

**Mestrado Integrado em Engenharia Informática e Computação**

**5º Ano**

**1º Semestre**

**Realidade Aumentada com Marcas Naturais**

**Realidade Virtual e Aumentada**

**2017/2018**

**Paulo Jorge Silva Ferreira,** [**up201305617@fe.up.pt**](mailto:up201305617@fe.up.pt)

**20 de dezembro de 2017**

Índice

[Introdução 3](#_Toc501556709)

[Programa de Preparação 3](#_Toc501556710)

[Programa de Aumento 4](#_Toc501556711)

[Resultados 6](#_Toc501556712)

[Conclusão 7](#_Toc501556713)

[Bibliografia 7](#_Toc501556714)

## Introdução

No âmbito da unidade curricular Realidade Virtual e Aumentada foi proposto o desenvolvimento de um conjunto de programas que possam ser usados para aumentar imagens de edifícios. O projeto está divido em dois programas: o de Preparação e o de Aumento.

O programa de Preparação permite a criação de uma base de dados com imagens dos edifícios que se pretende aumentar. Para isso o utilizador define uma linha delimitadora da região da imagem que pretende para aumentar outras imagens do edifício.

O programa de Aumento permite aumentar uma imagem do edifico numa pose diferente da imagem usada no programa de preparação. O programa mostra as marcas associadas ao edifício nas posições corretas, senão for possível deve notificar o utilizador.

## Programa de Preparação

No enunciado era pedido que o utilizador pudesse associar às imagens marcas, tais como linhas delimitadoras, setas ou etiquetas, mas neste programa optou-se apenas por implementar as linhas delimitadoras sob a forma de retângulos.



*Fig. 1 – Retângulo delimitador a azul*

O programa permite abrir uma imagem e selecionar um ou mais retângulos delimitadores através da função **void cv::selectROIs (const String &windowName, InputArray img, std::vector< Rect > &boundingBoxes, bool showCrosshair=true, bool fromCenter=false)**, da biblioteca OpenCV. Esta função cria uma janela com a imagem e permite ao utilizador selecionar as regiões de interesse (ROI) utilizando o rato. Utilizar a tecla ENTER para terminar a seleção da atual região e começar uma nova. Para terminara a seleção de regiões de interesse utilizar a tecla ESC.

Depois de terminar a seleção de ROI’s, a janela onde a imagem está a ser exibida é fechada e na consola o utilizador deve atribuir nomes as regiões que selecionou. Posteriormente as ROI’s são guardadas na base de dados com os nomes que o utilizador atribui.

O utilizador pode agora usar outra imagem para extrair as regiões de interesse ou então avançar para o programa de Aumento.

## Programa de Aumento

Inicialmente o programa de aumento permite ao utilizador escolher o Feature Detector (FAST, SIFT, SURF, ORB), o Descriptor Extractor (SIFT, SURF, ORB, BRIEF, FREAK) e o Descriptor Matcher (BFMatcher, FlannBasedMatcher). Por fim escolhe a imagem que pretende aumentar.

O programa começa por detetar os keypoints da imagem que pretende aumentar e extrair os descritores desses mesmos keypoints.



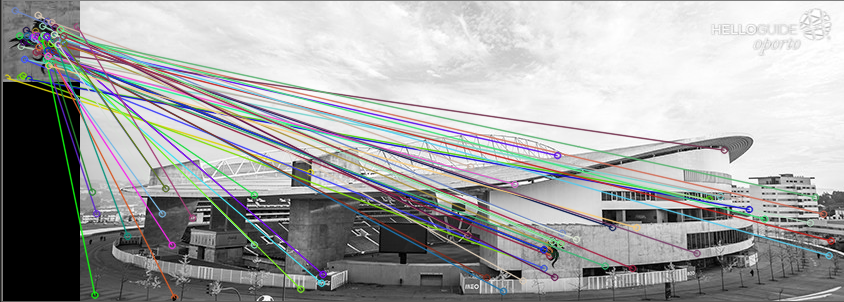
*Fig. 2 - Keypoints da imagem a aumentar*

Depois, por cada ROI guardada na base de dados, deteta os keypoint e extrai os descritores.



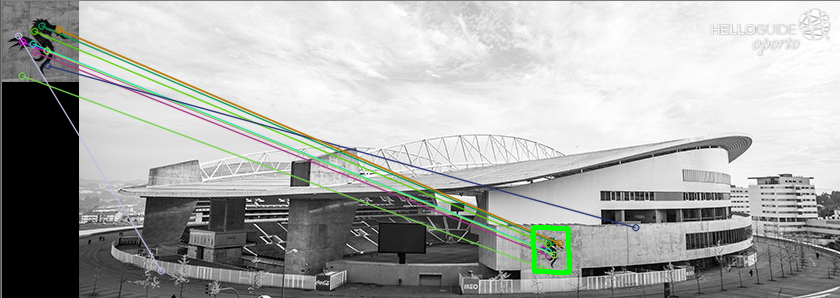
*Fig. 3 – Keypoints da imagem da base de dados*

De seguida faz corresponder todos os keypoints da imagem que pretende aumentar à região guardada na base de dados.



*Fig. 4 – Todas as correspondências*

Devido ao elevado número de correspondências tornou-se necessário reduzir esse mesmo número. De forma a reduzir as correspondências optou-se por se considerar apenas aquelas cujas distâncias entre os keypoints são menores que 3\*distância\_mínima. Se existirem correspondências suficientes é calculada a homografia utilizando o método RANSAC e os inliners (pontos que estão dentro da região de interesse na imagem a aumentar). De seguida são calculados os possíveis cantos da região de interesse na imagem a aumentar. Utilizando os cantos anteriormente calculados são desenhadas linhas entre eles, para delimitar a ROI.



*Fig. 5 – Apenas as melhores correspondências*

Por fim é desenhada na imagem escolhida para ser aumentada a região de interesse que estava guardada na base de dados, resultante do programa de preparação. Importa referir, que só são desenhadas as linhas delimitadoras se todos os inliners estiverem dentro da região delimitada pelos possíveis cantos, anteriormente calculados.



*Fig. 6 – Resultado final*

## Resultados

Foram realizados vários testes com diferentes tipos de imagens e diferentes Feature Detector, Descriptor Extractor e Descriptor Matcher.

Depois dos testes realizados chegou-se à conclusão que a combinação SIFT (Feature Detector), SIFT (Descriptor Extractor) e FLANN (Descriptor Matcher) é aquela que permite obter melhores resultados na maior parte das imagens utilizadas.

Em certas imagens só foi possível obter resultados utilizando a combinação FAST – SIFT -FLANN, uma vez que o método FAST permite obter muitos mais keypoints que o método SIFT, aumentando as hipóteses de correspondências.

Também foram obtidos resultados satisfatórios com a combinação SURF-SURF-FLANN.

Independentemente da escolha do Feature Detector e do Descriptor Extractor, não foi possível obter resultados utilizando como Descriptor Matcher o BFMatcher.

## Conclusão

Ao longo do desenvolvimento do programa de preparação surgiu um problema: como permitir ao utilizador escolher as regiões de interesse? Problema esse solucionado, após pesquisa no site da biblioteca OpenCV, com a função selectROIs (descrita anteriormente). Outros pequenos problemas foram solucionados com maior ou menor dificuldade recorrendo ao site da biblioteca OpenCV ou então ao site stackoverflow.com.

De um modo geral a solução proposta encontra-se finalizada no que aos principais objetivos diz respeito. Importa referir que a solução desenvolvida não permite outro método de seleção das ROI’s para além do retângulo, como inicialmente foi proposto através de setas ou legendas de texto.

## Bibliografia

1. <https://docs.opencv.org/2.4/doc/tutorials/features2d/feature_homography/feature_homography.html>
2. <https://docs.opencv.org/2.4/modules/features2d/doc/common_interfaces_of_descriptor_matchers.html>
3. <https://docs.opencv.org/2.4/modules/features2d/doc/common_interfaces_of_descriptor_extractors.html>
4. <https://docs.opencv.org/2.4/modules/features2d/doc/common_interfaces_of_feature_detectors.html>
5. <https://docs.opencv.org/3.3.1/d7/dfc/group__highgui.html#ga0f11fad74a6432b8055fb21621a0f893>
6. <https://stackoverflow.com/>