Journal of Signal and Information Processing, **2012, 3, 228-237**
http://dx.doi.org/10.4236/jsip.2012.32031 Published Online May 2012 (http://www.SciRP.org/journal/jsip)

Efficient Text Extraction Algorithm Using Color Clustering for Language Translation in Mobile Phone

Adrián Canedo-Rodríguez[1], Jung Hyoun Kim[2], Soo-Hyung Kim[3], John Kelly[2], Jung Hee Kim[2], Sun Yi[2], Sai Kiran Veeramachaneni[2], Yolanda Blanco-Fernández[1]

[1]Department of Telematics Engineering, University of Vigo, Vigo, Spain; [2]College of Engineering, North Carolina A&T State University, Greensboro, USA; [3]Department of Computer Science, Chonnam National University, Kwangju, South Korea.
Email: adri_canedo@hotmail.com, kim@ncat.edu, shkim@chonnam.ac.kr

Received March 22nd, 2012; revised April 17th, 2012; accepted May 11th, 2012

ABSTRACT

Many Text Extraction methodologies have been proposed, but none of them are suitable to be part of a real system implemented on a device with low computational resources, either because their accuracy is insufficient, or because their performance is too slow. In this sense, we propose a Text Extraction algorithm for the context of language translation of scene text images with mobile phones, which is fast and accurate at the same time. The algorithm uses very efficient computations to calculate the Principal Color Components of a previously quantized image, and decides which ones are the main foreground-background colors, after which it extracts the text in the image. We have compared our algorithm with other algorithms using commercial OCR, achieving accuracy rates more than 12% higher, and performing two times faster. Also, our methodology is more robust against common degradations, such as uneven illumination, or blurring. Thus, we developed a very attractive system to accurately separate foreground and background from scene text images, working over low computational resources devices.

Keywords: Text Extraction; Color Quantization; Text Binarization; Language Translation

1. Introduction

Within the general problem of Pattern Recognition, Text Information Extraction (TIE) in images and video exhibit characteristics that deserve individual analyses and solutions. Traditionally, TIE was related to the analysis of scanned documents, which provided a pseudo-ideal scenario: high-resolution, minimal character shape distortion, even and adequate illumination, clear, simple and known backgrounds, minimal blur, and so on. An Optical Character Recognition (OCR) technology was developed according to these ideal scenarios, achieving high recognition rates. However, this simple application did not fulfill the users' needs at all because scanners are slow and not portable. Moreover, the OCR technology can only process text embedded in documents, not in other objects.

The explosion of Handheld Imaging Devices (HIDs) represents an excellent opportunity to take advantage of TIE technology, and provide variety of useful solutions to the users' needs. These devices are portable, compact, able to capture images of any text in any scenario (usually called Scene Text Images, or Natural Scenes), and experienced a Moore's Law price reduction since they first appeared (specifically the digital cameras, either standalone or embedded into mobile phones or PDAs). Sign recognition and translation for travelers, automatic license plate recognition for law enforcement, driver assistance systems, assistance for visually impaired persons, or autonomous vehicle navigation represent just a small set of the wide range of possibilities of this new area.

However, the new TIE scenario that HIDs bring is far from being at the maturity level of OCR technology. The resolution is lower than in scanned documents, the surface of the object on which the text is embedded is arbitrary, the text can be distorted, the illumination is very difficult to control, and the background is often complex. Therefore, commercial OCRs present very low recognition rates on this kind of images, requiring preprocessing to improve performance, delimitating the text areas (text localization), and separating foreground from background (text binarization). These steps are usually very computationally expensive, so their implementation on HIDs is often unfeasible. On the other hand, those methods which are computationally efficient are not robust enough to cope with "real world" degradations, such as uneven illuminations, or lighting reflections.

In this paper, we propose a simple, fast, and accurate algorithm to separate foreground and background in text detected within natural scene images, so it can be im-

plemented as a part of an accurate, successful, and useful TIE system into these low computational devices. Instead of using just monochromatic information as in well-known fast algorithms, we use all the color information to ensure robustness against "real world" degradations. This adds complexity to the system, so we will use simple computations to perform the segmentation, in order to minimize the processing time. First, the color image is quantized to reduce the number of colors. Then, the Principal Color Components in the image are isolated, and from them, the foreground and background colors are separated based on the number of occurrences of each component, and the distance between them.

In Section 2, we introduce the general TIE system, with the main challenges with which it has to cope, its working scenario, and the different steps involved. After that, the text extraction step is explored in detail, and the different Text Extraction algorithms are introduced. In Sections 3 and 4, we describe and verify our algorithm, whose results are described and analyzed in Section 5. Finally, we summarize the contents of the paper, and discuss the convenience of our proposal, and our future research direction in Section 6.

2. Background

2.1. Overall TIE System Working Scenario

In this section, we will summarize the main background ideas related with the Text Information Extraction area [1-3]. Since the beginning of TIE research, there have been many proposals for specific applications. Due to the enormity of the challenges such as layout complexity, noise, distortions etc., no general system has been proposed so far capable of handling all the possible situations. The main factors of a TIE system are text characteristics, image scenario, and uneven image effects. Their importance depend on specific application of the system: if the goal of the system is to process scanned images from text documents, we will find that most of those challenges become insignificant, while if the goal is to process any picture that contains a scene text, most of them will be critical.

The images for which a TIE system can be divided into two major groups: traditional documents (subdivided into gray scale documents, or multicolor documents), and multi-context images (subdivided into superimposed text images, or scene text images). In this regard, gray-scale documents are less challenging, followed by multicolor documents, and superimposed text images. Scene text images are, by far, the most complex among the four.

A TIE system receives a still image or a sequence of images as an input, which can either contain text or not within them. The overall steps for the TIE system to recognize a text in an image or a video clip are:

- Text Detection: Determination of the presence of text.
- Text Localization: Determination of the location of the text.
- Text Tracking: In sequences of images, determination of the coherence of the text between frames, to reduce the processing time by not applying all the steps to every frame, and to maintain the integrity of position across adjacent frames.
- Text Extraction: Binarization of the image by separating the text components (foreground) from the background.
- Text Enhancement: Increasing of the quality of the binary image, mainly by increasing its resolution and reducing the noise.
- Text Recognition (OCR): Transformation of the binary text image into plain text using an Optical Character Recognition Engine.

Each step may include a pre-processing, and a post-processing part, to increase its overall accuracy. Depending on the application and its requirements, the TIE system will involve all the steps above, or a subset of them. Particularly, the text localization is one of the important steps for the TIE, especially the text extraction so that we have developed a new simple and fast text localization method [4].

2.2. Text Localization Developed for This Purpose

In Text Localization, high speed and locating the important text in the image are the most important things [4]. Many text localization methods have been proposed so far but none of them can be implemented in real scene text translation system by taking images using mobile phones. Images are generally stored in JPEG format because a mobile phone doesn't have much memory space. Thus, we have developed and proposed a new simple and efficient text localization method. The two expectations on the text localization method we proposed are the images are stored in JPEG format and the important text in the image is centered. A DCT block which contains characters present high frequency components both in horizontal and vertical directions for locating the text because of the variations in foreground and back ground.

The text localization algorithm is simple and affordable for the images taken from mobile phones. This algorithm shows high accuracy rates in different conditions and it can be implemented on devices with low computational performance.

2.3. Concept of Text Extraction

Text Extraction which also called Text Binarization is the part of the TIE system where, given an input text image, the background and the foreground are separated,

and a binary image is produced as the output. We will assume, without lack of generality, that the input text image contains a precisely localized text, although certain imprecision in the localization are also acceptable. The text extraction step is an essential part of every TIE system, as it will determine the accuracy of the text recognition step, depending on the quality of the foreground-background separation. In order to perform this separation, all of the text image characteristics can be used, such as color differences, character position, character shape, layout, and so on.

In general, text extraction methods use the colors of the foreground and the background as the main information source to separate them. That is, they divide the color space in groups, and each of them is classified as foreground or background, so are the pixels which contain those colors. The first generation of text extraction methodologies performed the segmentation using gray-scale images, assuming that the backgrounds were clean, and the degradations on the image were small enough not to be considered. Nevertheless, this is not the case for natural scenes, so soon new algorithms were developed, using the color information of the text pictures, allowing the possibility of dealing with more complex situations. Currently, text extraction methods can be classified into two main groups: Threshold-based, and Group-based. Thresholds based methods are calculated in order to classify the different colors as foreground, or as background. They can be sub-classified into three groups: Histogram-based, Adaptive or Local, and Entropy-based [5-14]. Group-based methodologies group pixels together according to certain criteria. Based on how the pixels are classified, they can be divided in three major groups: Region-based, Clustering-based and Learning-based [15-30].

3. Description of Proposed Methodology

Our algorithm has been designed as the text extraction part of a system for English to Spanish translation of the text present on signboard images, implemented on a mobile phone [31]. Our focus was to develop an accurate and efficient methodology. On the one hand, the method has to be accurate enough in order to ensure the usefulness of the TIE systems which utilize it as a part of them: in other words, the text extraction part should not be a major source of errors for the system. On the other hand, the method has to be efficient, so it could be implemented in any system over low computational resources devices: specifically, efficiency is critical when dealing with applications when is needed to give results instantly to the user, such as text images translation (our case), or text images to speech applications.

The algorithm assumes that there exists a reduced set of colors of the image's pixels, called Principal Components. Due to the contrast necessary for text images between text and background to ensure readability, all the Principal Components can be classified either as foreground, or as background, with one of the components as the centroid of the Principal Components Group. However, since the image can be degraded in various ways, each pixel color will be distorted in more or less grade, and converted on a different color. By knowing these components, the system will be capable of recovering the original value of each pixel, and then, segment the image.

Although the most efficient text extraction algorithms use gray-scale or monochromatic images to perform the separation, we decided to use the information of the whole *RGB* color space because of the accuracy benefits that it implies. However, this decision causes the problem to be much more complex, so the algorithm's computations were designed to be as simple as possible, to maintain the processing time as low as possible. First of all, the color space is reduced from a 24 bit (2^{24} colors), to a 12 bit (2^{12} colors) representation, to reduce the complexity of the problem, from which just the colors with a large number of appearances will be further considered. Then, to take into account possible distortions, the number of occurrences of each color is calculated as its own occurrences, plus the occurrences of its neighbors. Using that information, the technique isolates the Principal Components, decides the centroid of each Principal Components Group, and classifies each pixel either as foreground, or as background.

3.1. Notation

Equations (1) and (2) represent a digital image using the *RGB* (Red, Green, Blue) color space using a 3D matrix:

$$f(x,y)=\left\{ f_m(x,y)\middle| m\in\{R,G,B\},\ x\in\{0,\cdots,M-1\},\ y\in\{0,\cdots,N-1\}\right\} \tag{1}$$

$$f_m(x,y)=\begin{pmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & & \ddots & \vdots \\ f(M-1,0) & \cdots & \cdots & f(M-1,N-1) \end{pmatrix}, \quad f_m(x,y)\in\{0,\cdots,L-1\} \tag{2}$$

where $f_m(x,y)$ is a representation of the intensity of the color component m on the image. Therefore, the color of a pixel (x, y) will be a vector composed by the pixel's color component on each color space:

$$f(x,y)=\left\{ f_R(x,y),\ f_G(x,y),\ f_B(x,y)\right\} \tag{3}$$

Attending to this notation, the digital image $f(x, y)$ contains $M \times N$ pixels. Each pixel's color is represented

by a combination of three intensity values in three different color components (*R*, *G*, *B*), where a high intensity of a particular component stands for a high importance of that component in the pixel's color. Since the number of intensity levels is *L*, and the number of color components is three, the total number of possible colors in the image will be L^3.

3.2. Text Extraction

3.2.1. Color Reduction

Considering an ideal case, the text image just contains two colors (foreground and background colors), so it is easy to binarize. Nevertheless, in the case that we are considering, the number of colors is much larger because of the various difficulties. We can model the image as a combination of several principal colors, and their distortions, caused by blurring effects, uneven illumination, lightening reflections, and so on. Those principal components can be classified either as part of the foreground or as part of the background, and the image can be binarized.

First of all, in order to save computational time, we reduce the number of colors by performing a color quantization of the image. Given the image $f(x, y)$, each intensity component $f_m(x, y)$ is quantized from its L original levels to D levels ($D < L$), so the number of colors is reduced from L^3 to D^3. In our real implementation, the original images are represented with $L = 2^8$ levels (8 bits per *RGB* channel), and we quantize them into $D = 2^4$ levels (4 bits per channel). Following with the general notation, the quantization step is:

$$\Delta = \frac{L}{D}$$

And the quantized image and its quantized components are defined as in (4) and (5).

$$\tilde{f}(x,y) = \left\{ \tilde{f}_m(x,y) \middle| m \in \{R,G,B\}, \; x \in \{0,\cdots,M-1\}, \; y \in \{0,\cdots,N-1\} \right\} \tag{4}$$

$$\begin{aligned} \tilde{f}_m(x,y) &= \Delta \cdot \left(\frac{1}{2} + \left\lfloor \frac{f_m(x,y)}{L} D \right\rfloor \right) \\ &= \Delta \cdot \left(\frac{1}{2} + l_m(x,y) \right) \end{aligned} \tag{5}$$

where $l_m(x,y) \in \{0,\cdots,D-1\}$.

However, our goal is not to give a representation of the quantized image, but to cluster each pixel into a group. **Figure 1** shows a graphical example for the quantization of the red and green components of different pixels in an image and the calculation of its correspondent $l_r(x,y)$ and $l_g(x,y)$ values. Therefore, we can define the level of the image as in (6) which gives us a compact and efficient representation of the quantized image:

$$l(x,y) = \left\{ l_m(x,y), \; m \in \{R,G,B\} \right\} \tag{6}$$

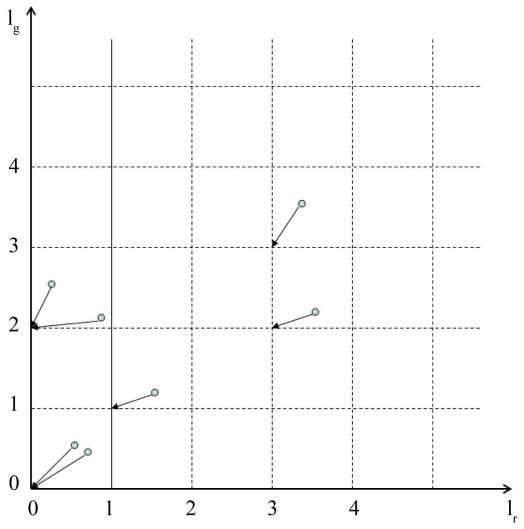

Figure 1. Graphical example: the quantization of the red and green components of different (*x*, *y*) pixels (represented by circles).

3.2.2. Principal Color Extraction

Each pixel (*x*, *y*) will be clustered into a group C'_{rgb}, depending on its level:

$$C'_{rgb} = \left\{ (x,y) \middle| l(x,y) = (r,g,b) \right\} \tag{7}$$

where $r,g,b = \{0,\cdots,D-1\}$.

From which we take into account (to reduce further computational complexity) the clusters whose number of elements (pixels) is larger than the number of pixels of the image divided by the number of groups C'_{rgb} in (8) that contain at least one pixel:

$$C_{rgb} = \left\{ C'_{r_i g_i b_i} \middle| \left| C'_{r_i g_i b_i} \right| > \frac{M \cdot N}{\left| \{ C''_{r_j g_j b_j} \mid \left| C'_{r_j g_j b_j} \right| \geq 1 \} \right|} \right\}, \quad i,j \in \left\{ 0,\cdots,\left| C'_{rgb} \right| - 1 \right\} \tag{8}$$

At this point, we will extract the Principal Color Components of the image. Each component contains one main color group, and its neighbors, which represent the distortions on the principal color of the component. The neighborhood of a color contains the color groups whose distance from the considered color is less than or equal to one using the Chebyshev distance. **Figure 2** shows the neighborhood (grey) of a cell (black), even though we consider the black cell as a part of its neighborhood. Therefore, the neighbors of the black cell are those located within a maximum Chebyshev distance of 1.

$$N_{r_i g_i b_i} = \left\{ C_{r_j g_j b_j} \middle| \max\left(\left| r_i - r_j \right|, \left| g_i - g_j \right|, \left| b_i - b_j \right| \right) \leq 1 \right\}, \quad \forall \, i,j \in \left\{ 0,\cdots,\left| C_{rgb} \right| - 1 \right\} \tag{9}$$

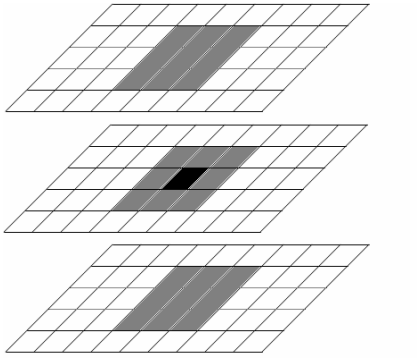

Figure 2. The neighborhood (grey) of a cell (black) within a maximum Chebyshev distance of 1.

$$N_{rgb} = \left\{ N_{r_i g_i b_i} \,\middle|\, i \in \left\{0, \cdots, \left|C_{rgb}\right| - 1\right\} \right\} \tag{10}$$

Using the definition of the neighborhood of a group, we define the importance of a color as the number of pixels which can be considered as formed by it:

$$p\left(C_{r_i g_i b_i}\right) = \left\{ \left| N_{r_i g_i bi} \right| , i \in \left\{0, \cdots, \left|C_{rgb}\right| - 1\right\} \right\} \tag{11}$$

By knowing the number of occurrences of each color on the image, we can decide which ones are the principals, that is, the most frequent ones. The following algorithm shows how the principal components are extracted. Initially, both the group of Principal Color Components P_{rgb}, and the group of colors which cannot be considered to be Principal Color Components E_{rgb} are empty. In each iteration *n*, the largest element of C_{rgb} not belonging to E_{rgb} is included on P_{rgb} (as its n^{th} component P^n_{rgb}) and excluded along with its neighbors from C_{rgb} for future selections, by including them in E_{rgb} (as its n^{th} component E^n_{rgb}). By doing this, we choose the Principal Color Components P_{rgb} as the union of the most important colors on each iteration (P^n_{rgb}), excluding those which can be considered as distortions of Principal Color Components (E_{rgb}).

$$\begin{aligned}
&P^0_{rgb} = \varnothing \\
&\text{while} \left(\bigcup_{i=0}^{n} E^i_{rgb} \neq C_{rgb} \right) \\
&\left\{ E^0_{rgb} = \varnothing, P^n_{rgb} = C_{r_i g_i bi} \,\middle|\, p\left(C_{r_i g_i b_i}\right) > p\left(C_{r_j g_j b_j}\right), \right. \\
&\quad \forall\, C_{r_j g_j b_j} \in \left(C_{rgb} - \bigcup_{i=0}^{n-1} E^i{}_{rgb} \right) \\
&\quad n = 0, P^n_{rgb} = C_{r_i g_i bi} \,\middle|\, p\left(C_{r_i g_i b_i}\right) > p\left(C_{r_j g_j b_j}\right), \\
&\quad \left. \forall\, C_{r_j g_j b_j} \in \left(C_{rgb} - \bigcup_{i=0}^{n-1} E^i{}_{rgb} \right), n = n + 1 \right\}
\end{aligned} \tag{12}$$

where $E_{rgb} = \bigcup_{i=0}^{n} E^i_{rgb}$ and $P_{rgb} = \bigcup_{i=0}^{n} P^i_{rgb}$.

Figure 3 shows an example for the extraction of the Principal Color Components (circles) and their respective neighborhoods (polygons) in a two dimensional color space, where each cell represents a color. The darker gray level of a cell is the larger amount of pixels belonging to it, so that the more important Component is the brown one, followed by the green one, and finally, the yellow one. The white color on a cell represents the absence of pixels belonging to it.

3.2.3. Text Binarization

We separate the foreground and the background by using the contrast between colors, and the importance of those colors in the image, represented by the number of pixels that belong to each color group.

In order to measure the contrast, we calculate the Euclidean distance (widely used in Text Extraction) between each pair (i, j) of groups of P_{rgb} as shown in (13):

$$\begin{aligned}
&d_e\left(P_{r_i g_i b_i}, P_{r_j g_j b_j}\right) \\
&= \sqrt{\left(r_i - r_j\right)^2 + \left(g_i - g_j\right)^2 + \left(b_i - b_j\right)^2}
\end{aligned} \tag{13}$$

where $i, j \in \left\{0, \cdots, \left|C_{rgb}\right| - 1\right\}, i \neq j$.

Also, the importance of each color group is calculated as the number of pixels of the group $p\left(C_{r_i g_i b_i}\right)$ as in (11). We assume that, even if several colors can form the foreground and the background, it is possible to select two of them as the main ones and the rest of them can be classified as variations of one of them (the most similar one). The foreground-background couple of main color groups will be those which maximize the combination of both contrast and color importance, represented by the Foreground-Background Centroid function:

$$\begin{aligned}
&C\left(P_{r_i g_i b_i}, P_{r_j g_j b_j}\right) \\
&= d_e\left(P_{r_i g_i b_i}, P_{r_j g_j b_j}\right) \cdot \left(p\left(P_{r_i g_i b_i}\right) + p\left(P_{r_j g_j b_j}\right) \right)
\end{aligned} \tag{14}$$

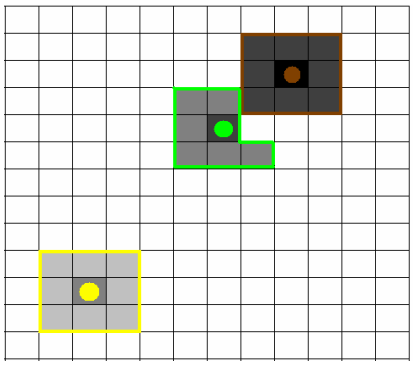

Figure 3. Extraction of the principal color components (circles) and their respective neighborhoods (polygons) in a two dimensional color space.

we decide for every group in $\left\{C'_{r_j g_j b_j} \middle\| C'_{r_j g_j b_j}\right| \geq 1\}$ that it belongs to the background if it is closer (Euclidean distance) to the main background group, and vice versa. After this decision, the foreground and the background will be separated, and the image can be binarized.

4. Verification of the Methodology

Our goal was to develop a new methodology suitable for its implementation under architectures with limited computational resources, reliable enough to handle the more common degradations present in natural scene images, and fast enough to work in a reasonably small time period. We developed an algorithm that can be classified both on the Histogram-based and on the Clustering-based ones. On the one hand, Histogram-based algorithm always construct one or several histograms (one for each color component), seeking for peaks on them (dominant colors), and defining thresholds on the valleys between peaks (modes). In this regard, we build a 3-D histogram and enhance the most frequent colors as well the number and location of the "peaks". On the other hand, the localization of these modes (Principal Color Components) is performed by iteratively clustering the different candidates into larger groups, until reaching the real number of important colors in the image.

4.1. Gray-Level Verses *RGB* Space

First of all, we decided to use all the color information of the image, on the contrary of other approaches, which just use gray-scale images or each channel independently although the color channels are usually correlated, as in the *RGB* space. It is easy to demonstrate that by just using the luminance component of the image it is more difficult to separate foreground and background, or even impossible. Consider the usual transformation from any *RGB* color representation to its equivalent Luminance component:

$$Y = 0.3R + 0.59G + 0.11B \tag{15}$$

As any linear equation with more than one unknown element, Equation (15) has infinite solutions; for every Y value, there can be several possible *RGB* combinations which produce it. If we constrain the *RGB* color space to a 24 bits representation (that is, 8 bits per channel), there are several possible combinations of *RGB* values which produce the same or closer Y values as shown in **Figure 4**. In other words, although the foreground and the background can be very different in color image, the Luminance transformation introduces distortion on the original image, making the segmentation more difficult, and even impossible. Thus, we decided to use the whole color information.

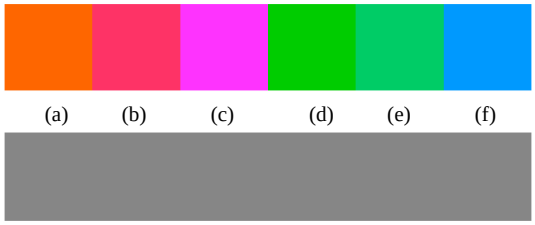

Figure 4. Example of six different colors which theoretically produce the same gray value: the *RGB* and *Y* values normalized on the interval [0, 1] are: (a) *R* = 1, *G* = 0.339, *B* = 0, producing *Y* = 0.5; (b) *R* = 1, *G* = 0.2458, *B* = 0.5, producing *Y* = 0.5; (c) *R* = 1, *G* = 0.1525, *B* = 1, producing *Y* = 0.5; (d) *R* = 0, *G* = 0.8475, *B* = 0, producing *Y* = 0.5; (e) *R* = 0, *G* = 0.7542, *B* = 0.5, producing *Y* = 0.5; (f) *R* = 0, *G* = 0.661, *B* = 1, producing *Y* = 0.5.

4.2. Image Quantization

Images usually contain a big amount of redundant color information, in order to enhance the quality of the human visual perception. Nevertheless, most of this information is not useful for the purpose of text extraction, and it just increases the processing time of the text extraction algorithms. The best solution seems to be a color quantization of each *RGB* color channel as we proposed in (5), especially if we choose D as a power of 2, since the quantization operation can be done by a simple bit dropping operation. **Figure 5** is an example of the difference between the original images, and the quantized ones. The quantization does not cause major differences between the images, so it does not cause any problem on the further Text Extraction.

4.3. 3-D Color Histogram

At a glance, histogram-based binarization algorithms are the most suited to work on our target devices (mobile phones) since they are very simple, fast, and quite reliable under controlled degradation conditions, but they often fail when implemented to deal with "real world" images. Also, they usually miss important information on the image, not only because they just process gray-scale images, but because they treat each color channel independently, as if they were not correlated. In order obtain uncorrelated color components, some authors apply the Fisher Discriminant Analysis to the image before constructing the histogram of each channel [32], but this processing becomes computationally prohibitive on our devices. Instead, our algorithm uses the idea of building a 3-D histogram [33-35], looking for the peaks by erasing the less-frequent colors for simplicity as in (8), and enhancing the most suited candidates depending on the importance of its neighbors as in (11).

4.4. Principal Components Extraction

Text and background are supposed to follow certain color

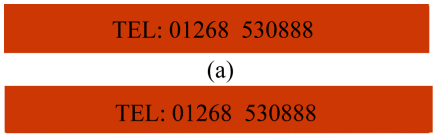

Figure 5. (a) Original image with a 24 bit representation; (b) Its quantized image with a 12 bit representation.

patterns, which we use to perform the binarization, although minimizing the importance of the assumptions made, for the system versatility. For example, texts are designed to be readable, so the contrast between the text and the background is high in general. Also, to ensure the readability, the foreground-background contrast exists even in the case of complex backgrounds. In an almost ideal case, there are just two colors on the image, and slight variations of them caused by small degradations. These two colors in the image would be the most common ones, so it would be easy to find, and separate the foreground and the background. However, not only the "real world" distortions are bigger than in this ideal case, but also the texts and the background can be designed with several colors. For these reasons, any assumption about the number of colors present on the image before knowing its nature just reduces the versatility of the technique. Our algorithm takes advantage of the aforementioned patterns, extracting the most important colors without making any assumptions on its number, ensuring the flexibility against a wide range of different situations. Other conventional classification schemes, such as k-means, fuzzy c-means, GMM, need to perform iterative algorithms involving complex statistical group calculations, and therefore they are computationally expensive to implement on mobile phones. On the contrary, our classification algorithm in (12) is simple enough to be implemented on these devices, without lack of the quality of the binarized results, as it will be seen in the Section 5.

4.5. Foreground-Background Centroid Function

Using the set of Principal Components, we have to binarize the image. We select one Principal Component as the centroid of the text group, and another one as the centroid of the background group, based on the Foreground-Background Centroid Function as in (14), maximizing the combination of contrast, and color frequency. In general, although the image can consist of several components, two of them will be the more frequent ones, corresponding with the foreground and background main colors. Also, as the foreground-background contrast is supposed to be high, the distance between these components should be high. So by maximizing a combination of both frequency and distance, the main foreground and background colors will be efficiently extracted.

4.6. Color Distance Selection

Although there exist many possible distances to measure the contrast between colors, the more widely used on Text Extraction are: Manhattan distance as (16), Euclidean distance as (17), Cosine distance as (18) or any combination of them.

$$d_m(a,b)=\sum_{i=1}^{n}\left|a_i-b_i\right| \tag{16}$$

$$d_e(a,b)=\left(\sum_{i=1}^{n}\left(a_i-b_i\right)^2\right)^{\frac{1}{2}} \tag{17}$$

$$d_{\cos}(a,b)=\frac{\sum_{i=1}^{n}a_i\cdot b_i}{\left(\sum_{i=1}^{n}a_i^{\,2}\right)\cdot\left(\sum_{i=1}^{n}b_i^{\,2}\right)} \tag{18}$$

where $a=\{a_1,a_2,a_3,\cdots,a_n\}$ and $b=\{b_1,b_2,b_3,\cdots,b_n\}$ are two general n-dimensional vectors.

Generally speaking, the Euclidean distance was the most robust one in our experiments, although the Manhattan distance showed similar results. The Manhattan distance has the advantage of its simplicity so that it requires less computing time. On the other hand, the Cosine distance shows less robustness than the others because of lacks of the difference measures between color intensity, although it gives a good measure of the hue difference of the colors which is robust against uneven lighting effects. However, a combination of the Cosine distance with the Euclidean or the Manhattan distance could improve the accuracy of the algorithm with measuring hue and color intensity at the same time. Nevertheless, this is not the aim of this paper, and we leave it for future improvements.

5. Experimental Results

Giving an objective measurement of the performance of any TIE system is a very complex task, mainly because of its strong dependence on the selected set of images with which the system has to work. A more realistic measure can be given by implementing another well-known binarization algorithm, and comparing both results. In our case, we have chosen Otsu's binarization algorithm, since it performs quite good in a very short amount of time, so it could be considered as a candidate for the Text Extraction step in devices with low computational resources. Also, since it is one of the most referenced algorithms in the literature, there exist implementations available in several programming languages.

The binarization results of both algorithms were recognized using a commercial OCR program (Hacking Tesseract 2.0), using the standard measures, Precision and Recal [36], to compare them:

$$\text{Precission} = \frac{\text{Correctly Recognized Characters}}{\text{Totally Recognized Characters}}$$

$$\text{Recall} = \frac{\text{Correctly Detected Characters}}{\text{Total Characters}}$$

In order to compare the two methodologies in terms of accuracy and processing time, we have programmed our algorithm using Matlab 7.6.9 (R2008a) on a Pentium 4 PC (CPU 3.8 GHz), and used Matlab's implementation of Otsu's algorithm. At the same time, we have built two different databases: one consisting on 83 text images (1008 characters in total), and other containing 302 characters. The first one measures the performance of the algorithm working globally in the text image, while the second one measures the performance in ideal "character splitting" situations, that is, an ideal local binarization. We avoided, on the first data base images, that could be very challenging for the OCR engine even under ideal binarization conditions, such as those containing very small, or artistic fonts.

The result in **Tables 1** and **2** shows that our algorithm performs better that Otsu's, both as a global and as a local algorithm. Roughly speaking, our algorithm's accuracy is about 12% higher than Otsu's with the first database (text images), and about 15% with the second one (individual characters). By analyzing carefully the binarized images, it is clear as well that our algorithm performs better than Otsu's when dealing with blurring images, uneven illuminations, and foreground-background colors which produce similar gray levels, which could explain the accuracy differences. Also, we have measured the processing times of both methods with the first database (text images). On this experiment, our algorithm performs about two times faster than Otsu's, with which we demonstrate that not only our algorithm is more accurate, but also faster. In **Table 3**, our algorithm is compared with different other algorithms and we achieved good success rate than other algorithms.

Figure 6 presents the comparison of the binarization results between our algorithm and the Otsu's algorithm

Table 1. Performance comparison with the first database.

	Otsu	Our
Precision	77.6%	89.08%
Recall	67.36%	80.15%
Average processing time	0.13 seconds	0.064 seconds

Table 2. Performance comparison with the second database.

	Otsu	Our
Precision	61.65%	77.94%
Recall	54.3%	70.20%

Table 3. Comparison of different text detection algorithms.

Algorithm	No of images	Result
Messalodi [7]	100 gray scale images of covers	54% precision, 91.2% recall
Otsu [5]	Experiment on gray level images of different sizes	77.6% precision, 67.36% recall
Gllavata [12]	175 images of various types with and without text	84.94% precision 85.94% recall (with local threshold extension)
Kim [37]	50 true color images	124 text lines, 107(86%) detected
Lee [38]	191 gray scale images	2096 characters, 1916 segmented correctly, 181 errors occurred
Proposed algorithm	83 text images	89.08% precision, 80.15% recall

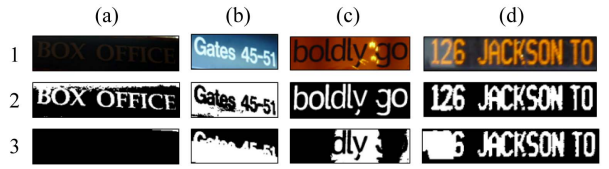

Figure 6. Comparison of the binarization results of our algorithm (2) with the Otsu's algorithm results (3) with 4 challenging images: low contrast (1-a), uneven illumination (1-b, 1-c), light reflections (1-d).

with four challenging images such as low contrast, distance variation, uneven illumination and light reflections. Nevertheless, there are various degradations with which our algorithm cannot cope, such as strong light reflections, and severe blurring and uneven illuminations, among others as in **Figures 7** and **8**.

However, this kind of situations has not been completely solved by any text extraction algorithms regardless on their complexity. Thus, it cannot be considered a major problem of our methodology.

6. Summary and Conclusions

In this paper, we have proposed a simple, fast, and accurate Text Extraction system to fit devices with low computational resources. Specifically, our algorithm has been developed on the context of language translation of scene text images on mobile phones. As it is known, these images suffer of multiple degradations, such as uneven illumination, reflections, blur, and so on, which cannot be handled by the existing simple algorithms. On the other hand, more complex algorithms can cope with some of these degradations, but they need unaffordable processing times on our scope. For these reasons, we developed a non-expensive algorithm which, for one part, provides accuracy higher than the existing simple algorithms, and

Figure 7. Examples of good binarization results of our algorithm in different situations.

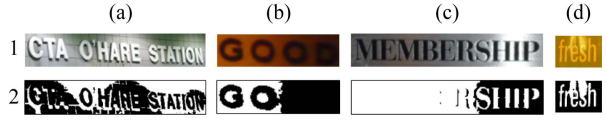

Figure 8. Examples of bad performance of our algorithm: severe uneven illumination (a, b), and severe light reflection (c, d).

for the other, performs even faster than them.

Even though using just the luminance component could accelerate the Text Extraction process, our algorithm uses all the color information of the image in order to perform the binarization, since it improves significantly the robustness of the system. Nevertheless, processing all this information adds a considerable complexity to our problem, so each step utilizes very simple computations to ensure that the quickness of the methodology is not compromised. First, taking into account that images contain a lot of redundant information which is not important for the text extraction, we quantize the image to reduce the colors from 24 to 12 bits. Then, we extract the Principal Color Components of the image from those colors with a larger number of occurrences. Second, since the texts are supposed to be readable, and mainly made up by two colors, we select to be the foreground-background main colors those components which maximize a combination of contrast and number of occurrences. Finally, the image is binarized, by clustering each pixel into its closer group, foreground or background.

We have compared our algorithm with the Otsu's one, one of the most well-known on the Text Extraction field, which can be considered as a candidate to be implemented on low computational resources devices, because of its simplicity, fast performance, and reasonably high accuracy on the most common situations. In the experimental results, our algorithm shows a 10% higher accuracy than Otsu's, and performs about two times faster. A detailed analysis of the results show the robustness of our algorithm against common degradations that cause to fail Otsu's, mainly uneven illuminations, blurring, and coincidence of different foreground-background colors in the gray scale domain.

Therefore, our algorithm is very appropriate for its implementation on devices with low computational resources. In addition, it works in a very short amount of time, so this shortage of computational resources is not a major problem. Furthermore, it shows high accuracy rates, which ensures its usefulness and its viability. Thus, the algorithm is already prepared to be part of a real TIE system as the one on which we based our efforts: a language translator of scene text on a mobile phone. Nevertheless, we will continue our research by trying to improve the accuracy of our methodology with non-expensive techniques, focusing on: using spatial information of the colors and their density on different areas, experimenting intensively with different distances to measure the contrast between colors, and finding an even better combination of color contrast and color importance on the Foreground-Background Centroid function, to separate the Foreground-Background Principal Components.

REFERENCES

[1] J. Liang, D. Doermann and H. P. Li, "Camera-Based Analysis of Text and Documents: A Survey," *International Journal on Document Analysis and Recognition*, Vol. 7, No. 2-3, 2005, pp. 84-104. doi:10.1007/s10032-004-0138-z

[2] C. Thillou and B. Gosselin, "Natural Scene Text Understanding," In: Croatia Ed., *Vision Systems*: *Segmentation and Pattern Recognition*, I-Tech Education and Publishing, 2007, pp. 307-332.

[3] K. Jung, "Text Information Extraction in Images and Video: A Survey," *Pattern Recognition*, Vol. 37, No. 5, 2004, pp. 977-997. doi:10.1016/j.patcog.2003.10.012

[4] A. Canedo-Rodríguez, J. Kim, S. Kim, *et al.*, "Simple and Efficient Text Localization for Compressed Images in Mobile Phone," *Submitted to IEEE Transaction on Image Processing*, 2009.

[5] N. Otsu, "A Threshold Selection Method from Gray Level Histograms," *IEEE Transactions on System, Man and Cybernetics*, Vol. 9, No. 1, 1979, pp. 62-66.

[6] C. Thillou, S. Ferreira and B. Gosselin, "An Embedded Application for Degraded Text Recognition," *Journal on Applied Signal Processing*, Vol. 2005, No. 13, 2005, pp. 2127-2135. doi:10.1155/ASP.2005.2127

[7] S. Messelodi and C. M. Modena, "Automatic Identification and Skew Estimation of Text Lines in Real Scene Images," *Pattern Recognition*, Vol. 32, No. 5, 1992, pp. 791-810. doi:10.1016/S0031-3203(98)00108-3

[8] H. Li and D. Doermann, "Text Enhancement in Digital Video Using Multiple Frame Integration," *Proceedings of ACM International Conference on Multimedia*, 1999, pp. 19-22.

[9] A. Zandifar, R. Duraiswami and L. S. Davis, "A Video-Based Framework for the Analysis of Presentations/Posters," *International Journal on Document Analysis and Recognition*, Vol. 7, No. 2-3, 2005, pp. 178-187. doi:10.1007/s10032-004-0137-0

[10] W. Niblack, "An Introduction to Image Processing," Prentice-Hall, Upper Saddle River, 1986, pp. 115-116.

[11] J. Sauvola and M. Pietikainen, "Adaptive Document Im-

age Binarization," *Pattern Recognition*, Vol. 33, No. 2, 2000, pp. 225-236. doi:10.1016/S0031-3203(99)00055-2

[12] J. Gllavata, R. Ewerth and B. Freisleben, "Finding Text in Images via Local Thresholding," *Proceedings of IEEE Symposium on Signal Processing and Information Technology*, Siegen, 14-17 December 2003, pp. 539-542.

[13] I.-J. Kim, "Multi-Window Binarization of Camera Image for Document Recognition," *Ninth International Workshop on Frontiers in Handwriting Recognition*, Inzisoft Co. Ltd., 26-29 October 2004, pp. 323-327.

[14] Y. Du, C.-I. Chang and P. D. Thouin, "Unsupervised Approach to Colour Video Thresholding," *Optical Engineering*, Vol. 43, No. 2, 2004, pp. 282-289.

[15] J. Kim, S. Park and S. Kim, "Text Locating from Natural Scene Images Using Image Intensities," *Proceedings of International Conference on Document Analysis and Recognition*, Seoul, August 31-September 1 2005, pp. 655-659.

[16] R. Lienhart and A. Wernicke, "Localising and Segmenting Text in Images, Videos and Web Pages," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 12, No. 4, 2002, pp. 256-268. doi:10.1109/76.999203

[17] H. Hamza, E. Smigiel and A. Belaid, "Neural Based Binarisation Techniques," *Proceedings of International Conference on Document Analysis and Recognition*, Seoul, August 31-September 1 2005, pp. 317-321.

[18] Z. Saidane and C. Garcia, "Robust Binarization for Video Text Recognition," *Ninth International Conference on Document Analysis and Recognition*, Vol. 2, 2007, pp. 874-879.

[19] K. Sobottka, H. Bunke and H. Kronenberg, "Identification of Text on Colored Book and Journal Covers," *Proceedings of the Fifth International Conference on Document Analysis and Recognition*, Bangalore, 20-22 September 1999, pp. 57-62.

[20] T. Perroud, K. Sobottka, H. Bunke and L. Hall, "Text Extraction from Colour Documents—Clustering Approaches in Three and Four Dimensions," *Proceedings of International Conference on Document Analysis and Recognition*, 10-13 September 2001, pp. 937-941. doi:10.1109/ICDAR.2001.953923

[21] D. Comaniciu, "Nonparametric Robust Methods for Computer Vision," Ph.D. Thesis, Rutgers University, Newark, 2000.

[22] D. Lopresti and J. Zhou, "Locating and Recognising Text in WWW Images," *Information Retrieval*, Vol. 2, No. 2-3, 2000, pp. 177-206. doi:10.1023/A:1009954710479

[23] B. Wang, X.-F. Li, F. Liu and F.-Q. Hu, "Colour Text Image Binarisation Based on Binary Texture Analysis," *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, Shanghai, 17-21 May 2004, pp. 585-588.

[24] A-N. Lai and G. Lee, "Binarization by Local k-Means Clustering for Korean Text Extraction," *IEEE International Symposium on Signal Processing and Information Technology*, Gwangju, 16-19 December 2008, pp. 117-122.

[25] C. Thillou and B. Gosselin, "Combination of Binarization and Character Segmentation Using Color Information," *Proceedings of the Fourth IEEE International Symposium on Signal Processing and Information Technology*, Mons, 18-21 December 2004, pp. 107-110. doi:10.1109/ISSPIT.2004.1433699

[26] J. Gao, J. Yang, Y. Zhang and A. Waibel. "Text Detection and Translation from Natural Scenes," 2001. http://reports-archive.adm.cs.cmu.edu/anon/2001/abstracts/01-139.html

[27] J. Park, G. Lee, A.-N. Lai, E. Kim, J. Lim, S. Kim, H. Yang and S. Oh, "Automatic Detection and Recognition of Shop Name in Outdoor Signboard Images," *IEEE International Symposium on Signal Processing and Information Technology*, Gwangju, 16-19 December 2008, pp. 111-116. doi:10.1109/ISSPIT.2008.4775652

[28] P. Berkhin, "Survey of Clustering Data Mining Techniques," Technical Report, Accrue Software, 2002.

[29] O. D. Trier and T. Taxt, "Evaluation of Binarization Methods for Document Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 3, 1995, pp. 312-315. doi:10.1109/34.368197

[30] S. A. Mingoti and J. O. Lima, "Comparing SOM Neural Network with Fuzzy c-Means, K-Means and Traditional Hierarchical Clustering Algorithms," *European Journal of Operational Research*, Vol. 174, No. 3, 2006, pp. 1742-1759. doi:10.1016/j.ejor.2005.03.039

[31] A. Canedo-Rodriguez, S. H. Kim, J. H. Kim and Y. Blanco-Fernandez, "English to Spanish Translation of Signboard Images from Mobile Phone Camera," *Southeast Conference*, Atlanta, 5-8 March 2009.

[32] M. Celenk, "A Color Clustering Technique for Image Segmentation," *Graphical Models Image Process*, Vol. 52, No. 3, 1990, pp. 145-170.

[33] R. M. Haralick and L. G. Shapiro, "Image Segmentation Techniques," *Computer Vision Graphics Image Process*, Vol. 29, No. 1, 1985, pp. 100-132. doi:10.1016/S0734-189X(85)90153-7

[34] B. Schacter, L. Davis and A. Rosenfeld, "Scene Segmentation by Cluster Detection in Color Space," University of Maryland, College Park, 1975.

[35] A. Sarabi and J. K. Aggarwal, "Segmentation of Chromatic Images," *Pattern Recognition*, Vol. 13, No. 6, 1981, pp. 417-427. doi:10.1016/0031-3203(81)90004-2

[36] M. Junker and R. Hoch, "On the eEvaluation of Document Analysis Components by Recall, Precision, and Accuracy," *Proceedings of the Fifth International Conference on Document Analysis and Recognition*, 1999, pp. 713-716.

[37] H.-K. Kim, "Efficient Automatic Text Location Method and Content-Based Indexing and Structuring of Video Database," *Journal of Visual Communication and Image Represent*, Vol. 7, No. 4, 1996, pp. 336-344. doi:10.1006/jvci.1996.0029

[38] C. M. Lee and A. Kankanhalli, "Automatic Extraction of Characters in Complex Images," *International Journal of Pattern Recognition Artificial Intelligence*, Vol. 9, No. 1, 1995, pp. 67-82. doi:10.1142/S0218001495000043

