Universitat de les Illes Balears

escola politècnica superior

Treball Final de Carrera

ENGINYERIA INFORMÀTICA

Ponster, tu app de realidad aumentada

Álvaro Medina Ballester

Tutor
Ramón Mas Sansó

Escola Politècnica Superior
Universitat de les Illes Balears
Palma, September 6, 2014

CONTENTS

CHAPTER 1

ABSTRACT

CHAPTER 2

THANKS

CHAPTER 3

INTRO

Augmented reality has become a very popular topic in the last five years. With the introduction of mobile smartphones, developers started to have the chance to develop applications using the powerful CPUs and GPUs of those devices.

CHAPTER 4

STATE OF THE ART

The techinique of mixing real world elements with virtual elements displayed on the screen of a device is what we call augmented reality. In the field of augmented reality, a lot of things have happened during the last years. The progress in the fields of computer vision and image processing have led to several new techniques of detection and tracking. This, combined with the increasing availability of powerful mobile devices, has enabled developers to build a plethora of high quality AR-based applications. Nowadays, modern mobile devices use integrated cameras, motion sensors and proximity sensors to make these AR-based experiences.

Augmented reality in mobile devices is slightly different from what can be seen in desktop environments. Although every year we have more powerful mobile smartphones (citation needed), processing and drawing into the device's screen is still an expensive operation in terms of computational cost. This is one of the reasons why cost-efficient computer vision algorithms and techniques have emerged in the past years.

Most of the augmented reality apps follow this behaviour:

- Get the input from the camera or a video.
- Search for an object of interest.
- Introduce our object into the scene, considering the camera or the input position.

In this chapter we are going to describe which are the different techniques that enables us to do them.

4.1 Object recognition

In order to provide an augmented reality experience, we have to know first which is the real world element that we are going to use as a reference to mix the real world input with our virtual elements. This reference can be from an image from the smartphone

camera to the user location. Ponster searches a particular image inside the camera input in order to draw the poster image. In computer vision, this is usually referred as object tracking.

In computer vision there are a lot of object recognition techniques. In the development of Ponster, two of these techniques have been tested: template matching and feature-based detection.

4.1.1 Template matching

Template matching consists of finding areas of an image that are similar to a provided template image. We have to provide a template image (the image that we want to look for) and compare it with the source image (the image in which we want to search). Template matching is also called area-based approach.

OpenCV provides a method to perform template matching with several methods, such as `SQDIFF` or `CCORR`. With the latest, `CCORR`, we use a correlation formula to check if the template is inside the image. Instead of applying a yes/no approximation, we can bring a positive match with a certain threshold.

Performing a template matching operation using OpenCV on mobile devices is fast enough to deliver a smooth 30fps-like detection. However, match template does not take account of scale, rotation and perspective invariance by itself. There are several approaches to bring invariation to match template. For instance, image pyramids are used to make match template scale and rotation invariant, but it is not part of the OpenCV match template function, although it provides some methods to implement image pyramids.

Match template has been tested during the development of Ponster. Also, a basic image pyramid system has been developed for scale-invariance, but match template has been discarded in favor of feature detection algorithms because rotation and scale invariance and perspective warp are required features.

4.1.2 Feature detection

Feature-based approach consists of detecting keypoints[1] in the source image and in the input image, provided by the camera in this case. Then, we have to match these two sets of keypoints in order to know if the source image is present. Source images with enough keypoints are easier to detect than more uniform images. This is why it's better to select a good source image with many features and good contrast. Usually, feature detection algorithms use corners, blobs or T-junctions [2]in the image as keypoints candidates.

As we've said before, in order to deliver a good augmented reality experience, we need to make our detection algorithm scale, rotation and perspective invariant. Feature detection techniques can be scaled invariant by extracting features that are invariant to scale, such as feature vectors computed from interest point neighbourhoods. For rotation invariance, algorithms can estimate the orientation of the keypoint.

There are plenty of feature-detection based algorithms, many of them based on Scale-Invariant Feature Transform, or SIFT. Cost efficiency is one of the most important features of these algorithms, as every new technique introduced tries to mantain robustness and reducing computation time. Robustness it's also very important, but less

robust algorithms are also been developed in favour of reducing computation time. One good example is FAST[orb paper] keypoint detector, which is not rotation invariant.

Speeded-Up Robust Features - SURF

Speeded-Up Robuts Features, also know as SURF, is a variant of Scale-Invariant Feature Transform, SIFT. SURF is actually

Features from Accelerated Segment Test - FAST

Oriented FAST and Rotated BRIEF - ORB

4.1.3 Descriptor extraction

4.1.4 Descriptor matching

Brute-Force Matcher

Fast Library for Approximate Nearest Neighbors - FLANN

4.2 Tracking

CHAPTER 5

DEVELOPMENT

CHAPTER 6

CONCLUSIONS