Smart Implementation of Text Recognition (OCR) for Smart Mobile Devices

Ondrej Krejcar
Department of Information Technologies,
Faculty of Informatics and Management,
University of Hradec Kralove,
Hradec Kralove, Czech Republic
ondrej.krejcar@remoteworld.net

Abstract **–The paper deal with a development of a mobile application for capturing digital photography and its subsequent processing by OCR (Optical Character Recognition) technologies.The developped solution adds to existing Smart Device a capability of a virtual keyboard to which it is possible to transfer recognized text for further work in SMS or text editor. For example, based on the limitation of mobile devices it is mainly targeting at short text sections (internet references, complex adresses, etc.). The accent is targeted on the simple, fast and intuitive working with a mobile device. Practical realization is verified at several Smart Devices with Windows Mobile OS.**

Keywords – OCR; Smart Device; Windows Mobile; Image Processing; Virtual Keyboard

I. INTRODUCTION

The Smart Phones, such as cell phones and PDA (Personal Digital Assistant), especially MDA (Mobile Digital Assistant) are the phenomenon nowadays. The number of cell phone users over 16 years old in the Czech Republic for the year 2009 climbed up to 91%. For the population in the age group from 16 to 54 years the number is equal to 98% [1]. A great boom in the field of cell phones and their performance was caused by using the OS (Operating System), such as Symbian, Android or Windows Mobile. Many of these devices use large colorful displays with touch screen and fast 32bit CPU. Moreover, the GSM module is usually integrated within the standard PDA together with WiFi module. The result is the incorporation of cell phones and PDA as Smart Phones. Based on the usage of efficient 32bit CPUs it is possible to develop power applications for computation.

The primary input system of these devices is the keyboard in a classic "physical" design or in the form of virtual keyboard on a display in the case of touch screen. These types of keyboards provide a comfortable method of information inscription. Nevertheless, the typing is approx. 4x slower than in the case of computer keyboards [2]. However, this typing speed may be insufficient if we would like to use a Smart Device as a tool for fast information recording (e.g. business card copying or copying parts of text). Most commonly integrated CCD (In many PDAs, more precisely in cell phones the cheaper CMOS sensors are used) chips enables the photographing or recording of a video-sequence. Therefore, it is a convenient and instant way of capturing information. Moreover, if this information is time-limited (e.g. it must return within certain time limit or it is only displayed for short time period) then it is the only method.

Nevertheless, sometimes there is a need to further process this captured text. The text retyping from these images is lengthy. Furthermore, if it is necessary to retype using the PDA it should be accounted for switching often between an application with displayed image and the text editor.

In these cases the usage of OCR (Optical Character Recognition) technology is the best solution. The first mobile application OCR was released to the market already in 2002 [3]. Certain factors complicate the usage of OCR in PDA which mostly originated from the low quality of copies acquired by CCM (Compact Camera Module, module with integrated CCD (Charge Coupled Device) or CMOS (Complementary Metal Oxide Semiconductor) sensor, simple optics and electronics). Finally, it is necessary to mention that the common source for OCR application is a scanner.

A PDA which is supplied by OCR has many options in a way of utilization. If the user notices an URL address in some printed document, he can look at it by taking a picture which consequently opens the link in a browser. After this picture the business card with user's data is saved into contacts, etc.

The problem we would like to deal with in this paper is based on a development of mobile OCR application for current Smart Phones at Windows Mobile platform. Such application is necessary for solving of problems mentioned before with the goal in development of virtual keyboard with embedded OCR engine.

Firstly an evaluation of existing solutions will be made in (Section II).

II. EXISTING OCR ENGINES FOR MOBILE DEVICES

The accuracy of OCR depends mainly on the quality of recognizable under layer. The most common usage of OCR on scanned documents achieves quite satisfactory results. Using of OCR in PDA with CCM as a data source recognizer carries number of problems [4], especially:

 ISBN: 978-1-61208-224-0

- Relatively low computational performance (Usually 1/10 of PC performance)
- Low quality of images for OCR (Generally meant as low resolution, blurring, background noise, anomalies caused by compression, etc.)
- Tilt (perspective deformation), skew and rotation
- Incoherent lighting and shadows

Mainly due to these complications is OCR in PDA limited to just small parts of text. Therefore, the insufficient quality of acquired images is compensated by the size proportion of symbols in the overall resolution. The existing applications may be good examples, because they are usually specialized on business card scanning.

A. Existing mobile applications

1) Nokia Multiscanner

Nokia Multiscanner [5] is a freeware application designed for cell phones with Symbian OS. The application supports picture taking and consequently sending it through MMS, Bluetooth or via infrared. It is possible to transfer the image into a text and save it and at the same time the selection of certain area can be made by dragging. Another possibility is to send the image for business card recognition. This option automatically recognizes contact details on the business card and fills in the details for adding a new contact. The OCR engine supports post-processing on the basis of language dictionaries (Technology for replacement of recognized words by words from a dictionary according to their relevance), including the Czech language.

However this solution do not support real virtual keyboard nor clipboard Copy/Past features. Also only Symbian OS is supported.

2) CameraDictionary OCR for Moto

An application [6] for cell phones with Android, Symbian and Windows Mobile systems. It operates on the basis of recorded text recognition and its immediate translation to another language. Even though, this recorded language is available in Chinese or English, the translation is extended by couple of other languages. Furthermore, it enables the text recording with consequent signing of the translated text or so called “Video” regime during which the cursor appears on the screen. The text below the cursor is immediately translated.

However, the main disadvantages are the price and the necessity of internet connection when used.

3) CamCard - Business Card Reader

CamCard [7] is an application specialized on reading business cards. It is targeted at cell phones which run on OS Android, iOS (OS of iPhone cell phones), or Windows Mobile and BlackBerry phones. Furthermore, the CamCard is an extensively automated business card reader with detection of a rotation and a language. The whole recording takes just couple of presses.

The main disadvantage is the narrow specialization on business cards and its price.

4) Babel Reader-LE

Babel Reader-LE [8] is a particular version of Babel Reader for Windows Mobile distributed as a freeware. It enables capturing of an image and subsequent storing of this image in a form of text. Babel Reader-LE is a very simple application. Moreover, it is possible to adjust the captured image before the actual recognition e.g. by background noise removal.

As in the case of Nokia solution a clipboard and keybard option is not possible.

B. Problems of Existing Mobile OCR Solutions

Nokia Multiscanner is the closest application to the one we needed. However, it is designed only for OS Symbian. CameraDictionary OCR and CamCard are commercial applications which are very specialized and not free. Finally, the last mentioned application called Babel Reader was only invented for text recognition. The selection of these applications with OCR for cell phones is significantly limited and the broader application with OCR which would work as an alternative for a virtual keyboard is still missing.

These reasons lead us to develop a new application which is described in this article. We expect to develop a solution which fills a space on current market.

C. Selection of OCR engine

Due to the extent of this application, it is planned to use the existing OCR engine. Following types of engines were chosen as the most suitable:

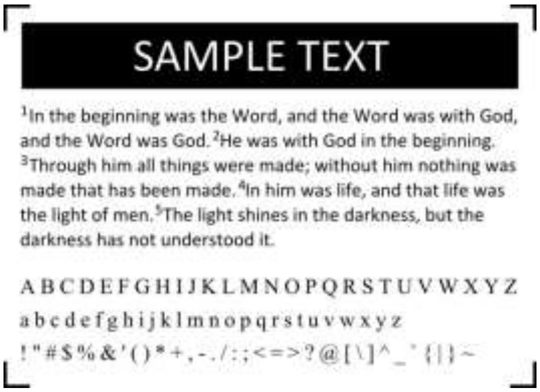
- **Tesseract OCR** [9] – OCR Engine developed by HP Company in since 1985 until 1995. Nowadays, it is being improved by Google. It is offered in C/C++ language.
- **Ocrad** [10] – another open-source OCR engine. One of his main advantages is mainly an automatic transformation of an input image. It does not accomplish post-processing on the basis of language dictionaries. It is written in C/C++ language.
- **Puma.NET** [11] – an engine for implementation in C# projects with .NET framework.
- **ABBYY Mobile OCR Engine** [12] – a commercial engine used here just for comparison of results. Not available for end users, tested by ABBYY FineReader Online service.

A script in PHP language was created in order to accomplish an objective comparison of recognition accuracy. This script is not included within the topic of this article and therefore will not be described in the text. The accuracy of the match is calculated by following formula.

The Greek letter ω is going to represent the number of symbols in a reference text and ω_{err} is the number of errors (substituted, missing symbols or additional symbols). Then the accuracy of match γ_{acc} is defined as:

$$\gamma_{acc} = \frac{\omega - \omega_{err}}{\omega} 100 \quad [\%] \qquad (1)$$

In order to identify the accuracy of recognition, the reference text [Fig. 1] was used.


SAMPLE TEXT

[1]In the beginning was the Word, and the Word was with God, and the Word was God. [2]He was with God in the beginning. [3]Through him all things were made; without him nothing was made that has been made. [4]In him was life, and that life was the light of men. [5]The light shines in the darkness, but the darkness has not understood it.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
! " # $ % & ' ( ) * + , - . / : ; < = > ? @ [\ ] ^ _ ` { | } ~

Figure 1. Reference image.

Furthermore, this sample was photographed by Canon PowerShot S3 IS and MDA HTC Touch 2. Consequently, this sample was transferred back to the text form using the above mentioned OCR and compared to the reference text. The accuracy of the match is expressed in percentages in [

]. Column “original” mean the reference text in form of an image.

TABLE I. COMPARISON OF OCR ENGINE ACCURACY

OCR	Original	Canon	HTC
Tesseract	89,78 %	94,72 %	85,25 %
Ocrad	93,30 %	92,71 %	74,36 %
Puma.NET	92,05 %	90,41 %	25,55 %
ABBYY	95,96 %	94,41 %	87,77 %

The comparison shows that the most exact engine is ABBYY Mobile OCR Engine. At the same time it may be noticed that the decreasing quality of sample results in a gradual increase in number of errors. The most significant is the rapid increase of errors when using the Puma.NET engine. In this case, the application was able to correctly recognize approximately one quarter of the text from an image taken by HTC Touch 2. On the other hand, the least sensitive engine considering the quality is Tesseract OCR. Even though, the Ocrad does not realize post-processing on the basis of language dictionaries, it was proven to be very precise.

The most suitable from the point of the evolving application would be to use the OCR engine Puma.NET. This engine is designed for .NET framework environment, contains an analysis of a document structure, writing styles and filter (It accomplishes filtering out of dot-matrix anomalies and contains a regime for processing documents which are sent through fax) [11]. Nevertheless, due to the insufficient recognition accuracy of low quality images, its use has to be denied. On the other hand, it is necessary to choose an engine with good results even for bad resolution photographs. Therefore, the Tesseract OCR engine will be used for this purpose.

III. IMPLEMENTATION

The programmating language C# with a connection to the developing environment Microsoft Visual Studio 2008 was designed for the development of the described application. Microsoft Windows Mobile 5.0 runs as the end platform. Therefore, the application should function well on a PDA with this OS or a higher type of OS.

The application is composed of couple components (by component we mean an incased object (UML expression), not the Visual Studio component), whose relationship is presented on a components’diagram in UML [13] on [Fig. 2].

The application consists of 5 parts of basic components as it may be noticed in this diagram. These are *CameraControl*, *ImageProcessing*, *MainApplication*, *OCRInterface* and *OCR*. The image data are obtained by CCM while considering the data flow. Consequently, the components *CameraControl* and *ImageProcessing* manage their accuracy and new modifications. After the start-up of the *CameraControl* application, it portrays the image data in preview regime on the device’s display (*ImageProcessing* is spanned in this stage – it is in bypass regime). After taking a picture and pressing the touchscreen, the *MainApplication* component records this event and ends the preview. A static image is on the output of *CameraControl*. This image is adjusted by *ImageControl* component and portrayed. At this moment, the user is able to choose the words for OCR processing. *MainApplication* sends the coordinates of the selection to the *ImageControl* component which crops the image and forwards it further through the *OCRInterface* to the OCR engine. Finally, the recognized data may be saved inside a folder or copied to a Windows folder.

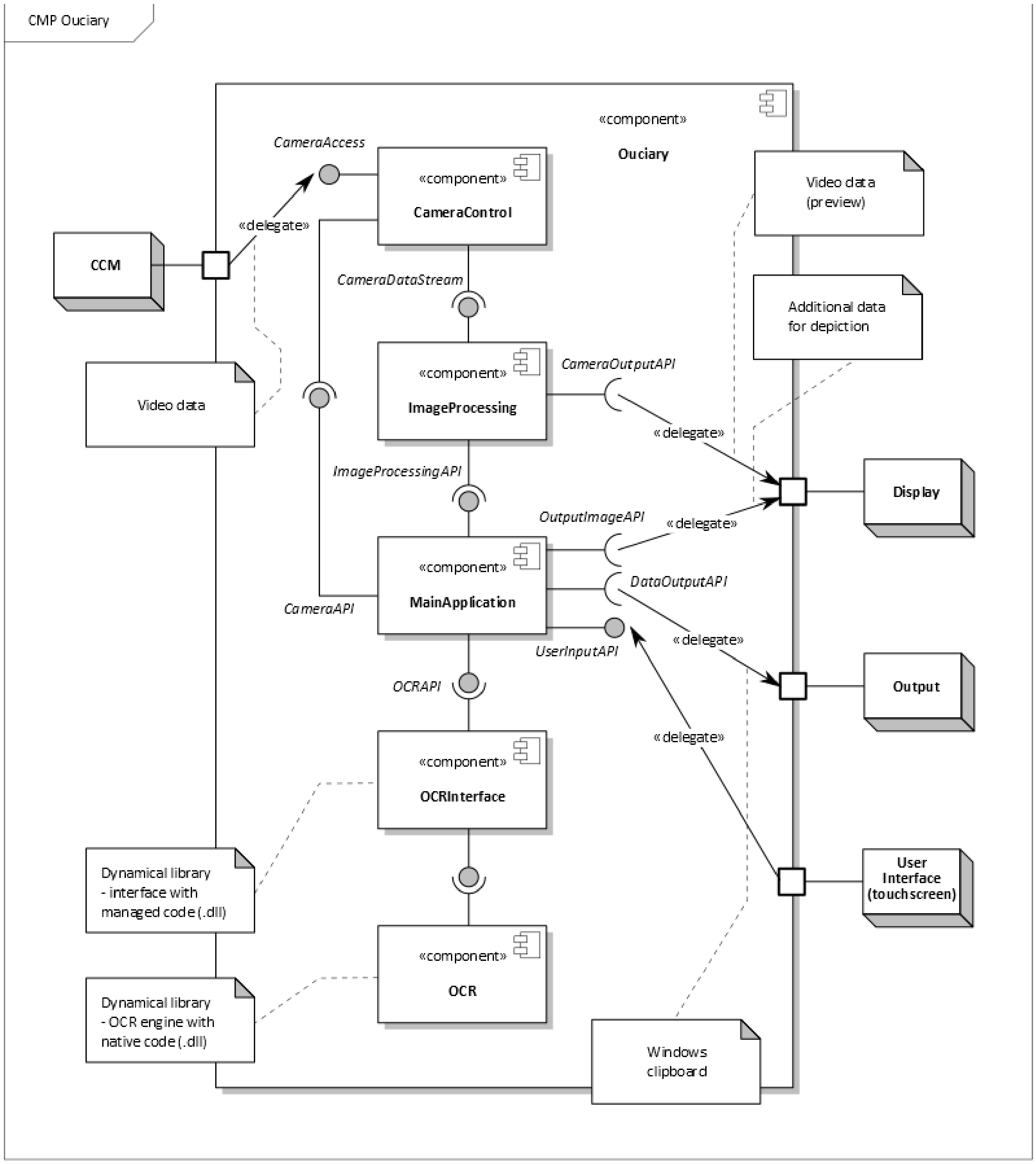


Figure 2. Components' diagram of application.

A. Ouciary Application

The application is practically created in form of a guide. After the initiation user is able to turn on the camera and capture the recognized image or simply choose from a file. This is displayed on the following images of the application [Fig. 3].

Consequently, the image is modified. According to the settings, the normalization, automatic rotation and saturation removal take place.

Furthermore, there is the area selection screen for recognition. Here it is possible to rotate the image manually and choose an area for recognition. Character recognition (described above) is very helpful during the text selection if this function is allowed. Moreover, this screen might be absolutely left out (In this case the image is modified according to the settings and the whole image is accounted for).

The selection happens by dragging ("rectangle drawing"). If it is necessary to cancel the whole selection it is enough to press anywhere on the image. If no area is chosen, the application automatically calculates with the whole image.

Figure 3. Application after the initiation and text capturing.

After the selection of the area it is possible to establish individual recognition process. The progress of recognition is shown here. After the termination of recognition process the application automatically moves on to the form for storage.

The resulting text can be seen here in a textbox and at the same time may be saved into a file or Windows mailbox.

As it was already mentioned, there is a space here for adding the functionality in a form of automatic events in relation to recognized text, eventually to a "templates" usage for contact creation according to a business card etc.

IV. Testing of Developed application

The application was tested continuously. The comparison of Tesseract with other OCR engines can be found in the section [*Section II*] and concretely in the table [*Table 1*]. Moreover, there is also a visible influence of the The testing of recognition quality is established by OCR engine which is a product of a third party and therefore is not a direct part of this work. The OCR engine was only a component of this work by a transfer (porting) to Windows Mobile (More precisely Windows Pocket PC 2003) in form of DLL library and to create suitable API. The function of recognition was not interrupted by anything; therefore the testing of quality recognition is not a direct component of this work.

Furthermore, it was established that the best results are obtained by using Tahoma font. In all of the tested samples the results were around 95% which is very sufficient. This is given mostly by the used language dictionary. In cases of graphically different fonts high results can be reached again, however Tesseract must firstly study these writings. The output is formed again by the language dictionary. It is possible to find more information about the preparation of language dictionaries on Tesseract webpage [9].

The results and quality of recognition are completely identical with Tesseract for PC.

A. *Testing of the speed of OCR on PDA*

According to the significantly smaller calculation performance of mobile devices it is advisable to undertake the measurement for the influence of an image characters number and the recognition of time of OCR process.

This measurement was accomplished using TesseractCLI application. The device which was used for testing was PDA HP iPAQ hx4700. This device has a 32 bite processor ARM which works using frequency of 624MHz and RAM memory of 64MB. Windows Mobile 6.5 was used as an operating system.

Moreover, 2 colored TIFF images were tested in resolutions of 268x240 (~64 kPx), 536x480 (~257 kPx) and 1072x960 (~1 MPx). All of these images were in variants with 81,202 and 335 characters. The tested images are shown on [Fig. 4].

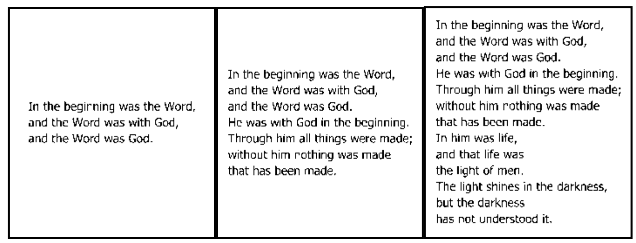


Figure 4. Tested images.

The testing was proceeding after the reset of the device. Each of the recognition measurements were established 5x and the final times were stated as an average. There were only small number of anomalies (maximum of 10% around average) among individual measurements and it was found out that the reset of the device does not have an influence on OCR operating period. The table [*Tab. 2*] gives the test results.

TABLE II. OCR Operating Period in Relation to the Number of Image Characters

Time OCR (s)		**Image resolution (px)**		
		268x240	*536x480*	*1072x960*
Number of characters	*81*	3,106	4,609	4,477
	202	9,529	8,450	10,523
	335	24,767	15,581	16,389

There was a strong dependence of the number of characters on the OCR operating period visible from the test's results. On the other hand the resolution dependence did not show to be as relevant, however the results here are still quite interesting.

The OCR time fluctuation concerning the recognition would be most likely given by the function of Tesseract. It can be expected that if the small resolution is used, there will be more faults in recognition and therefore the usage of dictionary will be increased in order to fix those faults. This may have impact on the time of transfer.

 ISBN: 978-1-61208-224-0

On the other hand, high resolutions will have effect on the on image processing time and recognition of individual characters.

According to the mechanism, the minimal time for the transfer is given by a compromise between image resolution (small resolution – dictionary usage, high resolution – long processing and image recognition). This assumption is identical to the measurement (the measurement which was intended for chosen combinations of resolution and for number of characters was repeated with practically same results.).

Generally, the processing is 10x slower than on PC, however it is bearable comparing to usage on PDA.

V. Conclusions

Main contribution of this project is development of mobile OCR application for Smart Devices with Windows Mobile OS. Used OCR technology can significantly speed up the work using text recognition where there is no requirement for manual transfer of text from an image (e.g. URL address capturing and consequent its display in the browser which is much faster than rewriting the address manually, especially in case of long and complicated addresses). Solution can be also used in the case of business cards digitalization or any other printed material which need to be rewritten. The mobile devices are always nearby [1] and therefore this method brings instant capturing of printed texts.

During the development phase some problem arose out from the real implementation, where the biggest one was the porting of selected desktop OCR Tesseract to the end platform (Windows Mobile). Nevertheless, due to its minimal demands on other components, Tesseract is one of the few free OCR engines which are possible to port using relatively small interferences.

Another inportant issue was in launching of CCM, where used DirectShow is quite complicated in the sense of a slow framerate of preview (around 2 fps).

Developped solution is very well functional, useful and positive.

Acknowledgements

This work was supported by „SMEW – Smart Environments at Workplaces", Grant Agency of the Czech Republic, GACR P403/10/1310. We also acknowledge strong support from Ales Kurecka during development phase and testing of application.

References

[1] J. Zelenka, "Information and Communication technologies in tourism – influence, dynamics, trends", In *E & M EKONOMIE A MANAGEMENT*, Vol. 12, Issue 1, pp. 123-132, 2009

[2] ZY. Liu, HN. Zhou, "Segmenting Texts From Outdoor Images Taken By Mobile Phones Using Color Features", Proceedings of SPIE, vol. 7874, Article Number: 78740B, DOI: 10.1117/12.872149, 2011

[3] J. Berclaz, N. Bhatti, SJ. Simske, and JC. Schettino,"Image-Based Mobile Service: Automatic Text Extraction and Translation", Proceedings of SPIE, Vol. 7542, Article Number: 754204, DOI: 10.1117/12.840279, 2010

[4] AP. Pozo, AW. Haddad, M. Boutin, and EJ. Delp, "A Method for Translating Printed Documents Using a Hand-Held Device", IEEE International Conference on Multimedia and Expo (ICME), JUL 11-15, 2011, DOI 10.1109/ICME.2011.6011940.

[5] Nokia Multiscanner - Recognize Texts With Camera. SymbianV3.Com. [Online] 6. 11 2008. http://symbianv3.com/nokia-multiscanner-recognize-texts-with-camera/.

[6] Camera-Dictionary Software (Net) for Android. [Online] 2010. http://www.hotcardtech.com/eng/OCRUserGuideen.doc.

[7] CamCard (Business Card Reader). IntSig Information. [Online] 3. 10 2010. http://www.intsig.net/home/us/android-/51-camcard-.

[8] Babel Reader-LE 1.0. WareSeeker - Search and Free Download PDA Software. [Online] 16. 10 2010. http://pda.wareseeker.com/Business/babel-reader-le-1.0.zip/15ca4f31ed.

[9] tesseract-ocr - Project Hosting on Google Code. Google Code. [Online] [Citace: 23. 10 2010.] http://code.google.com/p/tesseract-ocr/.

[10] Ocrad - The GNU OCR. Ocrad - GNU Project - Free Software Foundation (FSF). [Online] 11. 05 2010. http://www.gnu.org/software/ocrad/.

[11] Puma.NET. CodePlex - Open Source Project Hosting. [Online] 09. 01 2010. http://pumanet.codeplex.com/.

[12] ABBYY Mobile OCR Engine is a Software Development Kit (SDK). ABBYY - OCR, ICR, OMR, Data Capture and Linguistic Software. [Online] http://www.abbyy.com/mobileocr/.

[13] M. Kadavova, A. Slaby, and F. Maly, "Key factors involving the design of the system of virtual university", 7th WSEAS International Conference on Applied Computer and Applied Computational Science, pp. 678-683, april 6.-8. 2008

[14] F. Lefley, F. Wharton, L. Hajek, J. Hynek, V. Janecek, "Manufacturing investments in the Czech Republic: An international comparison", International Journal of Production Economics, vol. 88, Issue: 1, pp: 1-14, DOI: 10.1016/S0925.5273(03)00129-4, 2004

[15] P. Mikulecky, "Remarks on Ubiquitous Intelligent Supportive Spaces", 15th American Conference on Applied Mathematics/International Conference on Computational and Information Science, Univ Houston, Houston, TX, pp. 523-528, 2009.

[16] I. Bridova, M. Vaculik, and P. Brida, "Impact of Background Traffic on VoIP QoS Parameters in GPON Upstream Link", Elektronika ir Elektrotechnika, No. 8(104), pp. 113-118, 2011.

[17] V. Bures, "Conceptual Perspective of Knowledge Management", E & M Ekonomie a Management, vol. 12, Issue: 2, pp. 84-96, 2009

[18] D. Vybiral, M. Augustynek, and M. Penhaker, "Devices for position detection", Journal of Vibroengineering, vol. 13, Issue: 3, pp. 531-535, Sep. 2011

[19] J. Tariq, U. Nauman, M.U. Naru, "α-Soft: An English Language OCR". Second International Conference on Computer Engineering and Applications (ICCEA 2010), IEEE Xplore, DOI 10.1109/ICCEA.2010.112, 2010