Computer Vision Based Augmented Reality

Fangzhou Xia, Yingnan Cui

Department of Mechanical Engineering
Massachusetts Institute of Technology

Background Introduction

- Billboard 2D graph overlay for propaganda
- AR label pointers for industrial training
- IMU based AR for gaming
- April tag fiducial based AR

Billboard overlay

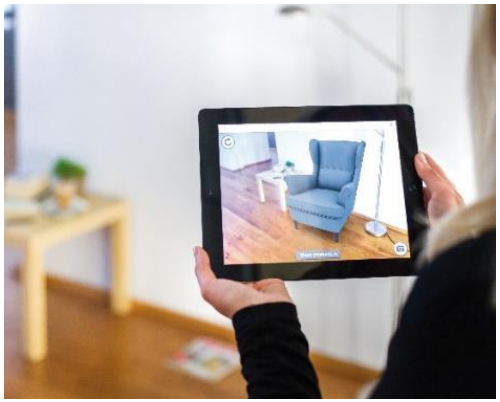
Engine with AR label for training

Pokemon Go

April tag based AR

Project Scope Description

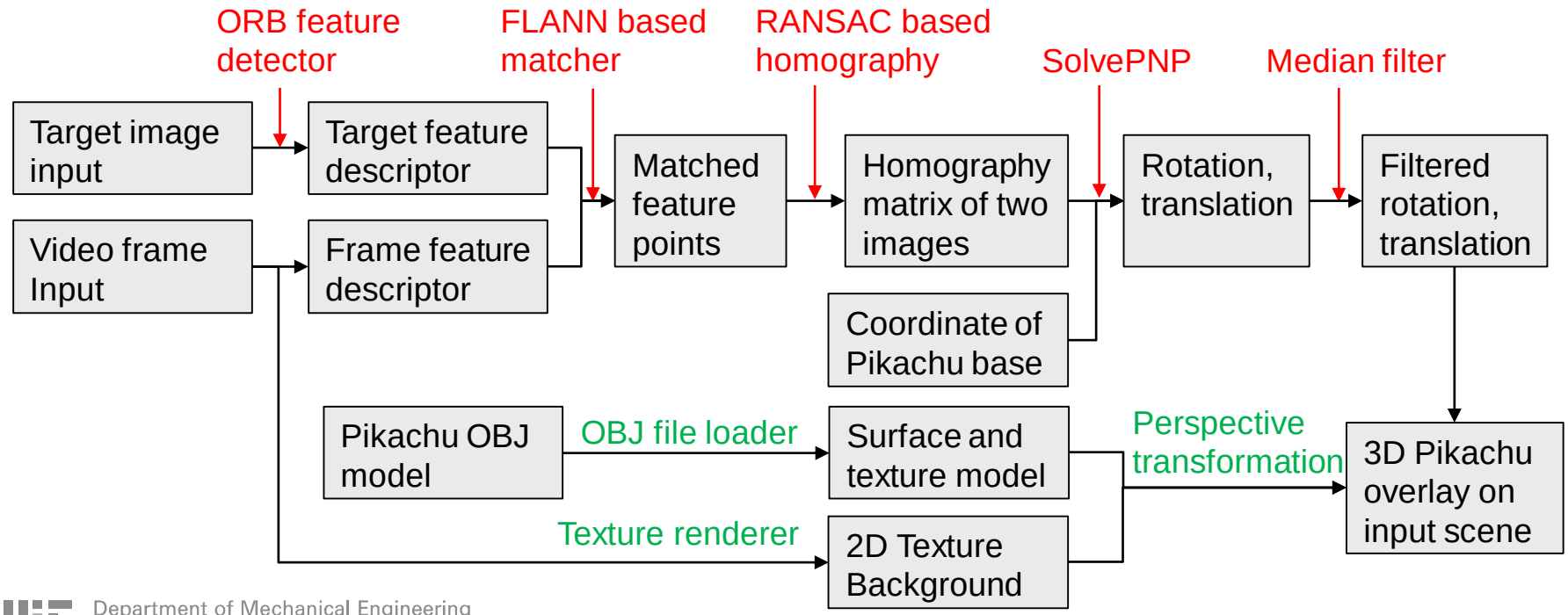
- Project Scope
  - Computer vision based AR
  - 2D rich texture surface tracking
  - 3D object model overlay
- Technical Aspects
  - Object detection
  - Pose estimation
  - 3D Pikachu model overlay
- Implementation Platform
  - Full implementation with Python on PC for demo
  - Computer vision only Java implementation on Android for feasibility



3D object overlay on 2D textured surface

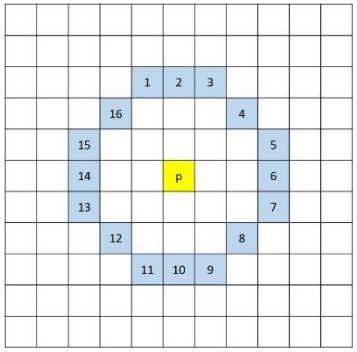
Implementation Details

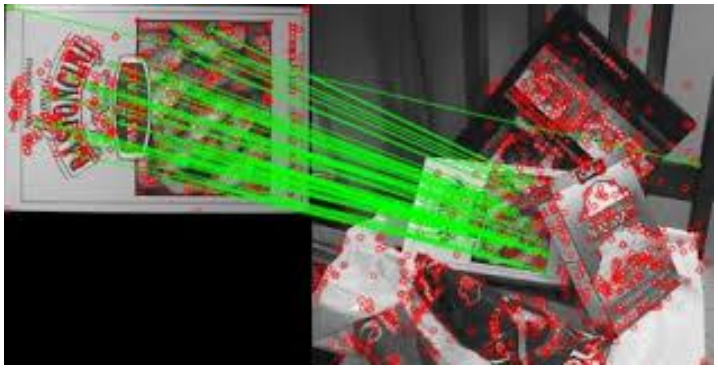
- OpenCV for object tracking
- OpenGL for 3D model graphics rendering



Feature Detection and Matching

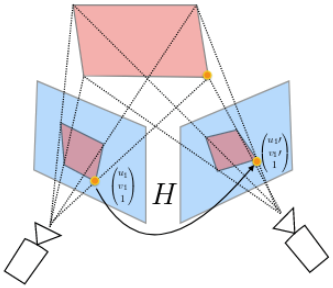
- Various detection methods available, SIFT, SURF, ORB
- ORB is used as it is fast and open source
- ORB stands for Oriented FAST and Rotated BRIEF
- FAST: Features from Accelerated Segment Test feature detector
- BRIEF: Binary Robust Independent Elementary Feature descriptor
- FLANN: Fast Library for Approximated Nearest Neighbor matching





Homography Matrix Estimation

- Homography matrix with 8 unknown parameters (scale ambiguity)
- Identify plane to plane transformation with at least 4 points
- Regression based method for optimal result
- RANSAC: RANdom Sample Consensus estimation
- Utilize RANSAC to remove outliers

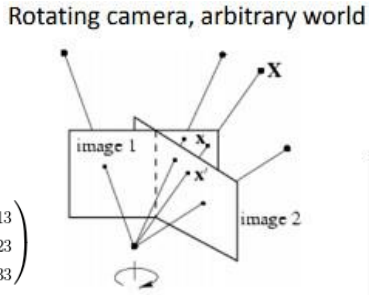


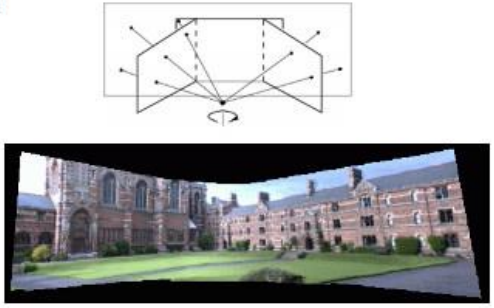
$$H_{4point} = \begin{pmatrix} \Delta u_1 & \Delta v_1 \\ \Delta u_2 & \Delta v_2 \\ \Delta u_3 & \Delta v_3 \\ \Delta u_4 & \Delta v_4 \end{pmatrix}$$

1-to-1 mapping

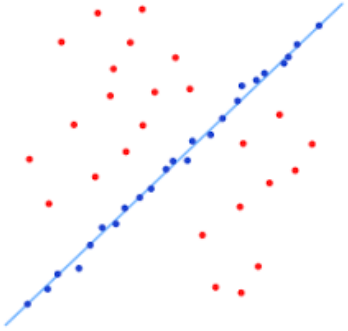
$$H_{matrix} = \begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{pmatrix}$$

Homography matrix





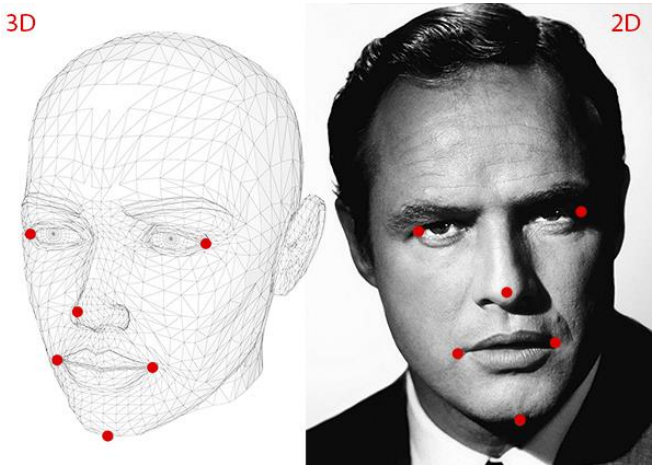
Application to panorama stitching



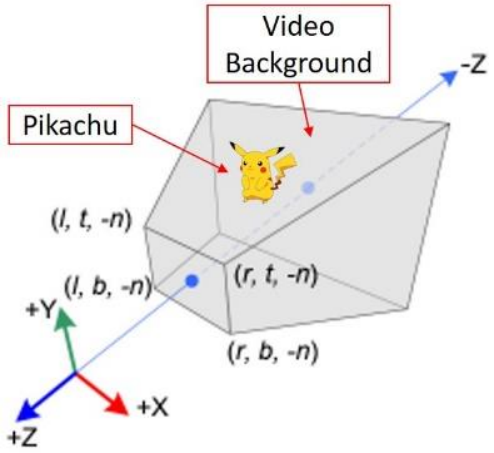
RANSAC illustration

Pose Estimation and Rendering

- Prospective N Points (PNP) problem for pose estimation
- Need 2D image points and 3D model correspondence
- Overlay 3D image



3D model and 2D image correspondence



Scene rendering in OpenGL

Project Demo

- OpenCV window for detection with 2D plane homography overlay
  - Python PC and Java Android Galaxy phone
- OpenGL window for Pikachu 3D model overlay with median filter
  - Python PC complete demo

Pikachu overlay with filter

OpenCV detection (no filter)

Android OpenCV Detection

Thank You!