ISEL
INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

2º Projeto Laboratorial

Visão Artificial e Realidade Mista

Trabalho realizador por:

Duarte Valente | A47657

Docente:

Eng. Pedro Jorge

Curso: MEIM

2023

Índice

Índice de ilustrações

Introdução

a) Descrição

O objetivo do trabalho laboratorial passa por criar uma aplicação onde seja possível de detetar marcadores e por sua vez projetar objetos sobre desses marcadores, de forma a pôr em prática os conceitos aprendidos sobre o tema de realidade aumentada, assim como ganhar experiência com com a biblioteca OpenCV e com marcadores ArUco.

b) Âmbito

O projeto laboratorial para pôr aplicar os conceitos lecionados na unidade curricular de Visão Artificial e Realidade Mista, do Mestrado em Engenharia Informática e Multimédia do DEETC do ISEL.

c) Documentação

Documentação de apoio à unidade curricular de Visão Artificial e Realidade Mista, assim como documentação das bibliotecas OpenCV e ArUco.

Desenvolvimento

a) Calibração da Câmara

Para pudermos colocar em prática o objetivo principal de integrar os objetos no mundo virtual o primeiro passo, passa por efetuar a calibração da câmara, pois esta desempenha o papel fundamental de captar o mundo real e de efetuar a deteção dos marcadores onde serão posicionados os objetos. Esta etapa é fundamental para obter informações mais precisas sobre os parâmetros de captação da câmara. Parâmetros esses, essenciais para uma interpretação correta das imagens capturadas e para assim, ser possível estimar com uma maior precisão onde se encontram os marcadores. Após captados estes parâmetros, podemos então corrigir as distorções geométricas e óticas criadas pela câmara, que caso não sejam corrigidas podem comprometer o desempenho dos próximos passos.

Para efetuarmos esta calibração, foi utilizado um tabuleiro em xadrez 8x6, pois é um padrão de possível identificação pela biblioteca do OpenCV através da função “findChessboardCorners”, podendo assim, com os pontos encontrados do tabuleiro calcular parâmetros como os coeficientes de distorção, os vetores de rotação, os vetores de translação e ainda a matriz de calibração, matriz essa, que estabelece a relação entre as coordenadas 3D do mundo real e as coordenadas 2D na imagem captada.

Assim, colocando este conceito em prática foram utilizadas dez imagens diferentes (figura 1) para melhorar a precisão da calibração da camara.

Figura 1 - Imagens para calibração

De seguida, foram convertidas estas imagens para Gray Scale possibilitando assim a utilização da função “cv.findChessboardCorners()” do OpenCV, para captar então os pontos do tabuleiro nas diferentes imagens.

Assim que encontrados, estes pontos são adicionados a um array (objpoints), array este que guarda os pontos 3D calculados a partir da imagem. Para alem dos pontos 3D tambem são guardados os pontos 2D que foram encontrados na imagem, para tambem ser possível desenhar e mostra estas imagens como mostra a figura 2, esta deteção, possibilita tambem analisar se este processo está a ser bem efetuado.

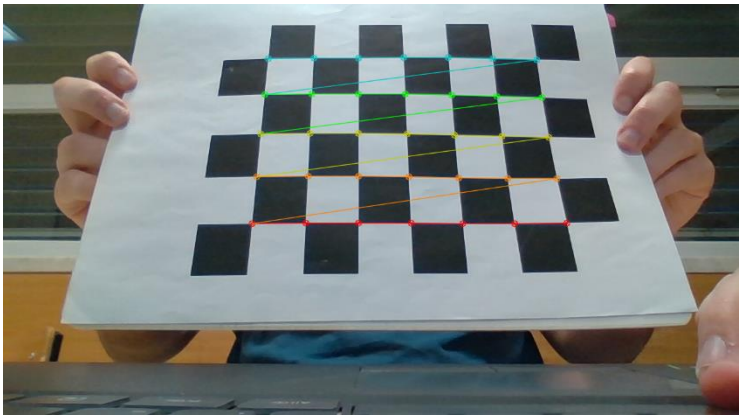

Figura 2 - Deteção dos pontos do tabuleiro

Por fim, após esta captação de pontos, podemos então obter a matriz de calibração, os coeficientes de distorção, os vetores de rotação e os vetores de translação da camara através da função “drawChessboardCorners” do OpenCV.

Resumindo este processo, a calibração da câmara é de extrema importância pois corrigir algum tipo de distorção e encontrar os parâmetros necessários para a interpretação precisa das imagens capturadas é fundamental para os próximos passos do projeto. Logo concluímos, que com esta calibração, agora o sistema de deteção dos marcadores em tempo real, irá melhorar consideravelmente a sua eficiência e qualidade.

b) Deteção e estimativa da posição da câmara com a biblioteca ArUco

A utilização dos marcadores ArUco, deve-se ao facto destes marcadores serem amplamente utilizados em várias aplicações de visão computacional, pois desempenham um papel fundamental na deteção do ambiente em que se inserem em tempo real. Essa técnica é tambem essencial em campos como no caso realidade aumentada, onde é feito tracking de objetos e onde a deteção de marcadores em tempo real desempenha um papel crucial para o bom funcionamento dessas aplicações.

Assim, passando ao processo implementado, foi utilizada a biblioteca disponibilizada pelo AruCo para utilização no projeto, em colaboração com o OpenCV. Desta forma, e de modo a explicar melhor o funcionamento geral do projeto, em seguimento da calibração da câmara, foi utilizada a mesma câmara para captar imagens em tempo real, imagens essas processadas, alteradas e por fim mostradas pela aplicação.

É então durante este ciclo de captação, alteração reprodução das imagens que é feita a deteção dos marcadores ArUco, onde inicialmente foi definido o tipo de marcador a detetar, que no caso deste projeto, foi um marcador 6x6.

Assim, para efetuar a deteção dos marcadores, todas as imagens captadas, são convertidas para Gray Scale para ser possível então, de utilizar a função detectMarkers disponibilizada pelo ArUco. Sendo, que como output recebemos, parâmetros como a posição 2D dos cantos do marcador na imagem e o id do marcador detetado.

De seguida, já com estes parâmetros detetados, foi então possível desenhar os contornos dos marcadores na imagem e o seu id como mostra a figura 3, através da função “drawDetectedMarkers” do ArUco.

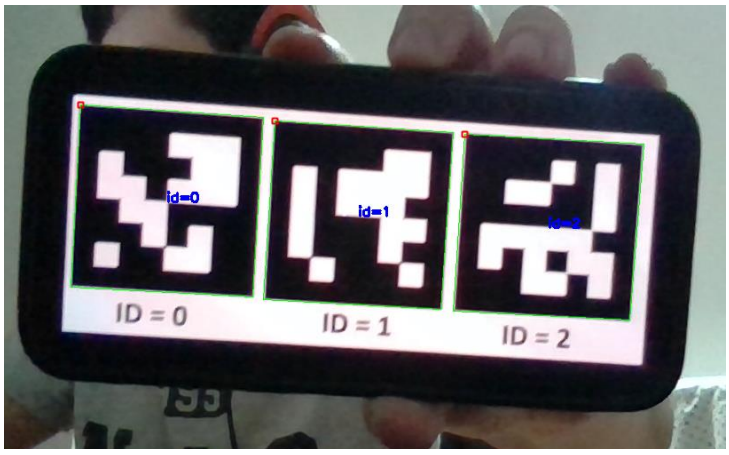

Figura 3 - Deteção dos marcadores

Foi também nesta fase, que foi estimada a posição 3D do marcador no ambiente através da função “estimatePoseSingleMarkers” do ArUco que utiliza os parâmetros calculados na fase de calibração da câmara para fazer esta estimativa mais precisa como foi mencionado anteriormente.

Concluindo assim a fase de deteção dos marcadores e estimativa dos mesmos no espaço 3D. Estava tudo encaminhado para efetuar o desenho dos objetos virtuais nas imagens.

c) Desenho dos objetos virtuais

Como última etapa deste projeto, existia o objetivo de desenhar objetos virtuais sobre os marcadores detetados em tempo real. Para tal, numa fase inicial, foram feitas algumas observações da posição estimada dos marcadores no espaço 3D através do desenho de simples linhas indicadoras dos eixos de coordenadas 3D como mostra a figura 4, de forma a verificar se a deteção e estimação destas posições estavam a ser feitas da forma correta.

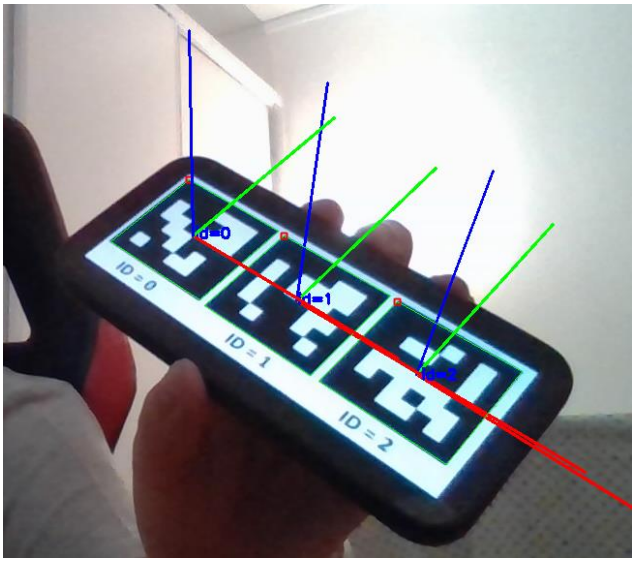

Figura 4 - Desenho dos eixos de coordenadas em cada marcador.

Assim que feitas as verificações iniciais, passou-se à fase do desenho do objeto 3D que passa por desenhar um cubo sobre o marker de id zero.

Para tal efeito, assim que a imagem é recebida e marcador de id zero é detetado, são definidos os oito pontos 3D do cubo em relação ao marcador, pontos esses definidos e guardados num array criado pela função “projectPoints” do OpenCV.

Assim que definidos os pontos relativos às arestas do cubo, são definidos com estes pontos, os planos que remetem às faces do cubo.

Por fim, para mostrar o cubo, são desenhados na imagem as arestas do cubo através do desenho de linhas entre os pontos dos vértices do cubo, assim como as suas faces. O desenho das arestas e das faces é feito em simultâneo para uma melhor precessão do objeto pois se desenharmos só as arestas não temos a noção de qual é a face de cima ou a de baixo, podendo criar uma ilusão de ótica, assim como se só desenharmos as faces, deixamos de ter a noção de onde se encontram as arestas do cubo.

De seguida, e como o objetivo do trabalho era desenhar múltiplos objetos no espaço, foi desenhada também uma pirâmide.

Para esta, foi definida que seria desenhada sobre o marcador de id um. Relativamente ao resto do processo, foi semelhante ao do desenho do cubo. Pois, foram também definidos os pontos 3D através da função “projectPoints”, só que neste caso foram definidos apenas cinco pontos, sendo os quatro da base e o vértice da pirâmide.

De seguida, foram também definidas as seis faces da pirâmide, seis pois para a base foram necessários dois planos, através da criação de planos com os pontos 3D definidos anteriormente.

E por fim foram desenhadas as arestas da pirâmide ao desenhar linhas entre os pontos da pirâmide. E também foram desenhadas as faces da pirâmide para um melhor contraste.

Assim, foi possível desenhar um cubo sobre qualquer marcador detetado com o id zero nas imagens captadas e uma pirâmide para qualquer marcador detetado com o id um nas imagens captadas. O que faz com que para qualquer outro tipo de marcador detetado com id diferente de zero ou um, não seja desenhado nenhum objeto virtual. A figura 5 mostra os resultados obtidos.

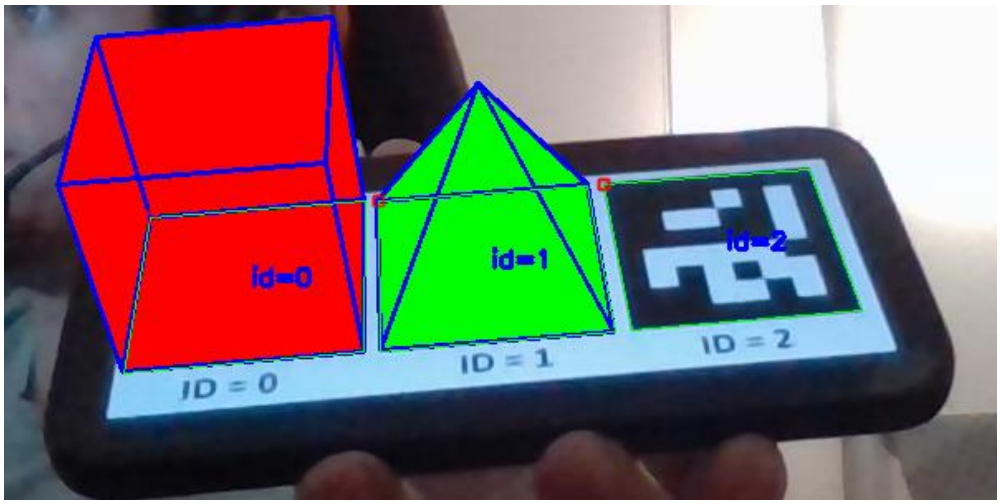

Figura 5 - Resultados obtidos

Conclusão

Em conclusão, no geral começou-se por efetuar a calibração da câmara através da utilização de imagens captadas de um padrão em xadrez, pois percebendo que esta poderia estar a distorcer as imagens, era necessário calcularmos os parâmetros que nos permitiriam efetuarmos a deteção dos marcadores em tempo real com maior eficiência e precisão.

De seguida, utilizou-se novamente a mesma câmara para captar imagens em tempo real, imagens essas onde efetuamos então a deteção dos marcadores ArUco.

Concluindo então por prever a posição 3D dos cantos dos marcadores no espaço. Definindo assim pontos nesse mesmo espaço que foram por sua vez utilizados para desenhar arestas e planos, também eles definidos, de um cubo e uma pirâmide para os marcadores de id zero e um respetivamente detetados.