

Realidade Aumentada com Marcas Naturais

Objetivos

O objetivo deste trabalho prático é desenvolver um conjunto de programas que possam ser usados para "aumentar" imagens de fachadas de edifícios. Ao realizá-lo, os estudantes terão oportunidade de aplicar os conhecimentos obtidos durante o curso, relativamente ao uso de técnicas de visão por computador em aplicações de realidade aumentada, nomeadamente para reconhecer marcas naturais.

Especificação

Devem ser desenvolvidos dois programas, usando a biblioteca OpenCV: um programa de preparação do sistema e um programa de "aumento" de imagens adquiridas.

Programa de preparação

Este programa deve permitir criar uma "base de dados" com imagens da fachada do edifício que se pretende "aumentar", sobre as quais devem ser assinalados pontos, linhas ou etiquetas que serão usados para "aumentar" outras imagens da mesma fachada.

O programa deve ter as seguintes funcionalidades:

- Obter uma ou mais imagens da fachada de um edifício (nota: embora uma imagem devesse, em princípio ser suficiente, a utilização de mais do que uma imagem, em diferentes poses, pode facilitar o emparelhamento com a imagem obtida na fase de "aumento"). A fachada deve ser, em grande parte, aproximadamente plana. Sugere-se que uma das imagens seja obtida numa pose aproximadamente frontal.
- Extrair pontos característicos das imagens adquiridas e selecionar, interativamente, um subconjunto de pontos da zona aproximadamente plana da fachada.
- Permitir que o utilizador associe, interativamente, marcas às imagens obtidas (ex: a linha delimitadora da entrada do edifício, uma seta apontando para a entrada ou para uma estátua existente na fachada, ...) ou etiquetas (ex: um texto descrevendo um dos elementos da fachada), como ilustrado na figura 1.

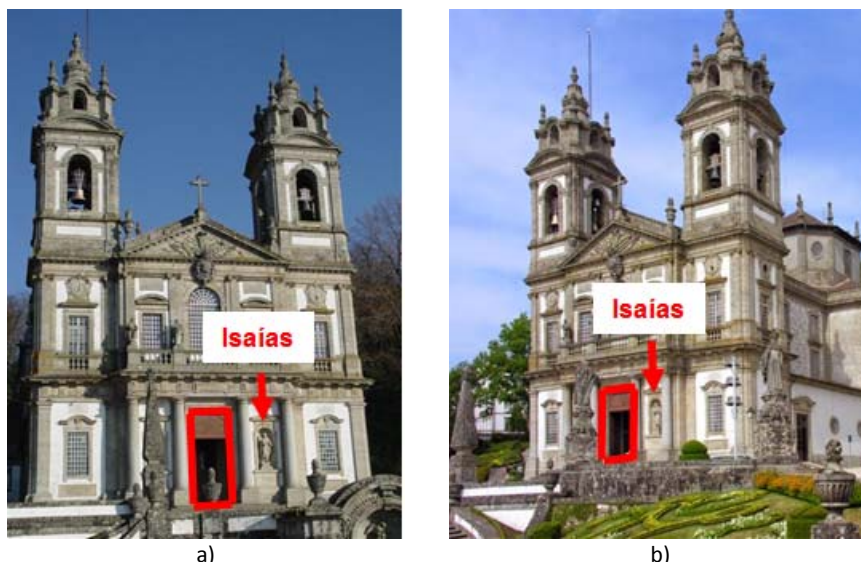


Figura 1 – a) Imagem da "base de dados"; a posição da porta e da estátua foram marcadas interativamente pelo utilizador, usando o programa de preparação.
b) Imagem "aumentada", obtida numa outra pose.

Programa de "aumento"

Este programa deve permitir "aumentar" uma imagem da fachada escolhida; deve ter as seguintes funcionalidades:

- Obter uma imagem, numa pose diferente das imagens da "base de dados" e/ou com uma ampliação diferente.
- Mostrar as marcas e/ou etiquetas associadas à fachada em causa, nas posições corretas; se não for possível estabelecer um número razoável de correspondências entre os pontos característicos da imagem adquirida e de pelo menos uma das imagens da "base de dados", informar o utilizador.

Notas sobre o desenvolvimento, entrega e avaliação

- O tipo de imagens a processar não está restringido a fachadas de monumentos, como ilustrado na especificação.
- Os programas devem ter dois modos de funcionamento; um modo normal, acima descrito, e um modo de teste, em que os resultados dos passos intermédios podem ser visualizados, sendo apresentada na consola uma pequena descrição textual do passo realizado.
- Deve ser elaborado um pequeno relatório (max. 4 páginas) que deve incluir: especificações adicionais (se necessário); a descrição dos algoritmos utilizados; comentários relevantes sobre a eficácia dos métodos utilizados, descrevendo os principais problemas encontrados e as soluções propostas; o estado da solução desenvolvida e o grau de cumprimento dos objetivos.
- O código dos programas, devidamente comentado, deve ser apresentado em anexo ao relatório.
- Os resultados deste trabalho prático serão avaliados de acordo com os seguintes parâmetros:
 - programa de preparação – 35%
 - programa de "aumento" – 40%
 - estrutura dos programas e comentários – 10%
 - relatório – 15%
- O código (fonte e executável), ficheiros de dados e relatório devem ser comprimidos num ficheiro "zip" e submetidos no "site" de RVAU, no Moodle da UP, até 2017/Dez/22, às 23:55.

Referências

- Notas das aulas da unidade curricular de "Realidade Virtual e Aumentada", Moodle de Universidade do Porto
- "Feature Matching", https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_matcher/py_matcher.html, 2017-11-21
- "Feature Matching + Homography to find Objects", https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_feature_homography/py_feature_homography.html, 2017-11-21