Real-time Detection and Recognition of Machine-Readable Zones with Mobile Devices

Andreas Hartl, Clemens Arth and Dieter Schmalstieg
Institute for Computer Graphics and Vision, Graz University of Technology, Inffeldgasse 16, 8010 Graz, Austria

Keywords: Machine-Readable Zone, Text, Detection, Recognition, Document Inspection, Mobile Phone.

Abstract: Many security documents contain machine readable zones (MRZ) for automatic inspection. An MRZ is intended to be read by dedicated machinery, which often requires a stationary setup. Although MRZ information can also be read using camera phones, current solutions require the user to align the document, which is rather tedious. We propose a real-time algorithm for MRZ detection and recognition on off-the-shelf mobile devices. In contrast to state-of-the-art solutions, we do not impose position restrictions on the document. Our system can instantly produce robust reading results from a large range of viewpoints, making it suitable for document verification or classification. We evaluate the proposed algorithm using a large synthetic database on a set of off-the-shelf smartphones. The obtained results prove that our solution is capable of achieving good reading accuracy despite using largely unconstrained viewpoints and mobile devices.

1 INTRODUCTION

Checking travel or identity documents is a common task. Especially in situations with a large throughput of individuals, the time for checking such documents is very limited. The machine readable zone (MRZ) found on documents such as passports, visas and ID cards was introduced with the goal to speed up identity checks and to avoid human error in reading textual ID data (ICAO, 2008). There are three different types of MRZ, usually placed on the identity page of machine-readable travel documents. They consist of a paragraph with two or three parallel lines of black OCR-B text (fixed width and size) with fixed inter-line distance. These lines contain personal information about the owner, information about the document, and various checksums.

Reading MRZ data usually requires dedicated machinery, be it stationary or mobile. In the context of mobile application, there is also additional hardware, which can be attached to standard mobile phones[1][2] Besides, there are mobile applications, which claim to support robust reading of MRZ data from the built-in camera of the device (Smart 3D OCR MRZ[3], ABBY on Device OCR[4], Keesing AuthentiScan[5] or Jumio FastFill/Netverify[6]). All approaches have in common, that the MRZ must be aligned with the image capture device before the actual reading operation can take place. This requirement prolongs reading time and thus runs against the original intention of machine-readable travel documents.

We want to stress the fact that although the MRZ is designed to be read by automatic machinery, solving the task in such a general setting as proposed in this work is far from trivial, as is the character recognition. As there is no prior knowledge about the presence of a MRZ, the algorithm has to identify the area of interest automatically in real-time, despite motion blur and all other adversities emerging in real-world mobile phone image acquisition. The subsequent character recognition algorithm is challenged by the need for *perfect* recognition performance, to make the overall system competitive - we will show that our approach provides an adequate solution to these problems.

The main contribution of this work is a real-time solution for detecting and recognizing Machine-Readable Zones on arbitrary documents using off-the-shelf mobile devices without additional hardware. In contrast to current mobile applications that use the

[1] http://www.access-is.com
[2] http://www.movion.eu/grabba
[3] http://smartengines.biz
[4] http://www.abbyy-developers.eu
[5] https://www.keesingtechnologies.com
[6] https://www.jumio.com

Hartl A., Arth C. and Schmalstieg D..
Real-time Detection and Recognition of Machine-Readable Zones with Mobile Devices.
DOI: 10.5220/0005294700790087
In *Proceedings of the 10th International Conference on Computer Vision Theory and Applications* (VISAPP-2015), pages 79-87
ISBN: 978-989-758-091-8

Figure 1: Top Row: Robust detection of Machine-Readable Zones despite perspective distortion. Bottom Row: Reading results and runtime (Samsung Galaxy S5).

built-in camera, much broader variations in the viewing direction are tolerated during reading. In addition, visual feedback about the status of the operation is given to the user in order to serve as visual guidance for the process (see Figure 1). Since there is no publicly available database for developing and evaluating MRZ reading algorithms, we also contribute a large database of synthetic MRZ data, covering a broad range of diverse acquisition settings, backgrounds and view points, which we will release publicly. The database is used to evaluate our approach, giving a baseline for future developments in MRZ reading.

2 RELATED WORK

Standard CV techniques for text detection can be mainly categorized into texture-based and component-based approaches. In texture-based approaches, sliding windows and a subsequent classifier are used for robust detection. However, the computation of text masks for an OCR stage may require additional effort. Component-based approaches tend to compute single characters through segmentation and group them together to form lines or words. Relevant approaches are often rather efficient and provide text masks as a by-product. However, such bottom-up approaches require region filtering operations for improved robustness. Although an MRZ has a regular structure, solving the detection and recognition task is closely related to reading text in natural scenes.

2.1 Text Detection and Recognition

(Liu and Sarkar, 2008) detect candidates for text regions with local adaptive thresholds (binarization). They perform a grouping step considering geometric properties, intensity and shape. (Zhu et al., 2007) segment and detect text using binarization and subsequent boosting in a cascade for reduction of runtime. (Liu et al., 2012) use an extended local adaptive thresholding operator, which is scale-invariant. Regions are filtered using character stroke features and are then grouped using a graph structure. Color clustering is used by (Kasar and Ramakrishnan, 2012) to produce text candidates. They use twelve different features (geometry, contour, stroke, gradient) in a filtering stage employing a Support Vector Machine (SVM).

Maximally Stable Extremal Regions (MSER), first proposed by (Matas et al., 2002) are used by (Merino-Gracia et al., 2012) in a system for supporting visually impaired individuals. They employ graph-based grouping on filtered regions for the final result. (Donoser et al., 2007) use MSER to track and recognize license plate characters. (Neumann and Matas, 2011) extend MSER using topological information and conduct exhaustive search on character sequences, followed by a grouping step and SVM-based validation. They consider powerful grouping to

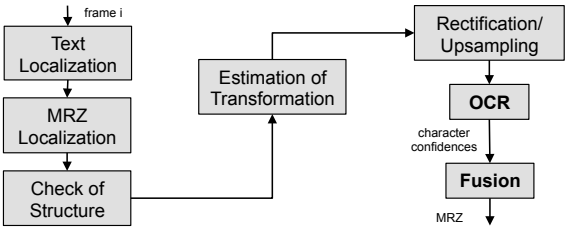

Figure 2: Outline of our algorithm for mobile MRZ reading. The MRZ structure is detected from text groups. Then, individual characters are rectified using an estimated transformation and fed into a custom OCR stage. Several frames are fused together for better performance.

be important for getting good results. (Álvaro Gonzalez et al., 2012) combine MSER with local adaptive thresholds and also use an SVM-based classifier for detection of characters.

There are several works which use morphological operations to segment text regions. (Fabrizio et al., 2009a) detect text in street-level images using toggle-mapping and SVM-based validation. (Minetto et al., 2010) extended this regarding scale-invariance. In addition, a Fuzzy HOG descriptor can be added for improved performance (Minetto et al., 2011).

(Epshtein et al., 2010) exploit the observation of constant character stroke width using a novel image operator called Stroke-Width Transform. This is based on the evaluation of opposing gradients on the basis of an edge map. They employ several filtering operations to obtain words. (Neumann and Matas, 2012) detect text using extremal regions, which are invariant regarding blurred images, illumination, color, texture and low contrast. Their approach employs a subsequent classification step (Boosting, SVM).

(Saoi et al., 2005) use wavelet-coefficients for text detection. (Mishra et al., 2012) first detect characters using HOG features and a SVM in a sliding window. They also use a lexicon-based prior and combine the available information in an optimization step. (Sun et al., 2010) evaluate several gradient images and verify the result using a visual saliency operator. (Yi and Tian, 2011) compute regions based on gradients and color information. They propose two different algorithms for grouping, which have a major impact on accuracy. (Pan et al., 2011) follow a hybrid approach by computing boosted HOG features and binarization with region computation.

2.2 Real-time Application on Mobile Phones

Real-time reading of MRZ data on mobile devices is different from performing this task on stationary devices. Due to limitations of the camera resolution and processing capabilities (CPU, GPU, RAM), only images with lower resolution can be processed, if constant feedback and responsive application behavior is desired. An efficient localization is desirable, because it allows to give immediate feedback to the user. For this reason, initial tests were carried out using component-based approaches due to the real-time requirements of the task. We experimented with several approaches such as local adaptive thresholding (Shafait et al., 2008), (Bataineh et al., 2011), Maximally Stable Extremal Regions (Matas et al., 2002) and Stroke-Width Transform (Epshtein et al., 2010). However, we found none to be suitable regarding a reasonable trade-off between segmentation performance and computing requirements. Subsequent experiments with segmentation based on Toggle-Mapping (Fabrizio et al., 2009b) gave promising results. Although this approach generates more noise than most competitors, this can be handled in a subsequent filtering stage.

3 ALGORITHM

We identified a set of properties for text on documents - in particular for the MRZ - which are useful for detection and reading. Text regions on documents are generally much smaller than text-like distortions in the background. A local region containing text normally consists of a single color with limited variation, and the stroke width of each character is roughly constant. All character boundaries are closed, and connecting lines on the contour are smooth. These boundaries correspond largely with edges detected in the input image. Single characters within text regions generally have very similar properties and are connected along an oriented line. In most cases, a minimum number of characters per text region can be assumed..

The approach we suggest for mobile MRZ reading works in four steps. First, the location of candidate text must be determined in the image. From this information, the MRZ is detected by considering the spatial layout between candidate groups. Then, a local transformation for each character is estimated, which can be used for rectification, followed by the recognition of characters, giving a confidence value w.r.t. each character of the relevant subset of the OCR-B font. Finally, information from several input frames is fused in order to improve the result (see Figure 2). We will now discuss these steps in more detail.

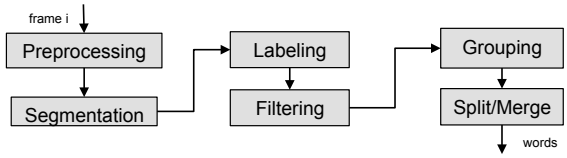

Figure 3: Outline of the text detection approach used in our framework. Connected components are obtained from an initial segmentation step, labeled and filtered. Then, they are pair-wise grouped and split into words, providing the basis for MRZ detection.

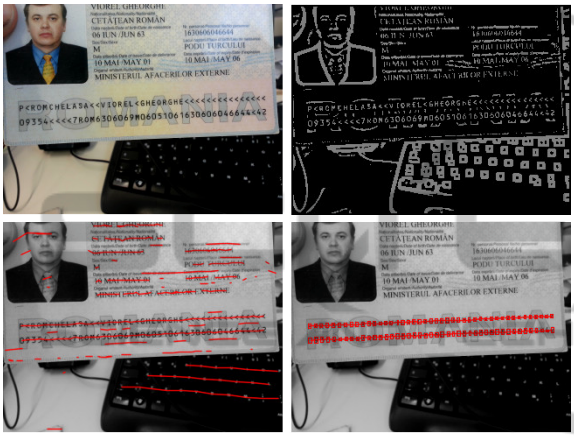

Figure 4: Steps in our algorithm. Input-image (top-left), filtered regions (top-right), filtered pairs from Delaunay triangulation (bottom-left), detection result (bottom-right).

3.1 Text Detection

We employ Toggle Mapping and linear-time region labeling as basic building blocks for initial generation of connected components (see Figure 3). Initial filtering is done based on region geometry and boundary properties (area, extension, aspect ratio, fill ratio, compactness). We also experimented with edge contrast and stroke width, but these did not improve results significantly at that stage.

Similar regions are grouped together based on region-properties and spatial coherence of characters. For reasons of efficiency, a Delaunay triangulation is used for getting an initial pair-wise grouping. Pairwise connections in the graph are then filtered using various relative criteria (height, distance, position-offset, area, angle, grey-value, stroke-width) followed by generation of strongly connected components (Tarjan, 1972). This gives a series of ordered groups, ideally representing single text words, but, depending on parametrization and document structure, several words can be contained (see Figure 4). Therefore, an additional filtering step is employed.

In a split/merge approach based on group properties (min. number of components, max./min. distances, direction, grey-value, area, stroke-width), final text groups are generated.

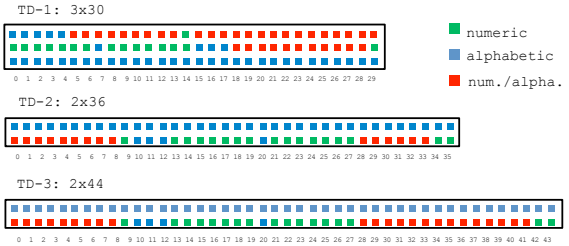

Figure 5: Structure of Machine-Readable Zones. There are three different types, which contain two or three lines of text. This corresponds to 90, 72 or 88 individual characters.

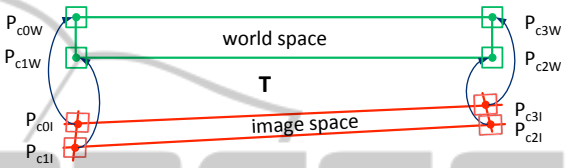

Figure 6: Rectification of Characters: First, a global transformation T is estimated using intersections points of fitted lines in image space and the corresponding world coordinates. Then, a local transformation can be estimated per character, which is then used for patch warping.

From the filtered groups, the individual components of the MRZ can be detected by analysis of their geometry. We search for groups fulfilling a minimum length requirement (30 characters). During selection, their horizontal and vertical distances are analyzed, finally giving a number of groups that are considered for processing in the optical character recognition stage.

3.2 Rectification

The detected characters can be rectified using MRZ structure information (see Figure 5). First, horizontal and vertical lines are fitted onto the detected MRZ components using linear regression on their centroids. These lines are further intersected in order to give improved estimates of the four outermost character centers P_{cI}. Using the known properties of the OCR-B font, corresponding coordinates P_{cW} can be computed in rectified (world) space, which allow to estimate a perspective transformation T. For each character centroid, as obtained from the intersection process, the limits of the patch can be determined in world space using font properties and then projected into the input image. Now a local transformation can be estimated for each character, which can be used for rectification (see Figure 6). In order to improve the input for the OCR stage, we perform up-sampling of character patches during warping.

3.3 Optical Character Recognition

The OCR stage uses the result of a subsequent binarization step as input data. We use Toggle Mapping for this task, label the obtained binary mask and estimate a minimum bounding box for the character.

Through a careful selection of frames, a small number of samples is sufficient for the recognition of single characters. We employ an overlap-metric for character recognition which is computed on a regular grid (Hu et al., 1999). We compute the local overlap for each cell and store it as a feature-vector. Using the L_1 distance, the similarity concerning a number of reference templates can be computed, which is also treated as a confidence value. We use ARM NEON[7] instructions in the matching stage in order to be able to deal with a higher number of template characters. We generated the initial samples by rendering true-type fonts and then added a small number of real samples, which were extracted using the proposed approach.

3.4 Frame Fusion

When working with live-video, several frames can be processed on the mobile device for improving robustness. For a subsequent fusion process, correspondences between characters must be established. In the fashion of tracking by detection, the structure of the initial detection result is considered whenever searching for suitable frames.

In each frame i, for every MRZ character j, distances $d_{i,j,k}$ concerning all known references k can be recorded. For each entry, the mean value w.r.t. all frames is computed: $d_{j,k} = mean(d_{i,j,k})$. The final result per character is then computed as the one having the smallest distance: $q_i = max(q_{j,k})$.

4 SYNTHETIC MRZ DATASET

Due to legal issues, it is not possible to get hold of a large number of identity documents for evaluation. Therefore a large database for developing and evaluating MRZ reading algorithms is not publicly available.

We collected a set of different ID documents and passports from Google images, using only images marked as free for modification and distribution. We sorted those documents according to their MRZ type and systematically removed the MRZ through impainting. We then use these document templates with different backgrounds and render both the document and a randomly generated MRZ string of the corresponding type. The MRZ string is generated by leveraging a public database of common names[8], using different nationality codes[9] and adding a random time stamp as the birth date, the date of issue and the date of expiry. Through this generic approach, we can create any number of example documents, single images and also entire frame sequences. The total number of different MRZ is over 11.000, the number of individual images is more than 90.000. The properties of the final database are listed in Table 1.

Table 1: Properties of the synthetic database. It contains over 11000 different Machine-Readable Zones in more than 90000 individual images.

Database Properties			
# Background Images	hard 10	medium 10	easy 10
Image Resolution		640x480	
# ID Documents	Type 1 10	Type 2 24	Type 3 4
# Single Images	24,000	57,600	9,600
# Image Sequences	100	240	40
# different MRZ	3,100	7,440	1,240
Total database size		22.5 GB	

Single Images. To generate realistic views of the documents, typical viewpoints are simulated by transformation and rendering of the current template-MRZ combination. In order to mimic typical user behavior, small local changes in transformation are introduced to create a number of images around a selected global setting. Noise and blur is added to the rendered document to increase realism. These documents are considered for the evaluation of algorithms based on single snapshots. Some sample images are depicted in Figure 7. To also allow for ID document *detection* algorithms to work on the proposed dataset, different backgrounds are used to reflect different levels of detection complexity.

Image Sequences. As mobile devices can be used to acquire entire frame sequences dynamically, we also created a set of frame sequences. We recorded several motion patterns of a mobile device over a planar target, storing the calculated pose for each frame (Wagner et al., 2010). The average length of these sequences is about 100 frames. For each frame, we render the template-MRZ combination using the previously recorded pose onto frames from a video taken

[7] http://www.arm.com/products/processors/technologies/neon.php

[8] https://www.drupal.org/project/namedb

[9] https://www.iso.org/obp/ui/#search/code

Figure 7: Single MRZ documents placed in front of a cluttered background image. Backgrounds with different complexities are used, starting from almost uniform to completely cluttered.

Figure 8: Top: Sequences of frames rendered onto a random background, and the corresponding camera trajectory. For better visibility, only every 25^{th} frame is drawn as a frustum. Bottom: Sample frames from two sequences. As the document is rendered into a video, the background changes with each frame.

at a public train station. Thereby we also allow the evaluation of approaches which are able to detect and track a document and combine the reading results over multiple frames. Sample camera paths and corresponding rendered image sequences are shown in Figure 8.

5 EVALUATION

In the following experiments, we determine the accuracy of MRZ detection, character reading and runtime for all relevant steps of the proposed approach. We evaluate a prototype of the MRZ reader on various mobile devices running Android and iOS operating systems with images from the aforementioned database[10].

5.1 Reading Accuracy

Reading accuracy is evaluated using single and multiple frames taken from the single-image database (see Table 1). While individual character recognition is barely affected by using more frames, the performance of MRZ detection is noticeably increased (see Figure 9). A MRZ detection rate of 88.18% is achieved by using five frames, along with a character

[10]A submission video can be found here: http://tinyurl.com/moq5ya2

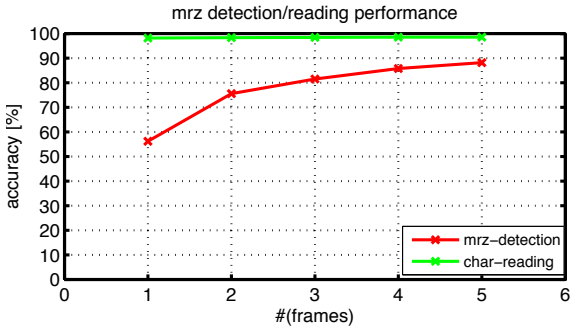

Figure 9: MRZ detection and character reading accuracy (single-image database): While individual character recognition is barely affected by using more frames, the performance of MRZ detection is noticeably increased. Note: Character reading results are given relative to successful detection.

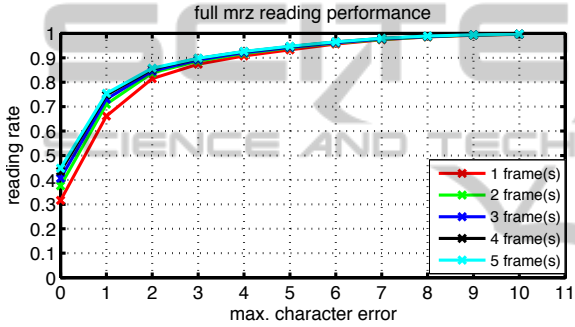

Figure 10: Full MRZ reading accuracy (single-image database): Despite reasonable character recognition rates, reading entire MRZs is still difficult, since no dictionary can be used for most parts. However, fusion of reading results from several frames improves reading rates by up to 15%.

reading rate of 98.58%. In terms of successful detection, this is a significant improvement over processing single shots (detection: 56.12%) on low-resolution mobile-images in real-time. Although the detection and reading of individual characters works reasonably well, getting correct readings for the entire MRZ is still a challenging task, since no dictionary can be used for large parts of the MRZ (see Figure 10). However, frame fusion helps to improve the results by up to 15%.

When considering image sequences, a detection rate of 99.21% is achieved. From these detections 98,9% of characters where successfully read. Considering all frames of a sequence, a fully correct MRZ can be read in 63.40% of all cases.

Obviously, MRZ detection performance and character reading are related to the input pose (see Figures 13, 14). We can observe that the proposed approach can detect and read MRZ data despite perspective distortion, saving document alignment time for the user. Most gaps seem to be caused by segmentation artifacts, which cause unresolvable ambiguities in the grouping stage. However, the largest gap for the exemplary sequence consists of just three frames, which corresponds to a maximum waiting time of 0.1 s for getting processable data, or 0.5 s when fusing five frames (assuming a framerate of 30 FPS).

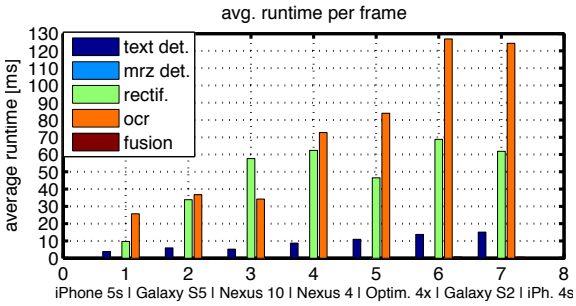

Figure 11: Runtime of the prototype for various mobile devices (iOS, Android): Runtime is dominated by patch warping and optical character recognition. In particular, the timespan needed for MRZ detection from text groups and the final fusion step is negligible. Detection and reading from a single frame takes around 35 ms on the iPhone 5s.

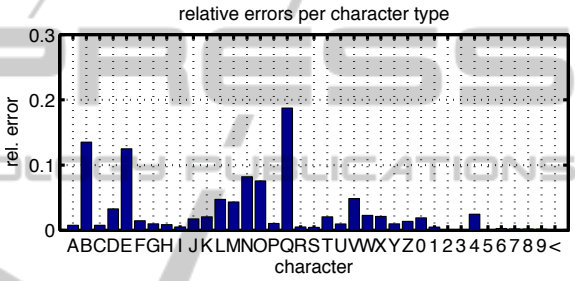

Figure 12: Errors for individual characters: In most cases, the characters B,E and Q are confused with others. Note: Individual errors are given relative to the sum of all errors across all characters.

5.2 Algorithm Runtime

Runtime is dominated by the OCR part of the algorithm, the rectification, segmentation and feature computation (see Figure 11), while the initial text detection and subsequent fusion operations take up only a fraction of the overall runtime.

In total, reading a single MRZ takes around 35 ms on the iPhone 5s (iOS). The closest competitor is the Samsung Galaxy S5 smartphone (Android), taking around 70 ms per frame. The iPhone 5s gains most of its speed-up during warping of individual characters. On our development machine (MBP i7, 2 GHZ), the overall runtime per frame is around 14 ms.

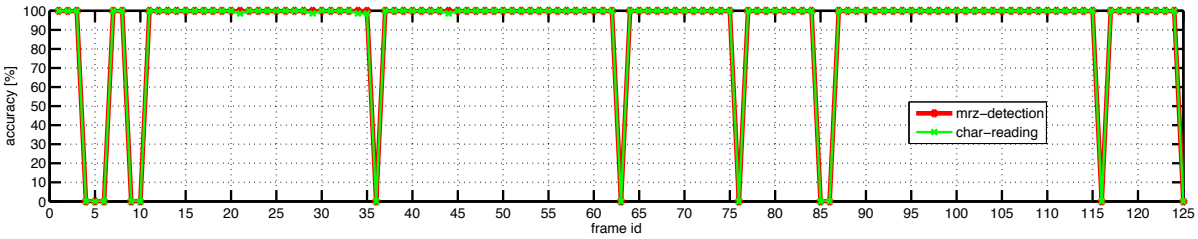

Figure 13: Exemplary result of processing an entire image sequence of the synthetic database. The maximum gap size is three frames, which corresponds to a waiting time of 0.1 s, until processable data arrives (assuming a framerate of 30 FPS).

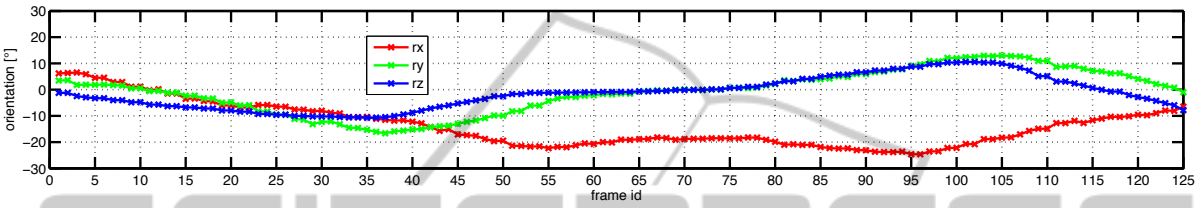

Figure 14: Corresponding orientation throughout the exemplary image sequence. The example document is captured from viewpoints that differ considerably from the ideal setting.

6 DISCUSSION

Based on the results of our experimental evaluation, some individual aspects deserve further discussion as follows.

MRZ Detection. Detection from a single frame is difficult, as it might fail if the document is viewed under steep angles. The overall MRZ recognition process therefore clearly benefits from using a continuous video feed (see Figure 9). Due to the efficiency of our approach, frames can be processed in real-time and instant feedback can be given to the user. Due to the larger amount of data, missing single frames is not critical for the task at hand.

Character Recognition. Although reasonable character recognition rates (exceeding 90%) could be obtained during our evaluation, a closer inspection reveals that in most cases, the current prototype confuses the characters B, E and Q with similar samples (see Figure 12). Beside character confusion, occasional issues in character segmentation make up most of the remaining cases due to region splits. This could be improved by using more advanced pre-processing or a machine-learning approach on the extracted patches (e.g., SVM).

It is important to note that for full MRZ reading, a heavily tuned character recognition engine has to be employed, suffering from a failure rate of at most $1e^{-4}\%$. Given the fact that real-world samples are hardly to be found in large quantities, this turns out to be a challenging problem on its own.

Image Resolution. We found that using a video stream with higher resolution (i.e., Full HD) in our mobile prototype only gives small improvements over fusing multiple frames with lower resolution, as proposed in this paper. When processing such a stream on Android, there is noticeable latency even though the full resolution is only used in the OCR stage. Due to this delay, there can be a lot of change between subsequent frames, causing occasional blur depending on user behavior. Since this is particularly undesirable regarding usability, it seems reasonable to stick with low or medium resolution images, employ an advanced frame selection strategy (e.g., dep. on sharpness or lighting) and to further improve the OCR stage. Our aim is to create synthetic character samples with different kinds of noise and other distortions in order to mimic all kinds of acquisition conditions and settings, and to employ different machine learning techniques to improve upon the current approach.

7 CONCLUSIONS

We presented an approach for real-time MRZ detection and reading, which does not require accurate alignment of the document or the MRZ. By initial MRZ detection and fusion of results from several input frames, our custom OCR stage produces reasonable character reading results despite having to deal

with unaligned input. For evaluation purposes, we introduced a new synthetic database, which covers many different document backgrounds, MRZ contents and viewpoints (available on request). Saving the time required for alignment, MRZ data can be extracted faster than with state-off-the-art mobile applications.

Our approach could be improved in various ways. If more character training data becomes available, the template matching could be replaced with a suitable classifier. This would certainly help to improve full MRZ reading results including runtime. Tracking the MRZ should increase robustness, since more input data would be available for the OCR stage. For practical reasons, slightly bent documents should also be handled.

ACKNOWLEDGMENTS

This work is supported by Bundesdruckerei GmbH.

REFERENCES

Álvaro Gonzalez, Bergasa, L. M., Torres, J. J. Y., and Bronte, S. (2012). Text location in complex images. In *ICPR*, pages 617–620.

Bataineh, B., Abdullah, S. N. H. S., and Omar, K. (2011). An adaptive local binarization method for document images based on a novel thresholding method and dynamic windows. *Pattern Recogn. Lett.*, 32(14):1805–1813.

Donoser, M., Arth, C., and Bischof, H. (2007). Detecting, tracking and recognizing license plates. In *ACCV*, pages 447–456, Berlin, Heidelberg. Springer-Verlag.

Epshtein, B., Ofek, E., and Wexler, Y. (2010). Detecting text in natural scenes with stroke width transform. In *CVPR*, pages 2963–2970.

Fabrizio, J., Cord, M., and Marcotegui, B. (2009a). Text extraction from street level images. In *CMRT*, pages 199–204.

Fabrizio, J., Marcotegui, B., and Cord, M. (2009b). Text segmentation in natural scenes using toggle-mapping. In *ICIP*, pages 2349–2352.

Hu, J., Kashi, R., and Wilfong, G. (1999). Document classification using layout analysis. In *1999. Proceedings of the International Workshop on Database and Expert Systems Applications*, pages 556–560.

ICAO (2008). Machine readable travel documents.

Kasar, T. and Ramakrishnan, A. G. (2012). Multi-script and multi-oriented text localization from scene images. In *CBDAR*, pages 1–14, Berlin, Heidelberg. Springer-Verlag.

Liu, X., Lu, K., , and Wang, W. (2012). Effectively localize text in natural scene images. In *ICPR*.

Liu, Z. and Sarkar, S. (2008). Robust outdoor text detection using text intensity and shape features. In *ICPR*.

Matas, J., Chum, O., Urban, M., and Pajdla, T. (2002). Robust wide baseline stereo from maximally stable extremal regions. In *BMVC*, pages 36.1–36.10.

Merino-Gracia, C., Lenc, K., and Mirmehdi, M. (2012). A head-mounted device for recognizing text in natural scenes. In *CBDAR*, pages 29–41, Berlin, Heidelberg. Springer-Verlag.

Minetto, R., Thome, N., Cord, M., Fabrizio, J., and Marcotegui, B. (2010). Snoopertext: A multiresolution system for text detection in complex visual scenes. In *ICIP*, pages 3861–3864.

Minetto, R., Thome, N., Cord, M., Stolfi, J., Precioso, F., Guyomard, J., and Leite, N. J. (2011). Text detection and recognition in urban scenes. In *ICCV Workshops*, pages 227–234.

Mishra, A., Alahari, K., and Jawahar, C. V. (2012). Top-down and bottom-up cues for scene text recognition. In *CVPR*.

Neumann, L. and Matas, J. (2011). Text localization in real-world images using efficiently pruned exhaustive search. In *ICDAR*, pages 687–691. IEEE.

Neumann, L. and Matas, J. (2012). Real-time scene text localization and recognition. In *CVPR*, pages 3538–3545.

Pan, Y.-F., Hou, X., and Liu, C.-L. (2011). A hybrid approach to detect and localize texts in natural scene images. *IEEE Transactions on Image Processing*, 20(3):800–813.

Saoi, T., Goto, H., and Kobayashi, H. (2005). Text detection in color scene images based on unsupervised clustering of multi-channel wavelet features. In *ICDAR*, pages 690–694. IEEE Computer Society.

Shafait, F., Keysers, D., and Breuel, T. (2008). Efficient implementation of local adaptive thresholding techniques using integral images. In *SPIE DRR*. SPIE.

Sun, Q., Lu, Y., and Sun, S. (2010). A visual attention based approach to text extraction. In *ICPR*, pages 3991–3995.

Tarjan, R. E. (1972). Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160.

Wagner, D., Reitmayr, G., Mulloni, A., Drummond, T., and Schmalstieg, D. (2010). Real-time detection and tracking for augmented reality on mobile phones. *TVCG*, 16(3):355–368.

Yi, C. and Tian, Y. (2011). Text string detection from natural scenes by structure-based partition and grouping. *IEEE Transactions on Image Processing*, 20(9):2594–2605.

Zhu, K.-h., Qi, F.-h., Jiang, R.-j., and Xu, L. (2007). Automatic character detection and segmentation in natural scene images. *Journal of Zhejiang University SCIENCE A (JZUS)*, 8(1):63–71.