Outdoor Object Recognition for Smart Audio Guides

Claudio Baecchi, Tiberio Uricchio, Lorenzo Seidenari and Alberto Del Bimbo
Media Integration and Communication Center, Università degli Studi di Firenze
{name.surname}@unifi.it

ABSTRACT

We present a smart audio guide that adapts itself to the environment the user is navigating into. The system builds automatically a point of interest database exploiting Wikipedia and Google APIs as source. We rely on a computer vision system, to overcome the likely sensor limitations, and determine with high accuracy if the user is facing a certain landmark or if he is not facing any. Thanks to this the guide presents audio description at the most appropriate moment without any user intervention, using text-to-speech augmenting the experience.

KEYWORDS

Computer Vision, Mobile Computing, Cultural Heritage

ACM Reference format:
Claudio Baecchi, Tiberio Uricchio, Lorenzo Seidenari and Alberto Del Bimbo. 2017. Outdoor Object Recognition for Smart Audio Guides. In *Proceedings of MM '17, Mountain View, CA, USA, October 23–27, 2017,* 2 pages.
DOI: 10.1145/3123266.3127923

1 INTRODUCTION

In this work, we present a novel wearable outdoor audio guide that adapts to the actions and interests of a city tourist. The proposed system can automatically gather interest points by exploiting the user GPS position and their relative descriptions obtained from the Internet without any supervision and present them to the user, but only when he is effectively close and in line of sight of an artwork. To accurately detect if the visitor is facing a point of interest we implemented a real-time computer vision system that constantly matches the image viewed by the user with an automatically obtained visual database of the surrounding artworks. On persistent matches, the guide starts the audio description generated by means of text to speech technology.

2 THE SYSTEM

The system is composed of three interacting modules: *i)* the *Location Module* that provides current location and nearby points of interest; *ii)* the *Content Provider* that is responsible to fetch interest point textual information; *iii)* the *Vision Module* that constantly acquires the user view and compares it against a set of expected point of interest appearances. Fig. 1 shows the architectural diagram comprised of each application module.

MM '17, Mountain View, CA, USA

DOI: 10.1145/3123266.3127923

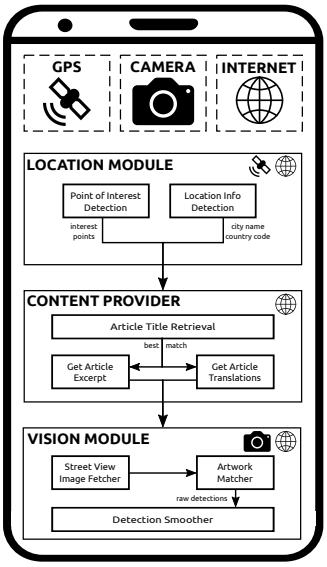

Figure 1: System Architecture

The system estimates an approximate user position via GPS and uses the camera to recognize the facing artwork. The three modules are working cooperatively in an Android application which controls the inputs and present the final information automatically using text-to-speech or interactively using the GUI.

2.1 Location Module

The location module is responsible of retrieving nearby points of interest to the application. It queries Google Places for a list of 20 interest points in a given radius, annotated with one or more `type` of interest point. The user can personalize the application to specify which type he is interested into (e.g. historical monuments) and exclude all the results containing unwanted types (e.g. cafes). In order to avoid finding results for interest points that share a common name but are located elsewhere, we explicitly specify the name of the current city in the query.

2.2 Content Provider

This module translates interest points into artwork descriptions. It queries Wikipedia for articles that contain both the point name and the city name, obtained by the Location module and localized in the local language.

The first result is selected as best candidate and a second query to Wikipedia is performed to collect the page extract which is then used as artwork description. To provide translations into other languages the module also performs an additional query to Wikipedia, this time requesting the Interlanguage Links for the retrieved page.

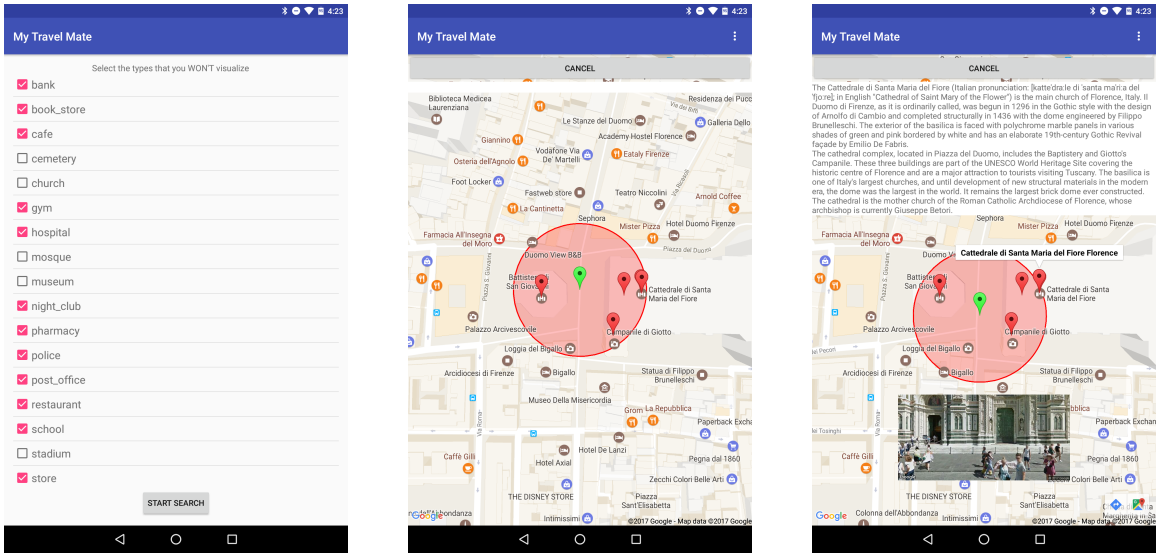

Figure 2: *(left)*, application view for selecting unwanted landmark types; *center* map view of surrounding landmarks; *right* textual description of a selected landmark.

2.3 Vision Module

Understanding when the user is actually facing a landmark is not a trivial task since user position and device orientation are not reliable information [2]. To address this issue we introduced in the system a computer vision algorithm that constantly observes the user perspective and matches it to the surrounding artworks provided by the *Location Module*. To determine if the user is facing one of the surrounding landmarks, the module queries Google Street Map. We retrieve an image taken considering the estimated point of view of the user plus 4 additional ones by varying the angle by ±10 degrees and the pitch by ±5 degrees.

We index SIFT descriptors on multiple kd-trees. To reduce the burden of RANSAC geometric verification, we filter descriptors according to the ratio proposed by [1]. To avoid throwing away good matches, each KD-Tree never stores images of the same artwork.

2.4 Temporal Smoothing

We apply temporal smoothing to prevent erroneous detections, applying a tracking strategy to provide only continuous output values. Given a sequence of input frames, first the *Vision Module* internally produces a series of artwork labels. We flag a detection valid once the same artwork id is emitted for T times consecutively. Each new detection is compared to the last valid one, if the new value is different then it is considered correct only if it persist for at least T frames.

We look at the continuity of the prediction. In case a sequence of labels s, lasting less than T frames, has differing values from the last valid one, we apply the following strategy. If the upcoming value matches the last valid one we assign labels in s to the last valid one. If this is not the case, we assign all the labels in s to background and restart the count. We set $T = 5$ since it gave the best results on our dataset.

2.5 System Implementation and Use Cases

The proposed system has been developed using a NVIDIA Jetson TK1 board, to test the performance of the vision system and then moved to an NVIDIA Shield Tablet K1. The two system are similar in specifics: they are based on an NVIDIA Kepler GPU with 192 CUDA cores, and an NVIDIA 4-plus-1 Quad-core ARM Cortex A15 CPU.

The application provides two possible use-cases. In the first, the user walks through the city with the device in a front pocket with the camera facing forward, and the audio description is provided automatically. In the second use-case, the user can interact with the application. As can be seen in Fig. 2, an user interface shows a map with the current position and a set of close interest points. By touching them, the textual descriptions will be shown and also the audio description can be started at will.

3 CONCLUSION

We have presented a mobile application able to deliver real-time audio information by exploiting Google APIs and Wikipedia for retrieving the relevant textual and visual information. We use the possibly imprecise GPS location to obtain a set of images of nearby entities in which the user may be interested. We than use local feature matching to find which of the landmarks is currently observed. The system have been deployed and tested on a NVIDIA Shield with TK1.

REFERENCES

[1] David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision* 60, 2 (2004), 91–110.
[2] Paul A Zandbergen. 2009. Accuracy of iPhone locations: A comparison of assisted GPS, WiFi and cellular positioning. *Transactions in GIS* 13, s1 (2009), 5–25.