Universidade de Brasília
Programa de Pós-Graduação em Informática

Processamento Morfológico e Momentos Invariantes

Disciplina: Processamento de Imagens
Professor: Ricardo Lopes de Queiroz
Aluno: Henrique Tibério Brandão Vieira Augusto (221101092)

Universidade de Brasília
Programa de Pós-Graduação em Informática

1. Introdução

O processamento de imagens desempenha um papel fundamental em diversas áreas, por exemplo: Reconhecimento de Padrões e Visão Computacional. Duas técnicas importantes nesse contexto são o Processamento Morfológico e o uso de Momentos Invariantes. Este relatório explora essas abordagens e suas aplicações. Este relatório trata de uma atividade consistente na aplicação de ambas para realizar uma tarefa de similaridade entre imagens.

2. Metodologia

A tarefa a ser realizada nesta atividade consiste em realizar as seguintes atividades, utilizando a imagem abaixo (Figura 1):

Figure 1: Imagem a ser utilizada como insumo da tarefa

1. Use processamento de imagens para gerar uma imagem binária da forma geométrica predominante e processamento morfológico para transformar em um objeto sólido;
2. Usando o método dos momentos invariantes descubra qual das imagens abaixo mais se encaixa na imagem binária resultante do seu processamento.

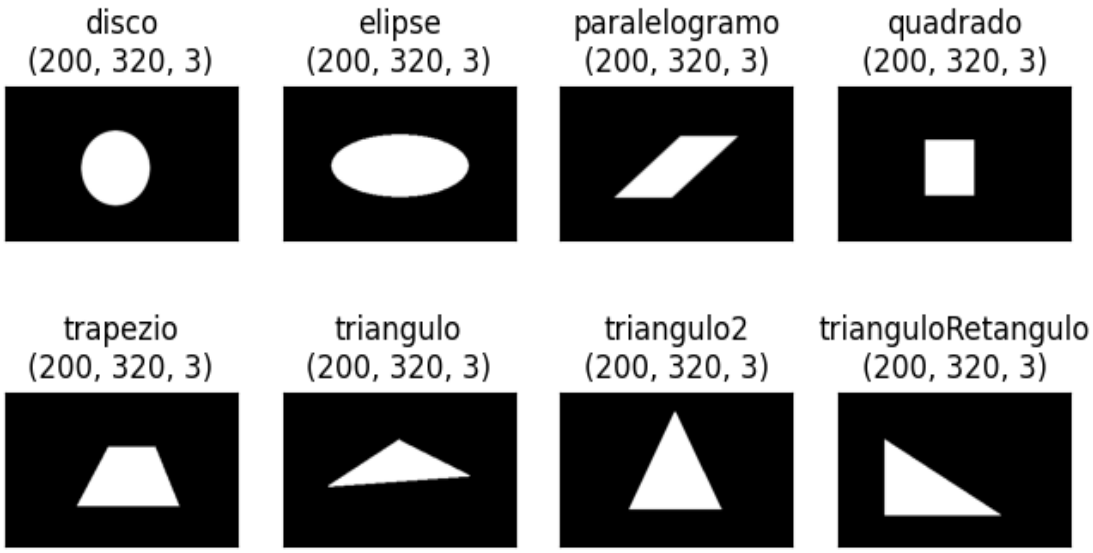

Figure 2: Sólidos para comparação

3. Processamento de Imagem

O processamento morfológico é uma técnica que analisa e transforma a forma e a estrutura das imagens. Pode ser utilizado para operações de pré-processamento, segmentação e reconhecimento de padrões. Dentre elas, as mais comuns incluem Erosão, Dilatação, Abertura e Fechamento:

- Erosão: Remove *pixels* da borda dos objetos, útil na remoção de ruídos e separação de objetos próximos.
- Dilatação: Adiciona *pixels* à borda dos objetos, útil na união de objetos próximos e preenchimento de pequenos espaços.
- Abertura: Combina erosão seguida por dilatação, útil na remoção de ruídos e na separação de objetos conectados.
- Fechamento: Combina dilatação seguida por erosão, útil na união de objetos próximos e no preenchimento de pequenos espaços.

Momentos invariantes são características matemáticas que permanecem constantes mesmo quando a imagem é transformada por rotação, escala ou translação. Também são utilizados em tarefas que envolvem reconhecimento de padrões e a classificação de objetos.

Para realizar esta tarefa, foi utilizada a linguagem Python em conjunto com as seguintes bibliotecas: Numpy, PIL, OpenCV e Matplotlib.

4. Resultados

De posse da imagem inicial, foram aplicadas as técnicas de Erosão, Dilatação, *Blur* e *Treshold*. A Figura 3 apresenta os resultados de cada operação aplicada sobre a imagem anterior, da esquerda para a direita, respectivamente. Enquanto a Figura 4 representa o resultado final.

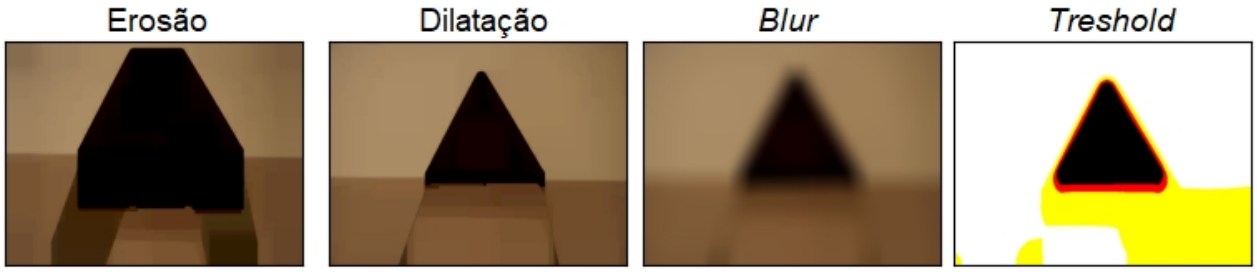

Figure 3: Resultados parciais das operações aplicadas

Figure 4: Resultado das transformações

De posse do resultado (Figura 4), foram calculados os momentos invariantes das imagens com o objetivo de determinar qual dos sólidos (Figura 2) seria mais próximo. A menor valor de distância foi de 5.97450393e-25, relativa ao sólido de nome “Triângulo 2”.

5. Conclusões

O processamento de imagens utilizando técnicas morfológicas e momentos invariantes desempenha um papel crucial na análise e interpretação de dados visuais. A combinação dessas abordagens oferece uma ferramenta poderosa para resolver desafios complexos em diversas áreas, desde a automação industrial até a medicina. O desenvolvimento contínuo dessas técnicas promete avanços significativos no processamento eficiente e preciso de informações visuais.

Anexo

Detalhes da implementação utilizando código em Python para processamento das imagens:

```
import cv2

import numpy as np
import seaborn as sns

from math import sqrt
from random import choice

from PIL import Image
from matplotlib import pyplot as plt

%matplotlib inline

kernel = np.ones((100,100), np.uint8)

erosao = lambda x: cv2.erode(x, kernel, iterations=1)
dilatacao = lambda x: cv2.dilate(x, kernel, iterations=1)
tresh = lambda i: cv2.threshold(i, 50, 255, cv2.THRESH_BINARY)[-1]

def momentos_invariantes(i1, i2):
    i1, i2 = cv2.cvtColor(i1, cv2.COLOR_BGR2GRAY), cv2.cvtColor(i2,
cv2.COLOR_BGR2GRAY)
    m1, m2 = cv2.moments(i1), cv2.moments(i2)
    hum1, hum2 = cv2.HuMoments(m1), cv2.HuMoments(m2)
    # dist = np.linalg.norm(hum1 - hum2)
    dist_hu7 = np.abs(hum1[-1] - hum2[-1])
    return dist_hu7
```