

# Projeto Final

## Representações B-spline a partir de imagens

Felipe Cadavez Oliveira 11208558

Giovanni Shibaki Camargo 11796444

João Victor Sene Araujo 11796382

Lucas Keiti Anbo Mihara 11796472

Prof. Dr. Antonio Castelo Filho

Bacharelado em Ciências de Computação  
Instituto De Ciências Matemáticas e de Computação  
Universidade de São Paulo

**GitHub do projeto:**

[https://github.com/giovanni-shibaki/SME0104\\_Bsplines](https://github.com/giovanni-shibaki/SME0104_Bsplines)

# Objetivos

Este projeto tem como objetivo, através de uma simples imagem, obter uma série de pontos e coordenadas que compõem os contornos dos objetos presentes nela de forma a encontrar a representação B-spline da curva fechada formada por esses pontos. Para isso, utilizaremos uma série de ferramentas e aplicações para a detecção de contornos a partir de uma imagem, além da edição dos pontos e coordenadas de forma a modificar a curva fechada de forma suavizada.

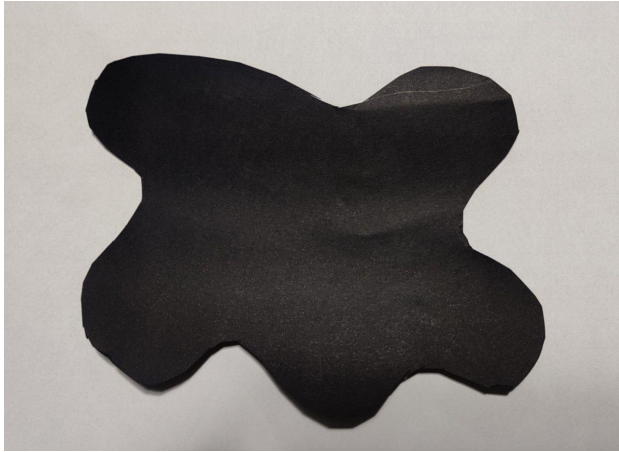
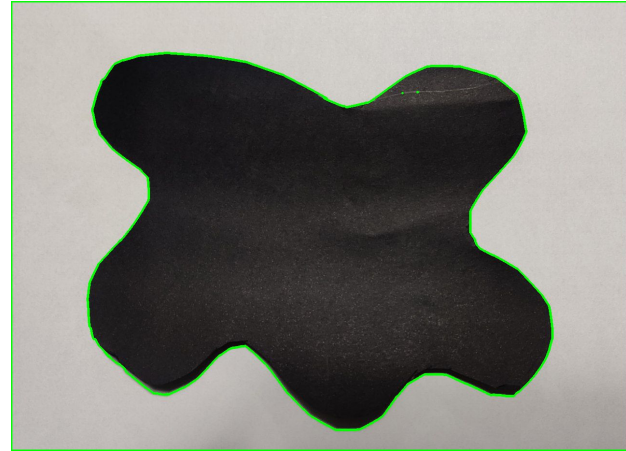


Imagem de teste



Contornos encontrados

# Biblioteca OpenCV

Biblioteca de código aberto que possui diversos algoritmos de visão computacional, oferecendo módulos de Processamento de Imagens, Análise de vídeo, reconstrução 3d, detecção de objetos, dentre outros...

Neste projeto utilizaremos apenas as funções relacionadas à detecção do contorno de objetos em imagens, para assim obtermos uma série de pontos e coordenadas que representam os contornos.

```
# Obtendo o Thresholding da imagem para que os possíveis objetos sejam detectados pela biblioteca
imgray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
ret, thresh = cv2.threshold(imgray, 127, 255, 0)

# Obter a lista de contornos encontrada pela biblioteca
contours, hierarchy = cv2.findContours(thresh, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

# Ordenando a lista de contornos encontrada de acordo com a área total do objeto contornado
largest_areas = sorted(contours, key=cv2.contourArea)
sortedValues = np.linspace(0, largest_areas[-2].shape[0]-1, num=numDots, dtype=int)
contourPoints = largest_areas[-2][sortedValues]
```

# Limiarização da imagem

Limiarização é o processo de segmentação de imagens baseada na **diferença dos níveis de cinza que compõem diferentes objetos de uma imagem**. A partir de um limiar estabelecido de acordo com as características dos objetos que se quer isolar, a imagem pode ser segmentada em dois grupos: o grupo de pixels com níveis de cinza abaixo do limiar e o grupo de pixels com níveis de cinza acima do limiar.

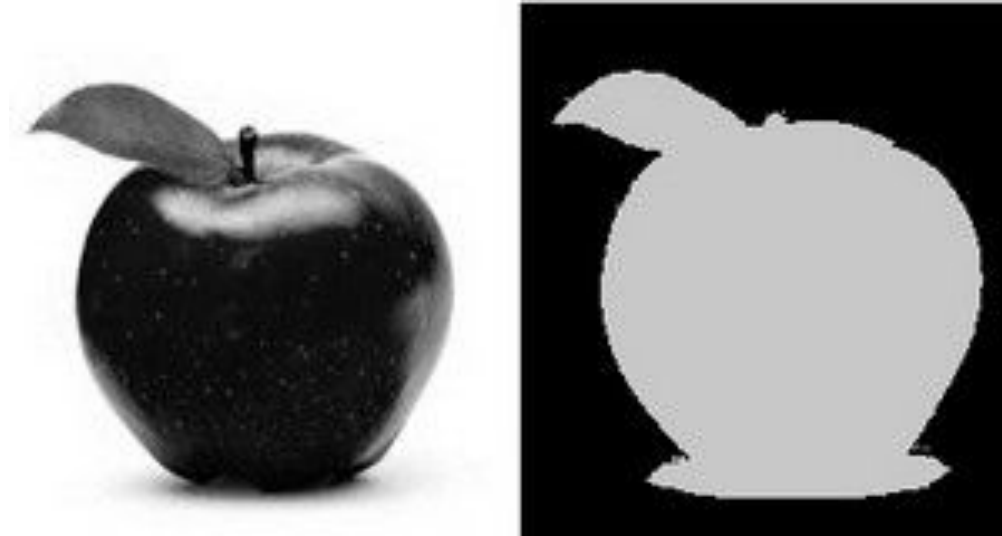
```
imgray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
```



# Limiarização da imagem

Dessa forma, através da limiarização da imagem é possível obter, através da função **findContours**, encontrar todos os contornos presentes na imagem.

```
ret, thresh = cv2.threshold(imgray, 127, 255, 0)
```



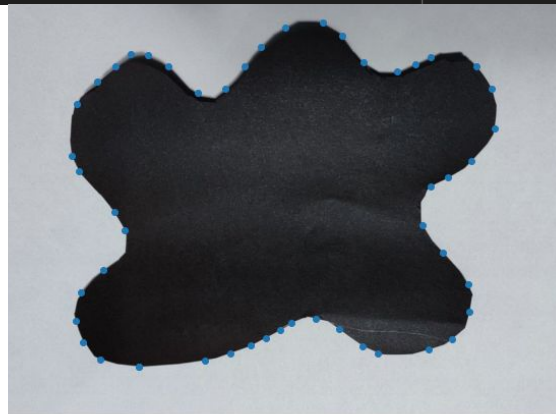
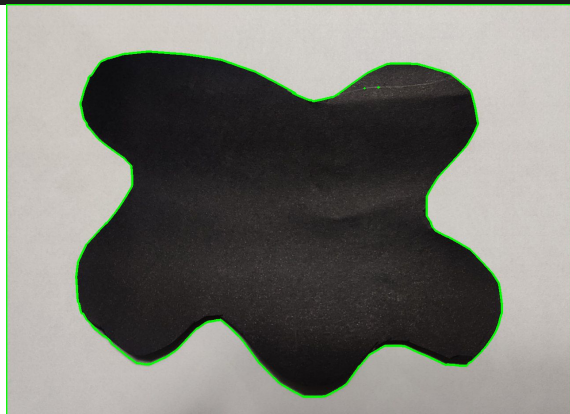
Limiarização da imagem de uma maçã

# Encontrando os contornos da imagem

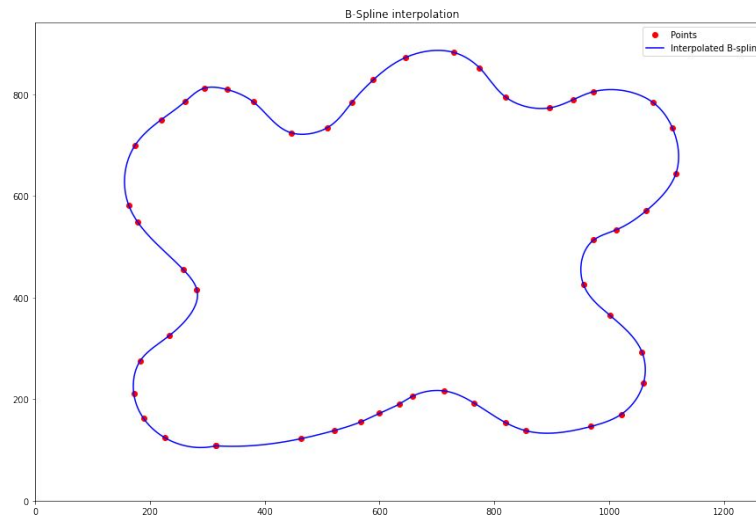
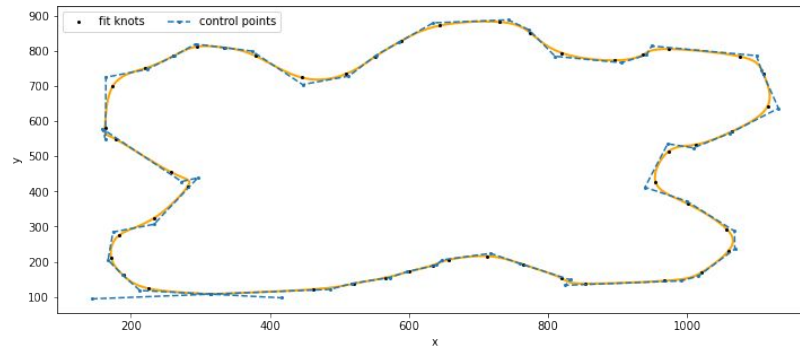
Finalmente, podemos encontrar a lista de todos os contornos presentes na imagem fornecida e selecionar N pontos destes que compõem o contorno para encontrar a representação B-spline da curva fechada.

```
# Obter a lista de contornos encontrada pela biblioteca
contours, hierarchy = cv2.findContours(thresh, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

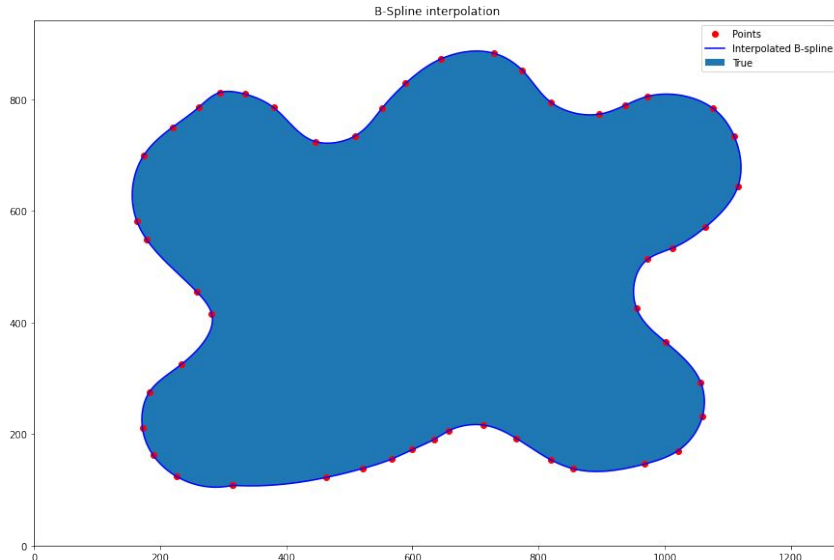
# Ordenando a lista de contornos encontrada de acordo com a área total do objeto contornado
largest_areas = sorted(contours, key=cv2.contourArea)
sortedValues = np.linspace(0, largest_areas[-2].shape[0]-1, num=numDots, dtype=int)
contourPoints = largest_areas[-2][sortedValues]
```



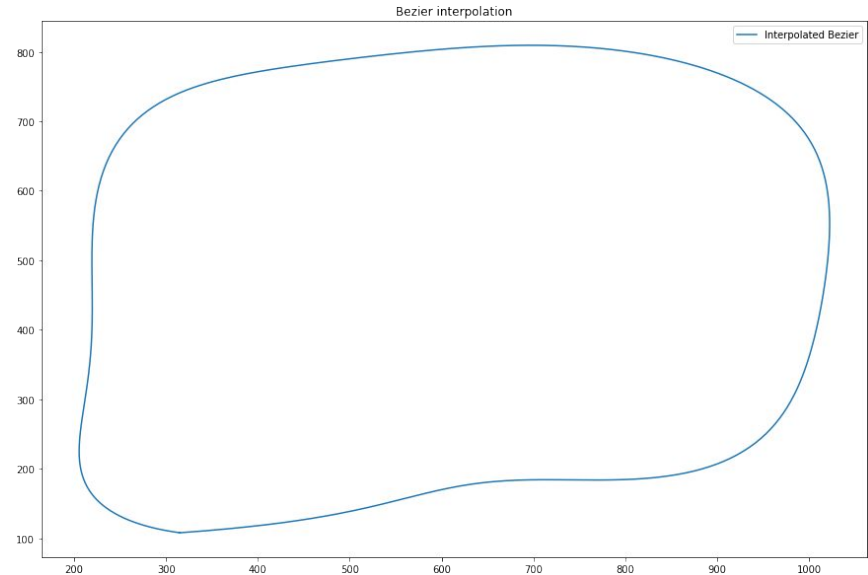
# Representações B-splines da curva fechada



# B-spline VS Bézier



Curva B-spline

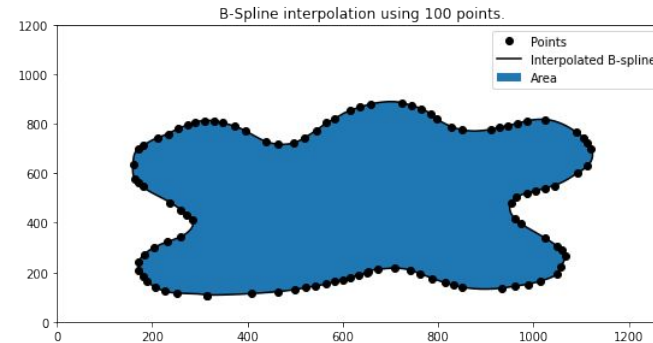
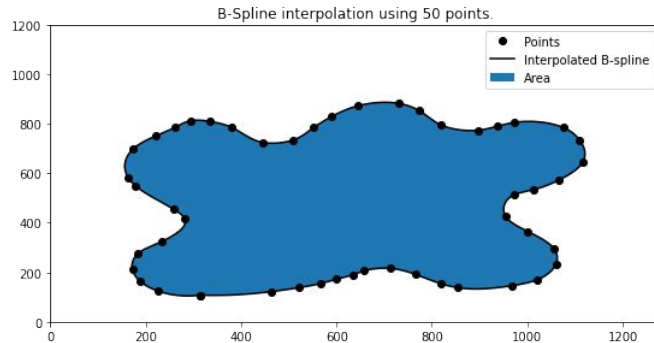
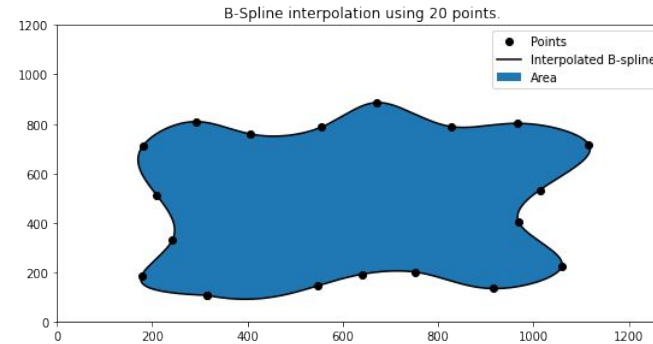
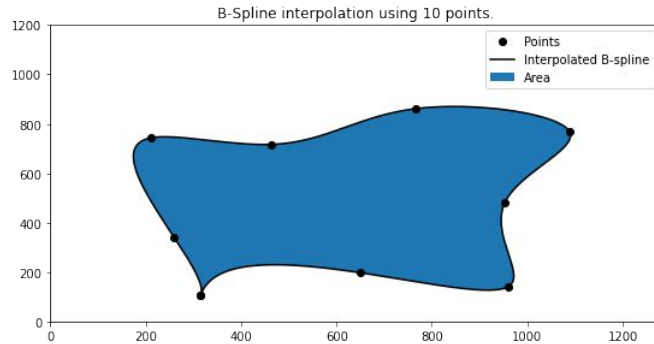


Curva de Bézier



# Comparações

Abaixo, um plot comparativo com as representações de B-splines das curvas fechadas de acordo com o número de pontos selecionados que formam o contorno da figura.



# Aplicações

Para este projeto desenvolvemos 2 aplicações com interface gráfica:

- A primeira aplicação faz a geração de curvas fechadas B-splines a partir de uma série de coordenadas XY fornecidas e permite a edição de pontos da curva e do grau utilizado na função B-splines da biblioteca Scipy.
- A segunda aplicação permite a geração de curvas fechadas B-splines por meio da inserção manual do usuário de pontos que representam os diversos contornos presentes na imagem, podendo escolher a cor de preenchimento das curvas geradas e também exportar as curvas geradas.

