

<https://eventos.utfpr.edu.br//sicite/sicite2019>

Desenvolvimento de um dispositivo para medida da posição de uma esfera sobre uma plataforma.

Development of a device for measuring the position of a sphere in a plate system

RESUMO

Neste trabalho são apresentados os estudos feitos para a detecção da posição de uma esfera sobre uma plataforma. Posteriormente, este método de detecção será acoplado a um sistema plataforma-esfera, ainda em fase de desenvolvimento, cujo objetivo será controlar uma esfera que rola sobre uma plataforma articulada, em uma posição pré-fixada. Resultados mostram que foi possível obter com grande precisão as coordenadas da posição da esfera numa plataforma utilizando-se uma webcam e um Raspberry Pi.

PALAVRAS-CHAVE: Plataforma-bola. Posição. Processamento de imagem. Raspberry.

Mateus Cesar Tosti

Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Paraná, Brasil

Kleber Romero Felizardo

Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Paraná, Brasil

ABSTRACT

THIS WORK PRESENTS THE STUDIES MADE TO DETECT THE POSITION OF A SPHERE ON A PLATFORM. SUBSEQUENTLY, THIS DETECTION METHOD WILL BE COUPLED TO A PLATFORM-SPHERE SYSTEM, STILL UNDER DEVELOPMENT, WHOSE OBJECTIVE WILL BE TO CONTROL A SPHERE THAT ROLLS OVER AN ARTICULATED PLATFORM IN A PRE-FIXED POSITION. RESULTS SHOW THAT IT WAS POSSIBLE TO OBTAIN VERY ACCURATELY THE COORDINATES OF THE SPHERE POSITION ON A PLATFORM USING A WEBCAM AND A RASPBERRY PI.

KEYWORDS: Ball-plate System. Position. Image Processing. Raspberry.

Recebido: 19 ago. 2019.

Aprovado: 01 out. 2019.

Direito autoral: Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



INTRODUÇÃO

O sistema plataforma-esfera apresenta desafios interessantes para o ensino da teoria de controle e tem como finalidade principal manter o equilíbrio de uma esfera em uma posição pré-definida, sobre uma plataforma plana. (RAHMAT; Fuaad,2016)

Este projeto consistiu no desenvolvimento de um sistema de vídeo-sensoriamento para detecção da posição de uma esfera sobre uma plataforma. Neste sistema foi utilizado um Raspberry Pi para aquisição das imagens de uma webcam e posterior pós-processamento destas imagens através da transformada de Hough para detecção da posição desta esfera.

O Raspberry Pi é um pequeno computador que possui um sistema operacional próprio que facilita a construção de projetos pois esse sistema já é otimizado para programação. O seu poder de processamento é capaz de compilar códigos escritos em linguagem Python com as bibliotecas de imagens e com o suporte para webcams. (RICHARDSON; WALLACE,2012)

Para a aquisição e processamento das imagens existe a biblioteca OpenCV. Esta biblioteca possui duas funções de interesse para este trabalho: uma função que realiza a transformada de Hough que pode ser utilizada na detecção de linhas e círculos e uma outra função que aplica um filtro para diminuir a quantidade de ruídos na imagens. (PAJANKAR,2015)

MATERIAIS E MÉTODOS

Para a realização deste trabalho utilizaram-se os seguintes recursos: um Raspberry Pi 3 modelo B, uma webcam Logitech c920 para aquisição de imagens, uma esfera de plástico, uma placa de EVA de 33x26 cm e um suporte de madeira.

Primeiramente foi necessário a instalação da biblioteca OpenCV no Raspberry Pi 3, para a utilização das funções da transformada de Houghs para detecção de círculos. Essa transformada trabalha com três parâmetros: coordenadas x e y e raio r do círculo.

Neste trabalho, x e y são as coordenadas da posição do centro da circunferência da esfera e r é o raio da esfera (parâmetro conhecido). O algoritmo da transformada de Houghs detecta circunferências e calcula valores para x e y , visto que x_0 e y_0 é a coordenada de cada ponto da imagem e será percorrido por todo o parâmetro da plataforma, ou seja, é a varredura de detecção.

$$(x_0 - x)^2 + (y_0 - y)^2 = r^2 \quad (1)$$

Essa detecção funciona como um sistema de votos, o algoritmo primeiramente detecta as bordas dos objetos em um fundo de escala binário, analisa cada ponto dessas e calcula a quantidade de círculos que passam por ele, quanto mais circunferências contidas naquele ponto, mais votos ele recebe e a chance de existir um círculo nessa região é maior.

A função pode ser escrita como `cv2.HoughCircles`, porém para que essa função seja aplicável é necessário introduzir um filtro preto e branco na imagem que está sendo capturada pela webcam. Para isto basta usar `cv2.cvtColor` que é um comando integrado da biblioteca e que faz a conversão de cores. Outros parâmetros também são importantes, dentro do comando `cv2.HoughCircles` existe a possibilidade de controle na detecção das circunferências, pode ser alterar o raio máximo e mínimo, quantidade de círculos entre outros dados da circunferência a ser detectada. Na Figura 1 é apresentado o código-fonte desenvolvido.

Figura 1 – Código de detecção de círculos.

```
import cv2
import numpy as np
cam = cv2.VideoCapture(0)
print ('Resolucao da camera ' + str(int(cam.get(3))) + 'x' + str(int(cam.get(4))))
w=640
h=480
cam.set(3,w)
cam.set(4,h)
print ('RESOLUCAO DEFINIDA ' + str(w) + 'x' + str(h))
while (True):
    ret , frame = cam.read()
    grey = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    blur = cv2.blur(grey,(5,5))
    circles = cv2.HoughCircles(blur,method=cv2.CV_HOUGH_GRADIENT,dp=1,minDist=500,param1=50,param2=13,minRadius=30,maxRadius=100)
    if circles is not None:
        for i in circles[0,:]:
            cv2.circle(frame,(i[0],i[1]),i[2],(0,255,0),2)
            print ('(X,Y)' + '(' + str((i[0])*0.0531) + ',' + str((i[1])*0.0541) + ')')
            cv2.imshow('CIRCULO DETECTADO',frame)
    if cv2.waitKey(1)==1:
        break
cv2.destroyAllWindows()
cam.release()
```

Fonte: Autoria Própria.

As informações ficam salvas em um vetor `i`, onde sua primeira componente armazena a coordenada `x` e a sua segunda a coordenada `y`. A unidade dessa posição inicialmente é dada em pixels, ou seja 640x480 pixels. Posteriormente é feita uma conversão para centímetros, utilizando aproximação linear, e as informações são dadas entre os valores extremos 33x26cm, sendo `x` o comprimento e `y` a largura da plataforma.

RESULTADOS E DISCUSSÕES

Os resultados das coordenadas da posição `x` e `y` da esfera na plataforma podem ser visto nas Figuras 2 a 4. Vale ressaltar que a origem está localizada no canto superior esquerdo, pois a câmera foi posicionada de ponta cabeça.

Na Figura 2, a esfera está bem perto da origem (0,0) cm da plataforma e o aplicativo desenvolvido nos fornece uma leitura de aproximadamente (1.778, 1.217) cm.

A figura 3 mostra a esfera no limite das dimensões da plataforma de 33x26 cm. Pode-se visualizar que a leitura de aproximadamente (33, 25.88) cm é bem precisa nestes limites de ponto máximo.

Na Figura 4 a esfera está em uma posição aleatória na plataforma.

CONCLUSÃO

Pode-se concluir que o desenvolvimento de um sistema de vídeo-sensoriamento para medição da posição da esfera na plataforma foi bem sucedido. O uso do Raspberry Pi para aquisição da imagem da webcam e posterior processamento da imagem utilizando-se a transformada de Hough garantiram resultados bem precisos da posição da esfera nesta plataforma.

REFERÊNCIAS

RAHMAT, Fuaad. Application of intelligent controller in a ball and beam control system. **International journal on smart sensing and intelligent systems**, v. 3, n. 1, p. 45-60, 2010.

RICHARDSON, Matt; WALLACE, Shawn. **Getting started with raspberry PI**. "O'Reilly Media, Inc.", 2012.

PAJANKAR, Ashwin. **Raspberry Pi computer vision programming**. Packt Publishing Ltd, 2015.