

Augmenting Information of Captured Image into the Real World

Shweta Kale¹, Manasi Ranadive², Athira Nair³, Tejaswee Newale⁴, Dr. Pravin Futane⁵

Computer Department, Savitribai Phule Pune University Pune, Maharashtra, India

Professor, Computer Department, Savitribai Phule Pune University Pune, Maharashtra, India

Abstract: The system consist of android application and sixth sense technology components in order to meet the objective. The input to the database would be images and corresponding information. The image is captured by the camera of smart-phone having our android application. Face will be detected from the captured image and characteristics will be extracted by the Local Binary Pattern algorithm and matched with the database images. If the characteristics are matched then information of corresponding image will be passed to the projector. Sixth sense is wearable gesture interface that augments the physical world around us with digital information and lets us use natural hand gesture to interact with that information. Image will be recognized by the android application and related information would be displayed in real world where the hand gestures are locating. Hardware devices used in sixth sense technology are Camera, colored marker, mobile component, mirror and projector. Camera will capture the object in view and will track the users hand gestures. It will send the data to the smart phone. It acts as a digital eye, connecting user to the world of digital information. The mirror will be used to reflect the scene of the projector into real world. The smart phone will process the input data (eg. hand gestures, captured image recognition and matching). A projector displays data sent from smart phone on any surface directed by gestures object, wall etc. The projector faces downward and project output on the mirror so that we can adjust the focus and project on desired surface

Keywords: Image display, Object-oriented Programming , Visual Programming ,Interactive environments, augmented reality

1. Introduction

Project Idea is by only taking the photo of student in real world the authority like Principle Sir will have information regarding particular student like his name, class, academic progress, current timetable, attendance etc in real world. Traditional working system includes lots of paperwork and it is very tedious. Our project idea which is inspired by sixth sense technology will provide instant results in real world on any surface. This project will save time. Desired result will be at tip of the fingers and will skip multiple steps of traditional systems hence will give faster results. The primary objective of this project is to augment reality. The Sixth sense technology blends physical reality and digital world in very creative, intuitive and compelling ways.

The model should be able to do the following: On server side database there will be images and corresponding information of student. When image is clicked, features will be extracted and would be compared with the database. If match is found then the information of student will sent to the projector. From projector using mirror focus will be adjusted and information of student would be in real world.

2. Literature Survey

A. Face Detection : Haar Cascade Classifier

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images.[3]

Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, haar features shown in below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under white rectangle from sum of pixels under black rectangle[3]

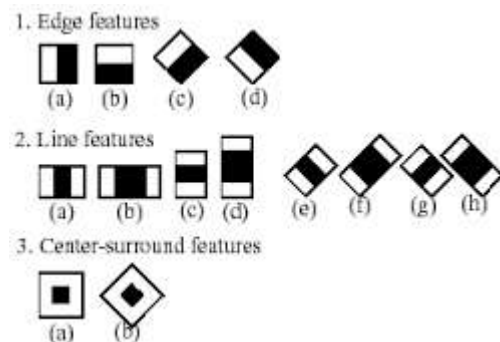


Figure 6.1: Techniques of face detection

Now all possible sizes and locations of each kernel is used to calculate plenty of features. (Just imagine how much computation it needs? Even a 24x24 window results over 160000 features). For each feature calculation, we need to find sum of pixels under white and black rectangles. To solve this, they introduced the integral images. It simplifies calculation of sum of pixels, how large may be the number of pixels, to an operation involving just four pixels. Nice, isn't it? It makes things super-fast.[3]

But among all these features we calculated, most of them are irrelevant. For example, consider the image below. Top row shows two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second

feature selected relies on the property that the eyes are darker than the bridge of the nose. But the same windows applying on cheeks or any other place is irrelevant. So how do we select the best features out of 160000+ features? It is achieved by Adaboost.



Figure 6.2: HAAR Features

For this, we apply each and every feature on all the training images. For each feature, it finds the best threshold which will classify the faces to positive and negative. But obviously, there will be errors or misclassifications. We select the features with minimum error rate, which means they are the features that best classifies the face and non-face images. (The process is not as simple as this. Each image is given an equal weight in the beginning. After each classification, weights of misclassified images are increased. Then again same process is done. New error rates are calculated. Also new weights. The process is continued until required accuracy or error rate is achieved or required number of features are found).

Final classifier is a weighted sum of these weak classifiers. It is called weak because it alone can't classify the image, but together with others forms a strong classifier. The paper says even 200 features provide detection with 95% accuracy. Their final setup had around 6000 features. (Imagine a reduction from 160000+ features to 6000 features. That is a big gain).

So now you take an image. Take each 24x24 window. Apply 6000 features to it. Check if it is face or not. Wow.. Wow.. Isn't it a little inefficient and time consuming? Yes, it is. Authors have a good solution for that.

In an image, most of the image region is non-face region. So it is a better idea to have a simple method to check if a window is not a face region. If it is not, discard it in a single shot. Don't process it again. Instead focus on region where there can be a face. This way, we can find more time to check a possible face region.

For this they introduced the concept of Cascade of Classifiers. Instead of applying all the 6000 features on a window, group the features into different stages of classifiers and apply one-by-one. (Normally first few stages will contain very less number of features). If a window fails the first stage, discard it. We don't consider remaining features on it. If it passes, apply the second stage of features

and continue the process. The window which passes all stages is a face region. How is the plan !!!

Authors' detector had 6000+ features with 38 stages with 1, 10, 25, 25 and 50 features in first five stages. (Two features in the above image is actually obtained as the best two features from Adaboost). According to authors, on an average, 10 features out of 6000+ are evaluated per sub-window.[3]

So this is a simple intuitive explanation of how Viola-Jones face detection works. Read paper for more details or check out the references in Additional Resources section.

Haar-cascade Detection in OpenCV

OpenCV comes with a trainer as well as detector. If you want to train your own classifier for any object like car, planes etc. you can use OpenCV to create one. Its full details are given here: Cascade Classifier Training.

Here we will deal with detection. OpenCV already contains many pre-trained classifiers for face, eyes, smile etc. Those XML files are stored in `opencv/data/haarcascades/` folder. Let's create face and eye detector with OpenCV.

3. Code Development Environment

Open-CV and MATLAB both can be used for code development of augmented reality. Study of both the utilities individually and comparison is done in order to find the most suitable for building augmented reality browser application.

Open-CV

Open-CV is the Open Source Computer Vision library. It is a machine learning software library. Released under BSD license hence it is free. Updation release in every 3 to 4 month. The library has more than 2500 optimized algorithms and 10 times more functions to support these algorithms. Open-CV is specially designed for achieving high efficiency and with strong focus to real time applications. Open-CV is written in C++.

Open-CV supports Windows, Linux, iOS, MacOS and Android. It has C, C++, Java, Python interfaces. It can take advantage of multi-core processing hence, parallel computing, multithreading is possible. A full featured CUDA and OpenCL libraries are being actively developed.

The algorithms in OpenCV are used for identifying objects, recognize faces, extract 3D models of objects, track camera movements, follow eye movements, recognize scenery and establish markers to analyze and establish markers to overlay it with augmented reality.

Android Studio

Android Studio is Android's official IDE. It is purpose built for Android to accelerate your development and help you build the highest-quality apps for every Android device. It offer tools custom-tailored for Android developers, including rich code editing, debugging, testing, and profiling tools.

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Instant Run to push changes to your running app without building a new APK
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems
- C++ and NDK support
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine

Android Studio uses Gradle as the foundation of the build system, with more Android-specific capabilities provided by the Android plugin for Gradle. This build system runs as an integrated tool from the Android Studio menu, and independently from the command line. You can use the features of the build system to do the following:

- Customize, configure, and extend the build process.
- Create multiple APKs for your app, with different features using the same project and modules.
- Reuse code and resources across source sets.

By employing the flexibility of Gradle, you can achieve all of this without modifying your app's core source files. Android Studio build files are named `build.gradle`. They are plain text files that use Groovy syntax to configure the build with elements provided by the Android plugin for Gradle. Each project has one top-level build file for the entire project and separate module-level build files for each module. When you import an existing project, Android Studio automatically generates the necessary build files.

SQLite

It is a relational database management system contained in programming library. In contrast to many other database management systems, it is not a client-server database engine. Rather, it is embedded into the end program. It is ACID-compliant and implements most of the SQL standard, using a dynamically and weakly typed SQL syntax that does not guarantee the domain integrity.

It is a popular choice as embedded database software for local/client storage in application software such as web browsers. It is arguably the most widely deployed database engine, as it is used today by several widespread browsers, operating systems, and embedded systems (such as mobile phones), among others. It has bindings to many programming languages.

Prototyping

The aim of this project is to augment the corresponding information of student. The overall project is divided into two parts first is android application which will run on smartphone and second the augmentation of output of screen in real world.

- The most important task of first part is to build database which will be having fields such as image, name and other information, extracted features to make comparison etc.
- About second part we can consider Sixth Sense Technology as a combination of computer and smart phone. It works as the device associated to attach at an appropriate angle around the neck of a person and thus the projection starts by means of the micro projector attached to the device. This means, we can act as a moving computer in ourselves and our fingers are working just like a keyboard and mouse.
- This SixthSense setup was built from a battery-powered 3M projector, with a mirror attached to it and, an ordinary webcam. All these devices are connected to an internet-enabled smart phone. According to Pranav Mistry the setup costs less than 20 thousand INR, permits user to display information from the phone to any surface like on the walls, on the body of another person or even on our hand.

Mistry wear the device on a lanyard around his neck, with four color Marker caps red, blue, green and yellow on four fingers that helps the camera to distinguish between four fingers and sense his hand gestures with software developed by Pranav Mistry.

4. Conclusion and Future Scope

In our proposed system user can take a snapshot of the particular activity i.e. road network, accident etc. The application will augment the current position where the picture is taken. The above augmented picture is sent to the concerned authority. Authority take appropriate action on that complaint within time.

References

- [1] Andrea Zanella, Lorenzo Vangelista, Internet of Things for Smart Cities, TRANSACTION ON INTERNET OF THINGS JOURNAL, VOL. 1, NO.1, FEB 2016.
- [2] A. P. Castellani, N. Bui, P. Casari, M. Rossi, Z. Shelby, and M. Zorzi, Architecture and protocols for the Internet of Things: A case study, in Proc. 8th IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PERCOM Workshops), 2010, pp. 678683.
- [3] C. E. A. Mulligan and M. Olsson, Architectural implications of smart city business models: An evolutionary perspective, IEEE Commun. Mag., vol. 51, no. 6, pp. 8085, Jun. 2013.
- [4] A. P. Castellani, M. Dissegna, N. Bui, and M. Zorzi, WebIoT: A web application framework for the internet of things, in Proc. IEEE Wireless Commun. Netw. Conf. Workshops, Paris, France, 2012.
- [5] Chris Thompson, Jules White, Brian Dougherty, Adam Albright, and Douglas C. Schmidt, Using Smartphones

- to Detect Car Accidents and Provide Situational Awareness to Emergency Responders,
- [6] HuiyongWang, Minglu Zhang, JingyangWang, An Emergency Search and Rescue System Based on WSN and Mobile Robot,2009 International Conference on Information Engineering and Computer Science 10.1109/ICIECS.2009.5362881
- [7] N. Walravens and P. Ballon, Platform business models for smart cities:From control and value to governance and public value, IEEE Commun.Mag., vol. 51, no. 6, pp. 7279, Jun. 2013.
- [8] H. Schaers, N. Komninos, M. Pallot, B. Trousse, M. Nilsson, and A.Oliveira, Smart cities and the future internet: Towards cooperation frameworks for open innovation, The Future Internet, Lect. Notes Comput. Sci.,vol. 6656, pp. 431446, 2011.
- [9] D.Cu, M. Hansen, and J. Kang, Urban sensing: Out of the woods,Commun. ACM, vol. 51, no. 3, pp. 2433, Mar.

