

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ - CÂMPUS
CURITIBA - SEDE CENTRO
BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

Giovanni Mito
Alexandre Alberto Menon
Jennyfer Mikaelly Melo

FUNDAMENTOS DE PROGRAMAÇÃO
ICSF13 - 2023-2 - PROJETO 3

CURITIBA
2023

1.INTRODUÇÃO

Esse relatório refere-se ao projeto 3 da disciplina de Fundamentos de Programação, atividade cuja qual abordou lógicas de programação referente ao tratamento e identificação de padrões em imagens, estudando matrizes de forma aplicada.

2. Se ambientando com a estrutura do projeto

Antes de iniciarmos a resolução do que fora pedido para solucionar no projeto, realizamos uma etapa já que já se tornou comum nos trabalhos da disciplina, procurando entender a estrutura do projeto, executar os testes mais básicos, bem como printar a matriz no terminal, e também analisar as imagens criadas nos testes.

Após esse processo, conseguimos estipular sumariamente o que deveria ser feitos e as etapas que tínhamos que seguir.

3. Sobre a questão

A questão do projeto 3 se resumia, a grosso modo, em identificar dois círculos contidos em uma imagem gerada apenas com pixels na escala de cinza, calcular os centros desses círculos e retornar o ângulo em radianos formados por estes.

3.1 Identificar as bolinhas

Foram necessárias algumas etapas para primeiramente identificar as bolinhas, sendo elas:

3.1.1 Binarização da imagem

Uma vez que a imagem era disposta em uma escala de tons de cinza, os arredores dos círculos poderiam conter pixels brancos ou cinza claro e as bolinhas poderiam conter pixels pretos ou cinza escuro, tornando mais árduo o trabalho de identificar o que era o círculo requisitado, justamente por poder haver aglomerados de pixels brancos em qualquer outra porção da imagem ou se o círculo tivesse pedaços em preto.

Assim, como a escala de tonalidades cinzas vai de 0 a 255, utilizamos como base 127 o parâmetro para distinguir as cores; cores que pertencem ao intervalo $[0,126]$ são definidas como preto, 0 na escala, e cores que pertencem ao intervalo $[127,255]$ são definidas como branco, 255 na escala.

Por conseguinte, esse método limpou grande parte da imagem, sobrando poucos aglomerados de pixels brancos e também consideravelmente menores.

3.1.2 Preenchendo a bolinha e limpado arredores

Assim como a estratégia da binarização, procuramos encontrar modos para tratar ainda mais a imagem. Prosseguindo desse ponto, percebemos que havia ainda pixels pretos dentro das bolinhas e seus arredores eram bastante

sombreados de pixels brancos, além de ainda haver alguns pixels brancos perdidos em outras regiões da imagem.

Dessa forma, aplicamos uma estratégia utilizada na atividade feita em sala de aula denotada com “desafio do jogo da vida”, onde trabalhamos bastante em cima de tratar uma posição da matriz com base nas posições vizinhas. Com isso, não foi diferente na nossa matriz binarizada, e funcionou da seguinte maneira:

Se uma posição de pixel branco (255) tivesse todos os seus vizinhos (superior, inferior, direita e esquerda) com pixels pretos, essa posição tornar-se-ia um pixel preto, e da mesma forma funciona para um pixel preto com vizinhos brancos.

3.1.2 Rotular as bolinhas

Essa foi a etapa que nos daria a diferenciação de cada bolinha presente na imagem, e que excluiria os aglomerados de pixels branco que sobraram.

A diferenciação das duas bolinhas era um passo que não poderia ser esquecido, visto que, se achássemos as coordenadas x dos centros, e posteriormente as coordenadas y, não seria possível relacionar qual coordenada pertencia a cada bolinha, podendo acabar ficando trocadas.

Dessa maneira, os rótulos nos pixels brancos conseguem diferenciar cada aglomerado da seguinte maneira:

Quando encontrado um pixel branco, analisa-se os vizinhos desse pixel, e verifica se não existe nenhum outro pixel branco rotulado. Caso haja um pixel rotulado, o pixel analisado recebe o menor rótulo desses vizinhos, e caso contrário, o pixel analisado recebe um novo rótulo único.

B C D
A ●

Nesse caso, verifica se as posições ABCD possuem rótulos, isso porque essa leitura é feita de cima para baixo, da esquerda para direita, ou seja, se essas posições possuíam pixels brancos, de certo que já estariam rotuladas com a mesma estratégia.

Ainda assim, há chances de na mesma bolinha haver mais de um rótulo, e por isso, fazemos uma varredura de baixo pra cima, da direita para esquerda, seguindo as mesmas regras, porém analisando outros vizinhos:

● A
D C B

Isso garante que se uma bolinha tiver mais de um rótulo, essa posição de rótulo diferente receberia menor rótulo dos vizinhos.

Em resumo, essa estratégia diferencia cada aglomerado de pixels brancos com rótulos começando de 1, sendo acrescentado de um em um, o que nos dá os dois maiores como resposta ao fazermos uma contagem, que são justamente as bolinhas que devemos analisar, e agora, de maneira distinguida. Assim, fizemos uma verificação e guardamos os dois números dos rótulos em maior quantidade, o que permite, enfim, buscar outras informações da bolinha.

3.2 Calcular do centro

Uma vez que temos as bolinhas diferenciadas entre si por conta dos rótulos, bastou pensar na estratégia de encontrar o centro das bolinhas. Inicialmente, acreditamos que o centro estaria relacionado com a linha e com a coluna que possuísse a maior quantidade do rótulo analisado. Apesar de fazer sentido até certo ponto, e a estratégia ter alcançado um resultado relativamente agradável, os testes retornavam um erro médio de precisão de aproximadamente 6 casas.

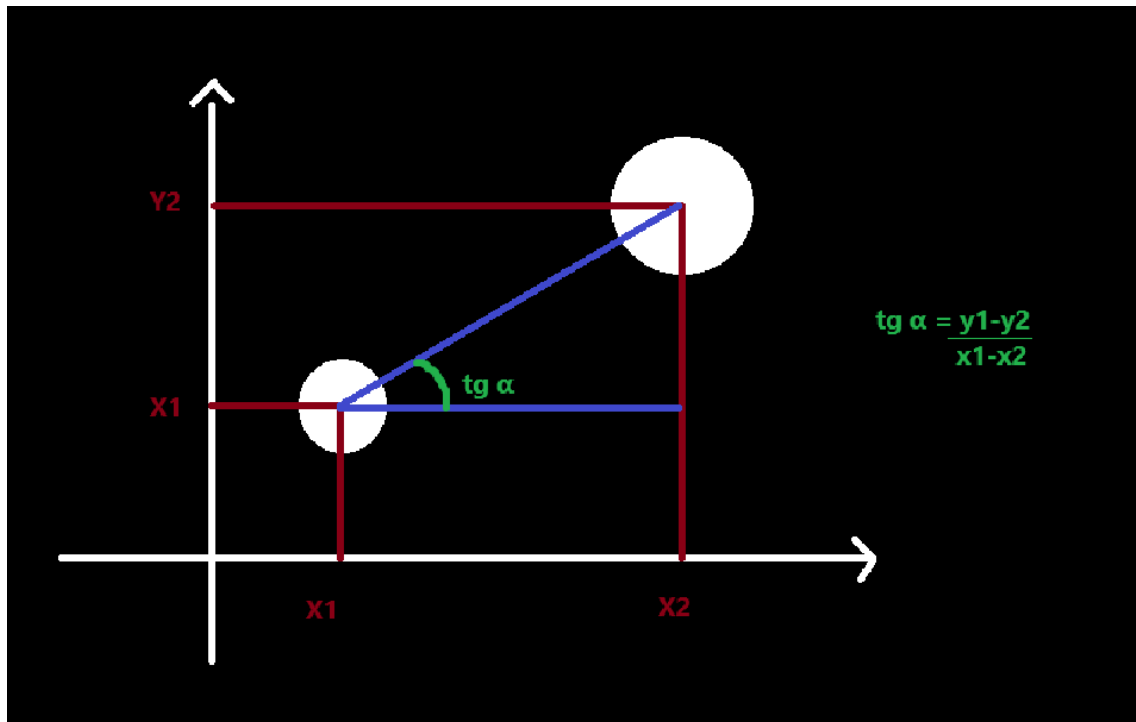
Portanto, decidimos mudar a estratégia de encontrar o centro da bolinha, o que nos tomou um tempo considerável. Assim, calculamos o centro de outra maneira, dessa vez, fazendo uma média entre a posição do primeiro pixel rotulado encontrado no eixo y lendo a matriz de cima para baixo, da esquerda para a direita, e a posição do primeiro pixel encontrado no eixo x lendo a matriz de baixo para cima, da direita para esquerda. A mesma estratégia é feita para encontrar a coordenada x do centro, apenas mudando o referencial, lendo a matriz de forma que as linhas são lidas se referenciando pelas colunas.

```
//Cálculo para achar a posição central das bolinhas com relação ao eixo y
*y1=( (*finaly_bola_1-*comecoy_bola_1)/2)+*comecoy_bola_1;
*y2=( (*finaly_bola_2-*comecoy_bola_2)/2)+*comecoy_bola_2;

//Cálculo para achar a posição central das bolinhas com relação ao eixo x
*x1=( (*finalx_bola_1-*comecox_bola_1)/2)+*comecox_bola_1;
*x2=( (*finalx_bola_2-*comecox_bola_2)/2)+*comecox_bola_2;
```

3.3 Calcular ângulo formado

Com as coordenadas do centro das bolinhas, colocamos as posições na equação da tangente:



Após realizar esse cálculo, chamamos a função “atan” da biblioteca math.h, função arco tangente que nos fornece o ângulo em radianos, bastando apenas retornar esse resultado na função.

4. Considerações finais

Ficamos muito satisfeitos com o que foi desenvolvido no Projeto 3 e em ver mais uma vez a aplicação direta dos conceitos aprendidos em sala de aula. Além disso, dada a estrutura do projeto, utilizando structs e matrizes e passando por diferentes funções, sentimos que foi muito benéfico para melhorar as habilidades relacionadas a conceitos como ponteiros e structs.