

# Objetos - Extração de Bordas e de Propriedades

Kleber Kruger

kleberkruger@gmail.com – RA: 230226

## Abstract—

This report describes the third work done as an assessment for the Introduction to Digital Image Processing course. The objective was to implement algorithms of color transformation, contour identification, property extraction and area histograms in images with geometric objects. The results, even in images with large number of objects, proved to be effective and efficient since the internal functions in OpenCV library presented low execution time.

**Keywords**—Thresholding, Object Contours, Object Properties, Area Histogram on Objects.

## Resumo—

Este relatório descreve o terceiro trabalho realizado como avaliação para a disciplina de Introdução ao Processamento de Imagem Digital (MO433). O objetivo foi implementar algoritmos de transformação de cores, identificação de contornos, extração de propriedades e histograma de áreas em imagens com objetos geométricos. Os resultados, mesmo em imagens com grande número de objetos, mostraram-se eficazes e eficientes uma vez que as funções internas da biblioteca OpenCV apresentaram baixo tempo de execução.

**Palavras-chave**—Limiarização, Contornos de Objetos, Propriedades de Objetos, Histograma de Áreas em Objetos.



## 1 Introdução

O processo de análise de imagens permite identificar formas e objetos presentes em uma imagem a fim de extrair informações relevantes conforme um domínio de aplicação. Uma etapa fundamental neste processo é a segmentação e a extração dos contornos dos objetos. O objetivo deste trabalho é obter algumas medidas de objetos presentes em imagens digitais. Os quatro algoritmos implementados permitem: transformar uma imagem colorida e de fundo branco em uma imagem em níveis de cinza (similar a uma limiarização global), extrair os contornos dos objetos de uma imagem, obter as propriedades dos objetos (área, perímetro, centroide, excentricidade e solidez), e classificar objetos a partir de sua área total.

### 1.1 Organização do Texto

O relatório está dividido da seguinte maneira: na Seção 2 fundamenta-se as técnicas de contorno e extração de informações de objetos; na Seção 3 apresenta-se o manual de utilização do programa.

Na Seção 4 são descritos os experimentos realizados juntamente a algumas considerações importantes, e por fim, na Seção 5 as conclusões.

### 1.2 Forma de Entrega

Os códigos e os resultados foram submetidos pela plataforma Google Classroom. O nome do arquivo é: “KleberKruger-Trabalho3.zip”. Descompactado é composto por este relatório em formato PDF e três diretórios: “in”, contendo as imagens de entrada utilizada nos testes; “out” com os resultados do processamento; e “src” com os códigos do programa.

## 2 Fundamentação Teórica

Esta seção apresenta conceitos básicos sobre identificação de contornos e extração de propriedades de objetos.

### 2.1 Limiarização

Limiarização é um processo de segmentação de imagens que se baseia na diferença dos níveis

de cinza que compõe 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. Em uma imagem limiarizada, atribui-se um valor fixo para todos os pixels de mesmo grupo. Esta técnica é útil na fase de pré-processamento da extração de contornos. Na Figura 1 podemos observar o resultado de uma imagem limiarizada.

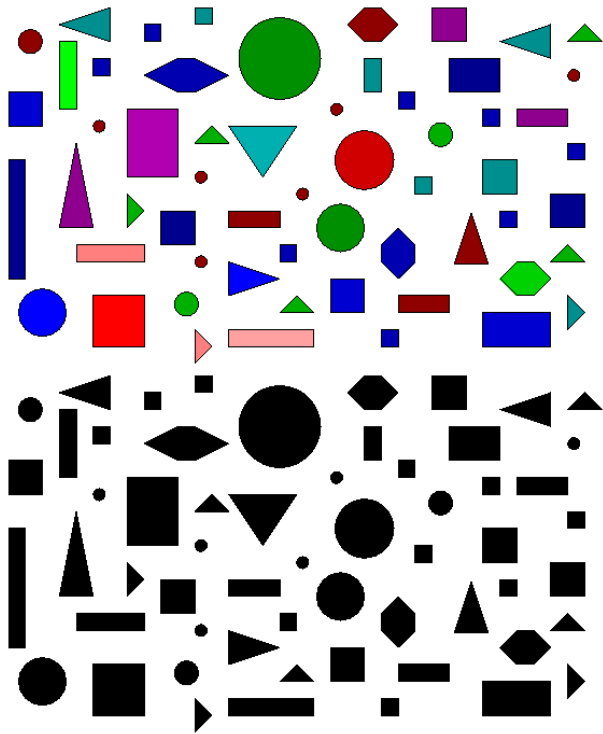


Figura 1: Método de limiarização global.

## 2.2 Contornos

Os contornos podem ser explicados simplesmente como uma curva que une todos os pontos contínuos (ao longo do limite), com a mesma cor ou intensidade. Por exemplo, considere a Figura 2.

Supondo-se uma imagem binária, o contorno é a curva que une todos os pontos brancos da borda. Portanto, ao se encontrar um contorno em uma imagem binária, encontra-se os limites dos objetos em uma imagem. Segundo a documentação da biblioteca OpenCV, “os contornos são uma ferramenta útil para análise de formas e detecção e reconhecimento de objetos” [?].

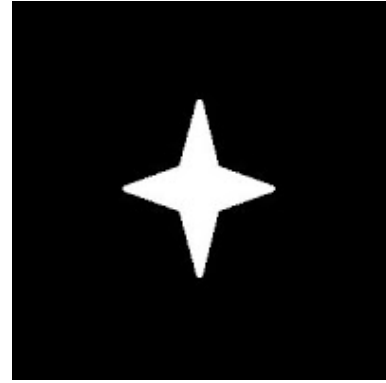


Figura 2: Contorno de uma imagem binária.

### 2.2.1 Momentos

Em processamento de imagens, na visão computacional e em campos relacionados, um momento da imagem é uma determinada média ponderada (momento) das intensidades dos pixels da imagem, ou uma função desses momentos, geralmente escolhidos para ter alguma propriedade ou interpretação atraente.

Momentos de imagem são úteis para descrever objetos após a segmentação. As propriedades simples da imagem encontradas por meio de momentos da imagem incluem área (ou intensidade total), seu centroide e informações sobre sua orientação.

### 2.2.2 Centroide

O centroide de uma forma é a média aritmética de todos os pontos em uma forma. Suponha que uma forma consiste em  $n$  pontos distintos  $x_1 \dots x_n$ , então o centroide é dado por:

$$\mathbf{c} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$$

No contexto do processamento de imagens e da visão computacional, cada forma é composta de pixels, e o centróide é simplesmente a média ponderada de todos os pixels que constituem a forma.

### 2.2.3 Propriedades dos Contornos

- Áreas de contorno: A área do contorno é igual ao número de pixels dentro do

contorno. Na biblioteca OpenCV, esta propriedade pode ser encontrada usando a função `cv2.contourArea()`.

- **Perímetro:** também é chamado de comprimento do arco. Pode ser encontrado usando a função `cv2.arcLength()`.
- **Excentricidade:** pode-se pensar na excentricidade como uma medida de quanto uma seção cônica se desvia de ser circular.
- **Solidez:** é a razão entre a área de contorno e a área convexa do casco.

## 2.3 Histograma de Áreas

Um histograma é um gráfico de frequência que tem como objetivo ilustrar como uma determinada amostra de dados está distribuída. O histograma mede quantas vezes tem-se um determinado valor dentro da distribuição de dados. Em imagens, esta técnica é comum para se obter métricas da distribuição dos níveis de cinza. Entretanto, neste trabalho, o histograma das áreas corresponde a distribuição dos tamanhos dos objetos conforme um conjunto de limiares.

# 3 Implementação

## 3.1 Bibliotecas Utilizadas

Este trabalho foi implementado na linguagem Python 3.7 com as seguintes bibliotecas:

- **Argparse:** utilizada para interpretar e extrair os argumentos de inicialização do programa;
- **OS, Glob e PathLib:** as classes importadas permitem manipular arquivos do programa a fim de obter informações do *filesystem*.
- **OpenCV:** biblioteca multiplataforma e *open-source* amplamente empregada em aplicações de processamento de imagens e de visão computacional. Seu uso foi para fazer o carregamento e a escrita de imagens no disco rígido, e para operações de contorno e extração de propriedades dos objetos;
- **Numpy:** oferece suporte para vetores e matrizes, além de conter um grande número de funções otimizadas para operar em elementos multidimensionais;

- **Matplotlib:** utilizado para fazer a plotagem dos histogramas;

## 3.2 Instruções de Uso

### 3.2.1 Parâmetros do Programa

O programa deve ser executado com os seguintes parâmetros:

- **-input/-in/-i:** caminho para o(s) diretórios ou a(s) imagem(s) de entrada;
- **-output/-out/-o:** caminho para o(s) diretórios ou a(s) imagem(s) de saída. A nomenclatura padrão para os arquivos de saída foram: <nome-original>-out-<tipolimiarização><parâmetro>;
- **-type/-t:** tipos das imagens (extensão do arquivo) a serem analisadas dentro dos diretórios de entrada;
- **-transform/-f:** executa o algoritmo de transformação definindo os objetos na cor informada neste parâmetro;
- **-contour/-c:** executa o algoritmo de contorno definindo os contornos dos objetos na cor e espessura informadas neste parâmetro;
- **-property/-p:** executa o algoritmo de extração de propriedades obtendo as propriedades especificadas;
- **-hist/-histogram/-g:** executa o algoritmo de histograma conforma os valores de limiar especificados;

### 3.2.2 Regras para os parâmetros

- 1) A quantidade de valores para o parâmetro <input> é de 0 (não informado) até n;
- 2) Os valores de cada <input> é o caminho para uma imagem ou um diretório;
- 3) Caso o valor de um <input> seja um diretório, apenas as imagens na sua raiz serão processadas;
- 4) Caso não informe nenhum valor, <input> será definido com 'in' (diretório de entrada padrão);
- 5) A quantidade de valores para o parâmetro <output> é 0 (não informado), 1 ou n (mesma quantidade de <input>);
- 6) Os valores de cada <output> é o caminho para um diretório de saída ou o nome de um arquivo resultante;

- a) Se o processamento de uma imagem de entrada é feito de  $k$  maneiras (aplicação de filtros, algoritmos, etc), cada imagem de entrada resultará em  $k$  imagens de saída, e o nome do arquivo seguirá a nomenclatura padrão conforme o nome de saída especificado;
  - b) Se nenhum nome foi especificado para o arquivo de saída, este seguirá a nomenclatura padrão conforme o nome original da imagem.
- 7) Um `<output>` associado a um `<input>` diretório, também é interpretado como um diretório, pois o resultado do processamento pode resultar em múltiplas imagens;
  - 8) Quando `<output>` não for informado pelo usuário:
    - a) Se `<input>` também não foi: `input = in`, `output = out`;
    - b) Caso o `<input>` tenha sido definido pelo usuário:
      - i) Se `<input>` for o caminho para uma única imagem, o valor de `<output>` seguirá a nomenclatura padrão, e a imagem de saída será salva no mesmo diretório da imagem de entrada;
      - ii) Se `<input>` for o caminho para um único diretório, `<output>` receberá o mesmo valor de `<input>`, e as imagens serão salvas conforme a nomenclatura padrão.
      - iii) Se `<input>` for um conjunto de valores (imagens e/ou diretórios), cada valor de `<input>` respeitará as duas sub-regras anteriores.
  - 9) Se a lista de `<input>`  $> 1$ :
    - a) `<output>` poderá estar vazio (não informado). Cada `input` seguirá a regra 8.2;
    - b) `<output>` poderá ter uma única entrada (obrigatoriamente um diretório de saída); - Observação importante: arquivos com mesmo nome resultante serão sobrescritos.
    - c) `<output>` poderá ter  $n$  entradas (em que ambas as listas têm o mesmo tamanho, pois os valores de `<input>``<output>` serão atribuídos na mesma ordem; cada elemento da lista deve respeitar a regra padrão.
  - 10) Caso apenas `<output>` tenha sido definido, `<input>` recebeu o valor padrão "in". Portanto, o valor de `<output>` precisa ser o caminho para um único diretório de saída;
  - 11) `<types>` define os tipos (extensões) de arquivos que serão incluídos na lista de imagens de entrada
  - 12) Se nenhum valor é informado pelo usuário, `types = '*'` (todos os tipos de imagens)
  - 13) Os parâmetros `-transform`, `-contour`, `-property` e `-histogram` possuem valor `None` como padrão. Ou seja, caso estas *flags* não sejam informadas, os respectivos métodos não são executados;
  - 14) Caso os parâmetros `-transform`, `-contour`, `-property` e `-histogram` sejam informados sem valores, (ex.: `-transform`), os seguintes valores serão definidos:
    - `-transform`: 0 - cor preta para os objetos no fundo branco
    - `-contour`: 0 0 255 1 - contorno na vermelha (0 0 255) de espessura 1
    - `-property`: "all" - todas as propriedades serão extraídas
    - `-histogram`: 1500 3000 - 3 faixas de objetos (objeto pequeno:  $area < 1500$ ; objeto médio:  $area \geq 1500$  e  $area < 3000$ ; objeto grande:  $area \geq 3000$ );

## 4 Experimentos

### 4.1 Entradas

Na pasta src, junto ao código do programa, existe um arquivo test.sh. Este arquivo descreve as execuções do programa durante os experimentos. Foram testadas com imagens de diferentes tamanhos, mas para não onerar o tempo de execução do teste, mantivemos as imagens de exemplo disponibilizadas no enunciado ([http://www.ic.unicamp.br/~helio/imagens\\_objetos\\_coloridos/](http://www.ic.unicamp.br/~helio/imagens_objetos_coloridos/)):

Os parâmetros de configuração para cada imagem de entrada foram: `-transform 0 -contour 0 0 255 1 -property all -histogram 1500 3000`. Ou seja, para todas as imagens, executou-se o método de transformação de cores usando-se a cor preta para os objetos em um fundo branco, o método de contorno usando-se cor vermelha e espessuras de tamanho 1, o método de extração de propriedades lendo todas as propriedades dos objetos, e o histograma com faixas de tamanho 0 a 1500, 1500 a 3000 e acima de 3000, para objetos pequenos, médios e grandes, respectivamente.

### 4.2 Saídas

Na pasta out é possível visualizar o resultado dos experimentos com as imagens de entrada fornecidas. Na Figura 3 o método de transformação de cores foi aplicado usando-se o valor preto e cinza (128), respectivamente.



Figura 3: Método de transformação.

Na Figura 4 são mostrados os resultados dos métodos de contorno para a mesma imagem da Figura 3. Foram utilizados os valores: 0 0 255 1, e 0 255 0, 2. Ou seja, bordas vermelhas de espessura 1 e bordas verdes de espessura 2, respectivamente.

O próximo experimento foi extrair as propriedades de área, centroide, perímetro, excentricidade e solidez dos objetos em uma imagem. Na

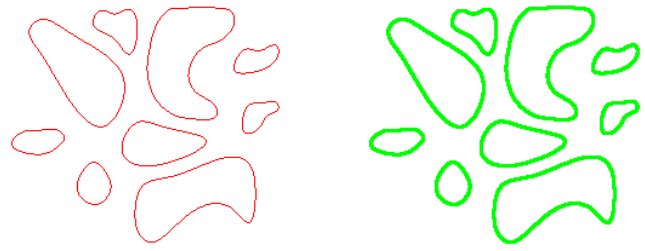


Figura 4: Método de extração de contornos.

Figura ?? são mostradas as informações impressas no terminal de execução, enquanto na ?? exibe-se a imagem resultante.

```
região 0: área: 4107.0 perímetro: 319.4213538169861 excentricidade: 0.7383454198792178 solidez: 0.7549632352941177
região 1: área: 843.5 perímetro: 125.63960921764374 excentricidade: 0.758542568466789 solidez: 0.9045576407506702
região 2: área: 3690.5 perímetro: 265.1198377689253 excentricidade: 0.9074588822575391 solidez: 0.9782637508283631
região 3: área: 584.0 perímetro: 104.91168713569641 excentricidade: 0.871986144022608 solidez: 0.9139280125195618
região 4: área: 478.0 perímetro: 94.42640566825867 excentricidade: 0.883524180063389 solidez: 0.925459825750242
região 5: área: 1761.5 perímetro: 179.78174459934235 excentricidade: 0.8759490714595861 solidez: 0.9718620689655172
região 6: área: 688.5 perímetro: 108.66904711723328 excentricidade: 0.8779532516093773 solidez: 0.972457627118644
região 7: área: 4067.0 perímetro: 311.0782079696655 excentricidade: 0.8819209459354206 solidez: 0.7806891256358576
região 8: área: 716.5 perímetro: 101.982754945755 excentricidade: 0.6253138152321981 solidez: 0.9801641586867305
```

Figura 5: Terminal na execução do método de propriedades de objetos.



Figura 6: Propriedades dos objetos.

O histograma de área pode ser conferido na Figura 7. A distribuição resultou em 5 objetos pequenos, 1 objeto médio e 3 objetos grandes.

Para não tornar este relatório extenso, as demais imagens de saída e os histogramas das imagens resultantes estão no Apêndice e no arquivo KleberKruger-Trabalho3.zip (diretório "out"), enviado via Google Classroom.

### 4.3 Tempo de Execução

O tempo de execução das funções podem ser conferidos no arquivo "execution\_time.txt" dentro

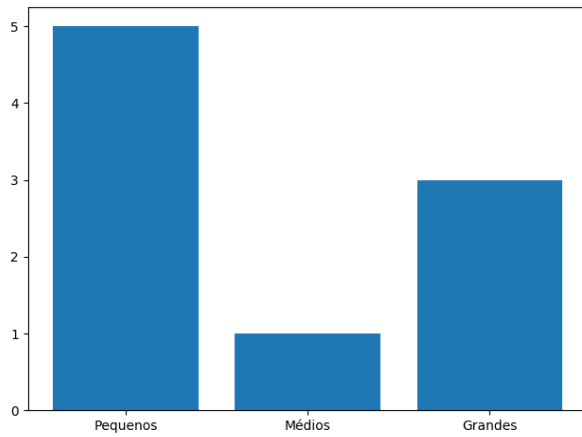


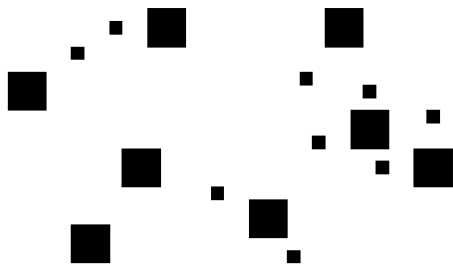
Figura 7: Histograma de área.

da pasta out. Como utilizamos funções internas do próprio OpenCV para as operações, o tempo de execução de todos os algoritmos foram rápidos, levando menos de 5 segundos, mesmo na imagem de maior dimensão e com maior número de detalhes.

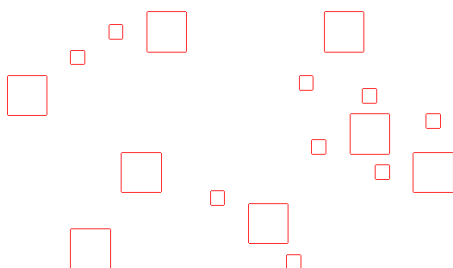
## 5 Conclusão

Este trabalho realizou tarefas essenciais na análise de imagens, tais como a limiarização, a extração de bordas e a identificação de propriedades (área, centroide, perímetro, excentricidade e solidez) de objetos de variadas formas geométricas. Essas tarefas nos permitiu aprender importantes funções da biblioteca OpenCV. Os resultados, mesmo em imagens com grande número de objetos, mostraram-se eficazes e eficientes, uma vez que as funções internas da biblioteca OpenCV apresentaram baixo tempo de execução.

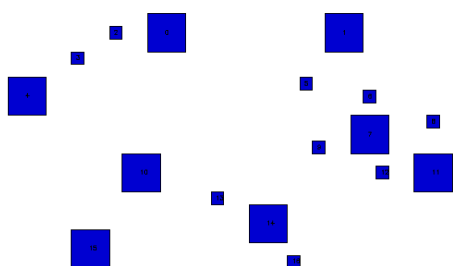
## Apêndice



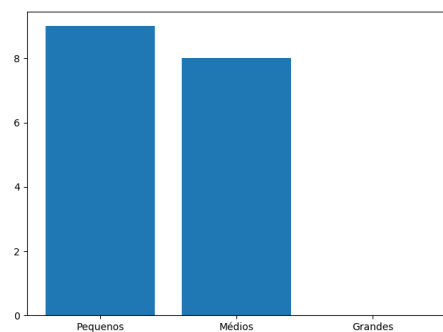
Transformação de objetos.



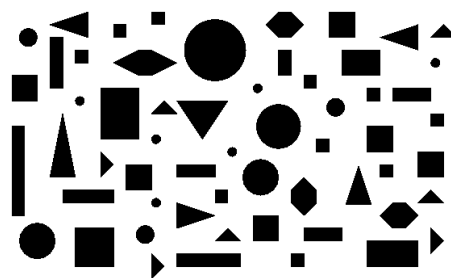
Extração de contornos.



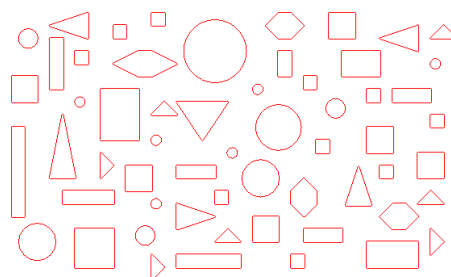
Propriedades dos objetos.



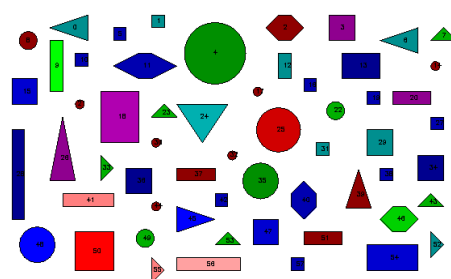
Histograma de área.



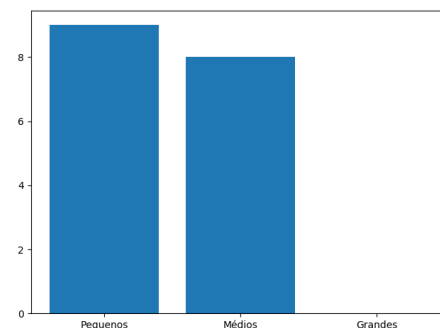
Transformação de objetos.



Extração de contornos.



Propriedades dos objetos.



Histograma de área.