

Realidade Aumentada

Visão Computacional - 2019/2

Andrew Kurauchi, Fábio Ayres, Igor Montagner, Raul Ikeda

A fusão entre o mundo virtual e o mundo real é designada realidade mista. Denominamos Realidade Aumentada (RA) as aplicações que adicionam experiências sensoriais (visão, audição, tato, etc.) do mundo virtual ao mundo real.



Figure 1: O jogo Pokémon GO possui um modo de jogo que utiliza realidade aumentada, no qual os pokémons são mostrados no mundo real. Imagem: <https://www.flickr.com/photos/iphonedigital/28286906571>

Diversos meios são utilizados para aumentar a realidade com elementos visuais virtuais: dispositivos móveis (smartphones e tablets), óculos de realidade aumentada, projetores, etc. Neste projeto vamos desenvolver uma aplicação simples de realidade aumentada utilizando uma câmera.

Detecção de marcadores visuais

Marcadores visuais são bastante utilizados em realidade mista para facilitar o processo de estimação de pose, ou seja, a estimação da posição e orientação de um objeto com relação à câmera.

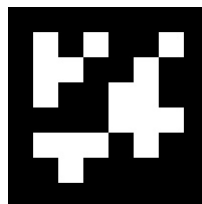


Figure 2: Exemplo de marcador visual. Imagem: https://docs.opencv.org/3.1.0/d5/dae/tutorial_aruco_detection.html

Neste projeto utilizaremos marcadores para indicar uma superfície (ex. uma mesa ou parede) onde serão renderizados (desenhados) objetos virtuais. O OpenCV já possui **funções para localizar marcadores**.

Definição do projeto

O projeto consiste em desenvolver uma aplicação simples de realidade aumentada em que marcadores são substituídos por objetos 2D ou 3D. Nosso projeto rodará em Android e usará a câmera do dispositivo para

capturar as imagens e a tela para posicionar as imagens na cena. Nosso objetivo final é usar o Vusix Blade do lab de RV para projetar as imagens na frente dos olhos do usuário.



Figure 3: Vusix Blade

Objetivos de aprendizado

Os principais objetivos de aprendizado avaliados neste projeto são:

- Compreender e aplicar o conceito de homografia para gerar visões de diferentes pontos de vista de uma mesma superfície.
- Ser capaz de calibrar uma câmera para obter os seus parâmetros codificados em uma matriz de parâmetros intrínsecos.
- Compreender o modelo de câmera pinhole e aplicá-lo na estimação de pose de objetos tridimensionais.

Restrições tecnológicas

O projeto é bastante livre com relação às bibliotecas a serem utilizadas, contanto que não interfiram na avaliação dos objetivos de aprendizado. Em especial, sintam-se livres para utilizar a biblioteca que preferirem para renderizar a cena/objeto 3D.

Na dúvida pergunte ao professor.

Entrega

O projeto deve ser entregue até o dia 29/11 às 23h59 através do Blackboard. Além da entrega do projeto, devem ser enviadas também as 4 atividades desenvolvidas em sala de aula.

Avaliação

O projeto receberá um conceito entre **I** e **A** dependendo da qualidade e quantidade de funcionalidades implementadas.

Importante 1: entregar uma documentação explicando o funcionamento do software é obrigatório e não fazê-lo resulta em conceito **D**. Seu texto deverá explicar como distribuir os marcadores visuais e como executar o programa.

- Conceito **D**:
 - o software tem algum erro conceitual grave **OU**;
 - o software não funciona para uso básico **OU**;
 - o software não está acompanhado de documentação explicando seu funcionamento **OU**;
 - o aluno não entregou as atividades desenvolvidas em aula.
- Conceito **C**:
 - o software é capaz de detectar marcadores na imagem da câmera;
 - o software é capaz de renderizar um objeto plano sobre uma superfície definida pelos marcadores considerando a sua posição relativa na imagem (ver Figura 4 para um exemplo);
 - a renderização é feita **SOMENTE** sobre a imagem da câmera, no monitor;
 - o software **NÃO** roda em Android
- Conceito **A**:
 - o software cumpre todos os requisitos do conceito C;
 - o software renderiza a imagem na tela do celular.
- Conceito **A+**:
 - o software cumpre todos os requisitos do conceito C;
 - o software renderiza a imagem na tela do Vusix
 - o software inclui uma etapa de calibração de câmera (que pode ser feita offline)
- Extra **+(1,0)**:
 - o software é capaz de renderizar um **objeto 3D** sobre a superfície definida pelos marcadores;
 - o software usa uma câmera calibrada para estimar a pose dos marcadores;
 - o software inclui uma etapa de calibração de câmera (que pode ser feita offline)

Checklist

Definições

- **Imagem de referência:** imagem com os marcadores, que você imprimiu;
- **Imagem da superfície:** imagem do objeto plano a ser mapeado na superfície (ex: foto do Insper);

Tarefas

Montei o checklist a seguir para te ajudar a guiar o desenvolvimento do seu projeto.

1. detectar marcadores na imagem da câmera;
2. obter coordenadas dos marcadores na imagem de referência (esse passo é realizado uma única vez);
3. construir lista de pares de pontos (coordenadas dos marcadores na imagem da câmera e na imagem de referência);
4. calcular a homografia;
5. aplicar a homografia na imagem da superfície e a desenhar sobre a imagem da câmera;
6. calibrar a câmera (esse passo é realizado uma única vez);
7. usar a matriz de câmera para **estimar a pose dos marcadores**;
8. renderizar um objeto 3D utilizando as informações da pose dos marcadores e os parâmetros da câmera;

Após implementar os itens de 1 a 5 você deve ter um resultado semelhante ao da Figura 4.

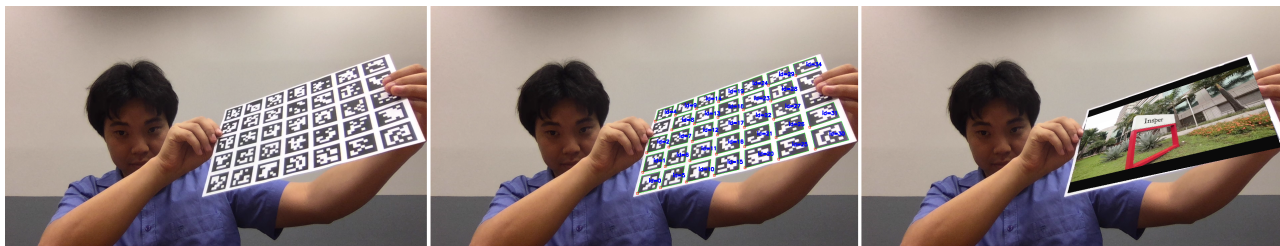


Figure 4: Os marcadores são detectados e a imagem da superfície é mapeada sobre o plano que os contém.