Mobile application for automatic translation with Augmented Reality

Joel-Omar Juárez-Gambino[1], Consuelo-Varinia García-Mendoza[1], Miguel Felix-Mata[2], Mario Aldape-Pérez[3] and Jorge-Emmanuel Morales-Díaz[1]

[1] Superior School of Computer Science, ESCOM-IPN, Lindavista, G.A. Madero, Mexico City, 07738, Mexico
[2] Interdisciplinary Professional Unit on Engineering and Advanced Technologies, UPIITA-IPN, Av. IPN 2580, Barrio La Laguna Ticomn, 07340, Mexico City, Mexico
[3] Center for Computing Innovation and Technology Development, CIDETEC-IPN, Nueva Industrial Vallejo, G.A. Madero, Mexico City, 07738, Mexico
omarjg82@gmail.com, cvgarcia@ipn.mx, mmatar@ipn.mx, maldape@ieee.org, jemd92@hotmail.com

Abstract. In this paper we describe a mobile application for automatic translation. This application is useful for people that need to translate short text, especially from traffic and emergency signs. Using the camera of the mobile device the text within the image is recognized, then it is translated, and finally it is shown superimposed into the original text via augmented reality. In order to automatically translate the text obtained from the camera, we used image analysis, optical character recognition, online translator and augmented reality. Several test were run to verify the performance of the application and we describe the obtained results considering different scenarios.

Keywords: Augmented Reality, Automatic Translation, Mobile Application

1 Introduction

When people travel to a foreign country one of the main problems is the communication, especially if they do not understand the local language. Nowadays there are some devices for helping people in this issue, for instance, the electronic dictionaries in which the text we want to translate is manually introduced, or smartphones with applications where the text is recognized via voice and translated to the selected language. One approach that has been studied during the last year is the automatic translation using an image taken with a camera as an input. Most of the works done in this approach follow the same stages: image analysis, text detection and optical character recognition.

In [1] a scene text extraction system for handheld devices is described. The system uses the built-in camera of a PDA to capture images and send these images to a server, where the text information within the scene is extracted. Using a commercial Chinese and Japanese to English translator the recognized words are translated and sending back to PDA where the words translation is shown over a augmented reality overlay. Another related work is presented in [2] where the authors developed an application for automatic translation designed for Windows Phone. The application identifies the text to be translated interpreting the user gestures on the display. The user can tap or swipe on the screen to point out the position of the text to be translated. Due to the computational resources of the smartphones all the processing is done locally, only the translation is done using the Bing dictionary web service. In [3] an application called TranslatAR was developed. This application shows the translated words obtained from a scene taken with a smartphone camera (Nokia N900). The translation is displayed over an augmented reality overlay. The Google Translate service was used for the translation and the augmented reality overlay was created with OpenGL. Besides there was implemented a tracking algorithm to keep track of the word of interest and present the translation in a live augmented reality overlay.

In this paper we describe the development of a mobile application for Android devices which uses the device camera for detecting and extracting the text included in the visualized image and show its translation with augmented reality. The KLT tracking algorithm [4] was implemented combined with the sensors of device to improve the tracking of the words. In the following sections we present an overview of the system and description of each one of the implemented phases(Section 2); the required features of the mobile device (Section 3); the experimentation and results(Section 4); and finally our conclusions and future work (Section 5).

2 System Overview

In Figure 1 we show the system architecture. The system implements all the phases shown in the figure using only the resources of the mobile phone, just the automatic translation is done with an external source. The system allows three kinds of input:

1. Text
2. Image from gallery
3. Image from live camera

The first one is used when the user provides the text that is wanted to be translated, this is the most simple scenario because no further processing is needed, so the translation phase is done directly from the provided text. In the second one, the user wants to translate a text contained in a stored image, in this case an image processing and optical character recognition is needed in order to extract the text from the image. Finally, the third one is the most challenging situation due to the fact that the input is not a static image, but a live scene that needs to be tracked in order to locate the text that will be translated. In this section, we describe in detail the implemented phases to process the live camera input (the other inputs share the phases).

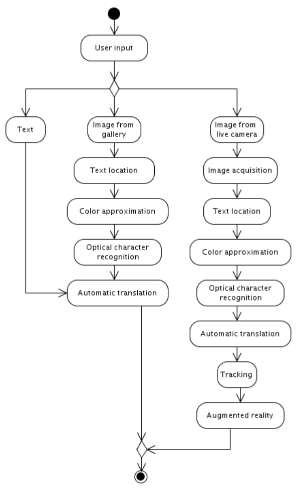


Fig. 1. System overview.

2.1 Image Acquisition

If the input of the system is the live camera, then there is not a static image to be processed instead there are video frames. In order to obtain an image we used OpenCV library [5]. Through a Listener provided by OpenCV we can detect when a frame changes and save the new image.

Once we have an image it is necessary to specify the area where the text is contained. There are two ways for the user to specify such an area; the first one is to set a reference point by touching the screen in the center of the text that wants to be translated; the second one is by drawing a rectangle using the touchscreen as shown in Figure 2, the text should be enclosed in a rectangle.

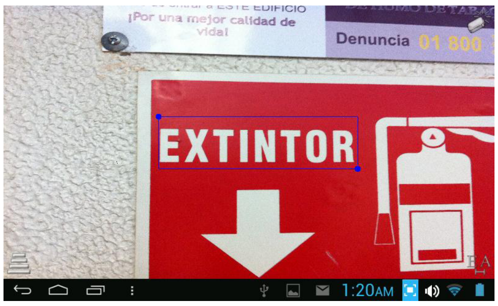


Fig. 2. Manual text selection of the word “EXTINTOR” (extinguisher).

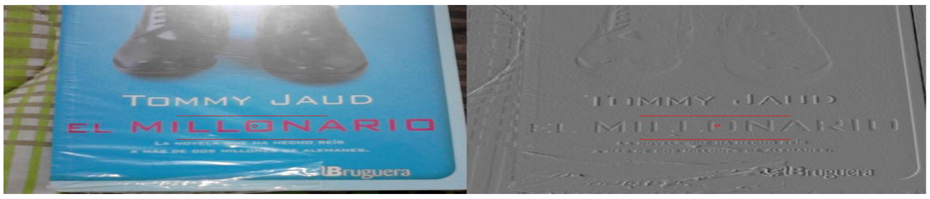


Fig. 3. Image segmentation of the word "MILLONARIO" (millionaire).

2.2 Text Location

After we have the reference point, a first segmentation is performed by identifying possible edges of words using image gradients. For this segmentation both vertical and horizontal tolerance is used to handle the spaces between characters in a word. The algorithm we use to locate the text in the image is based in [3]. In Figure 3 we show an example of the detection of the lower and upper limits using the gradient of the X axis.

2.3 Color Approximation

Every pixel in the image it is represented as a three component vector (RGB) and using a k-means clustering method [6] the background color of the image as well as the color of the text contained in it is identified. By taking the mean value of the clustering we select to wich of the two classes (text or background) the color belongs. This method allows to improve the character segmentation and is used to approximate the color displayed in the augmented reality overlay. For the proper operation of the above algorithm it is necessary that the background color is solid and contrasting with the color of the text contained in the image.

2.4 Optical character recognition

The segmented characters obtained in the above phases are stored as temporal images. These images are processed by the OCR module, we used the Tesseract OCR engine [7] for this purpose. Tesseract is a free distribution OCR engines supported by Google and distributed under Apache license. In Figure 4 we show the recognized characters for the word "DUCKING".

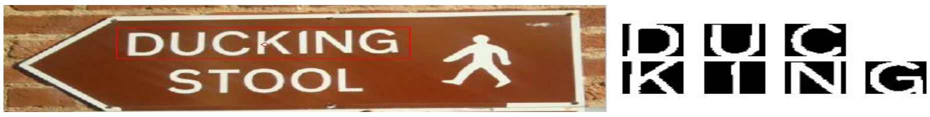


Fig. 4. Character recognition of the word "DUCKING".

2.5 Automatic Translation

Automatic translation is one of the major task in natural language processing. There has been a significant effort during the last decades [8] and despite of the problems of processing natural language, promising results has been obtained [9]. Nowadays there are some commercial software available for automatic translation, we have selected Google translate [10] and WordReference [11] as translate service because they provide a HTTP service. The text we have obtained using the OCR module is sent to the selected dictionary.

2.6 Tracking

Tracking is an important task in video processing, it is used to locate specific objects in a video stream [12]. In our system the object of interest is the text and this object is static, but we assume that the user can move the device when is trying to focus a specific area.

The KLT tracking algorithm was implemented to position the translated text over the original text location, even if the user moves the device during the process. We also use the accelerometer of the device to detect a relevant sudden change in motion and stop the tracking process in order to warning the user that is necessary for the correct function of the system to stay as motionless as possible during the translation process. By stopping the tracking process significant use of memory and processing time are reduced.

2.7 Augmented Reality

Once we have the translated text contained into the image and its position, this translation is displayed over the live video stream. For this purpose we implemented an augmented reality module. This module creates an additional 2D layer to display the translated text using the color of the font and background obtained during the Color Approximation phase. In Figure 5 we show an example of a Spanish-English translation.

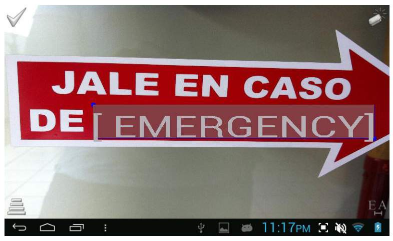


Fig. 5. Translation showed with AR for the Spanish word "EMERGENCIA" (emergency).

We also performed a coordinate conversion, because the resulting image has a standardized size of 320x240 pixels for all devices, however the devices screens have a very variable size and although the original image is scaled to cover the width and height of the screen, the matrix processed retains its original size. The conversion is performed as follow, given a point $P(x, y)$ located in a screen of size S_{HXW}, its position $P_1(x_1, y_1)$ into the original matrix of the image M_{hXw} is calculated by:

$$x_1 = xw/W$$
$$y_1 = yh/H$$

3 Mobile device features

The application was installed and tested successfully in two different mobile devices. One of these devices was a smartphone and the other one was a tablet. Both devices run over Android operating system and the minimum required version for the operating system is 2.3. In Table 1 we show the features of the smartphone and the features of the tablet are shown in Table 2 .

Table 1. Features of the smartphone Vodafone V860 Smart II.

Feature	Description
Display	320x480 px, 3.2 in.
Memory	1 GB ROM, 512 MB RAM
Video	VGA@20 fps
Chipset	Broadcom BCM21553
CPU	823 MHz ARMv6
Sensors	Accelerometer, proximity and magnetic field

Table 2. Features of the tablet WonderMedia XTAB-781+.

Feature	Description
Display	800x400 px, 7 in.
Memory	8 GB ROM, 1 GB DDR RAM
Video	VGA@30 fps
Chipset	WonderMedia WM8850
CPU	1 GHz ARMv7
Sensors	Accelerometer

4 Experiments and Results

The environment where the application has to run is highly variable. It includes changes in the distance, lighting, inclination, background, font and size of the text within the images. In order to verify the performance of the application in all these situations several test were run. All the test consider the text location and character recognition over some of the 8 different signs.

4.1 Distance test

In this test 24 images were selected from the 8 different signs using the live camera (video). For every sign 3 images were taken from different distances (20, 40 and 60 centimeters approximately). In Figure 6 images at different distances are shown and the obtained results are summarized in Table 3. The best result was obtained using medium distance, both the text location and the character recognition had 80% of success.

Table 3. Results obtained at short, medium and long distances.

Distance	**Text location**	**Character recognition**
Short	60%	73.3%
Medium	86.7%	80.0%
Long	73.3%	46.7%

4.2 Lighting test

The lighting conditions may affect the performance of text location and character recognition. For this test 3 different images with variations in the lighting

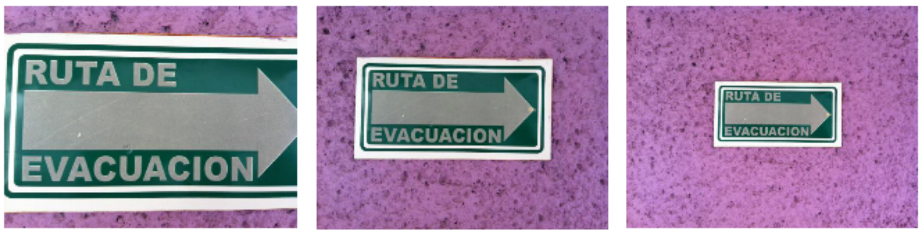
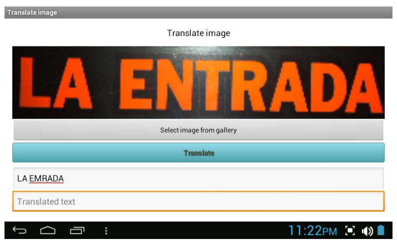


Fig. 6. Images taken (from left to right) at short, medium and long distance with a sign of "RUTA DE EVACUACIN" (evacuation route).

conditions where taken from each one of the 8 available signs. In Table 4 we show the results with the 3 lighting variations. As we can see the most affected functionality is the character recognition. Figure 7 shows an example of a wrong character recognition due to the use of an artificial lighting (smartphone integrated flash), the letters N and T directly receive the flash light which causes a reflection and this changes the color of pixels and finally these two letters are erroneously recognized as the letter M. The best result was obtained using natural environment lighting.Text location shows a 100% of success while character recognition had 80% of success.

Table 4. Results obtained using natural environment lighting, artificial lighting and night lighting.

Lighting	**Text location**	**Character recognition**
Environment	100%	81.3.3%
Artificial	68.8%	37.5%
Night	50.0%	56.3%



LA EMRADA

Translated text

Fig. 7. Wrong character recognition in a sign with the words "LA ENTRADA" (the entrance).

4.3 Inclination test

Another factor affecting the performance of the application is the inclination of the image that contains the text to be translated. The image could be taken from different angles so the text within would also be distorted. In order to run the test 11 images in different angles were taken from the same signal and the contained text was recognized. In Figure 8 we show the images taken from different inclination angles. As we can see in Table 5 the most affected functionality is the character recognition while text location shows no problem. The best results where obtained with an inclination angle between -30° to 30° because the text is less distorted.

Table 5. Results obtained using twelve different inclination angles.

Inclination angle	Text location	Character recognition
-75° to -45°	100%	11%
-30° to 30°	100%	87%
45° to 75°	100%	33%

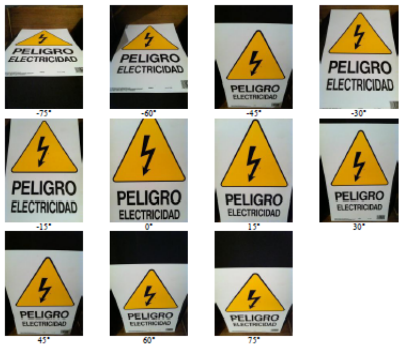


Fig. 8. Images taken from different angles of a sign with the words “PELIGRO ELECTRICIDAD” (danger electricity).

4.4 Miscellaneous tests

In addition to the tests performed in the above sections, characteristics of text that can affect the application performance were also tested. Three of these characteristics are background, font and size of the text. Table 6 summarizes the recommended conditions and text characteristics to get the best results.

Table 6. Recommended conditions and text characteristics.

Condition/characteristic	Recommendation
Font size	10% to 90% of the device display
Font	Sans-Serif family with a constant spacing between characters (without kerning)
Background color	Uniform background color in the area containing the text
Font color	Uniform font color and contrasting with the background
Lighting	Uniform environment light
Inclination	Tolerance of 30° to the horizontal plane passing through the center of the object

5 Conclusions and Future Work

The application has to deal with an uncontrolled environment and the changes in the environment make it very difficult to ensure proper operation all the time. The most challenging task is character recognition, even though OCR engines have improved more effort is required. Thanks to the current mobile devices resources all the image processing phases was locally implemented. The use of integrated sensors like the accelerometer helps to save valuable time and processing, improving the performance of the application. With the improvements on the capabilities of mobile devices and OCR engines these kind of applications will be fully operational and available in our every day life.

Acknowledgments

We thank the support of IPN, ESCOM-IPN, SIP-IPN project number: 20152102, COFAA-IPN, EDI-IPN. Special thanks to Alan Martinez and Carlos Rocha for helping with the application programming.

References

1. Haritaoglu, I.: Scene Text Extraction and Translation for Handheld Devices. In: CVPR (2), IEEE Computer Society (2001) 408–413
2. Du, J., Huo, Q., Sun, L., Sun, J.: Snap and Translate Using Windows Phone. In: ICDAR, IEEE (2011) 809–813
3. Fragoso, V., Gauglitz, S., Zamora, S., Kleban, J., Turk, M.: TranslatAR: A mobile augmented reality translator. In: WACV, IEEE Computer Society (2011) 497–502
4. Bouguet, J.Y.: Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm, Intel Corporation Microprocessor Research Labs (2000)
5. Bradski, G.: The OpenCV library. Dr. Dobb's Journal of Software Tools (2000)
6. MacQueen, J.B.: Some Methods for Classification and Analysis of MultiVariate Observations. In Cam, L.M.L., Neyman, J., eds.: Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability. Volume 1., University of California Press (1967) 281–297
7. Rice, S.V., Jenkins, F.R., Nartker, T.A.: (The Fourth Annual Test of OCR Accuracy)
8. Hutchins, W.J.: Machine translation: A concise history (2007)
9. Ney, H.: One decade of statistical machine translation: 1996-2005. In: Automatic Speech Recognition and Understanding, 2005 IEEE Workshop on. (2005) 2–11
10. Google: Google translate. `https://translate.google.com` (2014)
11. Kellogg, M.: WordReference. `http://www.wordreference.com` (2014)
12. Yilmaz, A., Javed, O., Shah, M.: Object Tracking: A Survey. ACM Comput. Surv. **38** (2006)