



Disciplina Introdução à Computação Visual	Semestre 2017/01
Professores Erickson R. Nascimento e William Robson Schwartz	

Entrega: 31/05/2017 até às 23h55 (via moodle)

Trabalho Prático 2: Realidade Aumentada

O objetivo deste trabalho é detectar e localizar alvos nos quadros (*frames*) de um vídeo e inserir na cena um objeto tridimensional sobre cada alvo detectado. Para este trabalho o objeto tridimensional será um cubo e um modelo 3D (o Pikachu). A Figura 1 mostra um exemplo de como um frame do vídeo final deve ficar após o processamento.

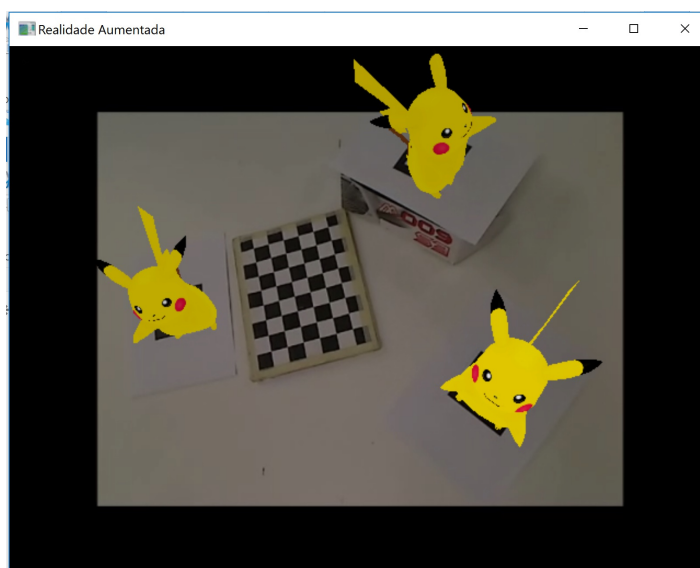


Figura 1: Exemplo do resultado de um frame processado após a inclusão de um objeto tridimensional acima de cada um dos alvos detectados.

O que deve ser feito

Aviso: O trabalho deverá ser implementado em um notebook Python.

O primeiro passo será calibrar a câmera utilizando o vídeo com o tabuleiro de xadrez como alvo de calibração. Nesta etapa será realizada a estimação dos parâmetros intrínsecos da câmera. Para fazer a calibração, o aluno poderá utilizar o toolbox de Jean-Yves Bouguet para Matlab ¹ ou o código disponibilizado na OpenCV ². A Figura 2 (b) mostra um frame do vídeo com alvo de calibração (o tabuleiro de xadrez).

Caso o aluno escolha usar o código disponibilizado na OpenCV, sugiro que ele use a seguinte linha de comando:

```
./calibration -w 6 -h 8 -s 0.03 -n 8 -V entrada.avi -o intrinsic.yml -zt
```

¹http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example.html

²http://docs.opencv.org/trunk/doc/tutorials/calib3d/camera_calibration/camera_calibration.html#

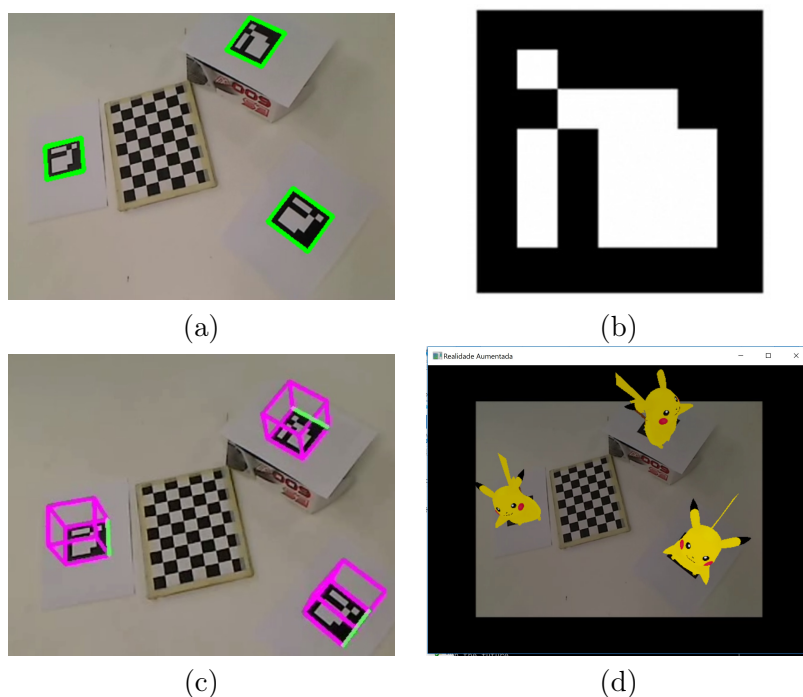


Figura 2: (a) Frame com todos os alvos detectados; (b) Imagem do alvo no qual o objeto 3D deve ser incluído sobre sua superfície; (c) Exemplo de um Frame do vídeo com o objeto 3D (um cubo) foi incluído. Note que o alvo de calibração também aparece junto com três alvos; (d) Exemplo de um Frame do vídeo com o objeto Pikachu.

Esta linha informa que o tabuleiro tem 6×8 quadrados, que cada quadrado tem 3 centímetros e que devem ser usados 8 frames para a calibração. Os parâmetros serão salvos no arquivo `intrinsic.yml`.

Depois de obtida a matriz de parâmetros intrínsecos, o aluno deverá implementar um método de detecção e localização da posição e orientação do alvo nos frames do vídeo. Pode-se utilizar funções do OpenCV para ler vídeos, imagens, calcular a matriz de homografia, recortar e colar, binarizar, detectar bordas (`Canny()`) e detectar quinas (`cornerHarris()`). A Figura 2 (a) mostra um frame do vídeo da cena com os alvos localizados.

O terceiro passo será codificar uma função para obter a pose da câmera (parâmetros intrínsecos). Para isso o aluno deverá utilizar a função `solvePnP()` da OpenCV.

Finalmente um cubo (objeto tridimensional) deverá ser incluído na cena na posição o orientado de cada alvo detectado. **A renderização deverá ser feita utilizando somente as funções da OpenGL (o aluno não deve usar funções da OpenCV para renderizar objetos na cena).** O objeto deve ser inserido sobre a superfície do objeto mostrado na Figura 1. O objeto 3D deve ser incluído em todos os frames do vídeo.

Para carregar o modelo 3d do Pikachu, o código fornecido precisará que seja instalada a biblioteca PyGame.

O que deve ser entregue

Deverão ser entregue um zip (ele não devem exceder o tamanho de 50Mb) com:

- O notebook Python. Cada uma das funções implementadas deve possuir um cabeçalho descrevendo seu objetivo, cada um dos parâmetros de entrada e suas saídas.
- O notebook deve conter a descrição das funções e cada um dos passos para a geração do vídeo final. A matriz de parâmetros intrínsecos estimada deverá ser mostrado no notebook. No caso do aluno ter utilizado o código da OpenCV para calibrar a câmera esse deve ser parte do código-fonte do TP e também deverá estar no notebook.

- Um arquivo AVI com o vídeo com todos os quadros com a detecção e localização do objeto e o cubo inserido na posição dos alvos detectados.

Não serão aceitos trabalhos atrasados. O Moodle irá fechar a submissão após o prazo de entrega..

O que será disponibilizado

- Um vídeo no qual aparece o alvo de calibração e os alvos para localizar as posições e orientações onde devem ser inseridos os objetos 3D: entrada.avi. As dimensões dos quadrados no tabuleiro de xadrez são $3cm \times 3cm$. Notem que esse vídeo deve ser utilizar para primeiro calibrar a câmera e depois para incluir o objeto 3D.
- Imagem jpg do alvo.
- Código em Python para carregar o modelo 3D do Pikachu.
- Modelo 3D com textura do Pikachu no formato OBJ.