Insper – Instituto de Ensino e Pesquisa



Documentação atividade Visão 2

Leonardo Pereira Medeiros Lucas Chen Alba

Engenharia da computação: 3º semestre

Disciplina: Robótica computacional

São Paulo, Setembro de 2016

Sumário

[Introdução 3](#_Toc475397170)

[Funcionalidade do programa 3](#_Toc475397171)

[Programas e materiais 3](#_Toc475397172)

[Programação 4](#_Toc475397173)

[Implementações futuras 5](#_Toc475397174)

[Resultados e conclusões 5](#_Toc475397175)

[Referências bibliográficas 6](#_Toc475397176)

# Introdução

Este é um projeto com o objetivo principal de aplicar a programação da biblioteca Opencv do python através de um programa que use a webcam de um notebook para detectar círculos e calcular a orientação e distância dos círculos até a webcam.

# Funcionalidade do programa

O nosso programa como dito anteriormente é feito integralmente da linguagem python focando na biblioteca opencv, e como bibliotecas secundárias, as numpy, matplotlib e time. Utilizamos um modelo pré pronto feito por Fábio Miranda e Matheus Dib, sendo aprimorado para focar a detecção apenas aos círculos desejados (no caso, três círculos impressos em uma folha A4) e adicionando as funções de cálculo de distância do papel até a webcam do computador e a orientação do papel.

# Programas e materiais

Programas utilizados:

-Programa Spyder para python 2 (contido no programa Anaconda python 2)

-Github

-Paint

Máquinas utilizadas:

-Computador

Materiais utilizados:

-Papel A4

# Programação

Como dito, a programação foi feita em cima de um modelo básico que já detectava círculos, porém este programa apenas detectava os círculos (e detectava não só os círculos do papel como detectava vários outros), com isso, foi nos dado a tarefa de melhorar o programa, no sentido de implementar um algoritmo de detecção de distância do papel até a câmera, a orientação do papel (horizontal ou vertical) bem como refinar a detecção de círculos ao ponto de não detectar, ou quase, círculos errados.

A função de cálculo da distância foi feito utilizando-se uma comparação entre o tamanho do objeto real e o medido no computador e a distância real e a distância focal (constante). Para isso coletamos uma série de dados com o objetivo de se chegar na distância focal, sendo elas: a distância inicial para o experimento, 34cm, a média do diâmetro dos círculos detectados no computador, 116 pixels, e a medida de um dos círculos no papel 6,5 cm. Obtido o valor da distância focal em pixels, pode se obter a partir da comparação citada acima as novas distâncias baseadas nos novos diâmetros medidos.

Quanto à orientação foram utilizadas as distâncias em x e y colocando uma função condicional para checar se o papel estava na orientação vertical ou horizontal.

Por fim, a otimização de detecção foi feita principalmente ajustando-se alguns parâmetros de transformação da imagem capturada para a imagem analisada, por exemplo o blur.

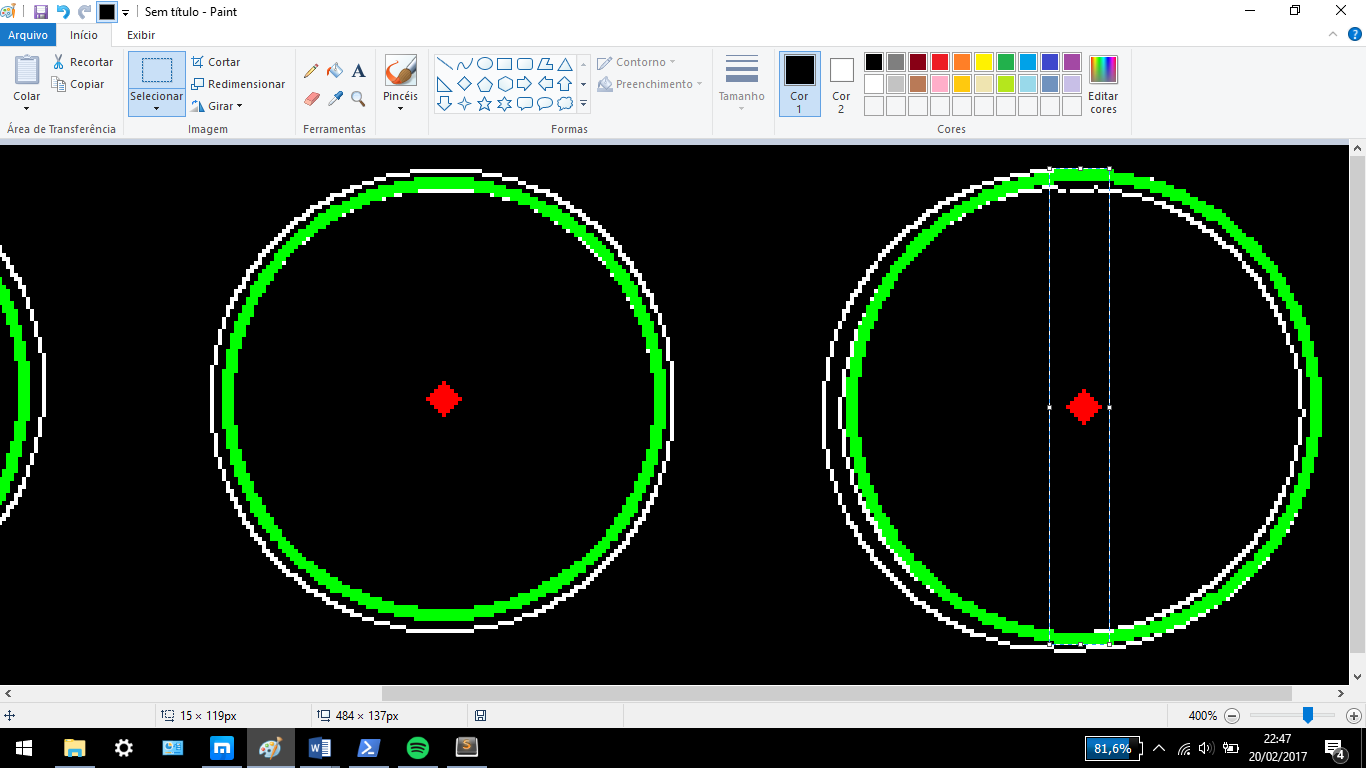


Figura 1 – Medindo o diâmetro do círculo em pixels, medido na tela do computador usando Paint.

[](https://www.youtube.com/watch?v=gbuZDNESCl0)

Figura 2 – O vídeo demonstrando as funcionalidades pode ser visto no Youtube, apenas clicar na imagem ou acessar o link: <https://www.youtube.com/watch?v=gbuZDNESCl0>

# Implementações futuras

Para implementações ou projetos futuros ao invés de detectar apenas círculos poderíamos fazer algo para detecção de rostos ou algo do tipo, e realizar algo mais ‘emocionante’ do que ‘apenas’ printar na tela a distância e orientação de círculos.

# Resultados e conclusões

Com esta atividade aprendemos várias funções de captura de imagens e transformação para figuras geométricas da biblioteca Opencv. Ela com certeza foi uma introdução para projetos maiores que teremos mais tarde.

# Referências bibliográficas

-Professor Fábio Miranda

-Ninja Matheus Dias

-Site: <http://www.pyimagesearch.com/2015/01/19/find-distance-camera-objectmarker-using-python-opencv/>