2011 International Conference on Document Analysis and Recognition

Snap and Translate Using Windows Phone

Jun Du, Qiang Huo, Lei Sun, Jian Sun
Microsoft Research Asia, Beijing, P. R. China
Emails: {jundu, qianghuo, v-lesun, jiansun}@microsoft.com

***Abstract*—We have developed a prototype of a mobile app called "Snap and Translate" on "Windows Phone 7". A person who is reading an English menu/sign and wants a Chinese translation of an English word or phrase or paragraph can use a Windows Phone to snap an image of the text, tap the word or swipe the phrase or circle the paragraph with a finger, and get a Chinese translation displayed on the screen of the phone. This is enabled by seamless integration of three Microsoft technologies: intelligent text extraction, OCR, and machine translation based on a client-plus-cloud architecture. The current prototype also supports Chinese OCR plus Chinese-to-English translation. In this paper, we highlight the UI design of the system and the corresponding user-intention guided text extraction approach to achieving a compelling user experience.**

***Keywords*-Camera-based OCR, mobile translation, user interface, text detection, augmented reality**

I. Introduction

For a language learner or a tourist traveling in a foreign country, OCR translation using a Smartphone with a camera will allow the person to read a menu/sign in a foreign language very conveniently. This has attracted much research activities in the past decade (e.g., [6], [17], [13], [9], [4]). Given the good quality of the built-in camera and the enough computational capability of Smartphones on the market [14], several commercial mobile OCR translation apps have been released, which can be divided into two broad classes: client only app and client-plus-cloud app. For the former type of apps, all the processing is done on the smartphone itself. Pleco [11] and Word Lens [15] are two popular examples. Pleco is a Chinese Dictionary app which is designed for language learning. Word Lens is an iPhone app which can translate a sign between English and Spanish with a live augmented reality (AR) overlay of the translation result. In both cases, only word by word translation (or dictionary lookup) is performed. For client-plus-cloud apps, memory and computation demanding tasks such as OCR and translation are done typically in the cloud, while other functions are run on the client-side. By definition, network access should always be available to enable a client-plus-cloud implementation. The translation feature in Google Goggles [5] is such a representative, which allows a user to translate sentence(s) by taking a picture and drawing a precise bounding box of the intended text, because more advanced OCR and translation technologies can be implemented in the cloud.

Recently, we have also developed a prototype of a mobile app called "Snap and Translate" on "Windows Phone 7" (WP7) based on a client-plus-cloud architecture. In one of the operation modes, a user can use one of three natural gestures, namely *tap* a word or *swipe* a phrase or *circle* a paragraph with a finger on the captured text image to indicate his/her intention explicitly. The intended text image patch will then be extracted automatically on the phone and sent to the cloud for recognition and translation. This makes our system different from the aforementioned apps with the following benefits: 1) a more accurate text extraction with the help of user's intention, 2) less computations on the phone without processing the whole image, therefore longer battery time, and 3) less network traffic and smaller latency because only a small image patch is sent over the network.

The remainder of the paper is organized as follows: In Section II, we give an overview of our OCR translation system and highlight its UI design. In Section III, we describe our user-intention guided text extraction approach. In Section IV, we report experimental results. Finally we conclude the paper in Section V.

II. System Overview

The overall system architecture is illustrated in Fig. 1. This is a bidirectional OCR translation system supporting both English and Chinese. In the following, we describe the UI design for different operation modes on the client side, and the web services running in the cloud, respectively.

A. Operation modes and UI design

To capture text, one can use both still-image mode and video mode. In still-image mode as illustrated at the top-left part of Fig. 1, one can take a still image by using the camera on the phone or load a pre-captured image from the picture gallery on the phone. The user can then use one of the three finger gestures to indicate the intended text. The first gesture is *tap*, with which one can select a single English word or a Chinese character. As illustrated in Fig. 2, the user only needs to tap anywhere inside the word or character with a finger. The second gesture is *swipe*, as illustrated in Fig. 3, with which one can select a short phrase or a single text line by swiping across the intended text with a finger. In both cases, a slightly modified version of the *Shift* technique [12] is used to help the user locate the intended target points more easily and precisely. Upon a finger tap, a

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.166

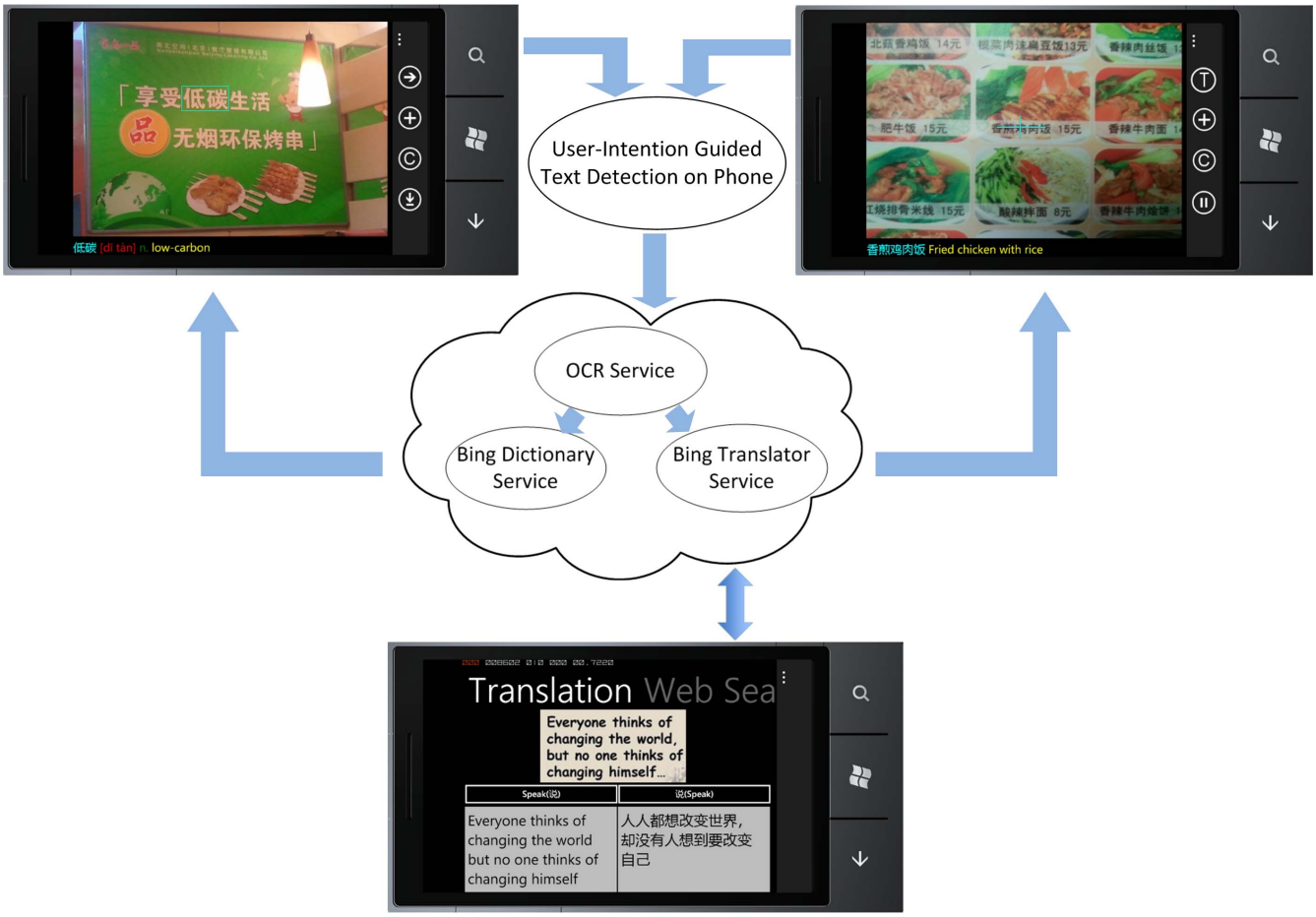

Fig. 1. Overall system architecture.

callout containing a copy of the occluded area with a pointer showing the finger selection point is displayed at a location (e.g., bottom-left corner of the image in Figs. 2&3) without occluding the intended text. The tap gesture is effected by allowing the user to make vertical corrective movements by keeping the finger on the display until the pointer is over the target to lift the finger for target selection. The swipe gesture is effected by allowing the user to make vertical corrective movements first to decide the starting target point, followed by horizontal corrective movements until the pointer is over the ending target point to lift the finger for target selection. The third gesture is *circle*, that allows the user to circle the intended text area, which may contain a paragraph with multiple text lines as illustrated at the bottom part of Fig. 1. Given the indicated user intention, a user-intention guided text extraction algorithm as elaborated in Section III is used to draw automatically the final bounding box of the intended text as illustrated in Fig. 2, Fig. 3 and Fig. 4.

As for video mode, an example is given at the top-right part of Fig. 1. A cross mark is used to indicate the user's intention whose horizontal line can be adjusted by horizontal corrective movements of a finger anywhere on the display to cover the intended word or phrase. This horizontal line can also guide the user to hold the camera appropriately to avoid the possible rotational distortions. Alternatively, the user can also switch the cross mark to a bounding box, whose width and height can be adjusted with corrective finger movements. This bounding box can be used to indicate the intention of capturing a paragraph with multiple text lines. Given the indicated user intention, the user can tap a "translation" (T) button on the app bar to capture the current video frame as a still image, which is then processed almost the same as in the still-image mode.

It is noted that both operation modes are useful and can deliver desirable user experience depending on the scenarios and the user's preference. The unified framework of user-intention guided text extraction for both modes is shown in Fig. 5, which is elaborated in Section III.

B. Cloud services

After the image patch with intended text is sent to the cloud from the phone client, OCR and translation services

Fig. 2. Word dictionary look-up with a tap gesture.

Fig. 3. Phrase translation with a swipe gesture.

Fig. 4. General translation with a circle gesture.

are invoked. A Microsoft OCR engine is used, which supports multiple languages including but not limited to English and Chinese. Users can experience English OCR capability by trying out Bing Vision feature in released Bing for iPhone app [2]. For translation, if the OCR result is a word, then we will invoke the Bing Dictionary service [1] to perform word dictionary look-up. Otherwise, the Bing Translator service [3] is used to get the text-to-text translation result. A TTS (Text to Speech) function to read out both OCR and translation results is also provided as illustrated at the bottom part of Fig. 1.

III. User-Intention Guided Text Extraction

The flowchart of our user-intention guided text extraction approach is illustrated in Fig. 5. In still-image mode, after loading an image, it is down-sampled to a resolution of 640×480 pixels to improve the efficiency of preprocessing. In video mode, after image capturing and downsampling, a

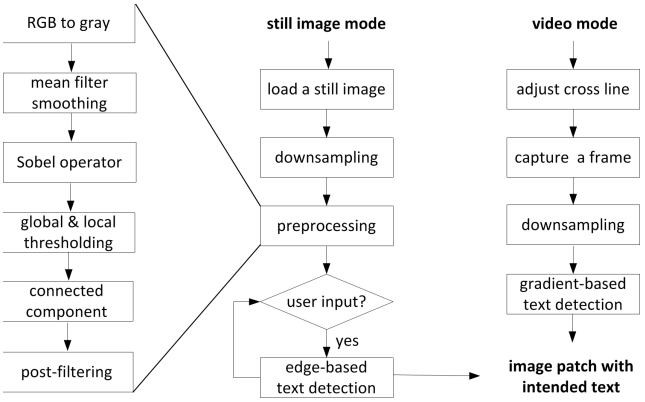

Fig. 5. Flowchart of user-intention guided text extraction.

gradient-based text detection is performed. The final output of the text extraction is an image patch with intended text, which will be sent to the cloud. If this image patch is small (e.g., width or height is less than 40 pixels), the corresponding original image patch before downsampling will be used. In the following, we elaborate on the modules of preprocessing, edge-based text detection, and gradient-based text detection.

A. Preprocessing

The goal of preprocessing in still-image mode is to obtain an edge map for subsequent text detection with user's input. The procedure is shown at the left part of Fig. 5, where each step is optimized computationally to make the preprocessing very efficient. First, the color image is converted to gray scale image. Then, a 3×3 mean filter is applied for image smoothing and noise removal. A Sobel operator is used to calculate the gradient for each pixel. The gradient magnitude is approximated as the $\max$ of the horizontal and vertical gradients. An edge map is initialized based on gradient features by using the non-maximum suppression to remove most non-edge pixels. Then the integral image of gradient features is precalculated for global and local thresholding. The global thresholding is performed to remove non-edge pixels with very small gradient magnitude by using a conservative global threshold. For local thresholding, we use hysteresis thresholding, where two (high and low) local thresholds are calculated from the integral image. Consequently, both strong and weak edges can be preserved while those non-edge pixels nearby an edge can be removed. After thresholding, a binary image morphology operation called *bridge* is applied to make the edge map more connective. Connected component analysis is then conducted using a very efficient algorithm [16]. Finally, post-filtering is applied to remove those non-text connected components by using geometry information such as the area and aspect ratio of each connected component.

Table I
PROCEDURE OF EDGE-BASED TEXT DETECTION

Input:	p_l, p_r, p_t, p_b are left, right, top and bottom positions of user's input, e.g., the bounding box of the cyan circle curve in Fig. 4. Tap ($p_\mathrm{l} = p_\mathrm{r}$ and $p_\mathrm{t} = p_\mathrm{b}$) and swipe ($p_\mathrm{t} = p_\mathrm{b}$) gestures are two special cases.
Output:	bb_t, bb_b, bb_l, bb_r are the corresponding top, bottom, left, and right positions of the final bounding box.
Predefined:	gr is the ratio of minimum gap between words or characters to height. $gr = 0.2$ for English word, and $gr = 0$ for Chinese character. $L = 32$ is a fixed length of a horizontal line segment.
Initialization: $bb_\mathrm{t} = p_\mathrm{t}$; $bb_\mathrm{b} = p_\mathrm{b}$; $bb_\mathrm{l} = p_\mathrm{l}$; $bb_\mathrm{r} = p_\mathrm{r}$	
Step 1: Localization of top and bottom positions Move a horizontal line segment, with the length set as the max of $p_\mathrm{r} - p_\mathrm{l}$ and L, starting from the top point $((p_\mathrm{l} + p_\mathrm{r})/2,\ p_\mathrm{t})$ or bottom point $((p_\mathrm{l} + p_\mathrm{r})/2,\ p_\mathrm{b})$, vertically upwards (decrease bb_t) and downwards (increase bb_b), respectively, until non-edge horizontal line segment is encountered, which contains no edge pixels of scanned connected components.	
Step 2: Localization of left and right positions Move a vertical line segment, with the length set as the $bb_\mathrm{b} - bb_\mathrm{t}$ starting from the left point $(p_\mathrm{l},\ (bb_\mathrm{b} + bb_\mathrm{t})/2)$ or right point $(p_\mathrm{r},\ (bb_\mathrm{b} + bb_\mathrm{t})/2)$, horizontally leftwards (decrease bb_l) and rightwards (increase bb_r), respectively, until more than consecutive $(bb_\mathrm{b} - bb_\mathrm{t}) * gr$ non-edge vertical line segments are collected.	
Step 3: Refinement of top and bottom positions Due to the case of word with ascending/descending part, bb_t and bb_b are probably the baseline positions, which are not the desired border line positions. Again, we move a horizontal line segment, with the length set as $bb_\mathrm{r} - bb_\mathrm{l}$, starting from the final positions in **Step 1**, vertically upwards (decrease bb_t) and downwards (increase bb_b), respectively, until non-edge horizontal line segment is encountered.	

B. Edge-based text detection

Whenever a user input is given in still-image mode, an edge-based text detection using previously obtained edge map is triggered to locate the bounding box of the intended text area. The procedure is listed in Table I. Because the main processing jobs have been done in the preprocessing stage, this text detection algorithm is very efficient.

C. Gradient-based text detection

In video mode, the intended text has been indicated by the user when an image is captured as a video frame, which makes processing the whole image unnecessary. Our gradient-based text detection approach uses a procedure similar to the one in Table I with only two main differences. First, gradient magnitude feature is used, which means only the first three modules of preprocessing in Fig. 5 are adopted. Second, the criterion of non-text horizontal/vertical line segment, which corresponds to the non-edge horizontal/vertical line segment in Table I, is changed to that the max gradient magnitude on the current line segment is less than an adaptive threshold thr. In **Step 1**, thr is related to the mean value of gradient magnitudes accumulated from scratch to the current horizontal line segment. In **Step 2** and **Step 3**, thr is fixed as the final value in **Step 1**.

This simple method works well for the text area with clean background. But compared with edge-based text detection, it is less robust to noise and complex background. Furthermore, the precise localization of top and bottom positions in **Step 1** is very important for the following steps and often more difficult than the localization of left and right positions. To address these issues, we propose to use a more robust localization method based on a convex hull technique in **Step 1**. As illustrated in Fig. 6, in this method, the difference of maximum and minimum gradient magnitude values on each horizontal line segment is extracted as a feature. As the line segment moves upwards and downwards starting from an initial point in black color, a feature profile in blue color can be formed incrementally. Obviously, the top and bottom positions correspond to the positions with two steepest slopes of the feature profile. Given the feature profile, a convex hull in red color can also be constructed incrementally and efficiently [8], from which the final top and bottom positions in green color can be localized accordingly with an appropriate stopping criterion.

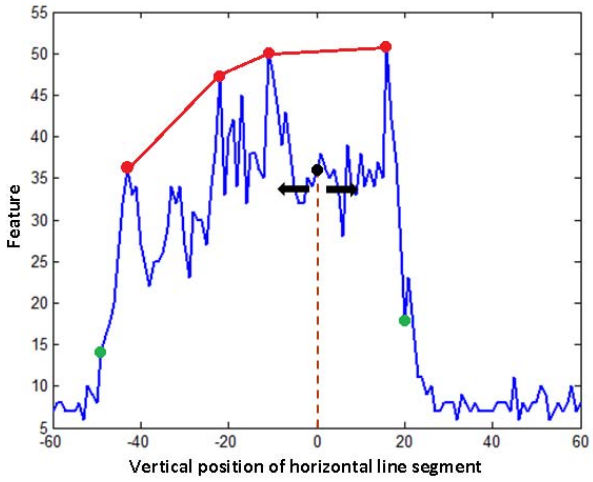

Fig. 6. A robust localization method using convex hull.

IV. EXPERIMENTS AND RESULTS

We implemented the client-side software of our "Snap and Translate" app using C# programming language with the Silverlight framework [10]. The client-cloud communications were implemented in WCF (Windows Communication Foundation) SOAP (Simple Object Access Protocol) and REST (Representational State Transfer). We tried our app on three models of WP7 Smartphones, namely Samsung Focus, HTC HD7, and LG Quantum [14] and found that they work equally well.

To evaluate our user-intention guided text extraction algorithm, we used the publicly available ICDAR 2003 dataset [7], which contains 258 images in the training set and 251 images in the test set with full-color and varying size from 307×93 to 1280×960 pixels. For traditional evaluation of text extraction, three metrics, namely precision, recall

Table II
THE PRECISION OF OUR USER-INTENTION GUIDED TEXT EXTRACTION ALGORITHM USING TAP AND SWIPE GESTURES RESPECTIVELY ON A SUBSET OF IMAGES IN ICDAR 2003 DATASET.

Gestures	Tap	Swipe
Precision	0.72	0.77

Fig. 7. Examples of tap (top row) and swipe (bottom row) based text detection results from the ICDAR 2003 dataset.

and F-measure are typically used [7]. However, for our experiments, the recall is always 100% because of an explicit indication of user's intention. We only need to calculate the precision measure. Our experiments are conducted as follows: 1) Volunteers are asked to make gestures (tap and swipe) to indicate the intended words in images randomly selected from ICDAR 2003 dataset, and finally 575 words are collected; 2) Our algorithm will output a rectangle bounding box for each word; 3) For each word, the match between the estimated and the ground-truth bounding boxes is defined as the area of intersection divided by the area of the minimum bounding box containing both rectangles; 4) The precision is the average of matches for all collected words.

The precision scores of our user-intention guided text extraction algorithm using tap and swipe gestures respectively are listed in Table II. It is noted that due to the lack of ground truth for multi-word phrases in ICDAR 2003 dataset, the swipe result is also based on words. It is observed that swipe outperforms tap, which is very reasonable because more information is provided by users with swipe gestures. Overall, our method achieves promising performance on ICDAR 2003 dataset where there are many challenging cases. A few examples are given in Fig. 7. Our algorithm can achieve almost perfect results in the cases with simple background (the first column). It is also robust to some challenging cases (the second column).

V. CONCLUSION AND FUTURE WORK

Currently, our "Snap and Translate" app works well for scenarios such as translation of text in printed newspapers, magazines, books, menus, and signboards with simple backgrounds under normal lighting conditions and careful image capturing. New researches are needed to improve text extraction and OCR engine in order to deal with more challenging scenarios such as text with complex backgrounds and uncommon font types under non-uniform lighting conditions and casual image capturing with large geometry distortions.

ACKNOWLEDGMENT

We thank our colleagues, Matt Scott and Gang Chen for their help in using Bing Dictionary web service, and Ivan Stojiljkovic, Magdalena Vukosavljevic and David Nister for their help in OCR engines.

REFERENCES

[1] Bing Dictionary, http://dict.bing.com.cn/api/service.svc

[2] Bing for iPhone, http://itunes.apple.com/app/bing/id345323231#

[3] Bing Translator, http://api.microsofttranslator.com/V2/Soap.svc

[4] V. Fragoso, S. Gauglitz, S. Zamora, J. Kleban, M. Turk "TranslatAR: a mobile augmented reality translator," *Proc. IEEE Workshop on Applications of Computer Vision (WACV) 2011*, pp.497-502.

[5] Google Goggles, http://www.google.com/mobile/goggles/#text

[6] I. Haritaoglu, "Scene text extraction and translation for handheld devices," *Proc. CVPR-2001*, pp.II-408-413.

[7] S. M. Lucas *et al.,* "ICDAR 2003 robust reading competitions: entries, results, and future directions," *Int. J. Document Analysis and Recognition*, Vol. 7, Nos. 2-3, pp.105-122, 2005.

[8] A. Melkman, "On-line construction of the convex hull of a simple polyline," *Information Processing Letters*, Vol. 25, No. 1, pp.11-12, 1987.

[9] H. Nakajima, Y. Matsuo, M. Nagata, K. Saito, "Portable translator capable of recognizing characters on signboard and menu captured by built-in camera," *Proc. ACL Interactive Poster and Demonstration Sessions*, 2005, pp.61-64.

[10] C. Petzold, *Programming Windows Phone 7*, Microsoft Press, 2010.

[11] Pleco, http://www.pleco.com/

[12] D. Vogel, P. Baudisch, "Shift: a technique for operating pen-based interfaces using touch," *Proc. CHI-2007*, pp.657-666.

[13] Y. Watanabe, K. Sono, K. Yokomizo, Y. Okada, "Translation camera on mobile phone," *Proc. ICME-2003*, pp.II-177-180.

[14] Windows Phone 7 models vs. other Smartphones, http://www.pcworld.com/zoom?id=207502&page=1&zoomIdx=2

[15] Word Lens, http://questvisual.com/

[16] K. Wu, E. Otoo, K. Suzuki, "Optimizing two-pass connected-component labeling algorithms," *Pattern Anal. Applic.*, Vol 12, pp.117-135, 2009.

[17] J. Yang, J. Gao, Y. Zhang, A. Waibel, "Towards automatic sign translation," *Proc. HLT-2001*, 6 pages.