Are Characters Objects?

Markus Diem and Robert Sablatnig
Vienna University of Technology, Austria
Institute of Computer Aided Automation
Computer Vision Lab
diem@caa.tuwien.ac.at

Abstract

This paper presents a character recognition system that handles degraded manuscript documents like the ones discovered at the St. Catherine's Monastery. In contrast to state-of-the-art OCR systems, no early decision (image binarization) needs to be performed. Thus, an object recognition methodology is adapted for the recognition of ancient manuscripts. The proposed system is based on local descriptors which are clustered in order to localize characters. Finally, a class probability histogram is assigned to each character present in an image which allows for the character classification. The system achieves an $F_{0.5}$ score of 0.77 on real world data that contains 13.5% highly degraded characters.

1 Introduction

State-of-the-art character recognition systems tread characters as texture or patterns on writing material. This generally implies that a correct binarization of characters is possible (to our knowledge, all state-of-the-art OCR systems need binarized images). However, character binarization is still challenging [5], if degraded characters are considered. In contrast to OCR systems, modern object recognition systems are able to classify and localize objects correctly without binarizing images. In this paper ancient characters are considered to be objects without sharp edges, low contrast and non-rigid transformations. This implies a substantial difference of the methodology compared to state-of-the-art OCR systems.

Offline character recognition systems generally consist of three steps [13]. First a pre-processing step is applied where the images are binarized and enhanced (e.g. background removal, line segmentation). Afterwards a segmentation of characters or words is performed. Finally, features of the fragments, characters or words are computed which are subsequently classified. This architecture has proven to give good results for general document recognition ($60.0\% - 99.3\%$) [13]. However, there exist alphabets like the glagolitic for which a generally designed character recognition system fails, since words are not separated and the manuscripts are degraded. In this paper Glagolica, the oldest slavonic alphabet, is considered. More precisely, the Missale (*Cod. Sin. Slav.* 5N) is analyzed which was written in the 11^{th} century and discovered in 1975 at the St. Catherine's Monastery [9].

The manuscripts are degraded due to bad storage conditions and environmental effects. The partially faded-out ink cannot be handled with state-of-the-art binarization methods [6, 11]. Figure 1 shows a sample of the glagolitic manuscript and the corresponding binary image when the Sauvola thresholding method is applied having tuned parameters (k, R). In addition the correct glagolitic character is overlaid. Preliminary tests showed that these characters can be recognized with the proposed system (Figure 1 c).

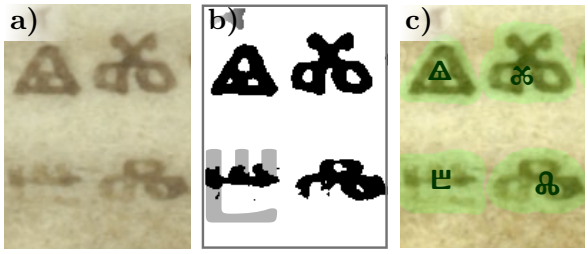

Figure 1. A detail of a glagolitic manuscript page with degraded characters a), binarized characters using Sauvola thresholding b) and the classification results of the proposed system c).

The system presented in this paper is based on local descriptors. Thus, the pre-processing step with the

image binarization can be completely rejected which allows a recognition of highly degraded characters. Having computed the local descriptors for a whole manuscript page, they are classified using a Support Vector Machine (SVM). Subsequently, the characters are located by clustering the interest points. A voting scheme of the classified local descriptors finally recognizes characters.

This paper is organized as follows. The subsequent section covers related work focusing on off-line handwriting recognition. In Section 3 the proposed system is detailed. Afterwards, in Section 4 the system's results on synthetic data and the investigated dataset are presented. Section 5 concludes the paper.

2 Related Work

In this section state-of-the-art OCR systems for degraded or ancient manuscripts are presented. The approaches differ according to the data investigated. Thus, two general data sets are differentiated: cursive handwritten documents and ancient manuscripts. Figure 2 illustrates documents of the particular datasets. The Georg Washington document in Figure 2 (left) can be correctly binarized since the background is homogeneous. However, a correct character segmentation is hard as stated in [7] because of the cursive script. In contrast, the Hebrew manuscript in Figure 2 contains background clutter because of ink on the reverse that bleeds through.

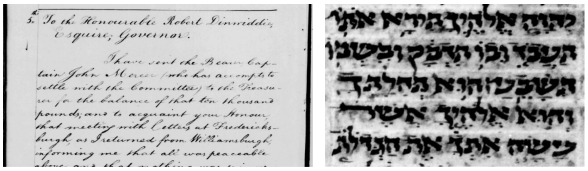

Figure 2. Cursive manuscript (left) [7] and Hebrew manuscript (right) [14].

Lavrenko et al. [7] try to directly recognize words from the George Washington collection. Therefore, previously segmented words need to be normalized according to the slant, skew and baseline. Then, scalar features such as the word's width or aspect ratio and profile-based features (e.g. projection profiles) are computed on the normalized word images. A Hidden Markov Model (HMM) with hidden states that represent words is used to classify the words. Lavrenko reports a precision on the George Washington collection of 0.603. This technique was later improved by Rath et al. [10] who propose to use dynamic time warping in order to compensate non-linear variations present in manuscripts.

Similar to the previously mentioned methods, a word recognition system is proposed by Frinken et al. [4]. They compute statistical moments from sliding windows that are applied to normalized word images. A Neural Network (NN) with one hidden layer is constructed for the classification. In addition the a priori data distribution is trained by means of semi-supervised learning that is fed with labeled and unlabeled data. Frinken et al. [5] additonally combine this methodology with HMM's in order to improve the word recognition.

In contrast to the word recognition methods, Alirezaee et al. [1] developed a character recognition system for medieval Persian manuscripts. They extract statistical features such as Pseudo-Zernike moments from previously binarized document images. In order to find features that are discriminant, the Fisher Linear Discriminant is used which transforms the data such that the inter-class variance is maximized. The resulting weight function classifies characters.

Arrivault et al. [2] propose a combined statistical and structural character recognition approach for ancient Greek and Egyptian documents. Therefore two statistical features namely Fourier moments and Zernike moments are extracted from binary document images. According to the dictionary's size a Bayes or k-NN classifier is used to label characters according to statistical features. Structural features such as attributed graphs are computed and classified for characters which are rejected during the classification of statistical features.

Another approach that aims at recognizing historical Greek documents is published by Vamvakas et al. [12]. Having binarized the image and segmented individual characters, zone features and character profile features are calculated. The first of which are constructed by tiling the character image into zones and accumulating the character pixel density to the normalized zone image. Unlabeled character features are then clustered according to the features extracted. In a manual step, labels are assigned to the clusters and clustering errors can be corrected. Finally, a SVM is exploited for character classification.

3 Methodology

In contrast to state-of-the-art systems, the proposed system has a fundamentally new architecture which is inspired by modern object recognition systems. It is designed to compensate the drawbacks that arise when dealing with ancient manuscripts. Instead of applying a binarization so as to compute features, they are directly extracted from the gray-scale image.

The system is divided into two major tasks: classification and localization. Both tasks are based upon the extraction of interest points. First local descriptors are

computed at locations of interest points which are directly classified. A comparison of local descriptors for character recognition tasks is given in [3]. Afterwards, the characters are localized by clustering the interest points.

3.1 Feature Extraction & Classification

In order to extract interest points – independent to scale changes – a scale-space is computed. It is constructed by convolving the image repeatedly with a Gaussian kernel. Differencing successive levels of the Gaussian scale-space results in a so-called Difference-of-Gaussians (DoG) [8]. Thus a second order derivative scale-space is constructed. This allows for detecting blob like regions at local extrema. Hence, the interest points are located at local extrema of the DoG scale-space. Therefore the interest points detect character attributes such as junctions, endings, stroke borders, corners and circles. Having found the interest points of a document image, local descriptors can be computed at their location.

The proposed system implements the Scale Invariant Feature Transform (SIFT) which was proposed by Lowe [8]. These high-dimensional features are computed by means of the image's gradient magnitude and gradient orientation. In order to compute them rotationally invariant, the main orientation is estimated for each interest point. Normalizing the feature vector according to the main orientation allows for a representation that is independent to rotational changes. SIFT descriptors are 128-dimensional gradient histograms. More precisely, 8 orientation histograms with 4×4 bins are created. Each orientation histogram represents gradient vectors with specific orientations $(0°, 45°, 90°, ..., 315°)$. Thus, gradient vectors of an interest point are accumulated to the gradient histograms according to their orientation and their spatial location (4×4 bins). A tri-linear interpolation guarantees a robust representation of a specific image region.

The local descriptors represent parts of characters such as junctions, corners or endings as well as whole characters and parts of text lines. Even though characters have similar shapes (e.g. ♌ ♍) their local structure alters since the stroke direction changes for different characters. Thus, the local descriptors are distinctive enough to recognize 36 characters.

That is why a multi-kernel SVM is trained using 20 samples per character class. The classifier consists of one Radial Basis Function (RBF) per character class. The local descriptors are classified by means of one-against-all tests. This results in a class probability histogram (see Figure 3) where each bin represents a character class and the bin's magnitude defines the probability of a local descriptor for belonging to the specific character class.

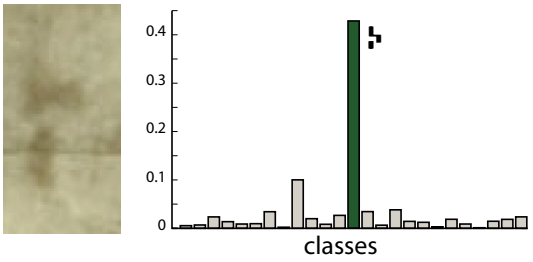

Figure 3. Class probability histogram

The classification results are presented in Figure 4. Gray blobs illustrate manually tagged ground truth data. Correctly classified interest points are illustrated by circular markers, where rectangles indicate falsely classified features. Small white rectangles illustrate features outside the evaluated domain. The black circles represent the interest points' scales.

Having classified the local descriptors, a spatially blurred character probability of a document page is known. However, the exact location of characters and the relation between local descriptors is not known. Thus, it is not known which local descriptors represent one and the same character.

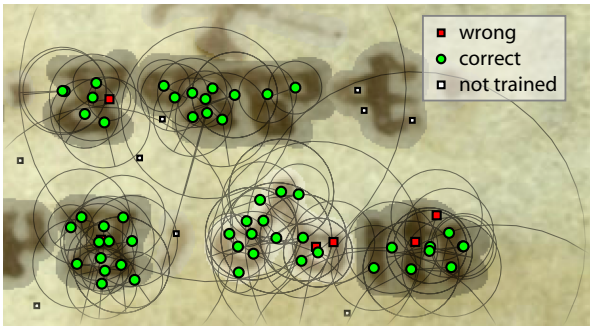

Figure 4. Classified local descriptors

3.2 Character Localization

For traditional OCR engines the characters or words are localized implicitly in the binarization step. If handwriting OCR engines are considered, an additional character segmentation step needs to be performed in order to detect concatenated characters. In contrast, the proposed system has no information about the positions of characters in a given image to the point of feature classification. Indeed, the positions of the classified features are known, but as a feature does not necessarily represent a whole character, their position and size is unknown.

The character localization is based on clustering the interest points. This approach benefits from the fact, that degraded characters are detected with local descriptors but not considered when the image is binarized. Thus, even degraded characters can be localized. Another advantage is the low computational complexity, since solely the interest points are considered (e.g. for a $436px \times 992px$ image that has totally $432512px$ 1543 interest points are detected).

Clustering algorithms such as the k-means cannot estimate the number of clusters k themselves. To overcome this problem the scales of interest points are exploited. Each character produces a single local maximum (interest point) in a certain scale level. These interest points are detected in a scale distribution histogram (see Figure 5). Having extracted interest points that correspond to characters, the number k of the k-means is known and at the same time initial cluster positions are obtained that improve convergence.

In order to find the minimum scale level s_t, the scale distribution of all features in a given image is regarded. Figure 5 shows the features' scale distribution. There, the abscissa shows increasing scales, particularly the radius of features measured in px. The ordinate gives the relative number of interest points corresponding to the scale interval. It can be seen that most interest points are detected in scale levels below $30px$. This results on the one hand from the higher resolution which decreases with increasing scale and on the other hand from the fact that manuscripts have high frequency features such as endings, junctions and corners. The scale levels corresponding to characters – which we are interested in – are within the second peak between $30px$ and $80px$. The third and last peak corresponds to interest points that represent text lines or low frequency features such as illumination changes or stains.

Thus, the minimum scale level s_t (rendered as a red vertical line in Figure 5) is defined as the inflection point after the first peak. This algorithm guarantees that even small characters[1] are localized for the k-means initialization. Generally more interest points are selected than characters are present in an image. This relies on the fact that background clutter – which produces interest points – is clustered together with characters, if too few initial cluster centers are obtained.

Afterwards, a k-means clustering is performed which is initialized with the previously found character interest points. Thus, the interest points are grouped together according to the subjacent characters.

For the final character classification a voting scheme is applied. Therefore, all local descriptors of a cluster are considered. As mentioned before, a probability histogram exists that indicates the class likelihood of each descriptor in the cluster. If these histograms are accumulated, the maximum bin indicates the most probable class label.

[1] $27 \times 34px$ compared to other characters which have $93 \times 33px$

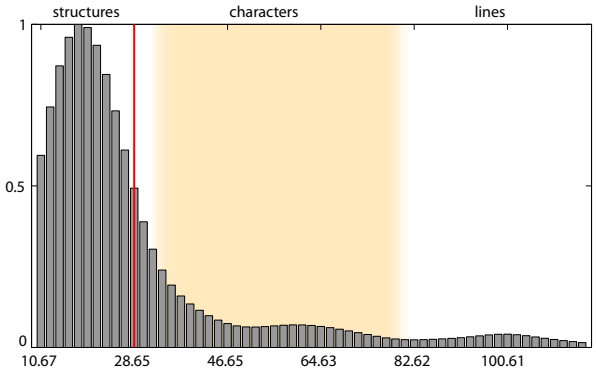

Figure 5. Interest points' scale distribution of a manuscript image.

However, directly averaging the descriptors' probabilities has drawbacks. Thus, descriptors which are larger than a character describe the structure of more than one character. Additionally, descriptors of background clutter are falsely clustered to characters. These incorrect descriptors adulterate the performance if a direct averaging is applied. That is why a weighting is developed that regards these observations:

$$w_i = 1 - \frac{s_i}{\max\limits_{j=0\ldots n}(s_j + c)} \tag{1}$$

where s_i is the ith descriptor's scale and w_i is the final weight. The constant $c > 0$ guarantees that the weight w_i is > 0 for all descriptors. Similarly the descriptors are weighted according to their distribution within the character cluster. Instead of the scale s_i, the descriptor's distance d_i to the cluster center is regarded. It turned out, that a robust cluster center (e.g. median) improves the weighting compared to the default center-of-mass. This is because the robust center penalizes outliers (the center-of-mass shifts towards outliers).

4 Results

In this section, results of the proposed system are given. It is intended to empirically evaluate the system by manually annotated real world data and synthetically generated data. The subsequent experiments show the strengths and drawbacks of the new character recognition methodology proposed in this paper. Three different experiments were carried out in order to analyze certain aspects which are detailed subsequently.

4.1 Synthetic Data

Tests with synthetic data are carried out to demonstrate the system's capability of being trained on different fonts. The training and test set are generated by rendering TrueType fonts into images. This allows for generating test images with arbitrary fonts and at the same time to automatically annotate the ground truth data which minimizes the human effort. The system is trained using Times New Roman (regular) and Arial (regular). These fonts are chosen in order to guarantee that the system is trained on Serif fonts and Sans Serif fonts. In all subsequent experiments 26 character classes (the English alphabet) are evaluated.

First the system is tested with the training set so as to guarantee that the implementation is correct. If all 52 characters are considered, 2 characters are falsely classified (precision: 0.962), namely: i and j when generated with Arial. This can be traced back to the fact that Sans Serif characters such as i, j, l exclusively produce SIFT features that represent corners with changing orientations. However, all remaining characters (e.g. h) produce the same corners at stroke endings. That is why the SVM cannot be trained properly for Sans Serif fonts.

In addition, the system's performance is evaluated if new fonts are presented. Therefore a test set containing three Serif fonts (namely: Times New Roman, Georgia, Garamond) and three Sans Serif fonts (namely: Arial, Helvetica, Tahoma) is generated. This results in 156 sample characters. In this experiment a precision of 0.763 is achieved. If weak character clusters are rejected ($m_b = 0.85$), the precision increases to 0.865.

Additionally white Gaussian noise is added to the synthetic characters. If the standard deviation σ of the Gaussian noise is set to 0.003, the precision is 0.923. Increasing the noise to $\sigma = 0.008$ decreases the system's precision to 0.904. Hence, the proposed system is robust with respect to Gaussian noise.

4.2 Character Evaluation

Single characters where extracted from the investigated dataset so that the classification and voting can be evaluated. Therefore two datasets are constructed that consist of single characters which were annotated and extracted from the *Cod. Sin. Slav.* 5N.

The first dataset (SETA) consists of 10 classes having 10 – 12 samples (totally 107) which are well preserved. This dataset is a reference for the evaluation with degraded characters. The second dataset which is referred to as SETB contains 9 characters per class. Degraded or partially visible characters where extracted to construct this set. It is used to demonstrate the systems' behavior when degraded characters need to be recognized.

Figure 6 shows examples of both datasets. It can be seen that some characters such as and  are similar to each other.

Figure 6. Examples of the datasets evaluated.

For the character classification an overall precision of 0.981 is achieved. Thus solely 2 characters out of 107 are falsely predicted. Both confused characters consist of two circles and a connecting stroke (see Figure 6 second and last column) which produce similar descriptors.

In contrast to SETA the degraded characters in SETB have a lower precision which is 0.789. Additionally, the ratio between local descriptors detected and descriptors classified is lower which is in this case 39% compared to 60% in SETA. These numbers indicate that it is harder for the system to classify degraded characters. On the other hand the system can cope with uncertainty which arises from the fact that fewer descriptors are classified in this case (see Table 1).

	#	# classes	precision
SETA	107	10	0.981
SETB	90	10	0.789

Table 1. Dataset, number of samples, number of classes and the system's precision

4.3 System Evaluation

In order to evaluate the system, 15 different pages containing 1055 characters are extracted from the *Cod. Sin. Slav.* 5N. The pages where chosen randomly. They contain faded-out ink, degraded characters and background noise. This is done to guarantee a statistically representative dataset of the investigated manuscripts. In the subsequent discussion results are presented that show the system's performance on good and degraded characters which were manually annotated beforehand. It is intended to show the system's behavior when solely good characters are considered and to draw conclusions about the character localization when degraded characters are considered.

Table 2 shows the system's recall, precision and F-score on the investigated dataset. It contains 142 degraded characters which are 13.5% of all characters

evaluated. If normal characters are regarded, a $F_{0.5}$-score of 0.79 is achieved. In contrast, degraded characters have a lower performance (namely: 0.38). This arises mainly from the fact that the recall is low due to 64 False Negatives which draws the conclusion that 45.1% of degraded characters are missed. When comparing these numbers to previous tests discussed in Section 4.2 where degraded characters were extracted, a performance loss can be observed. It can be attributed to the fact that no recall was obtained in this test since False Negatives do not exist if characters are manually extracted (all characters are classified).

	#	recall	precision	$F_{0.5}$-score
normal	913	0.732	0.862	0.792
degraded	142	0.296	0.539	0.382
SETB	198	-	0.712	0.712

Table 2. System's recall, precision and F-score when normal and degraded characters are considered. The last row shows the character evaluation from Section 4.2.

5 Conclusion

A new methodology for character recognition of ancient manuscripts was presented. The approach which is inspired by recent object recognition systems exploits local descriptors directly extracted from grayscale images. Hence, the system does not need any preprocessing of document images.

Experiments with synthetic data, show that the proposed system performs better on manuscripts since typewritten characters have the very same corners and junctions for different characters. Thus, the local information remains the same which is not the case if manuscripts are considered. A dataset was created that consists of highly degraded glagolitic characters. Experiments on this dataset proofed the system's capability to recognize degraded characters and the difference to well preserved characters. Additional tests with annotated ground truth allowed for analyzing the errors introduced by the character localization.

Since ancient manuscripts – in contrast to modern – exhibit stains, faded-out ink and rippled pages, new challenges are faced when trying to recognize characters. The system presented tries to consider these challenges by incorporating object recognition methods.

Acknowledgement This work was supported by the Austrian Science Fund under grant P19608-G12

References

[1] S. Alirezaee, H. Aghaeinia, K. Faez, and A. S. Fard. An Efficient Feature Extraction Method for the Middle-Age Character Recognition. In *Proceedings of the International Conference on Intelligent Computing*, pages 998–1006, 2005.

[2] D. Arrivault, N. Richard, C. Fernandez-Maloigne, and P. Bouyer. Collaboration Between Statistical and Structural Approaches for Old Handwritten Characters Recognition. In *Graph-based Representations in Pattern Recognition*, pages 291–300, 2005.

[3] M. Diem and R. Sablatnig. Recognition of Degraded Handwritten Characters Using Local Features. In *Proceedings of the 10th International Conference on Document Analysis and Recognition*, pages 221–225, Barcelona, Spain, 2009.

[4] V. Frinken and H. Bunke. Self-training Strategies for Handwriting Word Recognition. In *Proceedings of the 9th IEEE International Conference on Data Mining*, pages 291–300, 2009.

[5] V. Frinken, T. Peter, A. Fischer, H. Bunke, T. Do, and T. Artières. Improved Handwriting Recognition by Combining Two Forms of Hidden Markov Models and a Recurrent Neural Network. In *Proceedings of the 13th International Conference on Computer Analysis of Images and Patterns*, pages 189–196, 2009.

[6] B. Gatos, I. Pratikakis, and S. J. Perantonis. Adaptive degraded document image binarization. *Pattern Recognition*, 39(3):317–327, 2006.

[7] V. Lavrenko, T. M. Rath, and R. Manmatha. Holistic Word Recognition for Handwritten Historical Documents. In *DIAL*, pages 278–287, 2004.

[8] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[9] H. Miklas, M. Gau, F. Kleber, M. Diem, M. Lettner, M. Vill, R. Sablatnig, M. Schreiner, M. Melcher, and G. Hammerschmid. St. Catherine' s Monastery on Mount Sinai and the Balkan-Slavic Manuscript-Tradition. In *Slovo: Towards a Digital Library of South Slavic Manuscripts.*, pages 13–36, Sofia, Bulgaria, 2008. "Boyan Penev" Publishing Center.

[10] T. M. Rath and R. Manmatha. Word spotting for historical documents. *IJDAR*, 9(2-4):139–152, 2007.

[11] J. J. Sauvola and M. Pietikäinen. Adaptive document image binarization. *Pattern Recognition*, 33(2):225–236, 2000.

[12] G. Vamvakas, B. Gatos, N. Stamatopoulos, and S. Perantonis. A Complete Optical Character Recognition Methodology for Historical Documents. *IAPR International Workshop on Document Analysis Systems*, 1:525–532, 2008.

[13] A. Vinciarelli. A survey on off-line Cursive Word Recognition. *Pattern Recognition*, 35(7):1433–1446, 2002.

[14] I. B. Yosef. Input sensitive thresholding for ancient Hebrew manuscript. *Pattern Recognition Letters*, 26(8):1168–1173, 2005.