

UNIVERSIDADE DE SÃO PAULO  
Instituto de Ciências Matemáticas e de Computação

## **Reconstrução de curvas por meio de características robustas extraídas de imagens**

Relatório científico parcial de iniciação  
científica referente ao processo FAPESP  
nº 2020/07224-5 para o período entre  
01/10/2020 e 10/03/2021.

**Bolsista:** André Luís Mendes Fakhoury

**Orientador:** João do Espírito Santo Batista Neto

Brasil

2021

# Sumário

<b>1</b>	<b>Introdução . . . . .</b>	<b>3</b>
1.1	O projeto e cronograma original . . . . .	3
1.2	Reordenação do cronograma . . . . .	4
1.3	Organização do relatório . . . . .	5
<b>2</b>	<b>Atividades realizadas . . . . .</b>	<b>5</b>
2.1	Pré-processamento de imagens . . . . .	5
2.2	Análise da curvatura . . . . .	6
2.3	Reconstrução de curvas . . . . .	9
<b>3</b>	<b>Próximas atividades . . . . .</b>	<b>15</b>
	<b>Referências . . . . .</b>	<b>16</b>

# 1 Introdução

Este documento tem como objetivo apresentar as atividades realizadas pelo bolsista André Luís Mendes Fakhoury no período de outubro de 2020 a março de 2021, referente ao projeto de Iniciação Científica, processo FAPESP nº 2020/07224-5. O trabalho, intitulado “Reconstrução de curvas por meio de características robustas extraídas de imagens”, é parte de umas das linhas de pesquisa do projeto temático FAPESP de nº 2019/07316-0, que visa a reconstrução de faces humanas a partir de informações reduzidas do domínio.

## 1.1 O projeto e cronograma original

Um dos desafios encontrados em visão computacional é o mapeamento de características robustas entre espaços bidimensionais e tridimensionais. Este processo pode ser simplificado com a redução de informações a serem analisadas: como representar curvas a partir de algum conjunto reduzido de pontos.

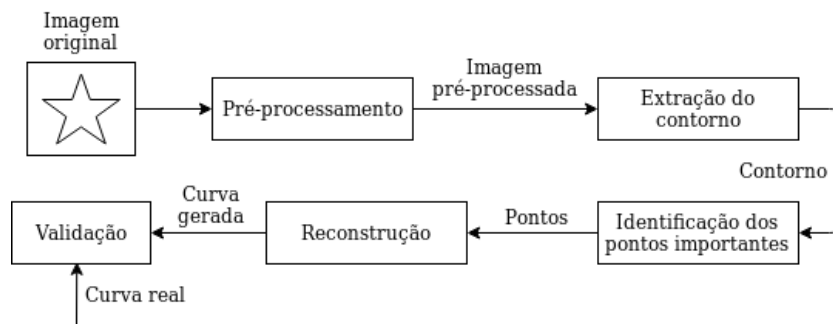
Este projeto visa, principalmente, a extração de características robustas de curvas, e posterior reconstrução delas a partir destas informações reduzidas com um erro mínimo. Os pontos que representam estas características robustas são denominados pontos importantes.

Para este processo, contornos de imagens podem ser extraídos e analisados como curvas discretas e, portanto, reduzidos a finitos pontos importantes. A partir desta premissa, os seguintes objetivos específicos podem ser definidos:

- Extrair características robustas em  $\mathbb{R}^2$  para reconstrução de curvas com alta precisão a partir de imagens.
- Pré-processar as imagens com eliminação de ruídos, binarização e consequente segmentação;
- Extrair atributos de formas das imagens (contorno e cálculo da curvatura);
- Determinar as características robustas por análise da curvatura;
- Reconstruir as formas a partir das características robustas, por meio de curvas poligonais e operadores Laplacianos como sugerido por Sorkine (2006) e
- Aferir a qualidade da reconstrução a partir da curva original.

As principais etapas de desenvolvimento deste projeto são ilustradas no diagrama de blocos da figura 1. O projeto, como inicialmente proposto, apresentou o cronograma representado na tabela 1.

Figura 1 – Diagrama de bloco das etapas de desenvolvimento



Fonte: Elaborada pelo autor.

Atividades	Meses de trabalho					
	1º e 2º	3º e 4º	5º e 6º	7º e 8º	9º e 10º	11º e 12º
Estudo das técnicas de reconstrução de curvas	•	•				
Pré-processamento e Extração de contornos		•	•			
Extração dos pontos importantes		•	•			
Redação do relatório parcial			•			
Implementação da reconstrução de curvas (Sorkine)			•	•	•	
Avaliação e testes				•	•	
Desenvolvimento do relatório final					•	•

Tabela 1 – Cronograma de atividades para 12 meses de trabalho.

## 1.2 Reordenação do cronograma

Durante o segundo semestre de 2020, o professor Dr. Antônio Castelo Filho, pesquisador principal do projeto temático, lecionou a disciplina “Modelagem Geométrica”, de código SME0271, para alunos de graduação e com espelho em “Tópicos em Análise Numérica II (Variedades Computacionais)”, de código SME5850, para a pós-graduação. Nela, foram abordados diversos tópicos referentes a variedades computacionais. Nas reuniões semanais de acompanhamento das atividades do temático por cada membro da equipe, o prof. Castelo e demais pesquisadores incentivaram ao aluno bolsista e demais colegas a cursar a disciplina, visto ser uma grande oportunidade para adquirir conhecimento sobre a modelagem geométrica e análise numérica das várias linhas de pesquisas relacionadas. No caso particular deste projeto, a atividade diretamente relacionada referiu-se à reconstrução de curvas pelo método descrito em Sorkine (2006), originalmente programada para ser desenvolvida na segunda metade deste projeto.

Além do aluno bolsista deste projeto, três alunos de mestrado do projeto temático cursaram a disciplina. Uma das formas de avaliação da disciplina era a co-autoria da escrita de um capítulo de um livro, que ainda será publicado.

Como resultado, a atividade 5 (implementação da reconstrução de curvas) foi antecipada e a atividade 3 (extração dos pontos importantes) postergada para os próximos

meses, sem prejudicar o desenvolvimento do projeto. Ressalto que cursar a disciplina foi muito compensador, tendo contribuído para um melhor entendimento da teoria de reconstrução de curvas a partir de um conjunto de pontos e, sobretudo, no momento da implementação, a ser apresentada neste relatório.

## 1.3 Organização do relatório

As próximas seções deste relatório serão estruturadas da seguinte maneira:

- Na seção 2 serão descritos alguns conceitos estudados e as atividades realizadas;
- Na seção 3 serão vistas as próximas atividades a serem realizadas;

# 2 Atividades realizadas

O objetivo desta seção é relatar as atividades desenvolvidas pelo aluno bolsista no período a que se refere este relatório. A seção 2.1 descreve o que foi desenvolvido na etapa de pré-processamento das imagens utilizadas (envolvendo a utilização de filtros para extração de ruído e a extração do contorno). Na seção 2.2 serão descritas as técnicas utilizadas para a análise da curvatura discreta, e a seção 2.3 terá detalhes sobre a reconstrução de curvas, utilizando o método descrito em Sorkine (2006).

## 2.1 Pré-processamento de imagens

A primeira parte deste projeto refere-se ao pré-processamento de imagens, visando a eliminação de pontos espúrios no contorno de forma a permitir a extração de curvatura que melhor corresponda ao contorno original. A biblioteca de visão computacional OpenCV (2000), por possuir diversas funcionalidades para esta finalidade, foi utilizada neste projeto na sua versão em linguagem de programação C++.

O pré-processamento consistiu das seguintes operações:

1. **Filtros e suavizações (Gaussiano e Perona-Malik).** Nesta etapa, foram testados dois tipos de suavização sobre a imagem original, aplicando o filtro Gaussiano e a difusão anisotrópica de Perona-Malik, objetivando a diminuição de ruídos e suavização da imagem. A difusão anisotrópica de Perona-Malik é mais adequada ao projeto, pois preserva as bordas da imagem (BORROTO-FERNÁNDEZ; MECIAS; HIDALGO, 2013), enquanto o filtro Gaussiano não (SZELISKI, 2011).

2. **Conversão da imagem para tons de cinza.** Sem perda alguma de generalidade para as imagens-teste de folhas utilizadas nesta etapa, foi realizada a conversão RGB-tons de cinza, de forma a simplificar a aplicação das etapas de pré-processamento posteriores.
3. **Binarização por Otsu.** Teve por objetivo segmentar a imagem de folha em duas regiões: *background* e *foreground*. Esta segmentação é feita a partir da delimitação de um limiar, encontrado automaticamente pelo algoritmo de Otsu (BURGER; BURGE, 2016).
4. **Operadores morfológicos (erosão e dilatação).** A partir da imagem binarizada, são realizadas as operações morfológicas de erosão (remoção de alguns pontos isolados e penínsulas na imagem) e a dilatação (preenchimento de alguns buracos indesejados que possam ter sido formados ou amplificados na erosão) (SERRA, 1984).
5. **Extração do contorno pelo método *contour-following*.** A última etapa do pré-processamento é a extração do contorno da imagem. Esta etapa consistiu da implementação do algoritmo denominado *contour-following*, definido em (COSTA; JR, 2009), que procura o primeiro pixel pertencente ao contorno e, a partir deste, localiza, no sentido anti-horário, os próximos pixels pertencentes ao contorno, em uma vizinhança 8-conectada. Para o caso específico das imagens de folha, o contorno obtido é sempre fechado.

Após o final deste fluxo de operações, obtém-se um vetor de pontos ou coordenadas  $(x_i, y_i)$ , referentes ao contorno do objeto segmentado. As principais etapas deste pré-processamento são ilustradas na figura 2.

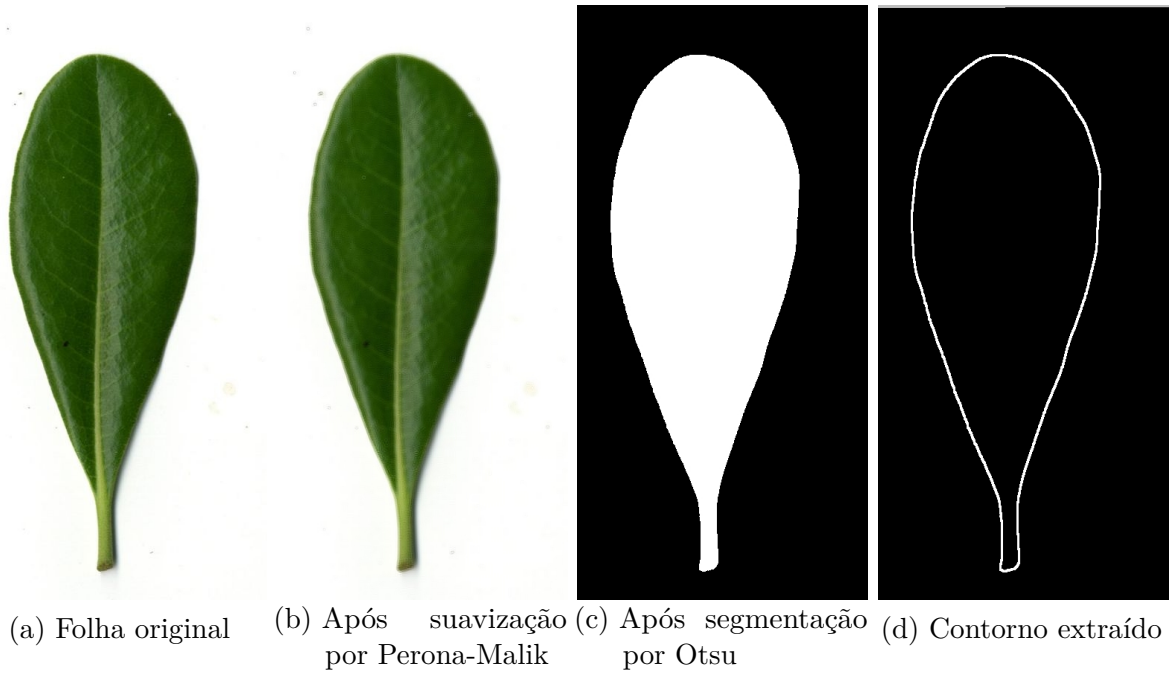
## 2.2 Análise da curvatura

A análise de curvatura é importante para a possibilitar a posterior extração de dos pontos importantes. A curvatura de uma curva regular parametrizada por uma aplicação  $t \rightarrow (x(t), y(t))$ , onde  $x(t)$  e  $y(t)$  são funções de classe  $C^2$ , é dada por (OLIVEIRA; MARROQUIM, 2016):

$$\kappa(t) = \frac{x'(t)y''(t) - y'(t)x''(t)}{(x'(t)^2 + y'(t)^2)^{3/2}}. \quad (2.1)$$

Porém, no escopo do projeto, como a curva é representada por um conjunto discreto de pontos  $(x, y)$  - que representam a posição de cada pixel - as derivadas devem ser calculadas de forma discreta. Este cálculo pode ser realizado a partir de métodos espectrais de Fourier (BRETHERTON, 2019).

Figura 2 – Resultados da etapa de pré-processamento.



Fonte: original de ImageCLEF (2011), e edições elaboradas pelo autor.

Seja  $u_j$  uma aproximação discreta da função  $u(x)$ , com  $n$  pontos de amostra  $x_j \in h, 2h, \dots, ih, \dots, 2\pi - h, 2\pi$ , onde  $h = 2\pi/n$ . Para o caso discreto, pode-se aplicar a versão computacionalmente otimizada da transformada de Fourier (FFT) em  $u_j$ , tal que  $FFT(u_j) \equiv \hat{u}_k$ , em que  $k \in \frac{-n}{2} + 1, \dots, \frac{n}{2}$ . Sabe-se que:

$$FFT\left(\frac{\partial u_j}{\partial x}\right) \equiv ik\hat{u}_k$$

Assim, para obter o valor da derivada, basta calcular a transformada rápida inversa IFFT. O código MATLAB apresentado abaixo calcula a derivada de uma função discreta  $y$  no domínio  $x$  e emprega um filtro Gaussiano com fator de suavização  $\sigma$ . A aplicação de um filtro Gaussiano no cálculo da derivada tem por objetivo evitar a amplificação do efeito de serrilhamento (*aliasing*), quando houver, causado pela transformada de Fourier (LI, 1987). O filtro Gaussiano cumpre este objetivo ao eliminar altas frequências presentes no sinal com a propriedade de preservar a localização de pontos importantes da mudança da curvatura (TOMASI, 2007).

---

```
function [ dy ] = calcDerivative(sigma, y, x)
% calcDerivative Calcula a derivada de uma funcao y(x) discreta

n = size(x, 2); % qtt de pontos
```

```

st = 1.0 / (x(1, n) - x(1, 1)); % step para x_i
X = (-floor(n/2):(n - floor(n/2) - 1)) * st; % shift do domínio
GAU = exp(-(X.^2)/(sigma^2))/2; % filtro gaussiano
GAU = fftshift(GAU);

% fft e aplicacao do filtro gaussiano
y = fft(fftshift(y));
A = (GAU .* y);

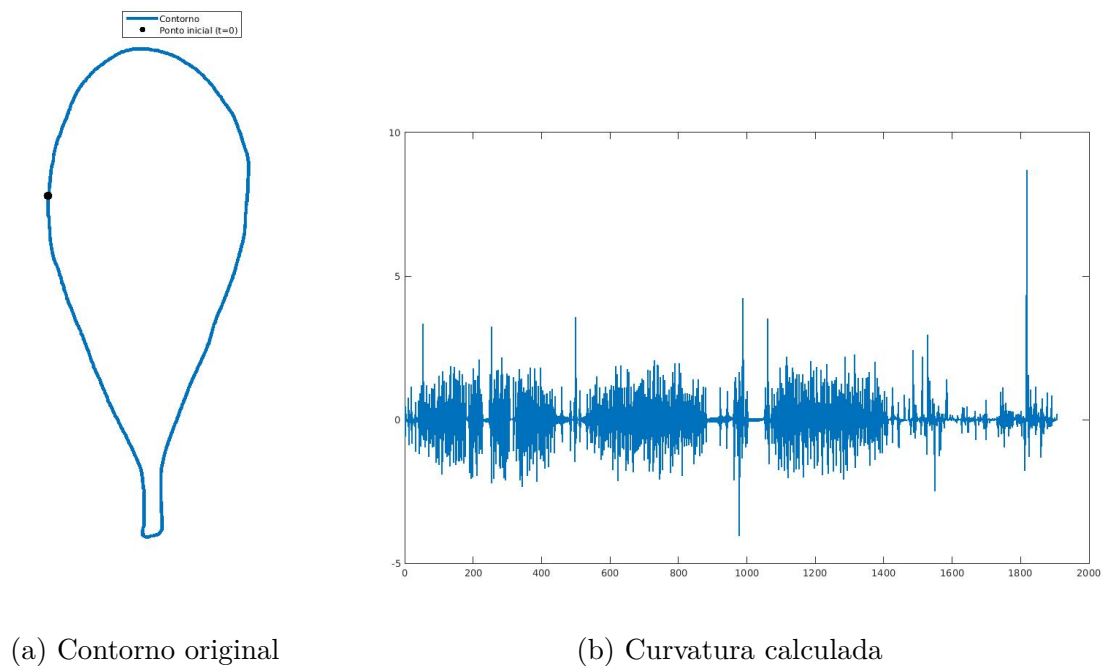
% funcao derivada
DER = 2*pi*1i*X;
DER = fftshift(DER);
dy = ifftshift(real(ifft(A .* DER)));
end

```

---

Um exemplo do cálculo da curvatura pode ser visto na figura 3. Nela, foi utilizado o contorno da imagem ilustrada na figura 2. Para facilitar a visualização do gráfico, o ponto destacado em preto é o ponto inicial da curvatura, e os demais pontos seguem sentido horário.

Figura 3 – Curvatura calculada para o contorno.



Fonte: Elaborada pelo autor.



## 2.3 Reconstrução de curvas

Nesta seção serão abordadas as atividades realizadas sobre a reconstrução de curvas a partir de poucos pontos e de informações de conectividade. Para isso, foram utilizadas coordenadas diferenciais, como descrito em Sorkine (2006).

Seja  $\mathcal{M} = (V, E, F)$  uma malha triangular caracterizada, respectivamente, por seus conjuntos de vértices, arestas e faces, e cada vértice  $\mathbf{v}_i \in V$  possui uma coordenada cartesiana dada por  $\mathbf{v}_i = (x_i, y_i, z_i)$ . As coordenadas diferenciais (ou  $\delta$ -coordenadas) são definidas como:

$$\delta_i = (\delta_i^{(x)}, \delta_i^{(y)}, \delta_i^{(z)}) = \mathbf{v}_i - \frac{1}{d_i} \sum_{j \in N(i)} \mathbf{v}_j, \quad (2.2)$$

em que  $N(i) = \{j \mid (i, j) \in E\}$  (ou seja, o conjunto dos vértices adjacentes a  $i$ ) e  $d_i = |N(i)|$  é o número dos vizinhos imediatos de  $i$  (grau de  $i$ ).

Caso a malha  $\mathcal{M}$  seja uma aproximação de superfície linear por partes, a coordenada diferencial  $\delta_i$  pode ser vista como a discretização da seguinte integral:

$$\frac{1}{|\gamma|} \int_{\mathbf{v} \in \gamma} (\mathbf{v}_i - \mathbf{v}) d\ell(\mathbf{v}) \quad (2.3)$$

onde  $\gamma$  é uma superfície simples e fechada em torno de  $\mathbf{v}$  e  $|\gamma|$  é o comprimento de  $\gamma$ . Sabe-se, de geometria diferencial (CARMO, 2014), que

$$\lim_{|\gamma| \rightarrow 0} \frac{1}{|\gamma|} \int_{\mathbf{v} \in \gamma} (\mathbf{v}_i - \mathbf{v}) d\ell(\mathbf{v}) = -H(\mathbf{v}_i) \mathbf{n}_i \quad (2.4)$$

em que  $H(\mathbf{v}_i)$  é a curvatura média em  $\mathbf{v}_i$  e  $\mathbf{n}_i$  é o vetor normal à superfície. Assim, as  $\delta$ -coordenadas podem ser vistas como a discretização do operador contínuo de Laplace-Beltrami, e a direção do vetor de coordenadas diferenciais aproxima-se à direção normal e sua norma aproxima a quantidade proporcional à curvatura média local. De maneira resumida, isto significa que as  $\delta$ -coordenadas encapsulam a forma da superfície local.

A transformação das coordenadas cartesianas absolutas para as  $\delta$ -coordenadas também pode ser calculada em forma de matriz. Seja  $A$  a matriz de adjacências da malha e  $D$  matriz diagonal tal que  $D_{ii} = d_i$  (grau do vértice  $i$ ), a matriz transformação de coordenadas absolutas para as diferenciais é:

$$L = I - D^{-1}A. \quad (2.5)$$

Porém, é mais conveniente (e computacionalmente eficiente) utilizar a versão simétrica  $L_s$  da matriz  $L$ , definida por:

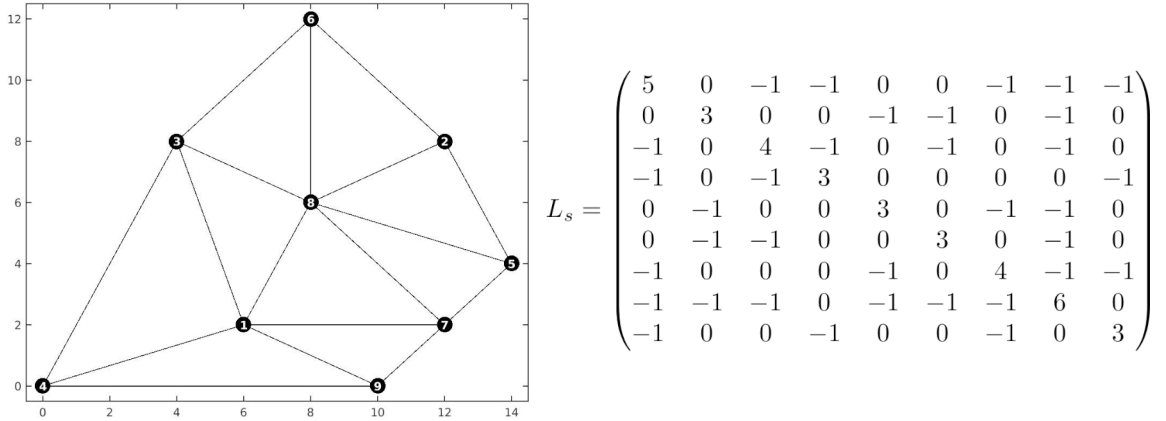
$$L_s = DL = D - A \quad (2.6)$$

em que cada célula pode ser calculada da seguinte forma:

$$(L_s)_{ij} = \begin{cases} d_i & i = j \\ -1 & (i, j) \in E \\ 0 & \text{caso contrário.} \end{cases} \quad (2.7)$$

Esta matriz  $L_s$  é denominada *Laplaciano topológico* da malha  $\mathcal{M}$ . Um exemplo dela pode ser visto na figura 4.

Figura 4 – Malha bidimensional e seu laplaciano topológico



Fonte: Elaborada pelo autor.

Porém, a conversão das  $\delta$ -coordenadas de volta para as cartesianas exige mais atenção. As coordenadas diferenciais são invariantes à translação, pois caso a malha  $\mathcal{M}$  seja transladada de acordo com um vetor  $u$  gerando novos vértices  $v'_i$ , tem-se:

$$\begin{aligned} L(\mathbf{v}'_i) &= \sum_{j \in N(i)} w_{ij}(\mathbf{v}'_i - \mathbf{v}'_j) \\ &= \sum_{j \in N(i)} w_{ij}((\mathbf{v}_i + \mathbf{u}) - (\mathbf{v}_j + \mathbf{u})) \\ &= \sum_{j \in N(i)} w_{ij}(\mathbf{v}_i - \mathbf{v}_j) = L(\mathbf{v}_i) \end{aligned}$$

Assim, as matrizes  $L$  e  $L_s$  são singulares, e  $\mathbf{x} = L_s^{-1}\delta^{(x)}$  é indefinido. Mais que isso, como cada componente da malha possui um grau de liberdade de translação, tem-se

que  $\text{rank}(L) = n - k$ , em que  $k$  é o número de componentes. Para que seja possível restaurar as coordenadas absolutas, é necessário que o sistema linear seja *full-rank* - e isto pode ser alcançado adicionando restrições para fixar a localização de alguns vértices, denominados pontos âncora. Suponha que sejam fixados a localização de  $m$  vértices com índices  $P = \{1, 2, \dots, m\}$ , em que são conhecidas as localizações  $c = \{\mathbf{v}_j \mid j \in P\}$ . Serão adicionadas restrições do tipo:

$$\mathbf{v}_j = \mathbf{c}_j, \forall j \in P$$

Com a denotação matricial, o novo sistema linear será:

$$\begin{pmatrix} L \\ \omega I_{m \times m} | 0 \end{pmatrix} \mathbf{x} = \begin{pmatrix} \delta^{(x)} \\ \omega c_{1:m}^{(x)} \end{pmatrix} \quad (2.8)$$

e o mesmo vale para os outros vetores de coordenadas. Com a utilização do vetor  $\omega > 0$ , cada restrição pode possuir um peso diferente, que pode ajustar a importância e relevância de cada vértice.

Graficamente, pode-se representar o sistema da seguinte maneira:

$$\begin{array}{c} L_s \\ I_c \end{array} \mathbf{x} = \begin{array}{c} \delta^{(x)} \\ c^{(x)} \end{array}$$

onde,  $I_c$  é a matriz identidade  $m \times m$ , com zeros à direita (e foi fixado  $\omega = 1$  por simplicidade).

A matriz dos coeficientes em 2.8 é denominada  $\tilde{L}$ . Por mais que o sistema tenha mais equações que incógnitas, ele é *full-rank* e possui uma única solução com a utilização do método dos mínimos quadrados:

$$\tilde{\mathbf{x}} = \underset{x}{\operatorname{argmin}} \left( \|L\mathbf{x} - \delta^{(x)}\|^2 + \sum_{j \in C} \omega^2 |x_j - c_j|^2 \right) \quad (2.9)$$

Um exemplo de fixação de pontos âncora e a respectiva matriz  $\tilde{L}$  é dado na figura 5.

Algumas aplicações interessantes são possíveis a partir do mapeamento entre coordenadas absolutas e diferenciais, como representação eficiente de formas, edição e interpolação de malhas. Neste projeto, com o objetivo de reconstrução das curvas, as coordenadas diferenciais serão utilizadas por representarem formas de maneira eficiente.



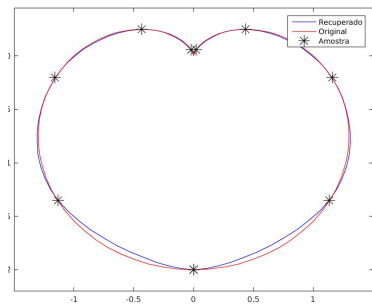
respectiva figura) é calculado pela soma dos erros de cada ponto, que por si são calculados como a distância Euclidiana entre os pontos recuperados e os pontos originais. Ou seja, para cada índice de vértice  $i$ , o erro dos pontos obtidos  $\mathbf{v}'$  em relação aos pontos originais  $\mathbf{v}$  é:

$$E_i(\mathbf{v}, \mathbf{v}') = \sqrt{(\mathbf{v}_i^{(x)} - \mathbf{v}'^{(x)})^2 + (\mathbf{v}_i^{(y)} - \mathbf{v}'^{(y)})^2}$$

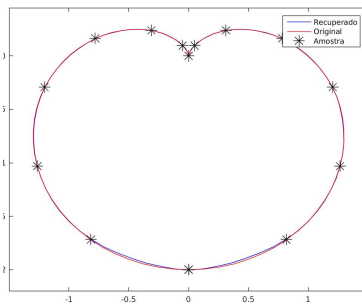
e o erro total da reconstrução é calculado como:

$$E(\mathbf{v}, \mathbf{v}') = \sum_{i=1}^{|V|} E_i(\mathbf{v}, \mathbf{v}')$$

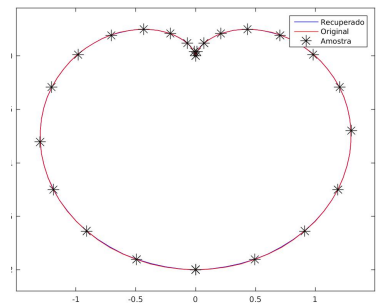
Figura 6 – Representação da função paramétrica  $(x(t), y(t)) = (\sin(t) + 0.5\sin(2t), -\cos(t) - 0.5 - 0.5\cos(2t))$  utilizando alguns pontos igualmente espaçados como amostra.



(a) 10 pontos. Erro = 3.7199



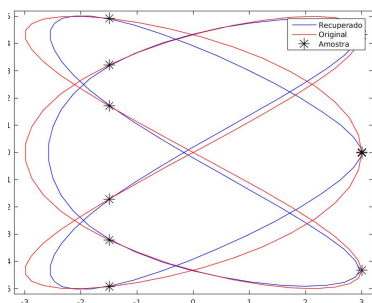
(b) 15 pontos. Erro = 0.4498



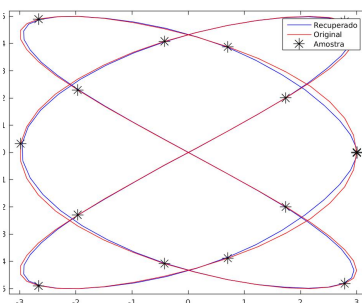
(c) 25 pontos. Erro = 0.2045

Fonte: Elaborada pelo autor.

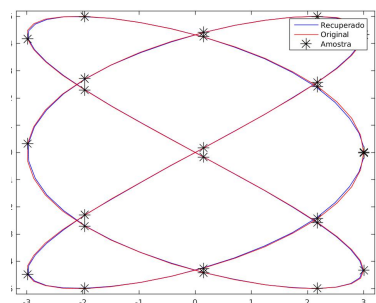
Figura 7 – Representação da função paramétrica  $(x(t), y(t)) = (3\cos(3t), 5\sin(2t))$  utilizando alguns pontos igualmente espaçados como amostra.



(a) 10 pontos. Erro = 75.2102



(b) 15 pontos. Erro = 4.8213



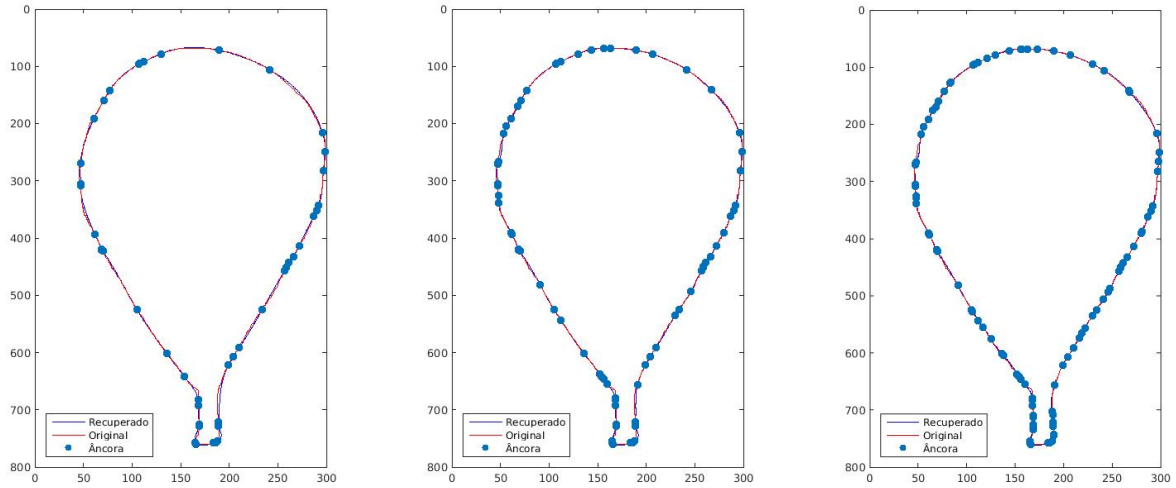
(c) 25 pontos. Erro = 2.0368

Fonte: Elaborada pelo autor.

Como dito anteriormente, neste projeto, os pontos âncora escolhidos serão as características robustas da curva. Embora isto ainda não tenha sido implementado, pode-se

ver na figura 8 a reconstrução da curva a partir da escolha de pontos de maior curvatura, calculada como mostrada na seção 2.2. Os pontos escolhidos estão destacados em azul, a curva original está representada em vermelho e a curva reconstruída, em azul.

Figura 8 – Representação da curva do contorno da folha, utilizando pontos de maior curvatura como amostra.



(a) 45 pontos. Erro 11971.34

(b) 70 pontos. Erro 762.87

(c) 100 pontos. Erro 712.67

Fonte: Elaborada pelo autor.

O contorno original da folha possui 1472 pontos, no intervalo  $x \in [46, 299]$ ,  $y \in [68, 761]$ . Por mais que a escolha dos pontos âncora não tenha sido feita de forma adequada (e será melhorada na próxima etapa do projeto), os erros apresentados foram baixos, considerando a curva em análise.

A escolha dos  $m$  pontos âncora foi realizada da seguinte forma: todos os pontos da curva foram ordenados de forma decrescente com relação ao valor da curvatura. Após isso, todos os  $n$  pontos foram divididos em  $K$  grupos linearmente espaçados (cada um com  $\lfloor n/k \rfloor$  pontos, e os  $n \bmod k$  primeiros grupos com um ponto a mais), e foram escolhidos os primeiros  $\lfloor m/k \rfloor$  pontos de cada grupo (e um ponto a mais dos primeiros  $m \bmod k$  grupos). Isto foi feito para aumentar a representatividade de pontos âncoras ao longo da curva, e não escolher apenas aqueles concentrados em região de maior curvatura que, no caso da folha, estão concentrados na porção inferior do contorno.

Esta escolha foi feita para teste, e será melhorada para a próxima etapa, com a escolha dos pontos importantes como pontos âncora.

### 3 Próximas atividades

De acordo com a organização inicial e levando em consideração as alterações discutidas na seção 1.2, as próximas atividades a serem realizadas serão as seguintes:

1. **Extração dos pontos importantes:** estudar e implementar técnicas de extração das características robustas, a partir da análise de curvatura.
2. **Combinação dos algoritmos:** juntar os códigos implementados de pré-processamento de imagens, extração dos pontos importantes e reconstrução de curvas em um único fluxo de desenvolvimento, para que eles se comuniquem e se facilitem os testes.
3. **Avaliação e testes:** executar o programa implementado em instâncias do banco de imagens ImageCLEF (2011), e inferir a qualidade das implementações feitas.
4. **Desenvolvimento do relatório final:** escrita do relatório final.

e seguirão o seguinte cronograma, visto na tabela 2:

Atividades	Meses						
	Março	Abril	Maió	Junho	Julho	Agosto	Setembro
Extração dos pontos importantes	•	•	•				
Combinação dos algoritmos			•	•			
Avaliação e testes				•	•	•	
Desenvolvimento do relatório final						•	•

Tabela 2 – Cronograma de atividades para os próximos meses de trabalho.

# Referências

- BORROTO-FERNÁNDEZ, M.; MECIAS, A. L.; HIDALGO, M. G. Image smoothing using the perona-malik equation with a new estimation method of the contrast parameter. 10 2013.
- BRETHERTON, C. *The Fourier spectral method*. 2019. Disponível em: <[https://atmos.washington.edu/~breth/classes/AM585/lect/DFT\\_FS\\_585\\_notes.pdf](https://atmos.washington.edu/~breth/classes/AM585/lect/DFT_FS_585_notes.pdf)>. Acesso em: 1 mar 2021.
- BURGER, W.; BURGE, M. J. *Digital Image Processing: An algorithmic introduction using java*. 2. ed. Londres: Springer, 2016. 253-263 p.
- CARMO, M. P. do. *Geometria Diferencial de Curvas e Superfícies*. São Paulo: SBM, 2014.
- COSTA, L. da F.; JR, R. M. C. *Shape Classification and Analysis*. 2. ed. Boca Raton: CRC Press, 2009. 4, 323-335 p.
- IMAGECLEF. 2011. Disponível em: <<https://www.imageclef.org/2011/Plants>>. Acesso em: 01 may 2020.
- LI, J. *Shape Recognition by Curvature Changing Points and Fourier Transform*. Dissertação (Mestrado) — Western Michigan University, 1987.
- OLIVEIRA, A.; MARROQUIM, R. *Processamento de Imagens*: aula 09 - curvatura. 2016. Disponível em: <<https://www.lcg.ufrj.br/marroquim/courses/cos756/lessons/09-curvature.pdf>>. Acesso em: 7 mar 2020.
- OPENCV. 2000. Disponível em: <<https://opencv.org>>. Acesso em: 20 feb 2021.
- SERRA, J. *Image Analysis and Mathematical Morphology*. Orlando: Academic Press, 1984. ISBN 9780126372427.
- SORKINE, O. Differential representations for mesh processing. *Computer Graphics Forum*, European Association for Computer Graphics, v. 25, n. 4, p. 789–807, 2006.
- SORKINE, O.; COHEN-OR, D. Least-squares meshes. *Proceedings of Shape Modeling International*, p. 191–199, 2004.
- SZELISKI, R. *Computer Vision: Algorithms and applications*. London: Springer, 2011.
- TOMASI, C. *Convolution, Smoothing, and Image Derivatives*. 2007. Disponível em: <<https://www2.cs.duke.edu/courses/fall07/cps296.1/convolution.pdf>>. Acesso em: 1 mar 2021.