Using Digital Cameras for Text Input on Mobile Devices

Frank Siegemund and Muhammad Haroon
European Microsoft Innovation Center
Ritterstrasse 23, D-52074 Aachen, Germany
{franksie|i-mharoo}@microsoft.com

Abstract

This paper presents novel text input methods for mobile devices. In particular, it shows how users can capture text from books, newspapers, and other objects by using the digital cameras that are integrated into an increasing number of smartphones. The idea is to select text from a live video stream generated by these digital cameras. Different interaction patterns are discussed that allow users to select text: (1) automatic detection of the word closest to a fixed position on the mobile device's screen, (2) delayed movement of the text selection to improve the precision of the selection procedure, and (3) using the stylus to select words and word groups. The proposed interaction patterns are evaluated based on the results of a user study, and their application is shown in an example scenario. The presented research has practical relevance because it can improve the usability of mobile applications such as online searching.

1. Introduction

In Pervasive Computing, a variety of tagging technologies – from simple bar codes to active and passive RFID tags – are used to augment everyday objects. While these technologies are well suited to link physical objects with virtual services, they often require a post-hoc augmentation of these objects. However, there is one "natural" tagging technology that is already present on many everyday things. This tagging technology is printed text. Printed text can be found not only in books and newspapers but also on posters and product packages. Also, with the increasing number of cameras integrated into cell phones and PDAs, these devices can serve as "tag readers." In other words, mobile devices can be used to capture text from the physical world, for transforming this information, and for using it as input for a range of applications. For example, a user reading a newspaper article about Roman history could capture words in the article with her mobile phone. Afterwards, the mobile phone could then automatically start an Internet search for more information about the captured words. The same mechanism applies when capturing text from posters or products in a supermarket.

As an example of this vision, this paper focuses on capturing text from printed documents. In the presented approaches, users point their mobile device at a printed text. The camera integrated in the mobile device then generates a live video stream that is analyzed in real-time for words that appear in the text. Users can then select a word and subsequently extend their text selection over multiple words. The result of the text selection is an identified area in a video frame that contains the selected words. This sub-image is preprocessed on the phone, and can then be sent to an application such as an online search engine. Using such methods for text input, a mobile device's digital camera can serve as an additional input channel for mobile applications.

The main contributions of the paper are as follows: First, it presents a thorough analysis of user interaction patterns for selecting text with a handheld-embedded digital camera. It is shown, for example, how to cope with unintentional camera movements, and how to improve the precision of text selections. Second, the paper proposes adaptive algorithms for selecting words in a live video stream. Third, it presents a concrete application that shows how the proposed concepts can be used in real-world scenarios.

The remainder of this paper is structured as follows: Sect. 2 summarizes related work. Sect. 3 describes user interaction patterns for selecting printed text with a camera phone, while Sect. 4 discusses their implementation. Sect. 5 presents algorithmic improvements. Sect. 6 evaluates our work based on a detailed user study, and Sect. 7 presents a selected application. Sect. 8 concludes the paper.

Proceedings of the Fifth Annual IEEE International
Conference on Pervasive Computing and Communications (PerCom'07)
0-7695-2787-6/07 $20.00 © 2007 IEEE
Authorized licensed use limited to: Reprints Desk. Downloaded on July 23,2025 at 20:21:52 UTC from IEEE Xplore. Restrictions apply.

2. Related Work

Due to the economic success of the Short Message Service (SMS), the problem of finding efficient text input methods for mobile devices has received considerable attention in Human Computer Interaction [4, 5]. The literature distinguishes between predictive text input and multitap [2] as well as between consecutive and concurrent text input methods [8, 9]. The main difference to the work presented in this paper is that we use digital cameras for text input instead of keys or a stylus. This requires that the text is available in a printed document in order to be captured by a handheld's camera.

Text detection in images deals with the location of text without performing an optical character recognition (OCR) [1, 3, 10]. The result is the identification of multiple rectangular areas in an image, whose contents can later be processed by an OCR program. Comparing existing approaches to our work, there are a range of differences in the assumptions made about images and video streams; there are also different goals targeted in our system. First, we focus on usability and responsiveness. Hence, the video stream provided by a mobile device's digital camera must be processed on the fly in order to appropriately react to the users' movements. Second, the cameras embedded into mobile devices provide a video stream of only low-quality and resolution; and we must be able to deal with varying lightning conditions. Third, we are interested in the user interactions that make text selection possible on a mobile device.

There are a range of scanner devices on the market – which typically have the form of a pen – that capture words by moving the tip of the scanner over a printed text. The pen is typically connected to a stationary computer by means of a cable or a wireless interface such as Bluetooth. The advantages of our system compared to these devices are as follows. First, mobile phones are much more pervasive than these pens. They have a much higher availability from a user perspective. Second, the mobile phone is independent from a nearby more powerful computer because it provides pervasive cellular connectivity. Third, a mobile phone offers typically more sophisticated user interface capabilities (e.g., a larger screen) than a pen.

A video published in the Internet [11] exists that shows an application that carries out an OCR analysis on images captured by a handheld-embedded digital camera. Although no information could be obtained about the underlying technical system, the main difference to this work is again our focus on usability aspects. Furthermore, our user interaction patterns do not require an OCR tool installed on the mobile device.

Roduner and Rohs [7] present algorithms for comparing images taken with a handheld-embedded digital camera. In their system, a user manually selects an area of interest on a mobile device and sends it for processing to a server. In our system, text is detected automatically in a video stream, and we focus on evaluating interaction mechanisms.

3. UI Strategies for Text Input

3.1. Text Selection at Fixed Screen Position

In this user interaction pattern, a live video stream is generated by a mobile device's camera and presented to the user on the mobile device's screen. Additionally, a crosshair is displayed in the middle of the screen. The user then points the camera to a printed text and moves with the crosshair over individual words. Whenever the crosshair is over a word in the text, this word is automatically highlighted by a rectangle that is drawn around it (cf. Figure 1.) The crosshair must now be moved to the word the user is interested in, which is again highlighted as a result of the user's movement. The user then presses the action button on her phone to finalize the selection (the action button is often the one in the middle of the cursor keys.)

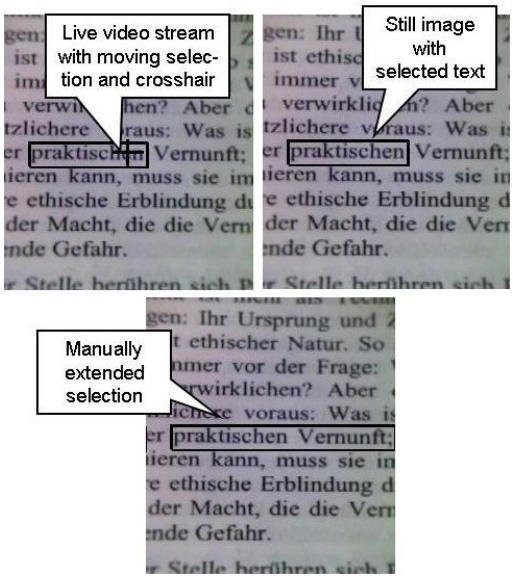

Figure 1. Selecting text from a live video stream

After the button has been pressed, the video stream stops and is replaced by a still image that shows the last frame of the stream with the selected word high-

Proceedings of the Fifth Annual IEEE International
Conference on Pervasive Computing and Communications (PerCom'07)
0-7695-2787-6/07 $20.00 © 2007 IEEE
Authorized licensed use limited to: Reprints Desk. Downloaded on July 23,2025 at 20:21:52 UTC from IEEE Xplore. Restrictions apply.

lighted. A user might unintentionally move the camera while pressing the action button or select a different word than the one intended because of other reasons. The user has therefore the possibility to move the text selection to a different word using the cursor keys of her mobile device. Pressing the action button once more confirms the currently selected word.

Afterwards, the cursor keys can be used to select multiple words. That is, by pressing the right or left arrow keys, the words to the right and to the left can be included in the selection. By pressing the action button again, the final text selection is confirmed. The result is a rectangular area of the displayed image, which can then be processed by other applications.

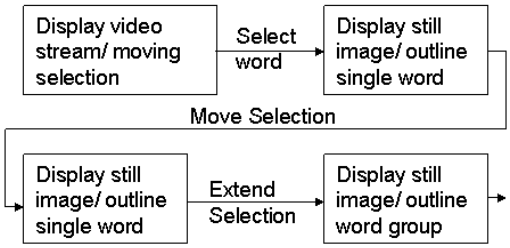

Figure 2. Text input from live video stream

3.2. Delayed Movement of Selection Boxes

The main difficulties in the above user interaction pattern are caused by unintentional camera movements. Although users can move the text selection using the cursor keys of their mobile devices after the initial selection has been made, it is desirable to simplify the process for making the initial selection. For doing this, we have to cope with the following problems:

First, the focal distance of the camera is important. The larger the distance between the printed text and the camera for getting a sharp image, the steadier a user has to be for selecting a specific word – and the more relevant are small movements of the camera. Second, it is important that the analysis of video frames can be done in real-time. If too much time passes before a new frame is processed, the text selection box "jumps" between video frames instead of moving smoothly from word to word. Third, the farther the camera is away from the text (this is often necessary for the user to see more of the printed text on the device's screen – and independent of the focal distance of the camera) the steadier a user has to be to keep the crosshair on a specific word.

Delaying the movement of the selection box can help to alleviate these problems. While the first user interaction pattern always highlighted the word closest to the crosshair, we now consider information about the word selected in the previous video frame. That is, if the user focuses on a word and unintentionally moves the camera so that another word is closer to the crosshair for only a short time period, the text selection does not change between video frames. This improves usability because it deals with unintentional camera movements and compensates shaking. Thus, it is simpler for users to carry out the text selection with a single hand or to select words from larger distances.

3.3. Using a Stylus for Text Selection

This interaction pattern differs from the two above in that no processing is done on the live video frames. Instead, the user focuses the camera on a particular text segment and freezes the video stream by tapping on the mobile device's screen. Technically, the video stream is then stopped and the last video frame is presented to the user as a still image. On the still image, words can now be selected by simply tapping on the word in the still image, and by moving the stylus over surrounding words in order to extend the text selection. This can be repeated several times. In order to finalize the text selection, a small graphical symbol can be displayed on top of the still image. By tapping on this symbol with the stylus a user confirms the selection.

4. Implementation

4.1. Technical Challenges

There are several technical challenges when implementing the presented interaction patterns: (1) the performance constraints of mobile devices, (2) the software support for video and image processing, (3) the video preview quality, and (4) the fixed-focus lenses integrated into most mobile devices. The last problem is the most complicated to deal with because it implies that pictures – and video frames – of nearby objects are typically blurred. There are three options to deal with this problem: auto-focus lenses, manual switches, or add-on lenses. While we think that more camera phones will contain auto-focus lenses in the future, simple add-on lenses are used in our prototype.

4.2. System Architecture

The system architecture consists of two main components: (1) the text detection algorithm on the mobile phone, which implements the user interaction patterns and preprocesses the selected text area, and (2) a server in the Internet or an application running on the phone for carrying out the OCR analysis.

Proceedings of the Fifth Annual IEEE International
Conference on Pervasive Computing and Communications (PerCom'07)
0-7695-2787-6/07 $20.00 © 2007 IEEE
Authorized licensed use limited to: Reprints Desk. Downloaded on July 23,2025 at 20:21:52 UTC from IEEE Xplore. Restrictions apply.

First, our text detection algorithm is responsible for the following: It (1) analyzes video frames for words closest to a given position in a frame, (2) it outlines words and word groups, (3) it allows the user to move and extend the text selection in a still image, and (4) it carries out a conversion of the selected image area so that it can be processed by an OCR tool or be transmitted over the Internet requiring the least amount of memory possible.

Second, we assume that there is an OCR tool available either on the phone itself or on a server. The last step of the text detection algorithm will convert the selected part of the image so that it consumes only very little memory. Thus, the image can be transmitted faster to a backend server in the Internet. In the current implementation, the Microsoft Office Document Imaging library, which is included in Microsoft Office 2003, carries out the OCR analysis. This application runs on a server, a normal PC in our case, and returns the result of the OCR to the phone.

4.3. Text Detection Algorithm

The basic structure of the text detection algorithm is as follows:

```
   VOID Transform(
        IMediaSample *pIn,
        IMediaSample *pOut)
   {
     RECT rcOut;

 7:  CopyVideoFrame(pIn, pOut);
 8:  DrawCrosshair(pOut);
 9:  CopyYInternal(pIn, m_rgCopy,
                   COPY_RECT);

 12: Thresholding(m_rgCopy);
 13: FindCharacterRegions(m_rgCopy);
 14: CalculateCharacterProperties();
 15: if (GetWordOutlineAt(m_ptCross,&rcOut))
     {
       DrawOutline(pOut, rcOut);
       CopyRectRegion(m_rgRes, pIn, rcOut);
     }
 }
```

Figure 3. Algorithm for transforming a single video frame

In DirectShow – the programming framework for multimedia in Windows Mobile – video frames flow through a so-called filter graph. Thus, the above algorithm describes how an input video frame (`IMediaSample *pIn`) is transformed to an output video frame (`IMediaSample *pOut`.) At first, the input sample is copied to the output sample, and the crosshair is drawn to the output sample (cf. lines 7-8.) Video is typically provided in the YUV format in Windows Mobile. This is an advantage for the text detection algorithm because it only requires the luminance component (i.e., the Y component), whereas the color components can be neglected (cf. line 9.) Furthermore, only a horizontal subset of the frame is copied to an internal buffer. The actual text detection takes place on this internal buffer (`m_rgCopy`.)

For carrying out the actual text detection the internal buffer is first converted to a black and white image (cf. line 12.) After the black and white conversion, all character regions inside the copied image area are identified; and simple properties are assigned to the different regions (cf. lines 13-14.) These properties include the number of pixels in a region as well as its minimum and maximum x and y coordinates. Assuming that each character is a single region in the image, it is now possible to calculate the outline of the word closest to the displayed crosshair (cf. line 15.)

5. Improvements

In order to ensure interactivity and usability, a number of optimizations must be applied in the presented algorithm: (1) only the luminance channel and a subset of a video frame are analyzed, (2) during the black and white conversion connected character regions are produced and a suitable threshold is determined in a feedback loop, (3) regions are analyzed for their suitability for an OCR conversion, (4) the size of the final image for the OCR analysis is reduced significantly for fast transmission over the Internet, and (5) history information is used to implement a delayed movement of the text selection box. In the following, improvements (2) and (5) are presented in greater detail.

5.1. Dynamic Thresholding in Video Frames

This improvement addresses the detection of single characters in a video frame. In the above description it was assumed that every character is a single region in the black and white converted image. Unfortunately, this is a simplification because differing lightning conditions and text fonts ultimately lead to different results with respect to character regions.

Figure *4* shows the results of the conversion process for two different words with different text fonts and under different lightning conditions. The thresholding algorithm also uses a threshold t that determines when pixels have to be transformed to either black or white. This threshold can either be (1) a concrete pixel value in case of a normal thresholding algorithm or (2) a threshold describing how a pixel value must differ from surrounding pixels in the case of adaptive thresh-

Proceedings of the Fifth Annual IEEE International
Conference on Pervasive Computing and Communications (PerCom'07)
0-7695-2787-6/07 $20.00 © 2007 IEEE
Authorized licensed use limited to: Reprints Desk. Downloaded on July 23,2025 at 20:21:52 UTC from IEEE Xplore. Restrictions apply.

olding. In our implementation an adaptive thresholding algorithm was used.

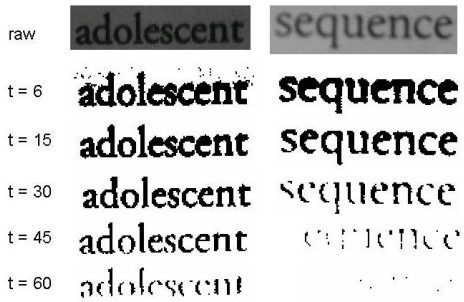

Figure 4. Result of thresholding under different lightning conditions

With respect to the word detection, we initially assumed that each character is a single region in the converted image. Considering Figure 4, this would be the case for t=30 and the word "adolescent" and t=15 for the word "sequence". However, while detecting the outline of a word, only the distance between words is important. Thus, if characters inside a word are connected its outline can still be detected. An example of this is the word "sequence" for t=6; in this case "seq", "u", "en", and "ce" are single regions. In fact, if characters in a word are thusly connected, this speeds up the whole word detection process because we have a smaller amount of regions to process in the algorithm. Also, if the lightning conditions change during word detection, a fixed threshold can lead to undesirable results.

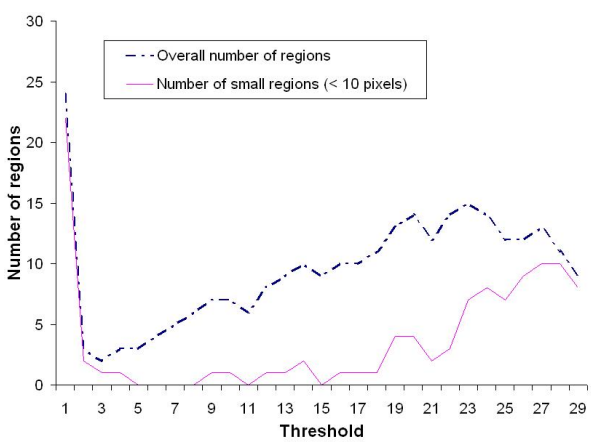

Figure 5. Number of regions wrt. threshold *t*

Thus, our algorithm dynamically tries to find a low threshold producing connected regions through a feed back loop – in order to speed up the word detection. The feed-back process starts with a low threshold and evaluates the regions found. If the result are many small regions, the threshold is increased before processing the next video frame. The heuristic is thus to choose the smallest threshold that produces no small regions. Figure *5* shows how the threshold *t* affects the number of small regions. Also, it is important to note that the dynamic adaptation of the threshold happens incrementally from video frame to video frame. The number and size of the regions found thus affect the threshold in the next video frame.

5.2. Delayed Text Selection Movement

When moving the crosshair, it is beneficial from a user point of view if unintentional camera movements do not have an effect on the selected word. Thus, the movement of the text selection box should be delayed (cf. Sect. 3.2.) We have implemented this usage pattern by considering information about selected words in previous video frames.

On the left hand side in Figure 6, the word "result" is just about to lose focus if the user moves the camera further to the right. In the image on the right in Figure 6, the word "but" should be selected, but this selection is delayed for several video frames. Only if the user moves further away from the initial selection, words are selected instantaneously.

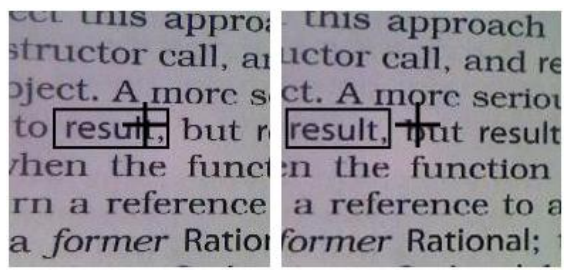

Figure 6. Delayed movement; if the crosshair focuses on a different word for only a short time period, the text selection box does not change[1]

Generally, there are two approaches to implement the delayed movement: (1) identify words in a video frame and decide whether the word closest to the crosshair is the one selected previously, and (2) consider only the position of the selected word in the previous video frame.

The first approach assumes that it is possible to identify words. This is difficult, because we cannot carry out an OCR analysis on multiple words due to time restrictions. Other methods to identify words would be to use second order moments of identified regions or other, simpler, properties such as the size of

[1] This picture is a reconstruction because it is difficult to capture screen contents at the moment the delayed movement occurs.

Proceedings of the Fifth Annual IEEE International
Conference on Pervasive Computing and Communications (PerCom'07)
0-7695-2787-6/07 $20.00 © 2007 IEEE
Authorized licensed use limited to: Reprints Desk. Downloaded on July 23,2025 at 20:21:52 UTC from IEEE Xplore. Restrictions apply.

a text box. It must be ensured that all properties calculated are rotation and scale independent.

In our implementation, the second approach is used because it requires fewer computations and is thus more suited for the analysis of video frames. On the other hand, the delayed movement implemented this way is only suitable for small camera movements, mainly in the horizontal direction. The simplified algorithm keeps track of the bounding box of a selected word. For a fixed number of consecutive video frames the logical position of the crosshair is then translated to the center of the previously identified bounding box.

Let us consider the example depicted in Figure 6. From a user perspective the crosshair always remains in the middle of the screen. On the left-hand side in Figure 6 the crosshair is close to the right border of the bounding box around the word "result." Now the user moves the camera slightly further to the right so that the crosshair (the center of the screen) is over the word "but". However, the algorithm still knows the bounding box of the selected word in the last video frame and computes the center of this bounding box. In the new video frame we would then look for the word at the center of the last bounding box. In the example, the effect is that the logical position of the crosshair (for the user it always remains in the middle of the screen) is shifted slightly to the left. Thus, the logical position of the crosshair is moved to the word "result" in the current video frame. From a user perspective this appears as a delayed movement.

6. User Study and Evaluation

In order to evaluate the presented user interaction patterns, we carried out a user study with people from different professional backgrounds and with different affinity to mobile devices. 19 users participated in the study, and a session with a single user took 25 to 60 minutes. We implemented the three user interaction patterns presented in Sect. 3 on a standard Windows Mobile-based smartphone, and asked each user to select 88 words using the different interaction mechanisms. Overall, we could thus collect over 1600 test cases. The youngest participant in the study was 23 years old, the oldest 61.

From a user perspective, there are two phases when selecting a word in a document: first, the word in the text must be located on the mobile device's screen, and second, the crosshair must be moved over the desired word in the text. To distinguish between the two phases, we printed out several words with no surrounding text and asked participants to select these words. The second step then consisted in selecting words within a paragraph.

The material collected in the study allows the following conclusions. The time users need to find a word in a paragraph varies significantly between different test cases of a single user. This is because the time essentially depends on how often a user switches between looking at the phone screen and the actual text document. Here, it must be distinguished between users that are accustomed to using mobile devices and those who almost never use a mobile device. Experienced users usually remained focused on the mobile device's screen during the whole text selection process. Other participants often switched between looking at the phone screen and the actual text. There was one participant who had problems to keep the phone steady, and moved away from a desired word even if the word had been located on the mobile device's screen. The amount of time required by experienced users for selecting isolated words varied from 3.3 s to 45.5 s; 2.2 s to 70.3 s were required to select words within a paragraph. For inexperienced users the time for selecting isolated words and words within a paragraph varied between 3.6 s to 85.3 s and 3.7 s to 72.1 s, respectively. The average time for experienced users was 7.8 s and 10.7 s; that for inexperienced users 14.1 s and 20.7 s. The median for experienced users was 6.7 s and 7.8 s; that for inexperienced users 11.9 s and 14.8 s (cf. Tab. 1.)

The next step in the user study was to evaluate the effect of unintentional camera movements. In order to do this we printed out words in different text fonts (from 8 to 14 pixels) and compound words. The idea is that small words are difficult to focus on because of their small size. On the other hand, when compound words shall be selected, the user has to move further away from the text in order for the words to fit on the mobile device's screen. Unintentional camera movements have then a bigger effect. Evaluating the collected data, user needed in average 64 % longer to select small words or long word sequences. The median for selecting those words was 28 % higher then the median for selecting normal words.

We also recorded the number of misses, i.e., the number of times people selected a wrong word or selected only parts of a required word sequence. The misses only refer to the initial selection. In all user patterns presented in Sect. 3, the selection can of course still be adapted in the still image after the first selection has been made. Together with the small words containing only one character (e.g., I) and very long word sequences containing up to four words we counted 154 misses. 115 of those resulted in no word being selected, 13 of those were wrong words, and 26 were

Proceedings of the Fifth Annual IEEE International
Conference on Pervasive Computing and Communications (PerCom'07)
0-7695-2787-6/07 $20.00 © 2007 IEEE
Authorized licensed use limited to: Reprints Desk. Downloaded on July 23,2025 at 20:21:52 UTC from IEEE Xplore. Restrictions apply.

partly selected word sequences. Small movements can be reduced by implementing delayed movements as discussed in Sect. 3. However, our study showed that the main problems were caused by movements when the user pressed the action button for selecting the text. This often resulted in considerable movements in the vertical direction. As a result of the user study, a way must be found to reduce these misses. We therefore propose to keep the identified text images (the area in the selection box) for a number of video frames, and use the last valid selection when the notification about the button press reaches the text detection algorithm.

Table 1. Results of user study (in seconds)

Test	Min	Max	Avrg	Median
Pattern 1 and 2 (cf. Sect. 3), experienced users				
Isolated words	3.3	45.5	7.8	6.7
In paragraph	2.2	70.3	10.7	7.8
Pattern 1 and 2 (cf. Sect. 3), novice users				
Isolated words	3.6	85.3	14.1	11.9
In paragraph	3.7	72.1	10.7	14.8

Table 2. Results for all users

Test	Normal	Delayed	Stylus
Median	9.9 s	8.7 s	7.8 s

With respect to the first two user interaction patterns presented in Sect. 3, the number of misses shows that the implementation of a delayed text selection is very useful. However, the algorithm that we implemented copes better with horizontal than with vertical movements. As a result of our user study, however, it must be said that unintentional vertical movements are much more frequent than horizontal ones. This is likely also a result of the phone model used in the user study, which has no conventional keypad and the center of gravity is therefore at the top of the devices. This results in increased vertical movements if the phone is kept in one hand. The median for selecting words inside a paragraph using the delayed movement was 8.7 s, while it was 9.9 s for the first user interaction pattern.

Several people favored the stylus-based user interaction presented in Sect. 3 – the median for selecting words in a paragraph was 7.8 s for this interaction pattern. Especially inexperienced users liked the fact that the word can appear at any position of the mobile device's screen and be selected by just tapping on it, which is a very intuitive way to select something.

Also the other two user interaction patterns support changing the initial selection. If no stylus is available, the advantage of giving the user feedback already in the video stream is thus that fewer key presses are necessary for changing the initial selection. Furthermore, the disadvantage of the stylus-based interaction is that it requires two hands: one to hold the phone, one to hold the stylus.

Last but not least, the performance of the proposed algorithm (cf. Sect. 4) was evaluated. The time for processing a single video frame is 40 ms; this includes the time for finding appropriate thresholds for the black and white conversion. The average time for carrying out the black and white conversion is 13 ms, 16 ms are required for the region detection, and 5 ms for assigning properties to regions.

7. Application: Mobile Online Searching

As text input mechanisms, the described user interaction patterns can be integrated into a range of mobile applications that rely on text input. One concrete application, which we prototypically implemented, is mobile online searching. The goal of this application is to make it easier for users to obtain additional information about words that are printed on real paper. One usage scenario would be a person that is reading a book or a newspaper and wants to get additional information about terms that are unfamiliar to her. Taking advantage of the pervasiveness of mobile phones, the mobile phone is an ideal platform for capturing the information printed in such a document and for displaying the results of a search.

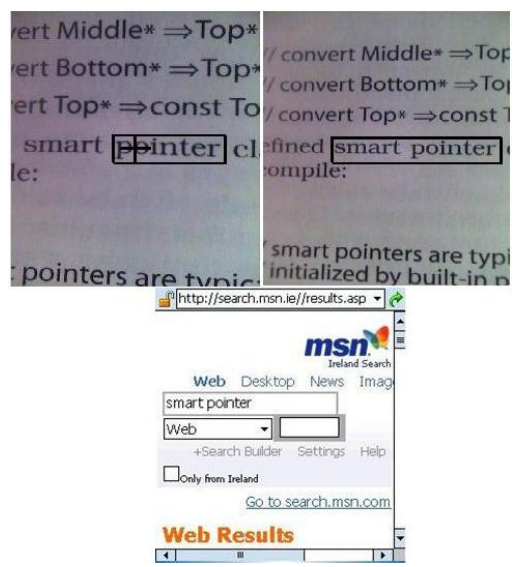

Figure 7. Automatically generating search queries from text captured in a book

In our implementation, the presented algorithms are running on a smartphone; they are responsible for cap-

Proceedings of the Fifth Annual IEEE International
Conference on Pervasive Computing and Communications (PerCom'07)
0-7695-2787-6/07 $20.00 © 2007 IEEE
Authorized licensed use limited to: Reprints Desk. Downloaded on July 23,2025 at 20:21:52 UTC from IEEE Xplore. Restrictions apply.

turing the text, and they prepare the selected image for an OCR analysis. Because we preprocess the text area containing the word(s) to reduce its memory size, the image area containing the selected text can be transmitted over the Internet to a stationary computer. This computer carries out the OCR analysis, and starts a corresponding online search. The results of this search are then displayed on the mobile device (cf. Figure 7.)

8. Conclusions

Pervasive Computing envisions a world in which everyday physical objects are augmented through the use of computing technologies. Considering everyday objects such as newspapers, books, or magazines, the mobile phone is an interesting platform for providing additional services that augment the capabilities of these objects. The digital cameras integrated into today's cell phones are thereby a tool for capturing data from the physical world (e.g., printed text), which can then be transformed and used as input for Pervasive Computing applications.

The main contribution of this paper was to propose and systematically evaluate different user interaction mechanisms for capturing text with a camera phone. In particular, we dealt with the practical constraints that are caused by the resource constraints of mobile devices, the low quality images produced by handheld-embedded cameras, and different usability restrictions. We think that the presented approaches have practical relevance and can be integrated into a variety of applications. In online searching applications, for example, it is possible to automatically generate search queries from the text images captured with a camera phone.

9. Acknowledgements

We would like to thank Wolfgang Manousek and Alain Gefflaut for giving us the opportunity to carry out this work. We also thank the anonymous reviewers for their valuable feedback.

10. References

[1] Li Huiping, D. Doermann, and O. Kia, "Automatic text detection and tracking in digital video", IEEE Transactions on Image Processing, Volume 9, Issue 1, 2000, pp. 147-156.

[2] C. L. James and K. M. Reischel, "Text input for mobile devices: comparing model prediction to actual performance", Proceedings SIGCHI Conference on Human Factors in Computing Systems, 2001, pp. 365-371.

[3] A.K. Jain and Bin Yu, "Automatic text location in images and video frames", Proceedings 14th International Conference on Pattern Recognition, Volume 2, 1998, pp. 1497-1499.

[4] I. Scott MacKenzie and R.William Soukoreff, "Text Entry for Mobile Computing: Models and Methods, Theory and Practice", Human-Computer Interaction, Lawrence Erlbaum Associates, 2002, pp. 147-198.

[5] Toshiyuki Masui, "An efficient text input method for pen-based computers", Proceedings SIGCHI Conference on Human Factors in Computing Systems, 1998, pp. 328-335.

[6] William Newman and Pierre Wellner, "A desk supporting computer-based interaction with paper documents", Proceedings SIGCHI Conference on Human Factors in Computing Systems, 1992, pp. 587-592.

[7] Christof Roduner and Michael Rohs, "Practical Issues in Physical Sign Recognition with Mobile Devices", Proceedings Workshop on Pervasive Mobile Interaction Devices (PERMID), 2006, pp. 297-304.

[8] Daniel Wigdor and Ravin Blakrishnan, "TiltText: Using tilt for text input to mobile phones", Proceedings 16th Symposium on User Interface Software and Technology, 2003, pp. 81-90.

[9] Daniel Wigdor and Ravin Balakrishnan, "A Comparison of Consecutive and Concurrent Input Text Entry Techniques for Mobile Phones", Proceedings SIGCHI Conference on Human Factors in Computing Systems, 2004, pp. 81-88.

[10] Victor Wu, R. Manmatha, and Edward M, "Riseman: Finding text in images", Proceedings 2nd International Conference on Digital Libraries, 1997, pp. 3-12.

[11] Camera Dictionary, http://www.dailymotion.com/video/10359, last accessed Aug. 2006.

Proceedings of the Fifth Annual IEEE International
Conference on Pervasive Computing and Communications (PerCom'07)
0-7695-2787-6/07 $20.00 © 2007 IEEE
Authorized licensed use limited to: Reprints Desk. Downloaded on July 23,2025 at 20:21:52 UTC from IEEE Xplore. Restrictions apply.