Hybrid Document Matching Method for Page Identification of *Digilog Book*s

Jonghee Park and Woontack Woo

GIST U-VR Lab.
Gwangju, 500-712, S.Korea
{jpark,wwoo}@gist.ac.kr

Abstract. *Digilog Books* are AR (Augmented Reality) books, which provide additional information by visual, haptic, auditory, and olfactory senses. In this paper, we propose an accurate and adaptive feature matching method based on a page layout for the *Digilog Books*. While previous *Digilog Books* attached visual markers or matched natural features extracted from illustrations for page identification, the proposed method divides input images, captured by camera, into text and illustration regions using CRLA (Constrained Run Length Algorithm) according to the page layouts. We apply LLAH (Locally Likely Arrangement Hashing) and *FAST+SURF* (FAST features using SURF descriptor) algorithm to appropriate region in order to get a high matching rate. In addition, it merges matching results from both areas using page layout in order to cover large area. In our experiments, the proposed method showed similar matching performance with LLAH in text documents and *FAST+SURF* in illustrations. Especially, the proposed method showed 15% higher matching rate than LLAH and *FAST+SURF* in the case of documents that contain both text and illustration. We expect that the proposed method would be applicable to identifying various documents for diverse applications such as augmented reality and digital library.

Keywords: Document matching, augmented reality, Digilog Book, page identification

1 Introduction

Since electronic books (e-book) started appearing in 1990s, they have been received much attention as possible replacements for paper books due to its mobility and relatively low price. However, most readers still more prefer paper books because of the emotional bond between book and reader, i.e., the aesthetic sensation of physical papers. These days, *Digilog Books* are greatly anticipated as the next generation of e-books because it can provide additional information by stimulating visual, auditory, tactile senses in Augmented Reality (AR) environment [4]. Since *Digilog Books* has been emphasized, tracking *Digilog Books* becomes an important research field because it is one of core procedures used within the specialty of AR and computer vision. Usually, tracking *Digilog books*

utilizing a camera, used for image retrieval, are comprised of page identification among many pages and tracking from camera image. Early *Digilog Books* usually have applied a marker tracking using ARToolkit [2]. However, there are two disadvantages in the marker tracking to apply it to published books. In order to utilize ARToolkit in tracking *Digilog Books*, artificial markers have to be attached on each page for finding page number and camera pose. In case of the published books, it would be exacting tasks. Secondly, there is a limit on the number of pages because of restriction in the number of possible patterns. Therefore, it would not be applied to the books that have large number of pages. In order to overcome these disadvantages, many researchers have applied NFT (Natural Feature Tracking) methods to *Digilog Books* because the NFT utilizes printed contents such as illustrations and text. Therefore, it can resolve problems of the marker tracking through natural manner.

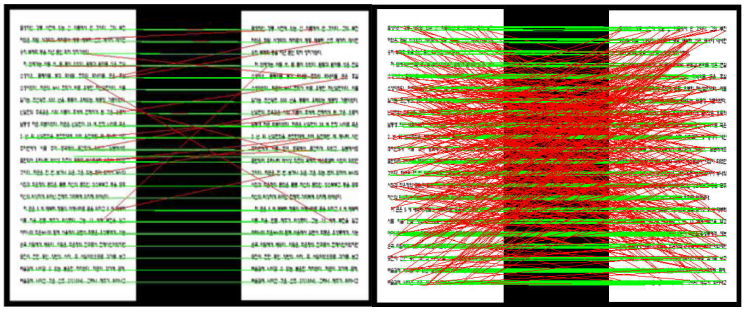
2 Related Works

Generally, there are mainly two categories to identify pages of *Digilog Books* in AR: a marker and a natural feature matching.

First, ARToolkit, which is applied to most of early *Digilog Books*, utilizes marker tracking proposed by Kato and Billinghurst [5]. In order to determine page, the sub-image within a marker is compared by template matching with patterns given the system before. Then, the marker that has the highest matching rate is selected and the designated marker number is the page number. Second, SIFT (Scale Invariant Feature Transform) is a representative method deploying illustrations (texture) in object or scene [7]. However, the processing time for feature extraction of SIFT still takes long time without Graphics Processing Units (GPU). In order to apply NFT to multi-objects tracking in real-time, Wagner applied FAST corners as feature points rather than SIFT features to reduce time in feature extraction [9] [12]. In addition, they trained the images in multi-scale to compensate scale invariant characteristics. Similarly, Kim utilized GPU and multi-threads to reduce feature extraction and matching time [6]. The NFT in Virtual Pop-up Book is another method utilizing template matching [10]. Templates around feature points are used in page determination due to robustness of color histogram in various camera rotations. Another approach in NFT is to utilize text printed on each page. LLAH (Locally Likely Arrangement Hashing) employs each center of words as feature points [8]. Then, geometric relation between features such as cross-ratio is used to generate feature descriptors. Finally, hash indexes that are employed in feature matching are created by the descriptors. Original LLAH has restricted in view point to recognize the page. So, Uchiyama proposed on-line learning method of LLAH to detect pages in various camera viewpoints [11]. However, existing NFT targeting illustrations are not applicable to text because of ambiguity as a result of color similarity. Similarly, NFT targeting text is not applied to illustrations due to unstable repeatability in feature detection. Therefore, if there is a page that consists of only text or illustrations, existing NFT would fail to track the page.

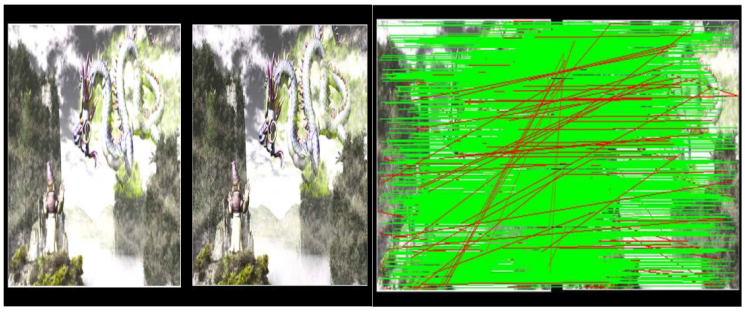
3 Hybrid Document Matching for Page Identification

Generally, NFT methods for illustration are not acceptable to a page which text is printed because of similarity of text descriptors. Figure 1(b) shows a matching result of *FAST+SURF* (FAST features using SURF descriptor) on a text image [1]. In the case, about 1300 features were extracted and only 48% features are correctly matched under 10 degree of rotation. The green lines represent inliers of matching and the red lines mean outliers. However, LLAH classify only few matching correspondences as outliers in figure 1(a). Similarly, NFT methods for text are not appropriate for illustrations. Figure 2(a) shows a matching result of LLAH under 10 degree of rotation. In the case, none of features are matched correctly due to low repeatability of features. In contrast, figure 2(b) shows the matching result of *FAST+SURF* in an illustration.



(a) LLAH. (b) *FAST+SURF*.

Fig. 1: Matching on a text image with LLAH and *FAST+SURF*.



(a) LLAH. (b) *FAST+SURF*.

Fig. 2: Matching on an illustration image with LLAH and *FAST+SURF*.

Figure 3 denotes overall procedure of the proposed method. Firstly, image is divided into two images that one has only text while the other has only illustration. In order to recognize the page, FAST corners and LLAH features are extracted from each image. Then, SURF descriptors are generated from FAST corners and LLAH descriptors are created from LLAH features for matching with pre-trained features.

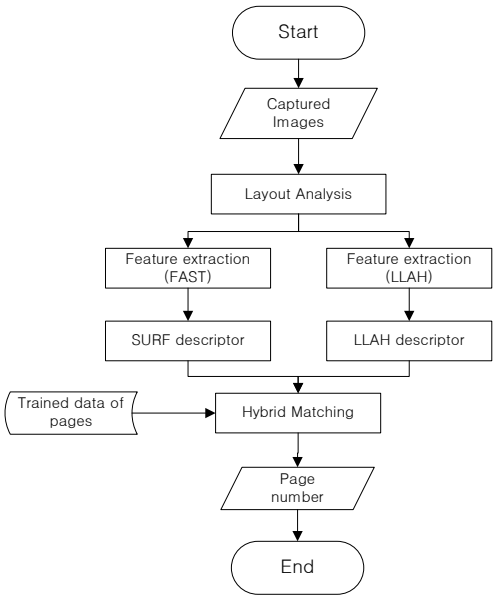


Fig. 3: Flow chart of the proposed method.

As the first step, an input image is divided into two parts: illustrations and text by layout analysis. The accurate layout of physical pages is not necessary because the matching is based on feature points. Therefore, rough layout is enough to identify each page of *Digilog Books*. After making a binary image from the image, CRLA (Constraint Run Length Algorithm) is applied to find long white lines in horizontal and vertical direction. From above-mentioned step, common regions in both directions are extracted. In order to make one text line into one block, the white small segments should be removed by applying horizontal direction again [13]. After finding some blocks, we find the contours of blocks except very small blocks that both width and height are less than 5 pixels. After all blocks are detected, each block is categorized into text or illustration. The normalized histogram of each block is utilized to determine the type of the block. Normally, the color of printed text is highly different from background color. Therefore, the histogram has a dominant peak if the background is not

mixed up much. If all blocks are categorized, text image and illustration image are generated from an input image according to the block types.

After classifying all blocks, proper features are extracted from each region. FAST corner detector is employed due to a fast processing time. In addition, the reference images are trained in multi-scale to compensate scale invariant factor. In the text case, we utilize all center points of words like LLAH. First, the text image is transformed into binary image and apply a Gaussian filter to remove noises. Then, dilation is conducted to make words into separated blocks. Finally, the center points of each block are considered as feature points. Figure 4 shows the result of feature extraction of both areas. The blue points denote FAST corners and red points represent LLAH features.

Fig. 4: Feature Extraction Result.

After finding matching correspondences from both regions, we need to merge both correspondences together. In the following equation, I is a correspondence set from text region and T is from illustration. Firstly, each set is sorted to satisfy following condition like PROSAC [3].

$$i_j, i_k \in I : j < k \Rightarrow q(i_j) \geq q(i_k), t_j, t_k \in T : j < k \Rightarrow p(t_j) \geq p(t_k) \qquad (1)$$

q and p represent correspondence quality function of illustration region and text region respectively. $q(i_j)$ is defined as $dist_{2nd}/dist_{1st}$ where $dist_{1st}$ is the nearest vector distance in k-d tree and $dist_{2nd}$ is the second nearest vector distance. For the text region, $p(t_j)$ is defined as V/L_{size} where V denotes the voting count when the point matched and L_{size} is the list size. After sorting each correspondence set, we create a set M by selecting correspondence alternately like under the following assumption.

$$M = I \cup T, M_{2i-1} = I_i, M_{2i} = T_i, min(\#ofI, \#ofT) > i > 0 \qquad (2)$$

$$q(i_j) \geq q(i_k) \Rightarrow P(i_j) \geq P(i_k), p(t_j) \geq p(t_k) \Rightarrow P(t_j) \geq P(t_k) \quad (3)$$

The reason why we select points from different region alternately is that homography from wide area is better than homography from small area. Next, we generate M_n which is subset of M and the number of elements is n. Then, M_n can be thought as the representative correspondence set which contains highest quality. Finally, initial homography is obtained from M_n by PROSAC. Figure 5 shows matching results on a page consisting of text and illustration by three different methods.

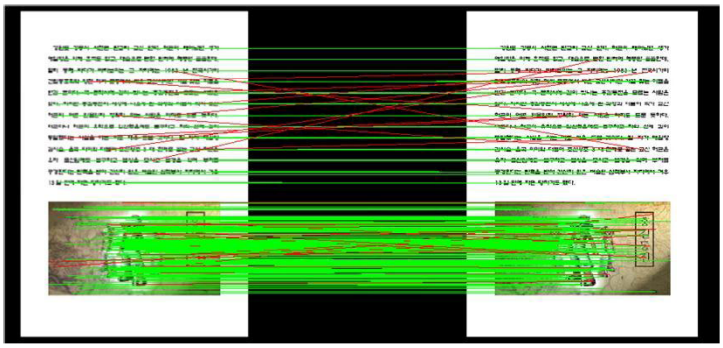
4 Experiments

We experiment the proposed method with LLAH and *FAST+SURF* in various layouts, different type of pages, and various noise levels. Intel Xeon 2.66GHz CPU and 3GB RAM were used for all experiments. Basically, the all image resolution was 640*480 and Gaussian smoothing was utilized for simulating noise test. In *FAST+SURF*, FAST (n=12) corners extracted with non-maximal suppression and 64 feature vectors for each feature was generated by SURF descriptor. In LLAH, 8 neighbor feature points were selected from each feature and each feature vectors discriminated by 15 levels that maximum value was 10. In addition, hash size was 200. The matching rate was defined as *"The number of inliers / The number of matching correspondences"*.

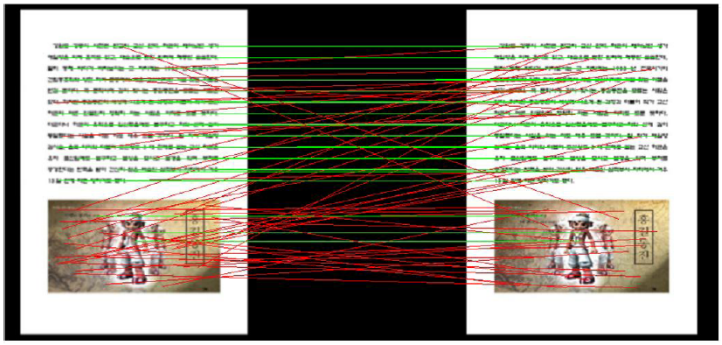
In order to check the performance of matching rate, we tested three methods on three types (text, illustration, both text and illustration) of image set without noise. The 9 test sets consisting of text images (two columns) was tested and each image was rotated by 10 degree increments along x axis. Therefore, 81 images, which content was only text, were tested. All methods recognized successfully from 10°to 50°and LLAH shown the best performance in text images. The performance of the proposed method was almost same as LLAH (Avg. 1.5% difference) as shown in figure 6. The reason of the difference was algorithm difference of finding matching algorithm.

The test set consisting of illustration images (one large figure) was rotated in the same way as text images. 81 images which content was only illustration were tested as well. In a different way with *FAST+SURF* in text images, LLAH was not able to detect reference images even in small rotation. Both *FAST+SURF* and the proposed method were able to detect pages from 10°to 60°. In addition, *FAST+SURF* showed better performance in illustration images than text images as we expected. The proposed method showed the similar performance (Avg. 0.2% difference) as *FAST+SURF* in illustration images. Figure 7 show the proposed method represents stable results according to the content type of pages.

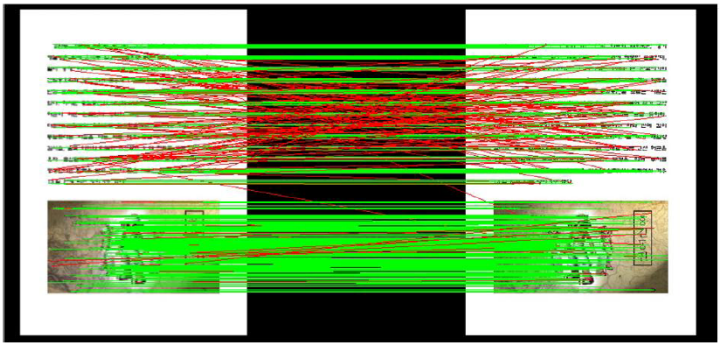
The matching rate of proposed method shown the better performance than *FAST+SURF* (Avg. 16.1% difference) and LLAH (Avg. 19.6% difference) respectively in the case of pages printed both illustration and text. Especially, in the case of 60°, the proposed method was able to detect pages using correspondences from both areas. It denotes that the performance of the proposed method



(a) The proposed method.



(b) LLAH.



(c) *FAST+SURF*.

Fig. 5: Comparison of matching results on a same document.

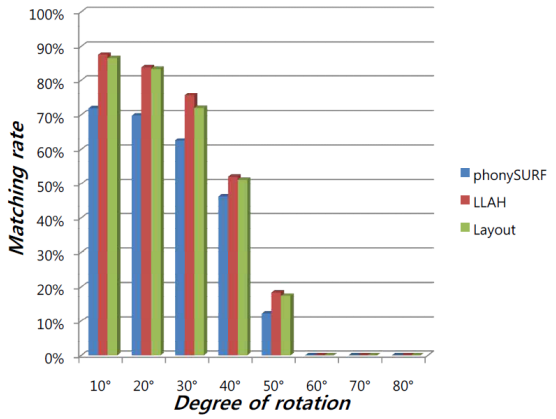


Fig. 6: Comparison of matching results on text images.

is not just sum of both methods. Figure 7 denotes overall performance of each algorithm according to content type. To sum up, layout based method shows reasonable performance in each type of pages according to the area of the green triangle in the graph. Moreover, the proposed method can be applied to three content types of image stably because its shape is almost regular triangle.

The second experiment was conducted in order to check robustness of the proposed method in noise images according to content type of pages. As the first experiment, each method was tested in three types of pages (text, illustration, and text & illustration). Each image was blurred with Gaussian smoothing with difference (from 0.5 to 2.5).

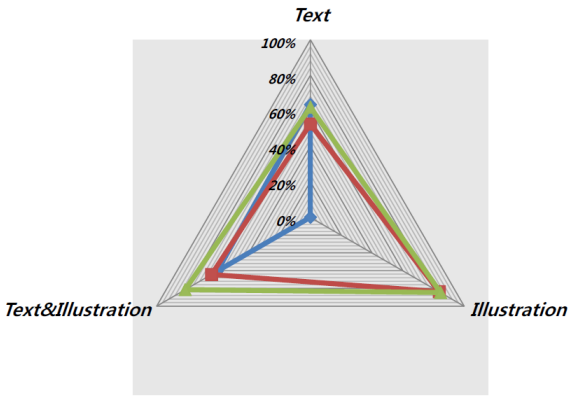


Fig. 7: Matching rate on various content types from 10 to 50 degree of rotation.

Figure 8 shows the matching rate of each method with different noise level in text images. Even though *FAST+SURF* showed reasonable performance in the first experiment, this graph verify why it cannot be applied to text images. The proposed method showed similar result (Avg. 2.6% difference) with LLAH and much better performance than *FAST+SURF* (Avg. 40.8% difference) in text images. The reason why the proposed method has different performance with LLAH was accuracy of layout analysis process. When images are blurred, layout classification did not work correctly. If some blocks were missed in layout analysis process, neighbor features of the blocks also cannot be matched correctly because of high dependency between features. As the result, the matching rate became lower than LLAH.

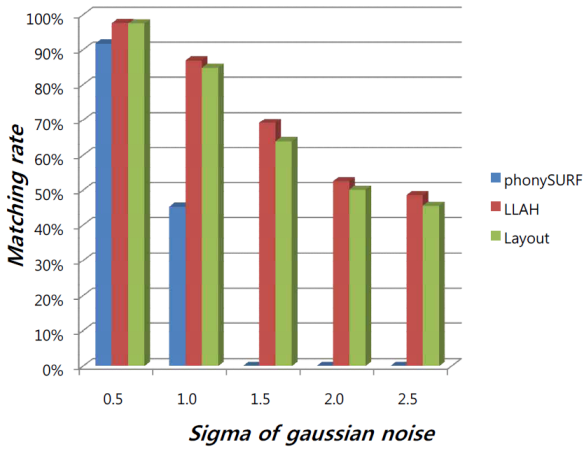


Fig. 8: Matching rate on text images according to various Gaussian noises.

Figure 9 denotes matching result under various Gaussian noises on illustration images. As we expected, LLAH was not able to detect the pages like *FAST+SURF* in text. *FAST+SURF* showed quite robust performance under various image noises. The proposed method showed very similar performance from 0.5 to 1.0. However, the gap with *FAST+SURF* was increasing as noise level increased. In the case of large size of illustration, some small blocks were generated inside the illustration because of fixed constraint in CRLA. In addition, when the blocks were classified as a text, the proposed method was not able to match in the blocks. However, the proposed method also shown reasonable performance in various noise level.

In summary of experiment about noise, figure 9 demonstrates the maximum detectable noise level of each algorithm. We decided the detectable noise as 40% matching rate. As the graph shown, the proposed method showed the most robust matching result in all types of pages. In the case of LLAH, it showed the higher matching rate than *FAST+SURF* except illustration case. The reason

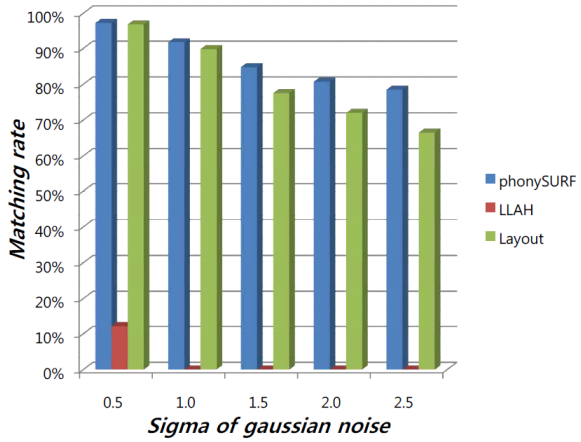


Fig. 9: Matching rate on illustration images according to various Gaussian noises.

why LLAH had much higher matching rate in mixed images was LLAH detected only few correspondences in illustration area. On the other hand, *FAST+SURF* detected many correspondences in text area and most of them were outliers.

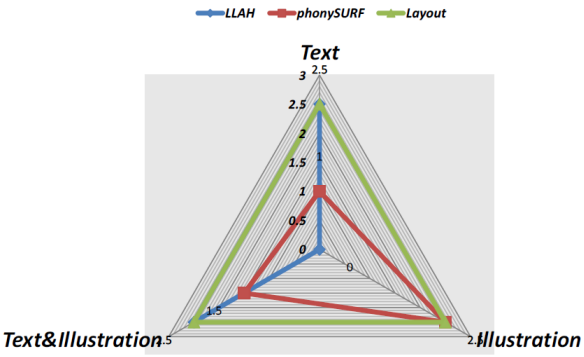


Fig. 10: Maximum detectable noise on various contents.

5 Conclusion and future work

The goal of this research is to transmute existing books into *Digilog Books*. Existing methods such as marker and natural feature, has some limitations to be applied for achieving the goal. Therefore, we proposed layout based matching for *Digilog Books*, which applies proper algorithms according to type of areas (*FAST+SURF* for illustrations and LLAH for text). The proposed method

showed higher matching rate if pages consist both text and illustration. Consequently, PROSAC finds homography matrix and removes outliers, took less time than single methods in mixed images consisting text and illustration while maintaining accuracy.

Acknowledgments. This research is supported by Ministry of culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA), under the Culture Technology (CT) Research & Development Program 2011.

References

1. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. Computer Vision–ECCV 2006 pp. 404–417 (2006)
2. Billinghurst, M., Kato, H., Poupyrev, I.: The magicbook-moving seamlessly between reality and virtuality. Computer Graphics and Applications, IEEE 21(3), 6–8 (2001)
3. Chum, O., Matas, J.: Matching with PROSAC-progressive sample consensus. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. vol. 1, pp. 220–226. Ieee (2005)
4. Ha, T., Lee, Y., Woo, W.: Digilog book for temple bell tolling experience based on interactive augmented reality. Virtual Reality pp. 1–15 (2010)
5. Kato, H., Billinghurst, M.: Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In: Proc. 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR '99). pp. 85–94 (20–21 Oct 1999)
6. Kim, K., Lepetit, V., Woo, W.: Scalable real-time planar targets tracking for digilog books. The Visual Computer 26(6), 1145–1154 (2010)
7. Lowe, D.: Distinctive image features from scale-invariant keypoints. International journal of computer vision 60(2), 91–110 (2004)
8. Nakai, T., Kise, K., Iwamura, M.: Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval. Lecture Notes in Computer Science 3872, 541 (2006)
9. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. Computer Vision–ECCV 2006 pp. 430–443 (2006)
10. Taketa, N., Hayashi, K., Kato, H., Noshida, S.: Virtual pop-up book based on augmented reality. LECTURE NOTES IN COMPUTER SCIENCE 4558, 475 (2007)
11. Uchiyama, H., Saito, H.: Augmenting Text Document by On-Line Learning of Local Arrangement of Keypoints. Proc. 8th IEEE/ACM International Symposium on Mixed and Augmented Reality ISMAR 2009 pp. 95–98 (2009)
12. Wagner, D., Reitmayr, G., Mulloni, A., Drummond, T., Schmalstieg, D.: Pose tracking from natural features on mobile phones. In: Proc. 7th IEEE/ACM International Symposium on Mixed and Augmented Reality ISMAR 2008. pp. 125–134 (Sep 15–18, 2008)
13. Wahl, F., Wong, K., Casey, R.: Block segmentation and text extraction in mixed text/image documents. Computer Graphics and Image Processing 20(4), 375–390 (1982)