

Departamento de Sistemas e Computação – FURB
Curso de Ciência da Computação
Trabalho de Conclusão de Curso – 2016/1

Tortuga: um protótipo para identificação de cágados da espécie *Phrynops williamsi*

Acadêmico: Guilherme Oecksler Bertoldi
guilhermebertoldi10@hotmail.com

Orientador: Prof. Aurélio Faustino Hoppe
aurelio.hoppe@gmail.com

Grupo de Pesquisa em Computação
Gráfica, Processamento de Imagens e
Entretenimento Digital
<http://www.inf.furb.br/gcg>



Roteiro

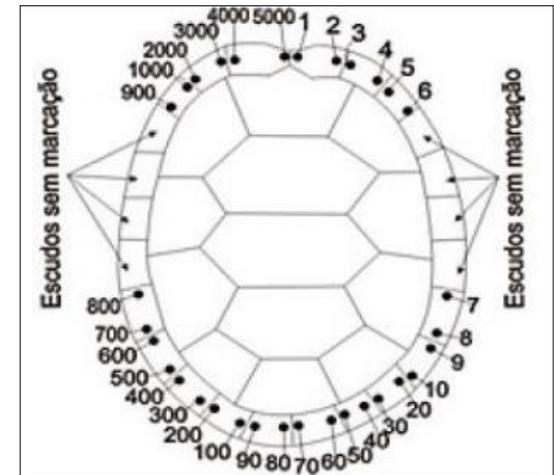
- Motivação
- Trabalhos correlatos
- Trabalho proposto
- Requisitos
- Desenvolvimento
- Testes
- Operacionalidade
- Conclusões
- Limitações
- Extensões
- Demonstração

Motivação

- A monitoração permite avaliar o impacto de doenças, caça e conservação de hábitat de uma espécie

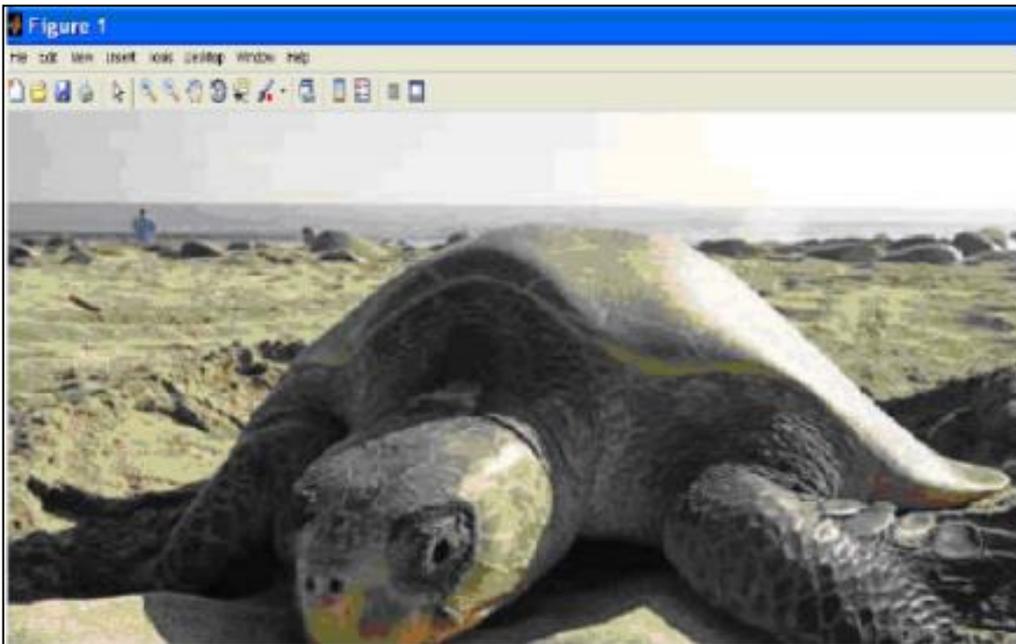


- A técnica de marcação mais comum de quelônios é a realização de furos ou cortes nos escudos marginais da carapaça
- Criar identificador único



Trabalhos correlatos

Título: Identification of olive ridley turtle using feature extraction
(BABOO e VIGNESWARI, 2014)



Objetivo:

Identificação de tartarugas da espécie *Lepidochelys olivácea* através das suas características a partir de uma imagem de entrada

Trabalhos correlatos

Título: Computer vision-based identification of individual turtles using characteristic patterns of their plastrons

(BEUGELING e BRANZAN-ALBU, 2014)



Objetivo:

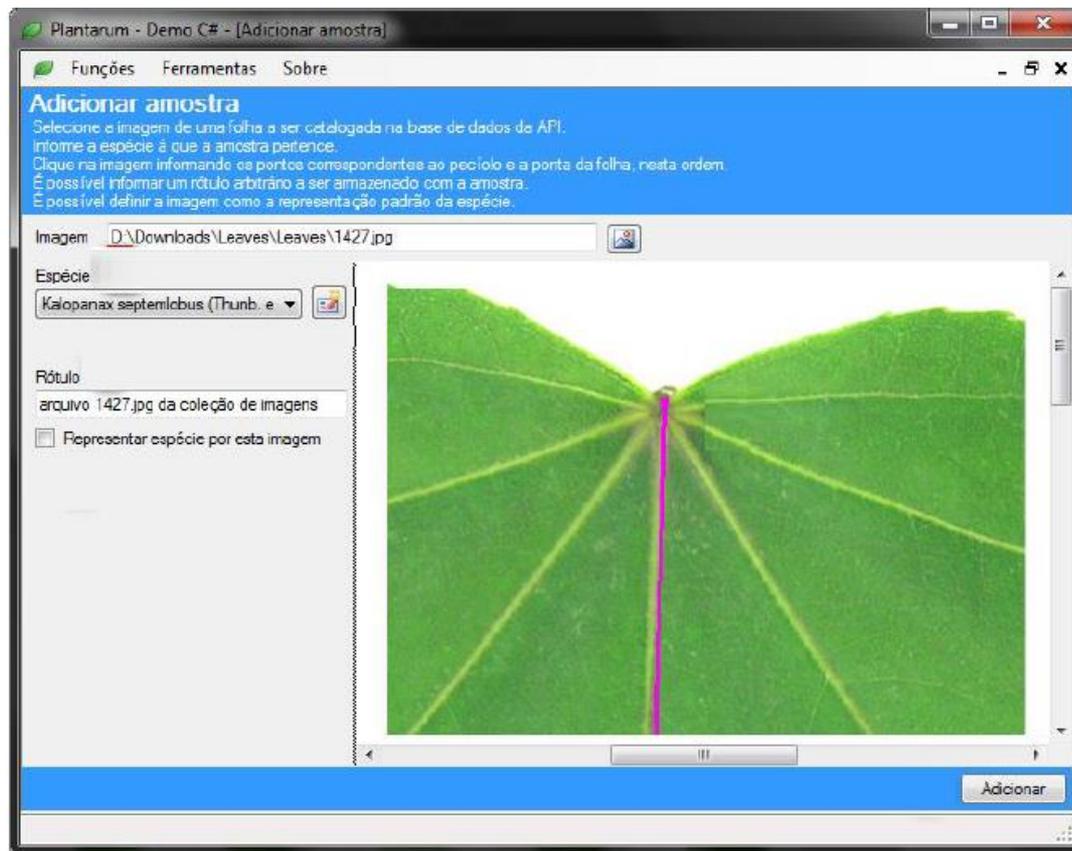
Identificação automática e individual de tartarugas baseado nas imagens dos seus plastrões



Trabalhos correlatos

Título: Planatarum: API para reconhecimento de plantas

(CASSANIGA, 2012)



Objetivo:

Desenvolver uma API para classificação de plantas a partir de imagens digitais das folhas

Trabalho proposto

Desenvolver um protótipo para identificação de cágados *Phrynops williamsi* por meio de um método não invasivo

Objetivos específicos:

- efetuar a segmentação da listra em formato de ferradura localizada na parte inferior da cabeça do animal
- estabelecer um identificador único a partir das características extraídas da ferradura
- disponibilizar uma interface gráfica que permita ao usuário realizar o cadastro e consulta dos cágados



Especificação

Requisitos funcionais:

- Identificar a ferradura localizada na parte inferior da cabeça do cágado
- Calcular o *bounding box* da ferradura, a centroide, dispersão e a circularidade das componentes da imagem
- Identificar a listra circular localizada próxima a ferradura, se existir
- Identificar a listra da ferradura e da listra circular através dos descritores de Fourier
- Possuir uma interface onde o usuário possa cadastrar, gerar um identificador único para os cágados e demonstrar as etapas do Processamento da Imagem
- Reconhecer os cágados de acordo com o seus identificadores únicos
- Armazenar em uma base de dados o identificador único para fins de reconhecimento futuros

Ferramentas utilizadas

- Linguagem C# (C-Sharp) na IDE Microsoft Visual Studio 2013
- EmguCV: *Wrapper* da biblioteca OpenCV para a linguagem C#.
- Windows Forms
- SQLite

Pré-processamento



Imagem original

```
CvInvoke.Imread(this.filePath,  
LoadImageType.AnyColor);
```

Imagem em escalas de cinza

```
CvInvoke.CvtColor(imgIn.ToMat(), imgDest,  
ColorConversion.Bgr2Gray);
```



Pré-processamento

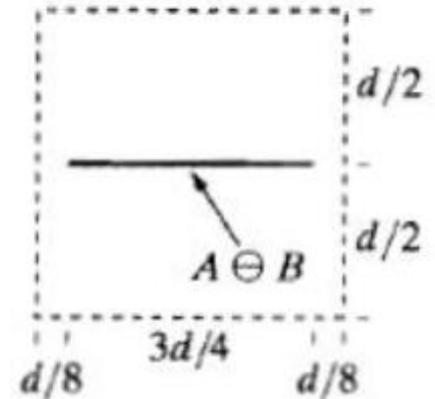
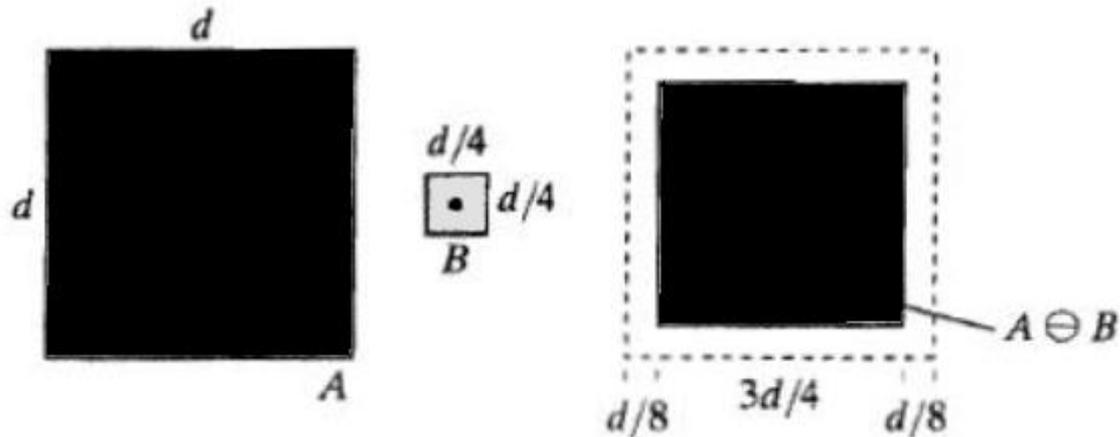
Imagem binarizada

```
CvInvoke.Threshold(imgIn, imgDest,  
threshold, 255,  
ThresholdType.BinaryInv);
```



Operadores morfológicos

Erosão:

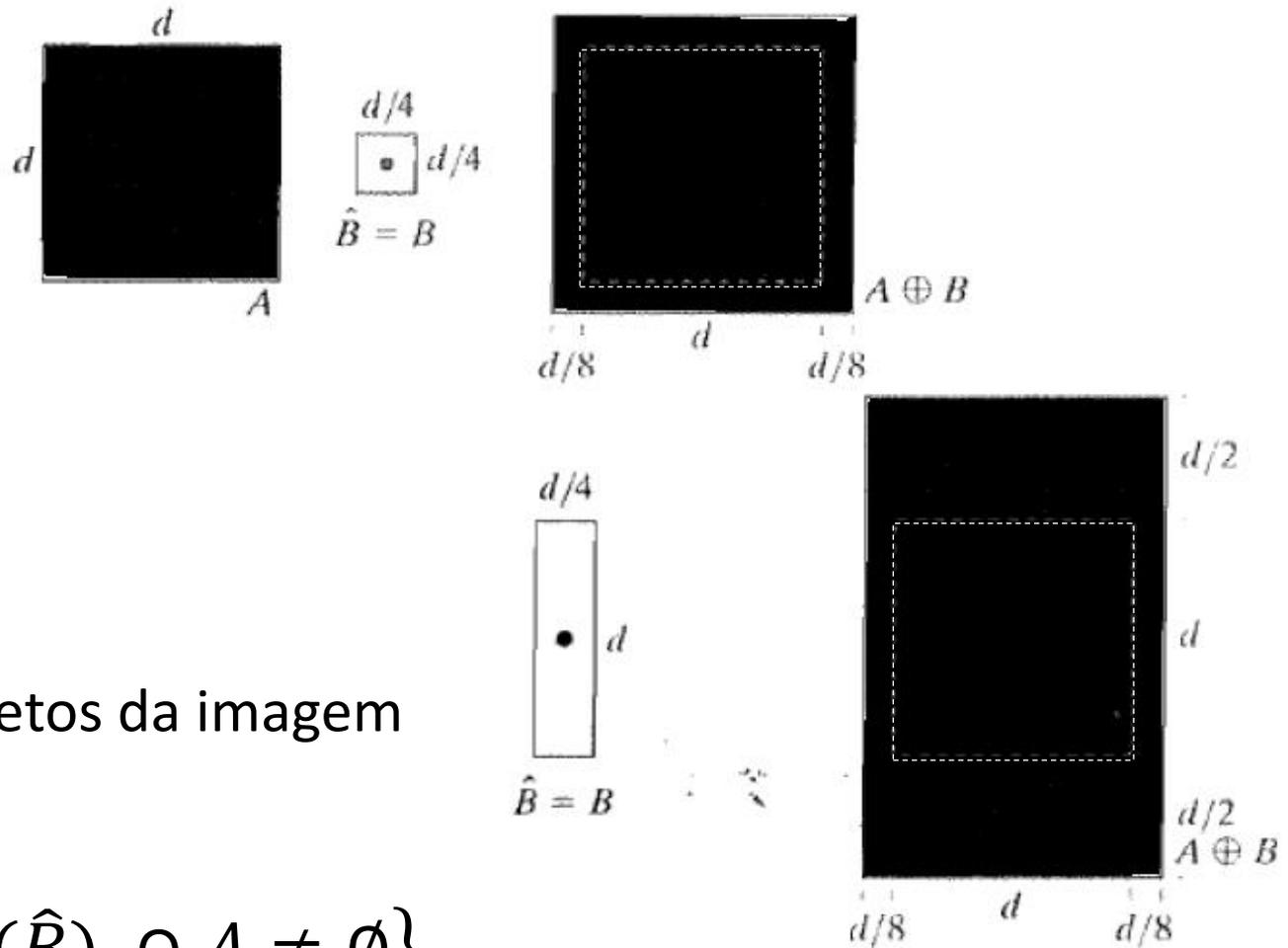


- Afina os objetos da imagem

$$A \ominus B = \{z | (B)_z \subseteq A\}$$

Operadores morfológicos

Dilatação:



- Engrossa os objetos da imagem

$$A \oplus B = \{z | (\hat{B})_z \cap A \neq \emptyset\}$$

Pré-processamento

Imagem erodida

```
CvInvoke.Erode(imgIn, imgDest, se, point, 1,  
BorderType.Default, new MCvScalar(1));
```



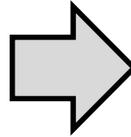
Imagem dilatada

```
CvInvoke.Dilate(imgIn, imgDest, se, point, 1,  
BorderType.Default, new MCvScalar(1));
```



Pré-processamento: Imagem filtrada

Imagem com menos pixels dispersos, mais homogênea e objetos com menos lacunas

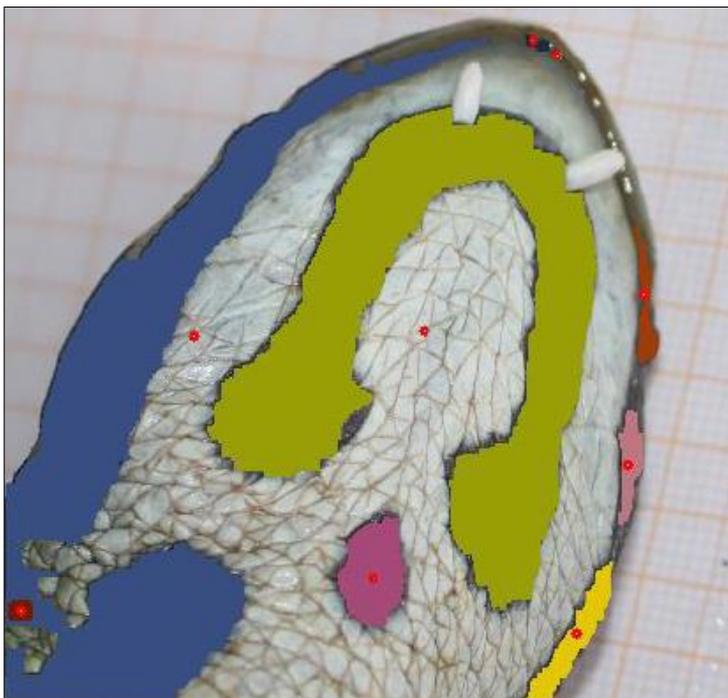


Melhor combinação:

erosão(3x3) - dilatação(12x12) - erosão(12x12) - dilatação(3x3)

Pré-processamento

Extração das componentes conexas através da extração do contorno



```
CvInvoke.FindContours(  
    imgIn,  
    contours,  
    hierachy,  
    RetrType.Ccomp,  
    ChainApproxMethod.ChainApproxSimple);
```

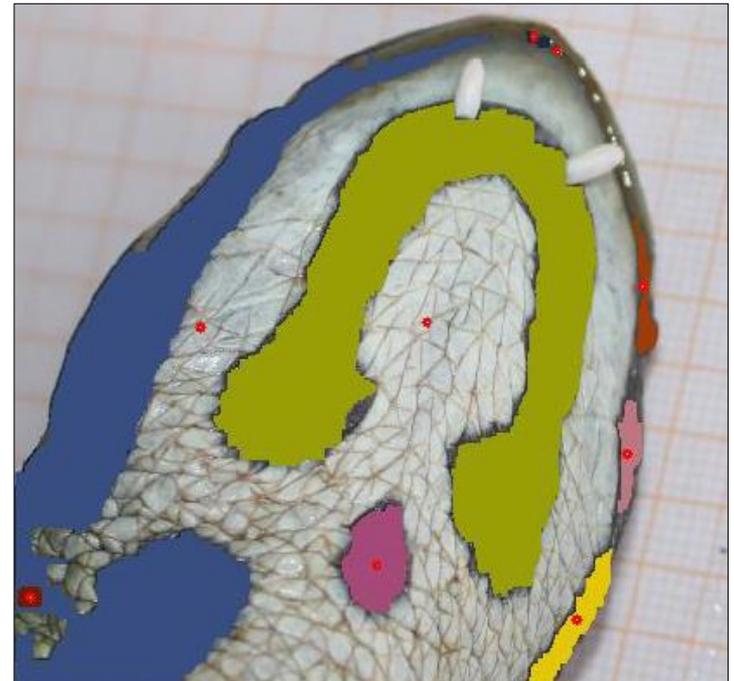
Seleção da ferradura

1ª etapa

- Tende a ser a segunda maior componente em relação à área
- Distância do centroide da ferradura até o ponto do contorno mais distante do tende a ser menor que a mesma distância da maior componente

2ª etapa

- Duas maiores componentes devem possuir uma área muito semelhante
- Dispersão da ferradura tende a ser menor que a dispersão da outra

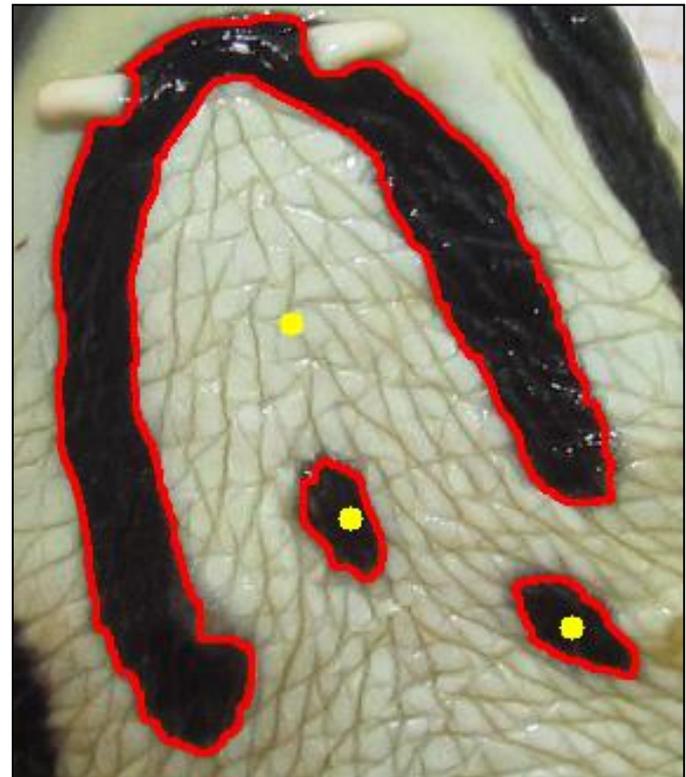


Centroide

- Centro da massa do objeto
- Média do somatório das coordenadas do contorno do objeto

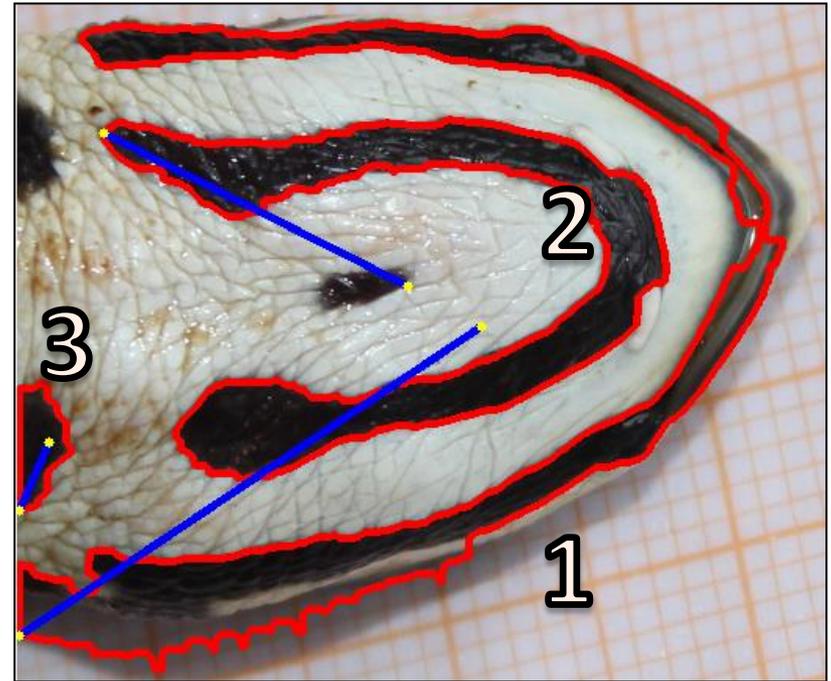
```
x = imgContour.Sum(p => p.X) / N;  
y = imgContour.Sum(p => p.Y) / N;
```

$$M = \frac{\sum_{n=0}^{N-1} u(n)}{N}$$



Seleção da ferradura (1ª etapa)

1. Criar lista com a área de todas as componentes
2. Ordenar decrescentemente e manter os 3 primeiros valores da lista
3. Criar lista com o quociente da área pela distância do centroide até o ponto mais distante do contorno
4. Primeiro item é considerado ferradura se a diferença do valor do primeiro item pelo valor do segundo item da lista é superior à 50% do valor do segundo item

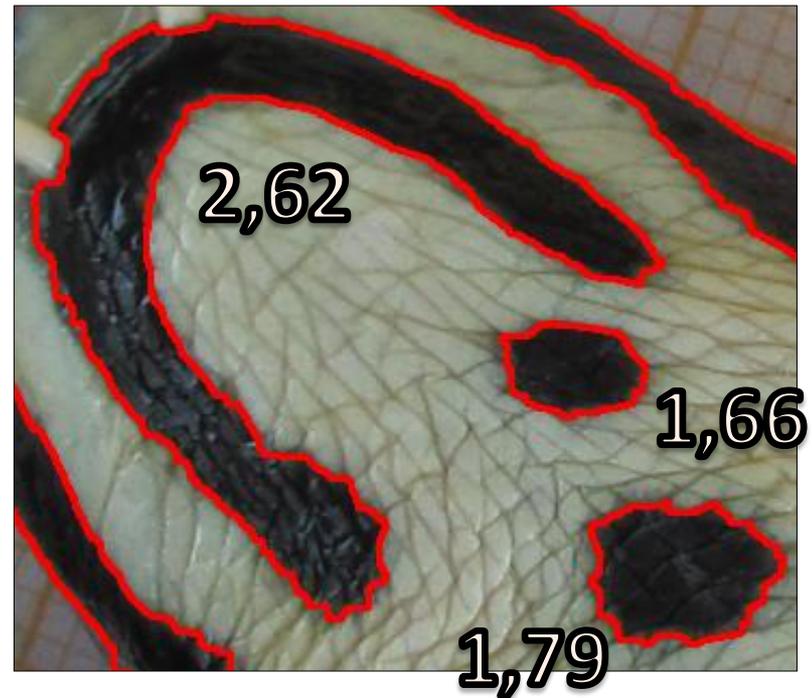


Dispersão

- Indica o nível de irregularidade do objeto
- Diferença entre a maior e a menor distância do centroide até o contorno do objeto

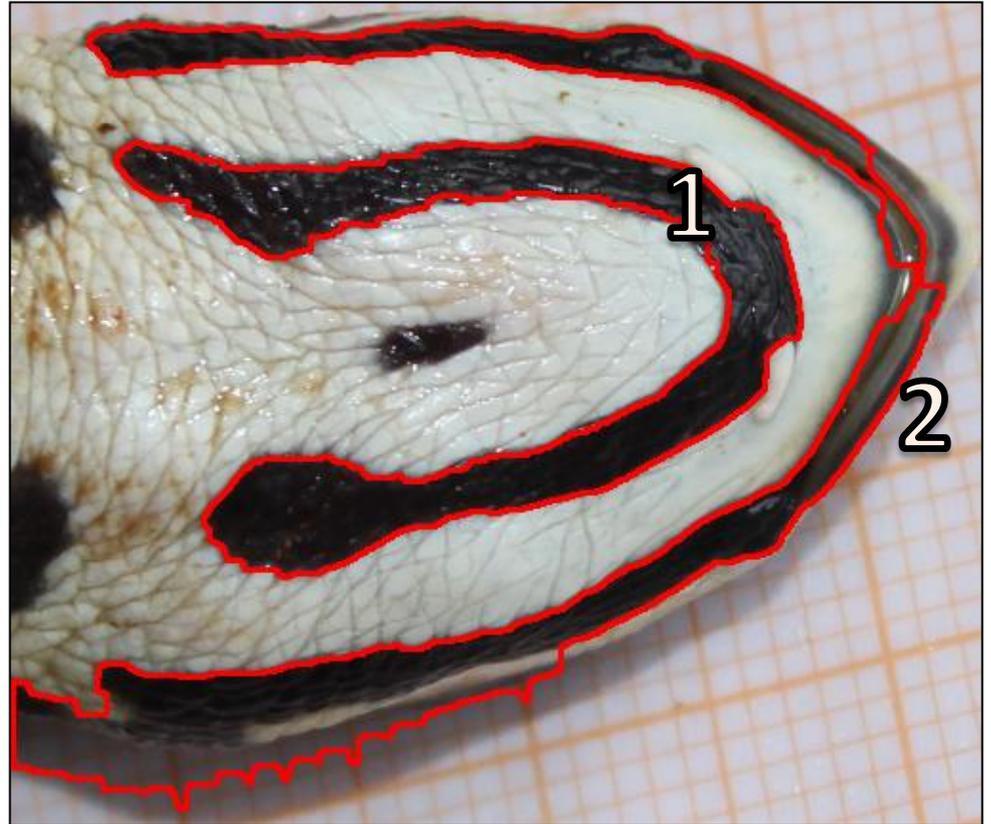
```
dist = imgContour.Select(x =>  
                        ED(x, cent));  
disp = dist.Max()/dist.Min();
```

$$disp = \frac{\max\left(\sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2}\right)}{\min\left(\sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2}\right)}$$



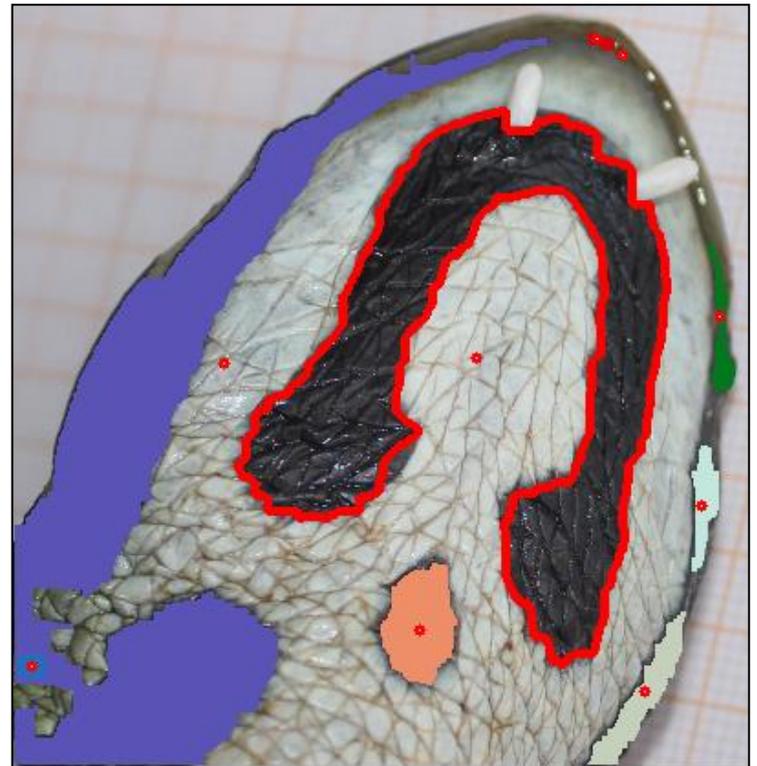
Seleção da ferradura (2ª etapa)

1. Remover terceira componente da lista
2. Criar lista com a soma da dispersão da componente com a sua área e ordenar ascendentemente
3. O primeiro elemento é considerado a ferradura, pois possui o menor valor



Seleção da listra circular

- Tende a ser de formato circular, com grau de circularidade maior que 0,45
- Deve possuir metade do perímetro da ferradura
- Ao menos um dos pontos da mancha circular tende a estar contido dentro do *bounding box* ferradura

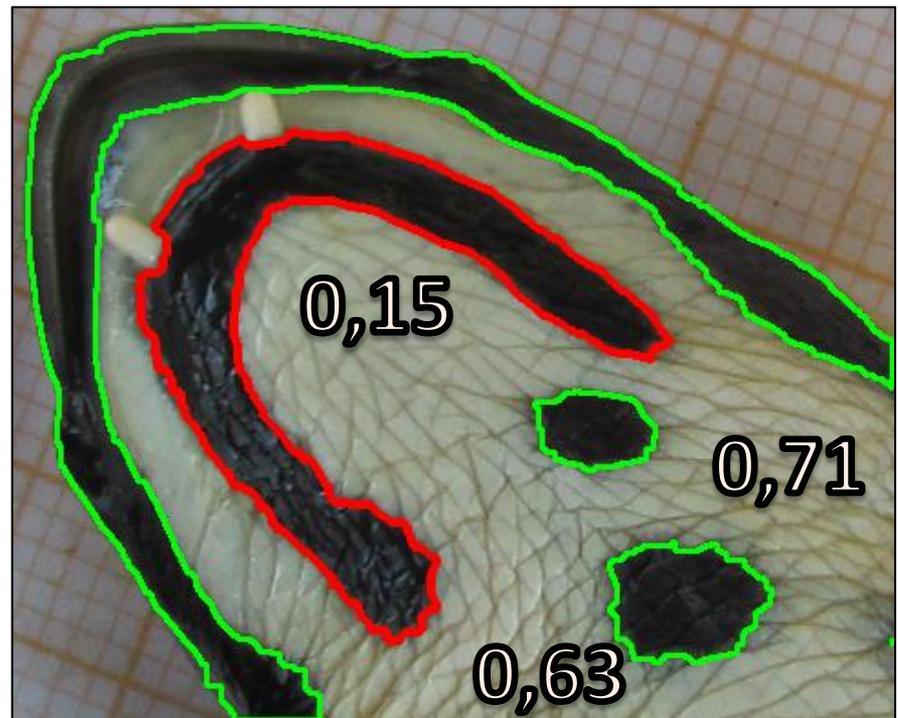


Circularidade

- Indica o quão circular é um objeto
- Aumenta conforme o formato do mesmo se assemelha a um círculo

```
roundness =  
(4 * Math.PI * area) /  
Math.Pow(perimeter, 2));
```

$$circ = \frac{4\pi A}{p^2}$$



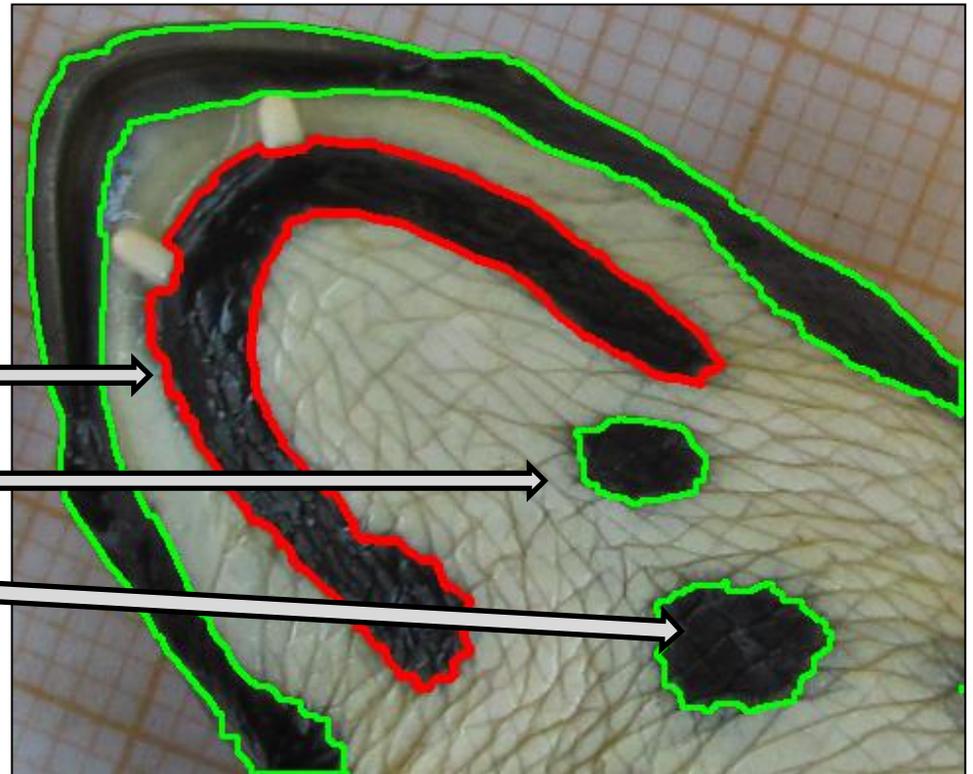
Seleção da listra circular (1ª etapa)

1. Verificar se perímetro é menor que 50% do perímetro da ferradura
2. Verificar se área é maior que 2% da área da ferradura
3. Verificar se grau de circularidade é maior que 0,45

P: 330,41 A: 24484

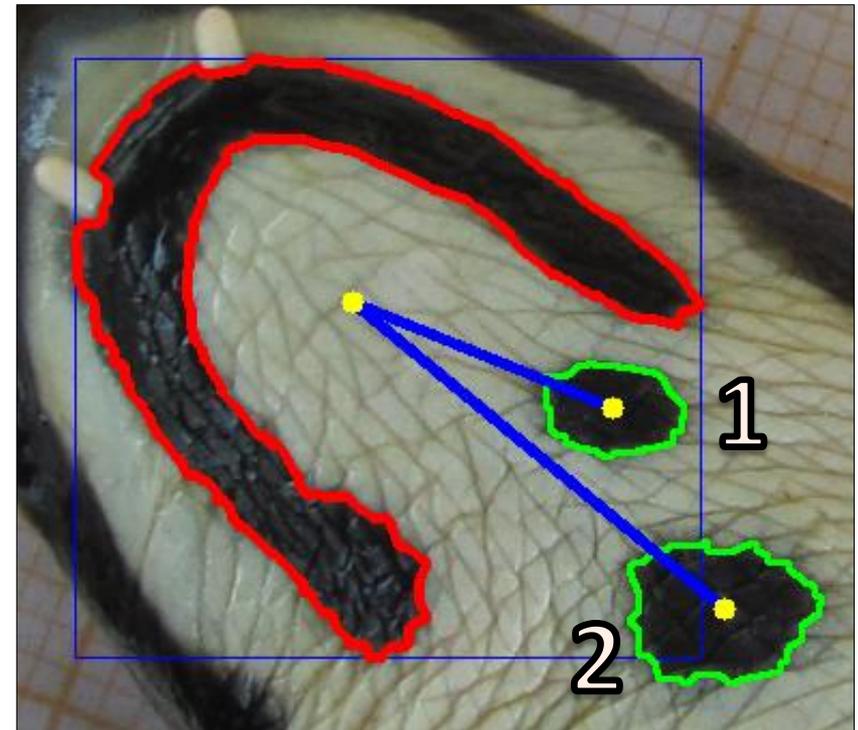
P: 196,57 A: 2170 C: 0,71

P: 298,51 A: 4443 C: 0,63



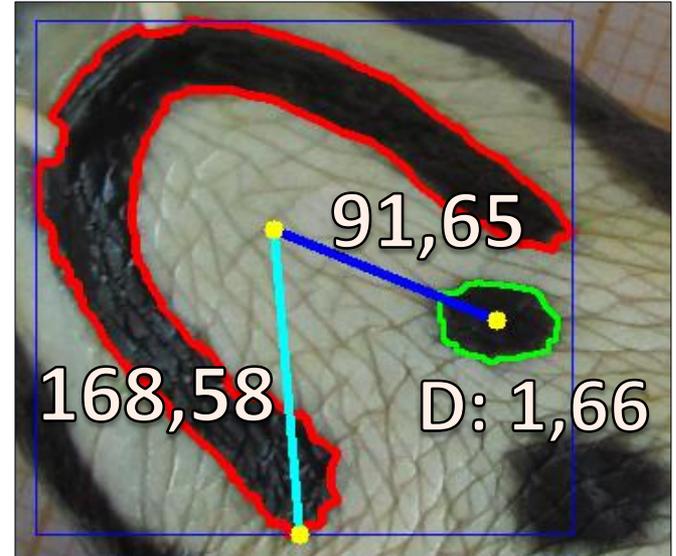
Seleção da listra circular (2ª etapa)

1. Calcular centroide das componentes
2. Verificar existência de pelo menos um ponto dentro da BB da ferradura
3. Calcular distância entre as centroides
4. Caso verificação 2 seja verdadeira, reduz essa distância em 30%
5. Ordena ascendentemente a lista de distâncias.



Seleção da listra circular (2ª etapa)

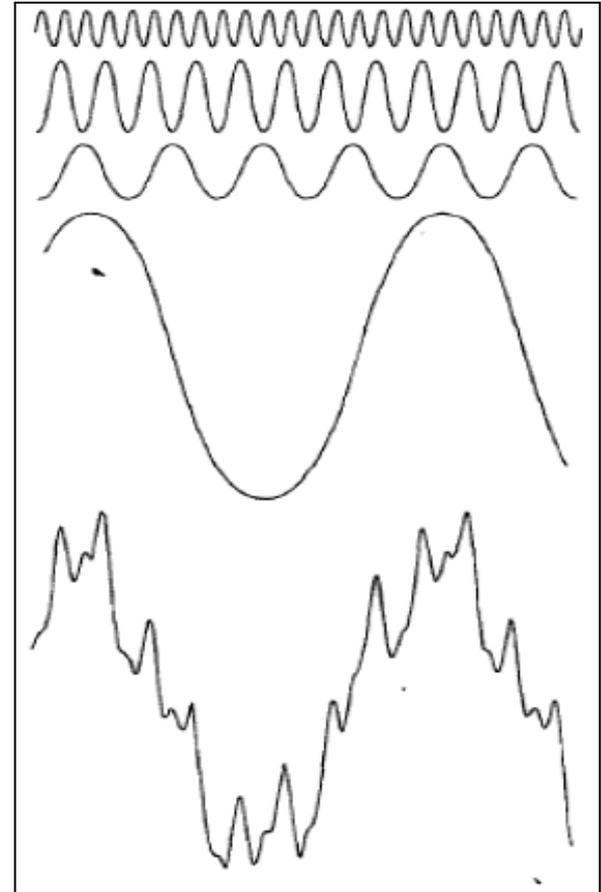
1. Verifica se primeiro item possui:
 - Dispersão menor que oito
 - Distância do centroide até o centroide da ferradura deve ser menor que 150% da distância entre do centroide da ferradura até o ponto do contorno mais distante da mesma
 - Distância entre o centroide e o ponto do contorno mais próximo maior que cinco ou, pelo menos um dos pontos esta contido dentro do BB da ferradura.



Descritores de Fourier

- Descritores de forma
- Soma de senos e cossenos em diferentes frequências
- Cada um deles multiplicado por um coeficiente distinto

```
theta = (2 * Math.PI * n * t) / T;  
...  
fR[n] += dist * Math.Cos(theta);  
fI[n] += dist * -Math.Sin(theta);  
...  
fR[n] = fR[n] / T;  
fI[n] = fI[n] / T;
```

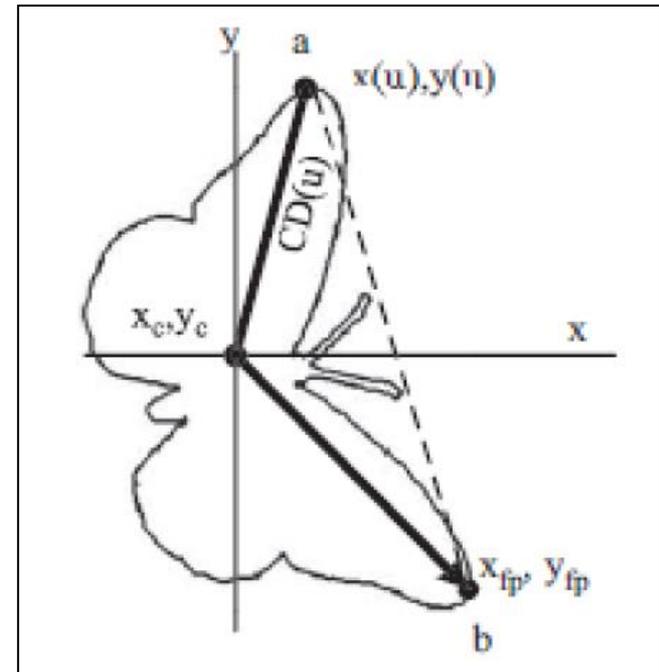


$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(u) e^{j2\pi ux/M} \quad x = 0, 1, 2, \dots, M - 1$$

Farthest Point Distance

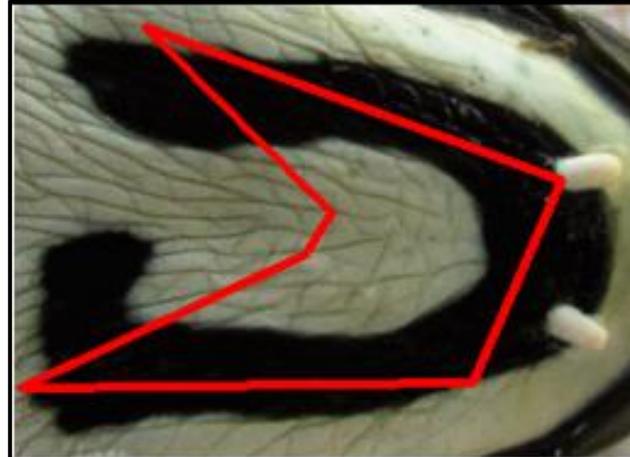
- Assinatura de forma utilizada no cálculo dos descritores de Fourier
- Distância do ponto analisado até o seu centroide, somada à distância deste centroide até o ponto mais distante do ponto analisado

```
for (int i = 0; i < contour.Size; i++)  
{ if((dist =  
    GetEuclDist(currentP, contour[i])) >  
    greaterDist)  
    greaterDist = dist;  
}
```

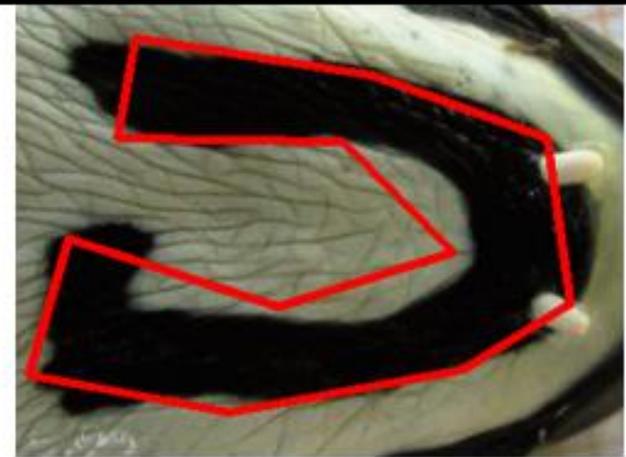


Descritores de Fourier

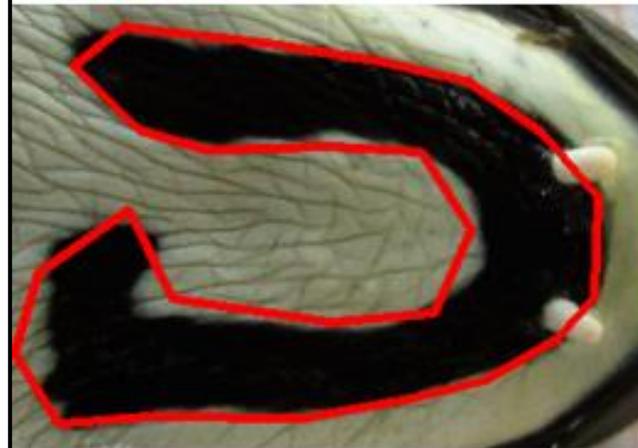
- Uma função expressa através de uma série/transformation de Fourier pode ser completamente reconstruída através do processo inverso.



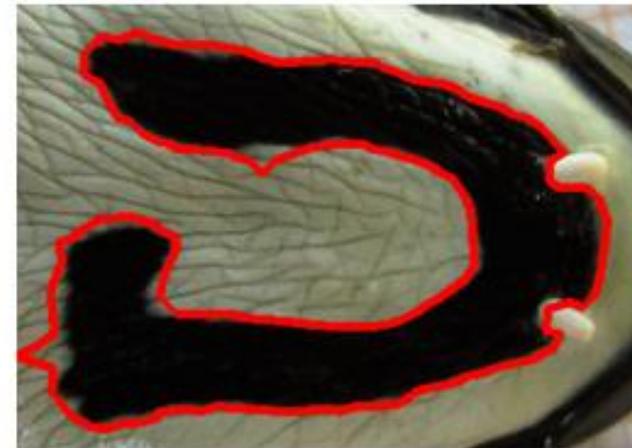
n=3 descritores



n=6 descritores



n=12 descritores



n=75 descritores

Identificador único

- É dado através da série de fourier calculada

11B = [14,99][1,27][0,18][5,88][2,24][1,00][0,61][0,94][0,96]
[23,10][0,67][1,22][0,80][0,44][0,42]

02A = [9,57][3,19][4,87][5,30][2,28][0,99][0,16][0,16][1,09]
[30,08][0,25][1,81][0,37][0,73][0,29]

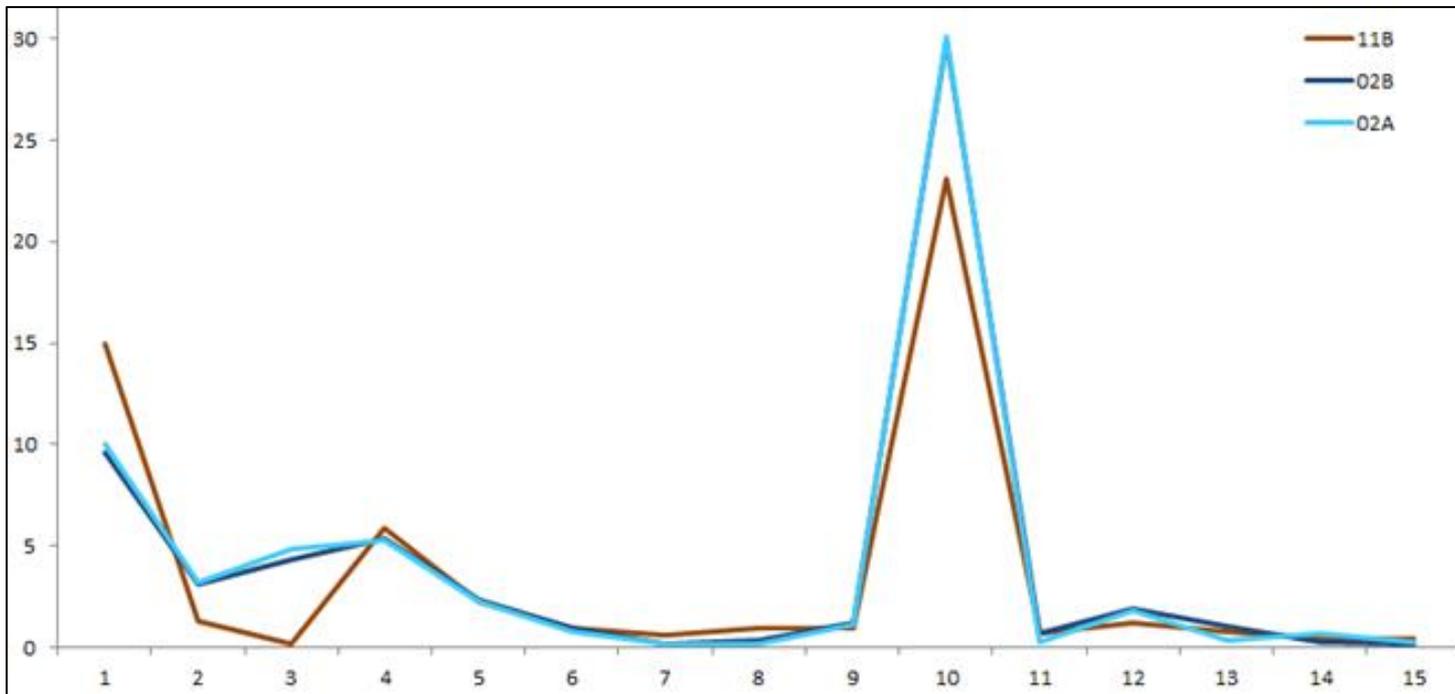
11B



02A



02B



Classificação

- Distância de Hamming.
- São consideradas apenas diferenças superiores a 10% do maior valor comparado.
- É utilizado um limiar de 12.

Exemplo:

11B = [14,99][1,27][0,18][5,88][2,24]

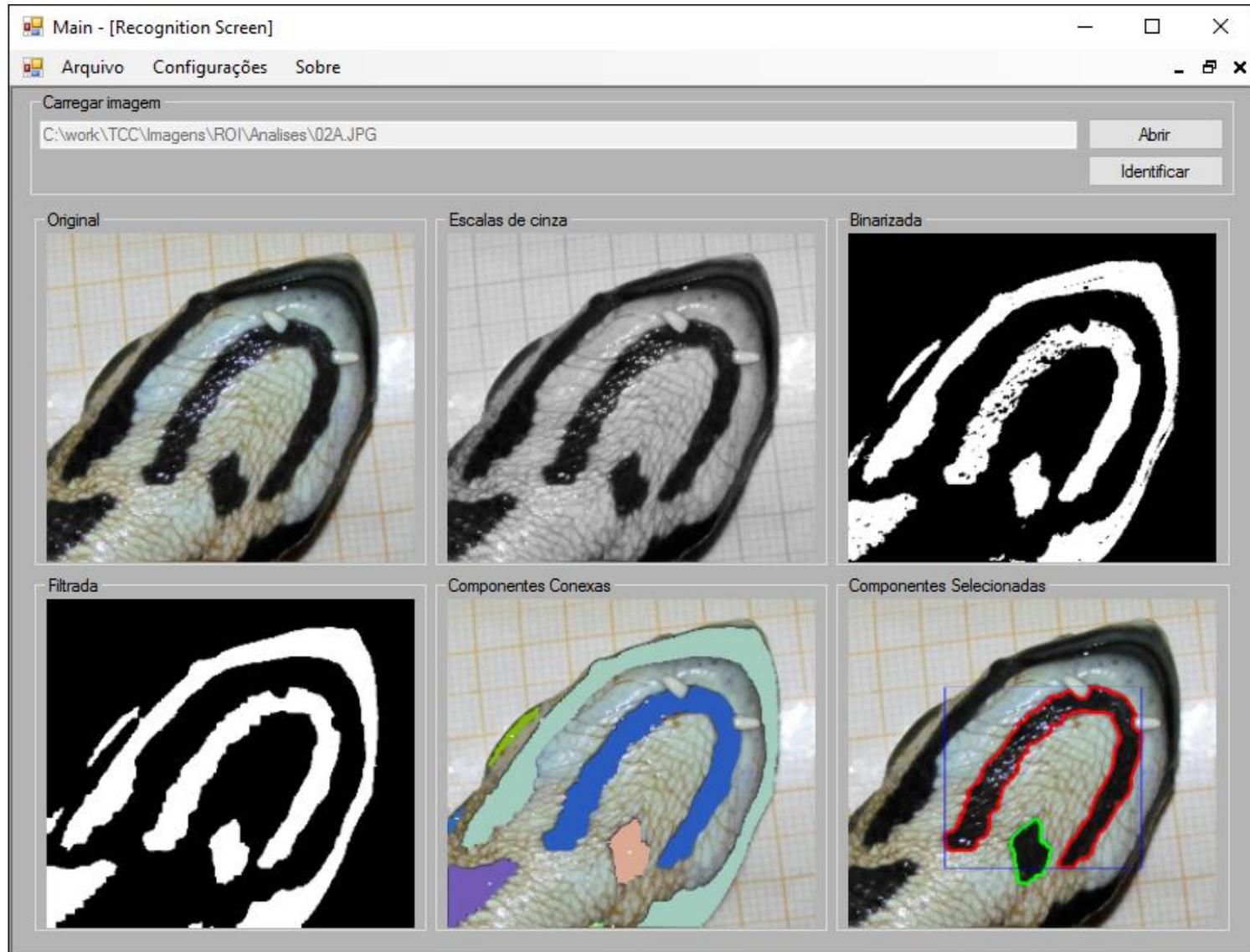
02B = [9,57][3,19][4,87][5,30][2,28]

Resultado = [5,42][1,92][4,69][0,58][~~0,04~~] = 12,61

< 12 = Mesmo código

>= 12 = Código distinto

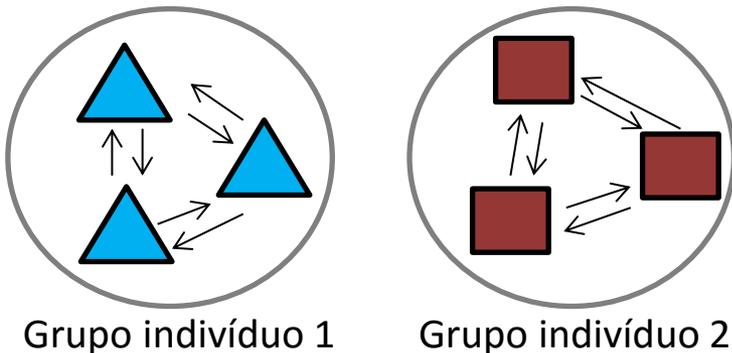
Operacionalidade do protótipo



Testes

- Testes efetuados utilizando uma base composta por 53 imagens de 26 cágados diferentes.
- Foram utilizados 23 descritores na ferradura e 6 na listra circular.
- Foram realizadas comparações dos tipos intra-classe e inter-classe.

Intra-classe



Inter-classe

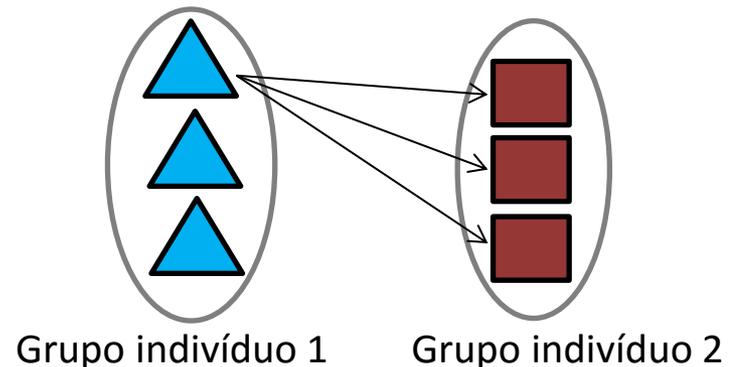


Tabela 1 - Resultado das comparações intra-classe e inter-classe

Individuos	Nº amostras	Nº classificações intra-classe	Nº classificações intra-classes corretas	Percentual de acerto intra-classe	Nº classificações inter-classes	Nº classificações inter-classes corretas	Percentual de acerto inter-classe
PW01	1	-	-	-	52	50	96,15%
PW02	3	6	6	100,00%	147	147	100,00%
PW03	1	-	-	-	48	48	100,00%
PW04	1	-	-	-	47	31	65,96%
PW05	1	-	-	-	46	27	58,70%
PW06	1	-	-	-	45	45	100,00%
PW07	1	-	-	-	44	28	63,64%
PW08	1	-	-	-	43	43	100,00%
PW09	1	-	-	-	42	30	71,43%
PW10	4	12	10	83,33%	152	146	96,05%
PW11	3	6	6	100,00%	105	104	99,05%
PW12	6	30	28	93,33%	174	109	62,64%
PW13	1	-	-	-	28	25	89,29%
PW14	2	2	2	100,00%	52	31	59,62%
PW15	1	-	-	-	25	16	64,00%
PW17	3	6	6	100,00%	66	65	98,48%
PW18	2	2	2	100,00%	40	32	80,00%
PW19	4	12	10	83,33%	64	55	85,94%
PW20	1	-	-	-	15	11	73,33%
PW21	3	6	6	100,00%	36	35	97,22%
PW22	2	2	2	100,00%	20	20	100,00%
PW23	2	2	0	0,00%	16	16	100,00%
PW24	1	-	-	-	7	7	100,00%
PW25	3	6	0	0,00%	12	8	66,67%
PW26	1	-	-	-	3	3	100,00%
PW28	3	6	6	100,00%	0	0	-
Total	53	98	84	85,71%	1329	1132	85,17%

Imagens com baixa taxa de acerto inter-classe

PW05 (58,70%)



PW14 (59,62%)



PW07 (63,64%)



PW15 (64,00%)



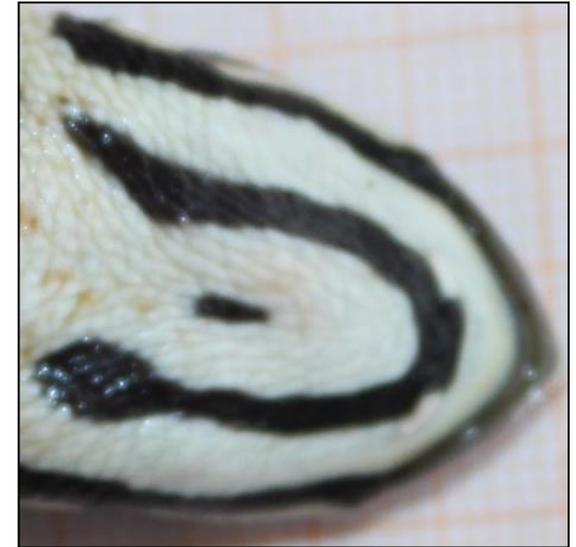
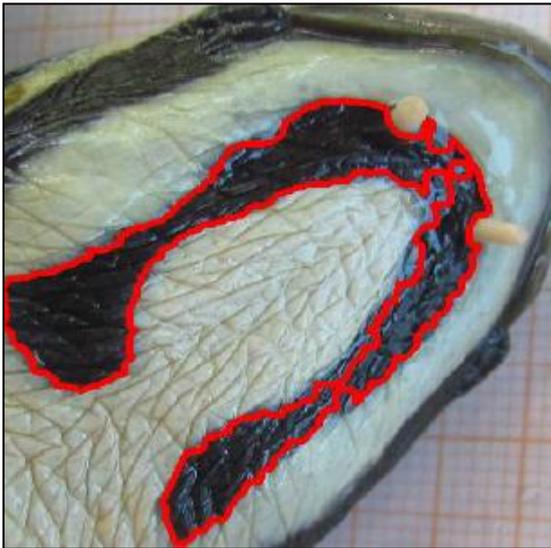
PW25 (66,67%)



Conclusões

Motivos que corroboraram com a queda das taxas de acerto:

- Presença de reflexo sobre a mancha
- Ângulo da imagem
- Falta de nitidez/foco da imagem

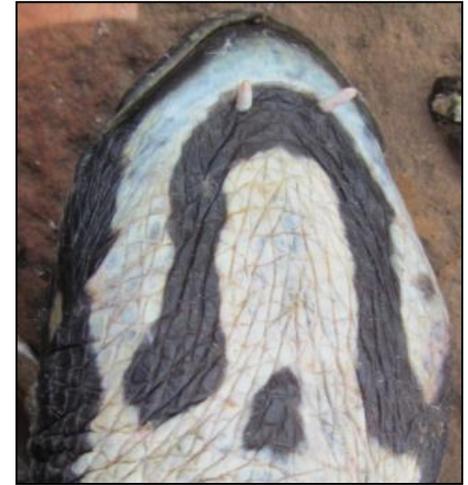


Conclusões

- O protótipo obteve taxas de acerto satisfatórias, 85,71% intra-classe e 85,17% inter-classe.
- Pode ser utilizado por pesquisadores e biólogos de campo que estudam essa espécie de cágado.
- Serve como base para outros trabalhos com foco na identificação de outras espécies de cágados e/ou tartarugas que utilizem a imagem das mesmas como entrada.

Limitações

- Não reconhece devidamente cágados que possuem a listra em formato de ferradura conexas com outras listras.
- Não realiza a extração da região de interesse (cabeça do cágado).
- Trabalha apenas com identificação de imagens de alto contraste das listras negras e da pele de cor clara.
- Funciona apenas na plataforma Windows.



Extensões

- Implementar a funcionalidade para seleção da região de interesse.
- Melhorar a técnica de seleção da componente, a fim de permitir o reconhecimento de cágados que possuam faixa em formato circular conexa à outras faixas.
- Realizar tratamento para remoção de reflexo das imagens.
- Permitir que o usuário possa cadastrar várias imagens para um mesmo cágado.

Extensões

- Exibir uma lista dos 10 cágados mais relevantes ordenadas de modo descendente pelo nível de similaridade, permitido a classificação por parte do usuário.
- Ajustar o protótipo para que seja multiplataforma.
- Transformar o protótipo em um App para dispositivos móveis.

Demonstração

Obrigado!