DOCENTE: Rui Jesus

**Trabalho Final**

**2425SV**

|  |  |
| --- | --- |
| Aluno | Número |
| Vitor Dinis | 46327 |

*13 de julho de 2025*

Índice

[1. Introdução 1](#_Toc200338836)

[1.1. Objetivo 1](#_Toc200338837)

[1.2. Conceitos teóricos 1](#_Toc200338838)

[2. Desenvolvimento 5](#_Toc200338839)

[2.1. Calibração da câmara 5](#_Toc200338840)

[2.2. Deteção e identificação de marcas 7](#_Toc200338841)

[2.3. Registo de objeto virtual 8](#_Toc200338842)

[3. Aplicação final 13](#_Toc200338843)

[4. Conclusões 15](#_Toc200338844)

[5. Bibliografia 17](#_Toc200338845)

Índice de Figuras

[Figura 1 - Pipeline de renderização AR utilizada na aplicação [5] 1](#_Toc200338828)

[Figura 2 - Matriz dos parâmetros intrínsecos da câmara 5](#_Toc200338829)

[Figura 3 - Imagem de calibração da câmara 6](#_Toc200338830)

[Figura 4 - Deteção e identificação de marcas 7](#_Toc200338831)

[Figura 5 - Estimação da pose de uma marca 8](#_Toc200338832)

[Figura 6 - Oclusão entre objetos virtuais 10](#_Toc200338833)

[Figura 7 - Renderização de dois modelos (esq. - draw\_polygon; dir. - draw\_wireframe) 11](#_Toc200338834)

[Figura 8 - Resultado da aplicação (registo de um objeto virtual numa marca) 13](#_Toc200338835)

Índice de Tabelas

[Tabela 1 - Lista dos modelos 3D usados na aplicação 8](#_Toc200338823)

# Introdução

No presente relatório consta um resumo de todo o trabalho realizado ao longo do 2º Trabalho prático da Unidade Curricular de Visão Artificial e Realidade Mista.

# Objetivo

Neste, o principal objetivo foi desenvolver uma aplicação que implementasse realidade aumentada (em inglês, *augmented reality*, ou *AR*), que permitisse a inclusão de elementos virtuais alinhados com marcadores fiduciais reais.

Para tal, recorreu-se à linguagem de programação Python [1], assim como às bibliotecas: *OpenCV* (Open-Source Computer Vision) [2], para processamento de imagem; *NumPy* [3], para cálculos matriciais e armazenamento de dados; e *matplotlib* [4], para auxiliar na observação de resultados durante a implementação.

# Conceitos teóricos

Texto aqui

# Desenvolvimento

Texto aqui

# Calibração da câmara

Texto aqui

# Deteção e identificação de marcas

Texto aqui

# Registo de objeto virtual

Texto aqui

# Aplicação final

Texto aqui

# Conclusões

Podemos concluir que a aplicação implementada cumpre com os objetivos propostos no enunciado do trabalho. As minhas maiores dificuldades concentraram-se na parte de registo de objetos virtuais, especialmente na implementação do método de renderização e na leitura de ficheiros *obj*.

Para o futuro, seria interessante corrigir o problema da distorção dos objetos que ocorre quando estes ficam muito próximos à câmara. Além disso, gostaria de otimizar a aplicação desenvolvida. Invés de usar o OpenCV para a renderização dos modelos (apesar de ter uma boa performance, ser rápido, não é o mais indicado para esta tarefa), usar uma biblioteca de computação gráfica, própria para esta tarefa, como, por exemplo, o OpenGL [12]. Finalmente, poderia implementar oclusão entre objetos reais e objetos virtuais, bastando para isso usar uma câmara que possua deteção de profundidade e acrescentar essa informação no *pipeline* de renderização; e a deteção de marcas naturais, invés de usar as marcas ArUco.

# Bibliografia

|  |  |
| --- | --- |
| [1] | Python Software Foundation, “Python,” 2001-2025. [Online]. Available: https://www.python.org/. |
| [2] | OpenCV team, “OpenCV - Open Computer Vision Library,” 2025. [Online]. Available: https://opencv.org/. [Acedido em Feb. 2025]. |
| [3] | NumPy team, “NumPy,” [Online]. Available: https://numpy.org/. [Acedido em Mar. 2025]. |
| [4] | The Matplotlib development team, “Matplotlib,” [Online]. Available: https://matplotlib.org/. [Acedido em Mar. 2025]. |
| [5] | The Khronos® Group Inc., “OpenGL,” [Online]. Available: https://www.opengl.org/. [Acedido em 2025]. |
| [6] | P. M. Jorge, “Displays - Chapter 2,” [Online]. Available: https://2425moodle.isel.pt/pluginfile.php/1275215/mod\_resource/content/1/MEIM\_VARM\_Chapter\_2.pdf. [Acedido em 2025]. |
| [7] | P. M. Jorge, “Tracking - Chapter 3,” [Online]. Available: https://2425moodle.isel.pt/pluginfile.php/1278724/mod\_resource/content/1/MEIM\_VARM\_Chapter\_3.pdf. [Acedido em 2025]. |
| [8] | P. M. Jorge, “Computer Vision for Augmented Reality - Chapter 4,” [Online]. Available: https://2425moodle.isel.pt/pluginfile.php/1280724/mod\_resource/content/1/MEIM\_VARM\_Chapter\_4\_new.pdf. [Acedido em 2025]. |
| [9] | O. Kalachev, “ArUco markers generator!,” [Online]. Available: https://chev.me/arucogen/. [Acedido em Mai. 2025]. |
| [10] | OpenCV team, “Camera Calibration,” [Online]. Available: https://docs.opencv.org/4.x/dc/dbb/tutorial\_py\_calibration.html. [Acedido em 2025]. |
| [11] | kyle-bersani, “GitHub - opencv-examples / CalibrationChessboard / chessboard-to-print,” [Online]. Available: https://github.com/kyle-bersani/opencv-examples/blob/master/CalibrationByChessboard/chessboard-to-print.pdf. [Acedido em 2025]. |
| [12] | Sketchfab, Inc., “Sketchfab,” [Online]. Available: https://sketchfab.com. [Acedido em 2025]. |
| [13] | Scratchapixel, “The OBJ File Format,” [Online]. Available: https://www.scratchapixel.com/lessons/3d-basic-rendering/obj-file-format/obj-file-format.html. [Acedido em 2025]. |
| [14] | Blender Foundation, “Blender,” [Online]. Available: https://www.blender.org/. [Acedido em 2025]. |