentra neste artefato para identificar se uma imagem foi manipulada e encontrar regiões adulteradas [4].

## 1.2. Contribuição

Do ponto de vista de mercado, o trabalho apresentado visa um nicho não explorado pelos concorrentes avaliados, já que fornece um serviço com resultados de detecção mais informativo do que outras plataformas disponíveis.

Em outras palavras, os concorrentes focam apenas em análises de baixo nível das imagens e o serviço proposto, além das funcionalidades encontradas nos concorrentes avaliados, também adiciona suporte a outras avaliações de alto nível, tais como, detecção de regiões na imagem que foram copiadas/coladas.

Por outro lado, do ponto de vista técnico e científico, o autor ressalta que o ELA tem uma desvantagem: a região de alta frequência na imagem dada afetará o resultado, já que a compressão JPEG na imagem, naturalmente, comprime mais as regiões de alta frequência e, sendo assim, a parte de alta frequência de uma imagem terá, geralmente, uma maior

queda nos níveis de erro, mesmo que não seja adulterada.

O autor propõe, então, superar esta deficiência do método criando uma máscara de baixa frequência que é gerada pela execução de um filtro gaussiano na imagem e medindo a diferença entre a imagem dada e a imagem filtrada, já que as regiões de baixa frequência terão alterações menores após filtro o gaussiano.

Em seguida, esta máscara é multiplicada em cada um dos três canais diferentes da imagem resultante do ELA, o que auxilia na percepção visual das diferenças. Além disso, regiões de baixa frequência serão destacadas para demonstrar regiões com alto risco de terem sofrido alterações.

Outras técnicas também foram apresentadas, com o intuito de integração na plataforma web que está sob desenvolvimento pela equipe em questão. Considerando o processo de codificação de uma imagem, deu-se foco na etapa de quantização, já que nesta etapa, o processo de codificação pode deixar rastros que permitem descobrir se uma imagem foi manipulada ou não. Uma delas é o *blocking artifact measurement* e a outra é o próprio ELA, com um realce no aspecto visual, no intuito de melhorar a experiência proporcionada pelo método original.

Uma outra técnica citada foi a análise de metadados, que consiste em avaliar algumas informações tais como, a ferramenta que foi utilizada para criar a imagem e, no caso de uma possível alteração, a ferramenta que alterou a imagem, que ficam gravadas no arquivo. Quando da alteração por um *smartphone*, o sistema operacional também fica armazenado. Outra informação importante é data de criação e/ou alteração da imagem. Apesar de serem características que podem ser um indicativo de que a imagem foi alterada, nada garante que terceiros tentem burlar, também, o conteúdo relativo ao metadado da imagem.

### 1.3. Metodologia

O método então utilizado pelo autor foi calcular a norma L2 dos dados da imagem resultante da aplicação do ELA. Em seguida, um *threshold* é aplicado na mesma imagem. Logo após este passo, o resultado do *threshold* do ELA é submetido a um filtro gaussiano para, então, produzir uma região contínua. Finalmente, a imagem resultante é suavizada e submetida a um novo *threshold* para remover o ruído.

Todo este processo cria regiões rotuladas de baixa frequência e regiões com alto índice de ELA tornam-se visualmente mais contundentes, quando da avaliação de qualquer tipo de violação da imagem.

De certa forma um novo algoritmo foi implementado, mas não foi apresentado pelos autores, apenas os passos que foram tomados para que pudessem ser gerados os filtros gaussianos, bem como sua aplicação na imagem. Apesar do método ter sido citado, não há teoremas detalha-

dos no artigo, apenas algumas técnicas, além do ELA, foram citadas, de forma que a ideia principal pudesse ser passada aos leitores. Além disso, o artigo tem um viés bem mais comercial do que técnico-científico, o que o torna muito parecido com um plano de negócio.

#### 1.4. Conclusão

Pode-se destacar que o foco principal do artigo é o ELA, onde a contribuição de um dos autores, Yan Zhao, fica mais evidente, tanto que uma seção foi inteiramente dedicada à sua colaboração no trabalho.

Dos experimentos realizados, uma imagem foi dada como exemplo para mostrar a técnica com *blocking artifact measurement*. Com o ELA, 3 *datasets* com imagens manipuladas foram utilizados: *CASIA2 Tampered Image Detection Evaluation Database* que consiste de 7491 imagens autênticas e 5123 adulteradas, o segundo *Columbia Uncompressed Image Splicing Detection Evaluation Dataset* formado por 180 imagens remontadas, fruto de cópia de um recorte da própria imagem ou de uma outra fonte, transposta para outra região e 180 imagens originais e o terceiro é o *Benchmark Dataset* composto, também, por imagens de alta qualidade com regiões adulteradas por cópia e cola.

O resultado do método ELA original elaborado por Krawetz [4] foi exibido e, posteriormente, a aplicação do novo método proposto com filtros gaussianos mostram resultados significativos.

Além do aspecto visual, Yan Zhao também criou um classificador capaz de decidir se uma imagem foi modificada ou não. A extração dos atributos foi feita com dados numéricos do ELA como a média, a mediana e a variância nos três canais da imagem resultante do ELA. O classificador foi criado com *support vector machine* e *adaptive boosting*, ambos do pacote *Scikitlearn*.

Os resultados obtidos com o classificador foram relevantes, sendo uma taxa de acerto de 84.34% com *support vector machine* e 85.54% com *adaptive boosting* e árvores de decisão.

Apesar dos resultados terem sido expressivos, o experimento com o *dataset Benchmark Dataset* não foi bem sucedido, conforme citado pelos autores, mas nenhuma métrica ou medida de erro foi referenciada.

Um *benchmark* também foi elaborado com duas empresas diferentes. A primeira, System of Methods and Tools of Digital Processing Technology LLC, conhecida como SMTDP, é uma empresa de tecnologia russa fundada em 2011 que se concentra em automação de processos de negócios e detecções de manipulação de imagens. Algumas das tecnologias usadas pela SMTDP são a análise de metadados de imagem e análise de compressão. A SMTDP concentra-se em parcerias com outras grandes empresas como a Belkasoft e PricewaterhouseCoopers.

A segunda empresa é chamada de Verifeyed e está localizada na República Checa. Tanto as ferramentas, quanto o mercado consumidor são diferentes se comparadas com a SMTDP. A Verifeyed tem um pacote de software, que é comercializado sob a venda de licenças diretamente aos usuários finais. A tecnologia principal concentra-se em análise de metadados e balística.

Ambas têm um posicionamento de mercado diferente do proposto pelos autores.

Em suma, o plano de negócios citado tem características de inovação, tanto do ponto de vista mercadológico quanto do tecnológico, já que os autores se preocuparam em melhorar técnicas já existentes com o objetivo de ter um diferencial competitivo.

Logo, as práticas já existentes na área podem ser impactadas com as mudanças indicadas, principalmente, pelo fato da plataforma web ser interativa e capaz de dar resultados mais analíticos do que as demais avaliadas como *benchmark*.

Os resultados obtidos não são de todo genéricos, já que é possível, ainda assim, alterar uma imagem sem que o método proposto pelo ELA detecte qualquer alteração.

Como trabalho futuro, o autor cita que novas ferramentas para auxiliar os usuários a obter uma melhor interpretação do resultado do ELA devem ser criadas e, também, extrair outros dados numéricos que possam ser incluídos como atributos de entrada para os classificadores.

## 2. Crítica

### 2.1. Capacidade de inovação

Com exceção do método composto pelo ELA com filtro gaussiano, todos os demais são apenas repetições do estado da arte em análise de imagens.

Não há preocupação em relação a comparação dos métodos com outros existentes, já que eles estão sendo utilizados como experimentos para compor as técnicas que estarão disponíveis na plataforma web posteriormente. Isso fica ainda mais evidente na seção *Intellectual Property*, onde os autores chegam à conclusão de que não é possível registrar patente da plataforma web já que ela não atende a dois requisitos das leis dos Estados Unidos para tal, que são:

1. A reprodução não pode ser óbvia, o que não ocorre, já que os próprios autores têm receio de serem copiados, uma vez que eles utilizam uma combinação de técnicas que estão disponíveis em outros artigos que estão publicados abertamente.
2. Inovação que, pelas mesmas leis, a invenção reivindicada deve ter sido antes de ser patenteada, descrita em uma publicação impressa, ou em uso público, em promoção, ou de outro modo disponível para o público, o que

também não ocorre neste caso e, por consequência, dificulta o processo de obtenção de patente.

## 2.2. Ausência de embasamento teórico

Nenhum algoritmo foi mostrado, poucos conceitos matemáticos, nenhuma prova, o que não garante que o método é totalmente funcional. É difícil até mesmo avaliar qualquer tipo de erro conceitual, uma vez que maiores detalhes não foram fornecidos, acredito que de forma propositada, para evitar que detalhes do método fossem publicados inadvertidamente, oferecendo risco ao negócio como um todo.

Outro ponto já observado, o experimento com o *dataset Benchmark Dataset* não foi bem sucedido, conforme citado pelos autores, mas nenhuma métrica ou medida de erro foi detalhada.

## 3. Sugestões

### 3.1. Mudança de paradigma para *open source*

Uma possível mudança de paradigma seria abrir o código para a comunidade científica para obter sugestões, críticas, elogios e possíveis novas funcionalidades para a plataforma, de forma que não fique limitada a métodos já existentes ou que possam ser facilmente copiados. Com isso, as técnicas apresentadas podem ser mais bem elaboradas e expostas, sem qualquer tipo de problema em relação a segredo industrial ou patente envolvida.

Com este mesmo viés, a contribuição de outros grupos de pesquisa podem incluir embasamento teórico e fundamentado para este e novos métodos propostos.

---

- Implemente o método ELA (Error Level Analysis) em Python (apresente o **algoritmo** na prova e anexe o código em Python no arquivo zip).

Abaixo o algoritmo que foi implementado onde utilizei o pacote *pillow* do python para as operações na imagem. O código completo está em **source/ela.py**.

---

**Algoritmo 1: ERROR LEVEL ANALYSIS**

---

**Entrada:** Caminho da imagem, escala

**Saída:** Imagem com ELA calculado

```

1 ImagemOriginal ← abrirImagem(Caminho da imagem)
2 se ImagemOriginal não é JPEG então
3   | retorna Nulo
4 fim

5 salvarImagem(Caminho da imagem comprimida, JPEG, qualidade)
6 ImagemComprimida ← abrirImagem(Caminho da imagem comprimida)

7 para x ← 0 até largura ImagemOriginal fazer
8   | para y ← 0 até altura ImagemOriginal fazer
9     | pixel_img_original = ImagemOriginal[x, y]
10    | pixel_img_comprimida = ImagemComprimida[x, y]
11    |  $R = \text{abs}(\text{pixel\_img\_original}[0] - \text{pixel\_img\_comprimida}[0]) * \text{escala}$ 
12    |  $G = \text{abs}(\text{pixel\_img\_original}[1] - \text{pixel\_img\_comprimida}[1]) * \text{escala}$ 
13    |  $B = \text{abs}(\text{pixel\_img\_original}[2] - \text{pixel\_img\_comprimida}[2]) * \text{escala}$ 
14    | ImagemELA[x, y] = [R, G, B]
15   | fim
16 fim

17 retorna ImagemELA
```

---

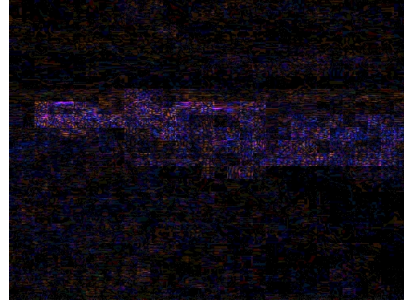
- Teste seu algoritmo com as imagens que deixei no paca para este exercício. Quantas imagens seriam consideradas modificadas por esse método? Comente o resultado, comparando com a sua intuição.

Uma vez submetida a imagem ao *Error Level Analysis*, pode-se perceber pelo resultado que a região manipulada terá um nível de erro diferente das regiões não manipuladas. Logo, o nível de erro irá expor a região manipulada rotulando as regiões com maior alteração após a imagem ser salva com um nível de qualidade inferior [4].

Na Figura 1, podemos ver o resultado do ELA na imagem dada. As regiões com maior chance de ter alterações são as que apresentam os pixels com maior brilho, uma vez que a alteração da imagem causa instabilidade nestas áreas.



(1a) Imagem original



(1b) Imagem com o resultado do ELA

Figura 1: Aplicação do método ELA

Os resultados foram avaliados de acordo com o brilho das bordas que devem ser semelhantes no resultado da aplicação do ELA na imagem. Além disso, regiões de cores e texturas semelhantes na imagem original, independentemente da cor, também devem ter cores aproximadamente similares no ELA [8].

Isto posto, considere que um total de 23 imagens foram alteradas de acordo com o método e avaliação a posteriori. A Tabela 1 lista a avaliação do método nas imagens dadas. As imagens originais, bem como as alteradas pelo método podem ser vistas no diretório **source/HoaxImages**.

**Q2.** Esta questão refere-se à transformada de Fourier.

– Encontre a transformada de Fourier da função:

$$f(x) = \begin{cases} 7 & \text{if } -5 < x < 5 \\ 0 & \text{caso contrário} \end{cases}$$

Por definição, temos que a transformada de Fourier de um pulso retangular de duração  $D$  e altura  $A$  tem a forma dada por:

$$F(\omega) = AD \operatorname{sinc}\left(\frac{\omega D}{2}\right) = AD \frac{\sin\left(\frac{\omega D}{2}\right)}{\frac{\omega D}{2}}$$

A função  $f(x)$  pode ser representada graficamente como (Figura 2):

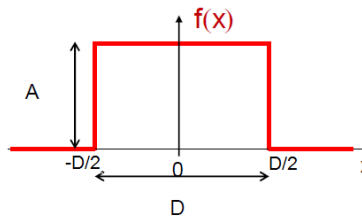


Figura 2: Função pulso retangular



Tabela 1: Tabela com o resultado da aplicação do método ELA

Imagem	Alterada?
2000_snowballcat.jpg	Y
bikefail.jpg	N
blacklion01.jpg	Y
bouncing_baby.jpg	N
broken_road.jpg	N
businahole.jpg	N
cows_on_beach.jpg	Y
cursor.jpg	N
daliatom.jpg	Y
dononwater.jpg	N
eagles.jpg	N
frozenvenice.jpg	Y
glass_butterfly.jpg	Y
goldfish_hitler.jpg	N
hatfield.jpg	N
hellephant.jpg	N
hitlerbaby.jpg	N
horseinahole01.jpg	N
houseboat01.jpg	Y
iceberg.jpg	N
jumping_giraffe.jpg	Y
kissing.jpg	Y
koala01.jpg	N
leap.jpg	Y
magic_tap.jpg	Y
manitoba_security.jpg	Y
moonmelon01.jpg	Y
nikolatesla.jpg	Y
queensguard.jpg	N
rainbow_tornado.jpg	N
rocket_bike.jpg	Y
sharkswim.jpg	Y
shark_roof.jpg	N
skiing_egypt.jpg	Y
skullrose.jpg	N
spacechair.jpg	Y
tandembike.jpg	N
tattooguy01.jpg	y
tennis.jpg	Y
tentacle_bldg.jpg	Y
trafficlights.jpg	N
tunnelface01.jpg	N
verydeep.jpg	N
vuitton.jpg	Y
whale.jpg	N
wienerplane.jpg	Y

Onde:

$$f(x) = \begin{cases} A, & x \in [-\frac{D}{2}, \frac{D}{2}] \\ 0, & x \notin [-\frac{D}{2}, \frac{D}{2}] \end{cases}$$

Logo, temos que  $A = 7$  e  $D = 10$  e, portanto, a transformada de Fourier da função  $f(x)$  é:

$$\begin{aligned} F(\omega) &= AD \frac{\sin(\frac{\omega D}{2})}{\frac{\omega D}{2}} \\ &= 7 * 10 \frac{\sin(\frac{\omega 10}{2})}{\frac{\omega 10}{2}} \\ &= 70 \frac{\sin(\frac{\omega 10}{2})}{\frac{\omega 10}{2}} \\ &= 70 \frac{\sin(5\omega)}{5\omega} \end{aligned}$$

- Encontre a transformada de Fourier da função  $g(x) = f(x) \cos \omega_0 x$ , sabendo que a transformada de Fourier de  $f(x)$  é dada por  $F(\omega)$

Tomando a propriedade da modulação:

$$\mathcal{F}[x(t) \cos(\omega_0 t)] = \frac{1}{2} [F(\omega + \omega_0) + F(\omega - \omega_0)]$$

Temos que a transformada de Fourier da função  $g(x)$  é:

$$G(\omega) = \frac{1}{2} F(\omega + \omega_0) + \frac{1}{2} F(\omega - \omega_0)$$

- Ache a inversa da transformada de Fourier de  $G(\omega) = 20 \frac{\sin 5\omega}{5\omega} e^{-3\omega i}$

Por ora, ignorando a exponencial complexa de  $G(\omega)$ , podemos obter os valores de  $A$  e  $D$ :

$$20 \frac{\sin(5\omega)}{5\omega} = AD \frac{\sin(\frac{\omega D}{2})}{\frac{\omega D}{2}}$$

$$5\omega = \frac{\omega D}{2}$$

$$10\omega = \omega D$$

$$D = \frac{10\omega}{\omega} = 10$$

$$AD = 20$$

$$A10 = 20$$

$$A = 2$$

Tomando a representação da função  $f(x)$  do pulso retangular de duração  $D$  e amplitude  $A$ :

$$f(x) = A.rect(x) = \begin{cases} A, & x \in \left[-\frac{D}{2}, \frac{D}{2}\right] \\ 0, & x \notin \left[-\frac{D}{2}, \frac{D}{2}\right] \end{cases}$$

Vimos no primeiro item do exercício 2 que:

$$A.rect\left(\frac{x}{D}\right) \xrightarrow{\mathcal{F}} ADsinc\left(\frac{\omega D}{2}\right)$$

O que nos dá a forma do pulso retangular:

$$f(x) = 2.rect\left(\frac{x}{10}\right) = \begin{cases} 2, & x \in [-5, 5] \\ 0, & x \notin [-5, 5] \end{cases}$$

Cuja representação gráfica é (Figura 3):

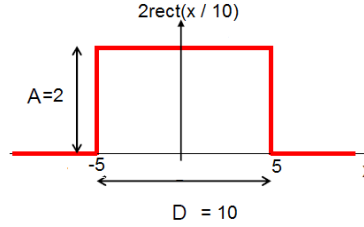


Figura 3: Função pulso retangular com  $D = 10$  e  $A = 2$

Considerando agora a exponencial complexa, sabemos que ela representa um deslocamento no tempo, que é 3 neste caso e, portanto:

$$2.rect\left(\frac{x-3}{10}\right) \xrightarrow{\mathcal{F}} 20 \frac{\sin 5\omega}{5\omega} e^{-3\omega i}$$

$$g(x) = 2.rect\left(\frac{x-3}{10}\right) = \begin{cases} 2, & -2 < x < 8 \\ 0, & \text{caso contrário} \end{cases}$$

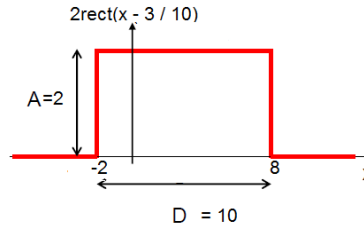


Figura 4: Função pulso retangular com deslocamento no tempo

- Calcule a DFT do sinal  $f = \{1, 3, 5, 3, 1\}$

A Transformada Discreta de Fourier (DFT) provém da Série Discreta de Fourier (DFS) e é dada por [1]:

$$X_k = \sum_{m=0}^{N-1} x_m e^{-\frac{2\pi i k m}{N}}, \quad \text{para } k = 0, 1, 2, \dots, N-1$$

Para realizar os cálculos devemos utilizar a identidade de Euler:

$$e^{-i\pi} = \cos \pi - i \sin \pi$$

Temos que  $N=5$ , ou seja, o tamanho do vetor do sinal dado por  $f$ , logo:

$$\begin{aligned} X_0 &= (1e^0 + 3e^0 + 5e^0 + 3e^0 + 1e^0) \\ X_1 &= (1e^0 + 3e^{-\frac{2\pi i 1}{5}} + 5e^{-\frac{2\pi i 2}{5}} + 3e^{-\frac{2\pi i 3}{5}} + 1e^{-\frac{2\pi i 4}{5}}) \\ X_2 &= (1e^0 + 3e^{-\frac{2\pi i 2}{5}} + 5e^{-\frac{2\pi i 4}{5}} + 3e^{-\frac{2\pi i 6}{5}} + 1e^{-\frac{2\pi i 8}{5}}) \\ X_3 &= (1e^0 + 3e^{-\frac{2\pi i 3}{5}} + 5e^{-\frac{2\pi i 6}{5}} + 3e^{-\frac{2\pi i 9}{5}} + 1e^{-\frac{2\pi i 12}{5}}) \\ X_4 &= (1e^0 + 3e^{-\frac{2\pi i 4}{5}} + 5e^{-\frac{2\pi i 8}{5}} + 3e^{-\frac{2\pi i 12}{5}} + 1e^{-\frac{2\pi i 16}{5}}) \end{aligned}$$

$$\begin{aligned} X_0 &= (1 + 3 + 5 + 3 + 1) \\ X_1 &= (1 + 3e^{-\frac{2\pi i}{5}} + 5e^{-\frac{4\pi i}{5}} + 3e^{-\frac{6\pi i}{5}} + 1e^{-\frac{8\pi i}{5}}) \\ X_2 &= (1 + 3e^{-\frac{4\pi i}{5}} + 5e^{-\frac{8\pi i}{5}} + 3e^{-\frac{12\pi i}{5}} + 1e^{-\frac{16\pi i}{5}}) \\ X_3 &= (1 + 3e^{-\frac{6\pi i}{5}} + 5e^{-\frac{12\pi i}{5}} + 3e^{-\frac{18\pi i}{5}} + 1e^{-\frac{24\pi i}{5}}) \\ X_4 &= (1 + 3e^{-\frac{8\pi i}{5}} + 5e^{-\frac{16\pi i}{5}} + 3e^{-\frac{24\pi i}{5}} + 1e^{-\frac{32\pi i}{5}}) \end{aligned}$$

Calculando cada exponencial complexa com a relação de Euler e substituindo os

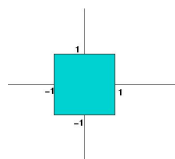
resultados na equação acima:

$$\begin{aligned}
 e^{\frac{-2\pi i}{5}} &= \cos\left(\frac{2\pi}{5}\right) - i\sin\left(\frac{2\pi}{5}\right) = 0.30902 - 0.95106i \\
 e^{\frac{-4\pi i}{5}} &= \cos\left(\frac{4\pi}{5}\right) - i\sin\left(\frac{4\pi}{5}\right) = -0.80902 - 0.58779i \\
 e^{\frac{-6\pi i}{5}} &= \cos\left(\frac{6\pi}{5}\right) - i\sin\left(\frac{6\pi}{5}\right) = -0.80902 + 0.58779i \\
 e^{\frac{-8\pi i}{5}} &= \cos\left(\frac{8\pi}{5}\right) - i\sin\left(\frac{8\pi}{5}\right) = 0.30902 - 0.95106i \\
 e^{\frac{-12\pi i}{5}} &= \cos\left(\frac{12\pi}{5}\right) - i\sin\left(\frac{12\pi}{5}\right) = 0.30902 - 0.95106i \\
 e^{\frac{-16\pi i}{5}} &= \cos\left(\frac{16\pi}{5}\right) - i\sin\left(\frac{16\pi}{5}\right) = -0.80902 + 0.58779i \\
 e^{\frac{-18\pi i}{5}} &= \cos\left(\frac{18\pi}{5}\right) - i\sin\left(\frac{18\pi}{5}\right) = 0.30902 + 0.95106i \\
 e^{\frac{-24\pi i}{5}} &= \cos\left(\frac{24\pi}{5}\right) - i\sin\left(\frac{24\pi}{5}\right) = -0.80902 - 0.58779i \\
 e^{\frac{-32\pi i}{5}} &= \cos\left(\frac{32\pi}{5}\right) - i\sin\left(\frac{32\pi}{5}\right) = 0.30902 - 0.95106i
 \end{aligned}$$

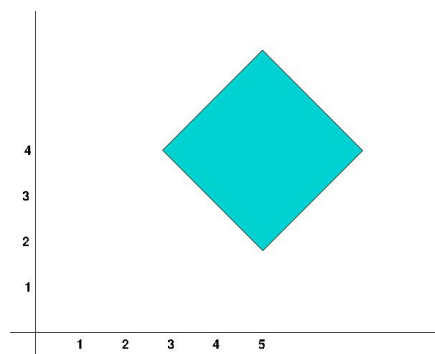
Temos então que o resultado da DFT é:

$$\begin{aligned}
 X[x] = 13.0, & -4.236067 - 3.077683i, 0.236067 + 0.726542i, \\
 & 0.236067 - 0.726542i, -4.236067 + 3.077683i
 \end{aligned}$$

- Q3.** – Calcule (apresente os cálculos) dos descritores de Fourier das figuras 5a e 5b. Lembre-se que os pontos da borda do quadrado serão representados por pontos no plano de Argand-Gauss. Isto é, cada ponto no plano passa a ser um número complexo e a borda passa a ser um vetor de pontos complexos, como num sinal, mas com valores complexos.



(5a) Quadrado de lado 1



(5b) Quadrado de lado 3

Figura 5: Figuras no plano cartesiano

**Resultados para a primeira imagem**

Para que possamos calcular os descritores de Fourier, devemos representar as coordenadas do quadrado (1,1), (-1,1), (-1,-1) e (1,-1), como coordenadas no plano de Argand-Gauss que, neste caso, são:  $(1 + i, -1 + i, -1 - i, 1 - i)$ .

Os descritores de Fourier podem ser calculados a partir da DFT [1]:

$$X_k = \sum_{m=0}^{N-1} x_m e^{\frac{-2\pi i k m}{N}}, \quad \text{para } k = 0, 1, 2, \dots, N-1$$

Da mesma forma que no item anterior, para realizar os cálculos devemos utilizar a identidade de Euler:

$$e^{-i\pi} = \cos \pi - i \sin \pi$$

Temos que  $N=4$ , ou seja, o número de pontos no plano de Argand-Gauss, logo:

$$\begin{aligned} X_0 &= (1+i)e^0 + (-1+i)e^0 + (-1-i)e^0 + (1-i)e^0 \\ X_1 &= (1+i)e^0 + (-1+i)e^{\frac{-2\pi i 1}{4}} + (-1-i)e^{\frac{-2\pi i 2}{4}} + (1-i)e^{\frac{-2\pi i 3}{4}} \\ X_2 &= (1+i)e^0 + (-1+i)e^{\frac{-2\pi i 2}{4}} + (-1-i)e^{\frac{-2\pi i 4}{4}} + (1-i)e^{\frac{-2\pi i 6}{4}} \\ X_3 &= (1+i)e^0 + (-1+i)e^{\frac{-2\pi i 3}{4}} + (-1-i)e^{\frac{-2\pi i 6}{4}} + (1-i)e^{\frac{-2\pi i 9}{4}} \end{aligned}$$

$$\begin{aligned} X_0 &= (1+i) + (-1+i) + (-1-i) + (1-i) \\ X_1 &= (1+i) + (-1+i)e^{\frac{-2\pi i}{4}} + (-1-i)e^{\frac{-4\pi i}{4}} + (1-i)e^{\frac{-6\pi i}{4}} \\ X_2 &= (1+i) + (-1+i)e^{\frac{-4\pi i}{4}} + (-1-i)e^{\frac{-8\pi i}{4}} + (1-i)e^{\frac{-12\pi i}{4}} \\ X_3 &= (1+i) + (-1+i)e^{\frac{-6\pi i}{4}} + (-1-i)e^{\frac{-12\pi i}{4}} + (1-i)e^{\frac{-18\pi i}{4}} \end{aligned}$$

Calculando cada exponencial complexa com a relação de Euler e substituindo os resultados na equação acima:

$$\begin{aligned} e^{\frac{-2\pi i}{4}} &= \cos\left(\frac{2\pi}{4}\right) - i \sin\left(\frac{2\pi}{4}\right) = -i \\ e^{\frac{-4\pi i}{4}} &= \cos\left(\frac{4\pi}{4}\right) - i \sin\left(\frac{4\pi}{4}\right) = -1 \\ e^{\frac{-6\pi i}{4}} &= \cos\left(\frac{6\pi}{4}\right) - i \sin\left(\frac{6\pi}{4}\right) = i \\ e^{\frac{-8\pi i}{4}} &= \cos\left(\frac{8\pi}{4}\right) - i \sin\left(\frac{8\pi}{4}\right) = 1 \\ e^{\frac{-12\pi i}{4}} &= \cos\left(\frac{12\pi}{4}\right) - i \sin\left(\frac{12\pi}{4}\right) = -1 \\ e^{\frac{-18\pi i}{4}} &= \cos\left(\frac{18\pi}{4}\right) - i \sin\left(\frac{18\pi}{4}\right) = -i \end{aligned}$$

Temos então que o resultado dos descritores de Fourier da imagem é:

$$X_0 = 0$$

$$X_1 = (1 + i) + (-1 + i) * (-i) + (-1 - i) * (-1) + (1 - i) * (i)$$

$$X_2 = (1 + i) + (-1 + i) * (-1) + (-1 - i) * (1) + (1 - i) * (-1)$$

$$X_3 = (1 + i) + (-1 + i) * (i) + (-1 - i) * (-1) + (1 - i) * (-i)$$

$$X[x] = 0.0, 4 + 4i, 0.0, 0.0$$

### Resultados para a segunda imagem

Assim como fizemos no item anterior, devemos calcular as coordenadas da imagem para encontrarmos os descritores de Fourier. Sabemos a coordenada x dos pontos inferior e superior do quadrilátero, que é 5 e, também, conhecemos a coordenada do eixo y para os pontos mais à esquerda e à direita que é igual a 4.

Para as demais coordenadas, basta calcular o valor da diagonal do quadrilátero que, neste caso, é a hipotenusa dos dois triângulos retângulos formados ao se dividir o quadrilátero ao meio, cujos lados possuem comprimento 3.

Com base no teorema de Pitágoras temos que a hipotenusa c desse triângulo é dado por:

$$c^2 = a^2 + b^2$$

$$c^2 = 3^2 + 3^2$$

$$c = \sqrt{18}$$

Consequentemente, temos que a diagonal é  $d = \sqrt{18} = 4.24264$  e as coordenadas dos pontos da imagem, que denominei de  $f$ , são:

$$f = (5, 4 + \frac{4.24264}{2}), (5 - \frac{4.24264}{2}, 4), (5, 4 - \frac{4.24264}{2}), (5 + \frac{4.24264}{2}, 4)$$

$$f = (5, 6.12132), (2.87868, 4), (5, 1.87868), (7.12132, 4)$$

Agora, devemos representar as coordenadas por pontos no Plano de Argand-Gauss, que são:

$$f = (5 + 6.12132i, 2.87868 + 4i, 5 + 1.87868i, 7.12132 + 4i)$$

Os descritores de Fourier podem ser calculados a partir da DFT [1]:

$$X_k = \sum_{m=0}^{N-1} x_m e^{\frac{-2\pi i k m}{N}}, \quad \text{para } k = 0, 1, 2, \dots, N-1$$

Novamente, utilizando a identidade de Euler:

$$e^{-i\pi} = \cos \pi - i \sin \pi$$

Temos que  $N=4$ , ou seja, o número de pontos no plano de Argand-Gauss, logo:

$$X_0 = (5 + 6.12132i)e^0 + (2.87868 + 4i)e^0 + (5 + 1.87868i)e^0 + (7.12132 + 4i)e^0$$

$$X_1 = (5 + 6.12132i)e^0 + (2.87868 + 4i)e^{\frac{-2\pi i}{4}} + (5 + 1.87868i)e^{\frac{-2\pi i}{4}} + (7.12132 + 4i)e^{\frac{-2\pi i}{4}}$$

$$X_2 = (5 + 6.12132i)e^0 + (2.87868 + 4i)e^{\frac{-2\pi i \cdot 2}{4}} + (5 + 1.87868i)e^{\frac{-2\pi i \cdot 2}{4}} + (7.12132 + 4i)e^{\frac{-2\pi i \cdot 2}{4}}$$

$$X_3 = (5 + 6.12132i)e^0 + (2.87868 + 4i)e^{\frac{-2\pi i \cdot 3}{4}} + (5 + 1.87868i)e^{\frac{-2\pi i \cdot 3}{4}} + (7.12132 + 4i)e^{\frac{-2\pi i \cdot 3}{4}}$$

$$X_0 = (5 + 6.12132i) + (2.87868 + 4i) + (5 + 1.87868i) + (7.12132 + 4i)$$

$$X_1 = (5 + 6.12132i) + (2.87868 + 4i)e^{\frac{-2\pi i}{4}} + (5 + 1.87868i)e^{\frac{-4\pi i}{4}} + (7.12132 + 4i)e^{\frac{-6\pi i}{4}}$$

$$X_2 = (5 + 6.12132i) + (2.87868 + 4i)e^{\frac{-4\pi i}{4}} + (5 + 1.87868i)e^{\frac{-8\pi i}{4}} + (7.12132 + 4i)e^{\frac{-12\pi i}{4}}$$

$$X_3 = (5 + 6.12132i) + (2.87868 + 4i)e^{\frac{-6\pi i}{4}} + (5 + 1.87868i)e^{\frac{-12\pi i}{4}} + (7.12132 + 4i)e^{\frac{-18\pi i}{4}}$$

Calculando cada exponencial complexa com a relação de Euler e substituindo os resultados na equação acima:

$$e^{\frac{-2\pi i}{4}} = \cos\left(\frac{2\pi}{4}\right) - i\sin\left(\frac{2\pi}{4}\right) = -i$$

$$e^{\frac{-4\pi i}{4}} = \cos\left(\frac{4\pi}{4}\right) - i\sin\left(\frac{4\pi}{4}\right) = -1$$

$$e^{\frac{-6\pi i}{4}} = \cos\left(\frac{6\pi}{4}\right) - i\sin\left(\frac{6\pi}{4}\right) = i$$

$$e^{\frac{-8\pi i}{4}} = \cos\left(\frac{8\pi}{4}\right) - i\sin\left(\frac{8\pi}{4}\right) = 1$$

$$e^{\frac{-12\pi i}{4}} = \cos\left(\frac{12\pi}{4}\right) - i\sin\left(\frac{12\pi}{4}\right) = -1$$

$$e^{\frac{-18\pi i}{4}} = \cos\left(\frac{18\pi}{4}\right) - i\sin\left(\frac{18\pi}{4}\right) = -i$$

Temos então que o resultado dos descritores de Fourier da imagem é:

$$X_0 = (5 + 6.12132i) + (2.87868 + 4i) + (5 + 1.87868i) + (7.12132 + 4i)$$

$$X_1 = (5 + 6.12132i) + (2.87868 + 4i) * (-i) + (5 + 1.87868i) * (-1) + (7.12132 + 4i) * (i)$$

$$X_2 = (5 + 6.12132i) + (2.87868 + 4i) * (-1) + (5 + 1.87868i) * (1) + (7.12132 + 4i) * (-1)$$

$$X_3 = (5 + 6.12132i) + (2.87868 + 4i) * (i) + (5 + 1.87868i) * (-1) + (7.12132 + 4i) * (-i)$$

$$X[x] = 20.0 + 16i, 8.428528i, 0.0, 0.0$$

- Para confirmar que seus cálculos estão corretos, implemente um programa em Python que receba como entrada um vetor de números complexos (que são as coordenadas das bordas) e retorne os descritores de Fourier do vetor de entrada. Você pode usar as funções fornecidas pela biblioteca NUMPY para facilitar a programação.

O programa foi implementado em Python, com a biblioteca Numpy e está localizado em **source/fourier\_descriptors.py**.



- Para reconstruir a curva, faça uma função que receba um vetor com os descritores de Fourier, um número  $N$  de descritores a serem usados e grafique os pontos num plano cartesiano (para fazer a mesma figura que fizemos nos slides das aulas 15 e 16).

O programa foi implementado em Python, com a biblioteca Numpy e está localizado em **source/ifd.py**.

Basta executar o arquivo e passar um array com os descritores para que os pontos sejam desenhados. O tamanho  $N$  do vetor é calculado pelo programa.

A Figura 6, por exemplo, foi gerada com o vetor de entrada:

$0 + 4j, 2 + 4j, 3 + 3j, 3 + 2j, 2 + 1j, 2 - 1j, 1 - 2j, 0 - 3j, -1 - 3j, -2 - 2j, -2 + 0j, -2 + 1j, -1 + 3j$

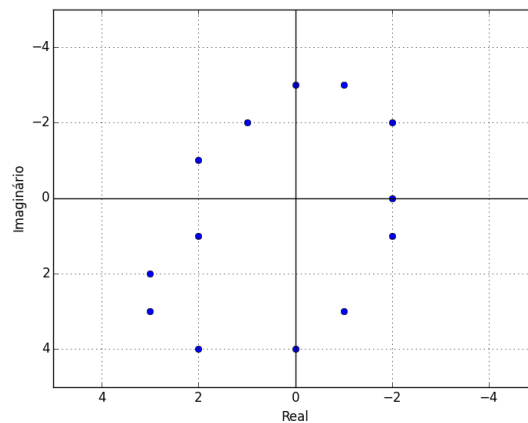


Figura 6: Exemplo de pontos de contorno a partir de descritores de Fourier

#### Q4. Apenas para os alunos de pós-graduação

- Leia o artigo do Torre e do Poggio <ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-768.pdf> e faça um resumo de acordo com as indicações que deixei no paca (artigos sobre como fazer um resumo).

---

# Resumo: *On Edge Detection*

Rodrigo Augusto Dias Faria

Instituto de Matemática e Estatística – Universidade de São Paulo (IME-USP)  
Departamento de Ciência da Computação

rofaria@ime.usp.br, rodrigoadfaria@gmail.com

## 1. Resumo

### 1.1. Motivação

Uma borda em uma imagem é caracterizada por um conjunto de pixels conectados que ficam na fronteira entre duas regiões de uma imagem com propriedades relativamente distintas de nível de cinza [3]. Em outras palavras, as bordas caracterizam-se por mudanças na intensidade da imagem em termos de aspectos físicos que as originaram [7].

Em processamento de imagens, a percepção da mudança de intensidade pode ser obtida por vários métodos que, em geral, utilizam de um operador local diferencial para identificar as bordas.

Torre e Poggio subdividem a detecção de bordas em duas etapas: uma etapa de filtragem para eliminação de ruídos da imagem e uma segunda etapa que caracteriza-se pela avaliação das derivadas da intensidade da imagem, sendo esta última classificada como um problema de diferenciação numérica.

Sabendo que a diferenciação numérica não é robusta contra ruído, Torre e Poggio mostram que a diferenciação de uma função  $f(x)$  é um típico problema mal-posto e pode ser visto como a solução do problema inverso

$$g(x) = Af(x) \tag{1}$$

onde  $Af(x)$  é o operador integral

$$\int_{-\infty}^x f(\tilde{x})d\tilde{x} = \int_{-\infty}^{\infty} h(x - \tilde{x})f(\tilde{x})d\tilde{x} \tag{2}$$

e  $h$  é a função degrau.

Uma vez que a diferenciação numérica é um problema mal-posto no sentido de Hadamard, conforme mostrado pelos autores, a motivação dá-se, então, em regularizar o problema para que ele se torne bem-posto por meio de uma operação de filtragem antes da diferenciação.

Pode-se dizer que não há nenhuma crise evidente na área pesquisada pelos autores, mas há demonstrações de que a proposta de realizar uma operação de filtragem antes da diferenciação reduz a sensibilidade a ruído a que esta etapa está suscetível, algo não considerado por alguns métodos,

tais como o proposto por Shanmugam, Dickey e Green que não faz nenhuma referência explícita à etapa de diferenciação.

Vale ressaltar que, mesmo uma pequena quantidade de ruído na imagem produz um efeito crítico na diferenciação, causando perturbação nos dados.

Em suma, Torre e Poggio deixam claro que para tornar o problema de diferenciação bem-posto é necessário que as intensidades da imagem sejam regularizadas, cujo processo pode ser feito por uma operação de filtragem antes da diferenciação e, para tal, eles estudaram propriedades de diferentes tipos de filtros, bem como a relação entre vários operadores diferenciais bidimensionais. Além disso, propriedades geométricas e topológicas desses operadores também são avaliadas com o objetivo de obter bordas mais suaves e contornos fechados, dentre outras características interessantes.

## 1.2. Contribuição

Há três métodos principais de regularização de acordo com a definição de Bertero, além dos operadores de estabilização de Tikhonov.

Poggio *et al.* [6], com base em dois métodos dos propostos por Bertero e no funcional de Tikhonov como  $P = d^2/dx^2$ , reformulou os resultados obtidos por Schoenberg, em um trabalho que, à época, ainda não havia sido publicado, provando que a solução para a regularização do problema da diferenciação numérica em dados não exatos, pode ser obtida pela convolução dos dados com um filtro, que neste caso é uma spline cúbica.

O resultado foi estendido por Torre e Poggio, citado em forma de teorema neste trabalho, provando que, além da convolução poder ser realizada com um filtro spline cúbico, ele é muito similar à uma Gaussiana. Essa é a prova mais simples e rigorosa de que um filtro Gaussiano representa a correta operação a ser realizada antes da diferenciação para detecção de borda.

Essa justificativa dá o potencial de inovação e importante contribuição para mostrar que a filtragem seguida pela diferenciação podiam ser reconhecidas como operações presentes na maioria dos métodos de detecção de borda existentes até então.

Vale ressaltar, ainda, que Poggio [6] analisa o papel do parâmetro de regularização  $\lambda$ , sua conexão com a escala do filtro Gaussiano e discute métodos para encontrar o parâmetro  $\lambda$  ótimo.

Outra contribuição importante feita por Torre e Poggio foi a observação de três tipos de filtros: passa banda, suporte limitado e de incerteza mínima, sendo que o filtro passa banda, bem como de incerteza mínima são bons operadores de regularização para a diferenciação no sentido de

Tikhonov.

### 1.3. Metodologia

Os autores fundamentam o trabalho, a princípio, postando a natureza do problema de diferenciação numérica como um problema mal-posto de encontrar  $x$  dos dados  $y$  tal que  $Az = y$ , sendo que, para a regularização é necessário a escolha de normas adequadas  $\|.\|$  e de um funcional de estabilização  $\|Pz\|$ .

Logo, três métodos de regularização propostos por Bertero são mostrados. O primeiro consiste em encontrar a função  $z$  que minimiza  $\|Az - y\|$  e satisfaz a restrição  $\|Pz\| < C_1$ , onde  $C_1$  é uma constante.

O segundo calcula a função  $z$  que está suficientemente próxima dos dados e é mais regular, minimizando  $\|Pz\|$  e obedecendo à restrição  $\|Az - y\| \leq C_2$ , onde  $C_2$  é uma constante.

O último método consiste em encontrar a função  $z$  que minimiza  $\|Az - y\|^2 + \lambda\|Pz\|^2$ , onde  $\lambda$  é um parâmetro de regularização que controla o grau de regularização da solução e a aproximação dos dados.

Outra forma de regularizar o problema de diferenciação são os operadores de Tikhonov que são equivalentes à filtragem dos dados com filtros passa-baixa.

Conforme supra citado, Poggio *et al.* [6] reformulou os resultados obtidos por Schoenberg, na forma de um teorema que diz que a interpolação de uma spline cúbica em uma estrutura regular, satisfazendo o segundo método de regularização com  $P = d^2/dx^2$ , pode ser obtida pela convolução dos dados com um filtro spline cúbico, que é uma função  $L^4$  de Schoenberg, onde  $P$  corresponde a forma mais simples do funcional de Tikhonov. Logo, a regularização pode ser obtida realizando a convolução dos dados com a primeira derivada do filtro  $L^4$  de Schoenberg.

No caso de dados não exatos, Torre e Poggio utilizaram o terceiro método de regularização que, por conseguinte, originou um novo teorema proposto neste artigo provando que a solução pode ser obtida pela convolução dos dados com um filtro, o qual é uma spline cúbica e é muito similar a uma Gaussiana.

Essa implicação é determinante para demonstrar que a convolução dos dados pode, então, ser realizada com um filtro Gaussiano.

Em seguida, Torre e Poggio avaliaram três tipos de filtros que podem ser utilizados nessa etapa, observando o tipo da derivada, se direcionais ou invariantes à rotação, e o tipo de representação, se zeros ou extremos. Os filtros de banda limitada satisfazem todas as condições de Tikhonov para a regularização da diferenciação, bem como os filtros de incerteza mínima.

Já o caso dos filtros de suporte limitado, os autores mostram que o *blurring* é uma classe desse tipo de filtro que falha em atender algumas propriedades para que possa ser aplicado à convolução. Em particular, a condição  $\tilde{F}(\omega, \alpha)j\omega$  pertence a  $L_2(-\infty, \infty)$ , não é satisfeita, uma vez que a diferenciação introduz de volta altas frequências na mesma quantidade em que elas foram removidas por este tipo de filtragem.

Vale lembrar que, a função Gaussiana  $e^{-x^2/\sigma^2}$  é a função real  $f \in L^2$  que minimiza a incerteza definida por  $\Delta U = \Omega X$  no domínio da frequência e do espaço e, por essa razão, foi escolhida como o filtro ótimo por Marr e Hildreth na elaboração do trabalho que originou o operador Laplaciano do Gaussiano.

Torre e Poggio fazem ainda uma comparação entre a função prolato e a Gaussiana, com o objetivo de mostrar uma aproximação satisfatória entre ambas, de acordo com parâmetros pré definidos. Além disso, eles mostram a robustez das propriedades de regularização do filtro Gaussiano, comparando a convolução do mesmo com uma imagem  $I(x, y)$  vista como a solução da equação do calor no caso bidimensional.

Tendo devidamente estudado a etapa de regularização e filtragem, a etapa de diferenciação foi dividida nos operadores diferenciais direcionais e os operadores diferenciais invariantes à rotação.

O primeiro provoca manchas nos contornos com passagens em zero, mas não pelo uso do operador e sim pela distorção introduzida por um operador de largura demasiada.

No caso dos operadores invariantes à rotação, dois deles merecem destaque por serem amplamente utilizados em função de possuírem características interessantes que são o Laplaciano  $\nabla^2$ , que é um operador linear e a derivada segunda ao longo do gradiente  $\partial^2/\partial n^2$ , que é um operador não linear.

Propriedades dos operadores invariantes à rotação são, então, derivadas, especialmente àquelas relacionadas a passagens em zero que mostram, por exemplo, que em dadas condições, as derivadas tanto do Laplaciano quanto da derivada segunda ao longo do gradiente coincidem. Outra propriedade interessante destes operadores é que eles são rotacionalmente simétricos, conforme visto em suas respectivas representações em coordenadas polares.

Um estudo das propriedades geométricas e topológicas dos contornos também foi fornecido mostrando que, em geral, operadores invariantes à rotação detectam bordas mais suaves, curvas fechadas, características normalmente não encontradas nos operadores direcionais.

Esse estudo se faz importante, já que contornos são frequentemente obtidos a partir de bordas e representam fronteiras entre objetos na imagem.

Sendo assim, eles precisam ser curvas fechadas e quando são obtidos a partir das bordas, é necessário interligá-las, a fim de se obter um contorno fechado, evitando que apenas pontos ou extremos da imagem resultante da derivação sejam a representação das bordas observadas.

Por fim, os autores estabeleceram uma comparação com os principais resultados obtidos e um *overview* de alguns métodos de detecção de borda como o *Difference of Boxes* proposto por Binford, o operador de Shanmugam, Dickey e Green, o Laplaciano do Gaussiano (LOG) de Marr e Hildreth, o método de Haralick e de Canny.

É evidente que o artigo está extremamente bem embasado, com referências a importantes trabalhos que contribuíram para o aperfeiçoamento dos métodos de detecção de borda ao longo dos anos, além de ter demonstrações matemáticas e teoremas consistentes que fundamentam o objetivo principal de encontrar técnicas de regularização para o problema de diferenciação e mostrar que há uma dada ordem na operação desta última com a filtragem da imagem.

#### 1.4. Conclusão

Conforme já mencionado, Torre e Poggio concluíram que a filtragem tem a atribuição de regularizar o problema mal-posto da diferenciação e que ela deve ser realizada antes desta última.

Outro ponto importante é que funções prolato regularizam melhor a imagem já que elas deixam a imagem "inteira" e com banda limitada, porém, o filtro Gaussiano tem duas vantagens sobre funções prolato. A primeira é que elas não geram passagens em zero quando o tamanho do filtro é incrementado. A segunda, no caso bidimensional, a Gaussiana pode ser decomposta no produto de dois filtros unidimensionais, o que reduz drasticamente a quantidade de cálculos envolvidos na aplicação do filtro.

Uma outra observação importante é que os filtros rotacionalmente simétricos garantem uma alta probabilidade de contornos fechados e bordas mais suaves, o que torna a escolha desses operadores bastante óbvia em detrimento dos filtros direcionais.

Na etapa de diferenciação, quando da aplicação de operadores direcionais, duas derivadas em apenas duas direções são necessárias para encontrar as mudanças de intensidade. No caso de operadores invariantes à rotação, a derivada de segunda ordem ao longo do gradiente tem uma melhor performance do que o Laplaciano, porém, este último é comutativo com a convolução.

Finalmente, propriedades topológicas e geométricas são derivadas de forma a caracterizar os tipos de mudanças de intensidade da imagem em termos de propriedades físicas que as geraram.

As práticas de detecção de borda podem ser alteradas de forma a considerar os resultados obtidos pelos autores, a fim de gerarem bordas mais suaves e com menos ruído, o que faz os resultados serem, de fato, genéricos.

Torre e Poggio deixam uma questão em aberto em relação ao funcionamento do sistema visual humano e os operadores diferenciais, ou seja, quais operadores e em que combinações eles devem ser organizados para realizar tarefas sob diferentes condições.

Em suma, pode-se tomar como lições do artigo que a identificação de transições de intensidade na imagem para detecção de bordas é um problema de diferenciação numérica que precisa ser regularizado, no intuito de reduzir a sensibilidade a ruídos. Além disso, para se obter resultados satisfatórios, tais como curvas fechadas e bordas mais suaves, características de diferentes filtros e propriedades geométricas dos operadores diferenciais devem ser detalhadamente avaliadas.

---

- O que é um problema mal-posto?

Um problema matemático é bem-posto, de acordo com a definição de Hadamard, se ele cumpre as seguintes condições:

1. Existe solução;
2. A solução é única;
3. A solução tem uma dependência contínua com os dados de entrada (quando o problema não é somente bem-posto, mas também, bem-condicionado, portanto, é robusto contra ruído);

Logo, um problema é dito mal-posto se alguma das condições acima não é satisfeita.

No artigo em questão, o problema mal-posto trata-se da diferenciação de uma função  $f(x)$  que é um típico problema mal-posto e pode ser visto como a solução do problema inverso visto no resumo em (1) e (2).

- O que é regularização?

Regularização, em linhas gerais, é a aproximação de um problema mal-posto por uma família de problemas bem-postos [2].

Conforme visto no item anterior, como temos um problema de diferenciação numérica e é sabido que ele é mal-posto, os autores avaliaram alguns métodos de regularização para, então, aproximar uma solução.

Um dos possíveis métodos são os operadores de Tikhonov que, para equações de convolução como em (2), os operadores correspondem à convolução de  $g(x)$  com um filtro  $F(x, \alpha)$ , onde  $\alpha$  é um parâmetro.

- Qual a importância do teorema apresentado no artigo?

No caso de dados não exatos, Poggio *et al.* [6] utilizou o terceiro método de regularização proposto por Bertero que, por conseguinte, originou um novo teorema proposto por Torre e Poggio provando que a solução para o problema de diferenciação pode ser obtida pela convolução dos dados com um filtro, o qual é uma spline cúbica e é muito similar a uma Gaussiana.

Essa implicação é determinante para demonstrar que a convolução dos dados pode, então, ser realizada com um filtro Gaussiano.

- O que são filtros de banda limitada? Qual a sua importância no artigo?

Também conhecidos como filtros passa-banda, permitem a passagem das frequências localizadas em uma faixa ou banda específica, enquanto atenua ou completamente suprime todas as outras frequências [5].

Sua importância no artigo se dá na capacidade de identificar bordas, uma vez que as baixas frequências são suprimidas, ao mesmo tempo que o ruído é reduzido pela atenuação de frequências altas. Além disso, eles atendem a todas as condições de Tikhonov na regularização da diferenciação numérica.



- Quais são os métodos de encontrar borda apresentados no artigo?  
Os métodos de detecção de borda apresentados no artigo são:

1. **Difference of Boxes - DOB**

Proposto por Herskovitz e Binford, o filtro DOB é do tipo suporte limitado e utiliza a função de Haar para filtragem direcional ou diferença de funções para filtragem rotacional. A função de Haar tem a propriedade de ser um filtro ótimo de suporte limitado que maximiza a relação sinal-ruído.

2. **Shanmugam, Dickey and Green**

O operador proposto por Shanmugam, Dickey e Green, como citado por Torre e Poggio, é um filtro ótimo na medida em que produz o máximo de energia dentro de um intervalo de um dado espaço ao redor da borda. Eles analisaram os casos contínuo e discreto e concluíram que a implementação pode ser realizada por meio da Transformada Rápida de Fourier (FFT).

3. **Laplacian of Gaussian - LOG**

Proposto por Marr e Hildreth, o Laplaciano do Gaussiano combina a filtragem Gaussiana para a suavização da imagem, com o operador Laplaciano que localiza as bordas pela presença de um cruzamento em zero na derivada segunda com um pico acentuado correspondente à derivada primeira [5].

A saída do operador é dada pela operação de convolução

$$(\nabla^2 G(x, y)) * f(x, y) \quad (3)$$

onde  $f(x, y)$  é uma imagem suavizada por uma função Gaussiana.

4. **Haralick**

O método proposto por Haralick consiste em marcar um pixel como parte integrante da borda se na sua vizinhança há passagens em zero da segunda derivada ao longo do gradiente. Com o propósito de calcular as derivadas, Haralick aproxima e interpola os valores de intensidade amostrados com polinômios discretos de Chebychev.

5. **Canny**

De acordo com Pedrini [5], com o intuito de otimizar a localização de pontos de borda na presença de ruído, o método de Canny suaviza a imagem por meio de um filtro Gaussiano e, em seguida, a magnitude e a direção do gradiente são calculadas utilizando aproximações baseadas em diferenças finitas para as derivadas parciais.

Após o cálculo do gradiente, a borda é localizada utilizando supressão não-máxima, ou seja, apenas os pontos cuja magnitude seja localmente máxima na direção do gradiente.

Fragmentos espúrios causados pela presença de ruído ou textura fina são então removidos com a aplicação de dois limiares diferentes, cujo processo é conhecido como *limiarização por histerese*.

## Referências

- [1] S Broughton. *Discrete Fourier analysis and wavelets : applications to signal and image processing*. Hoboken, N.J: Wiley, 2009. ISBN: 978-0-470-29466-6.
- [2] Heinz W. Engl. “Necessary and sufficient conditions for convergence of regularization methods for solving linear operator equations of the first kind”. Em: *Numerical Functional Analysis and Optimization* 3 (2 jan. de 1981). DOI: 10.1080/01630568108816087. URL: <http://www.tandfonline.com/doi/pdf/10.1080/01630568108816087>.
- [3] Rafael Gonzalez. *Digital Image Processing*. Upper Saddle River, N.J: Prentice Hall, 2002. ISBN: 0-201-18075-8.
- [4] Neal Krawetz. “A Picture’s Worth....” Em: *Black Hat Briefings DC* 2 (2008), pp. 16–20.
- [5] Hélio Pedrini. *Análise de imagens digitais princípios, algoritmos e aplicações*. São Paulo: Thomson Learning, 2008. ISBN: 978-85-221-0595-3.
- [6] T Poggio, H Voorhees e A Yuille. “A regularized solution to edge detection”. Em: *Journal of Complexity* 4.2 (1988), pp. 106–123. ISSN: 0885-064X. DOI: [http://dx.doi.org/10.1016/0885-064X\(88\)90024-6](http://dx.doi.org/10.1016/0885-064X(88)90024-6). URL: <http://www.sciencedirect.com/science/article/pii/0885064X88900246>.
- [7] Tomaso A. Torre Vincent; Poggio. “On Edge Detection”. Em: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8 (2 1986). DOI: 10.1109/tpami.1986.4767769. URL: <http://cbcl.mit.edu/people/poggio/journals/torre-poggio-IEEE TPAMI-1986.pdf>.
- [8] Yan Zhao, Anthony Sutardja e Omar Ramadan. *Digital Image Manipulation Forensic*. 2015. URL: <http://www.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-85.html>.