TECHNISCHE FAKULTÄT

Department Informatik

Technical Reports / ISSN 2191-5008

Robert Nagy, Anders Dicker, Klaus Meyer-Wegener

Definition and Evaluation of the NEOCR Dataset for Natural-Image Text Recognition

Technical Report CS-2011-07

September 2011

Please cite as:
Robert Nagy, Anders Dicker, Klaus Meyer-Wegener, "Definition and Evaluation of the NEOCR Dataset for Natural-Image Text Recognition," University of Erlangen, Dept. of Computer Science, Technical Reports, CS-2011-07, September 2011.

Friedrich-Alexander-Universität Erlangen-Nürnberg

Friedrich-Alexander-Universität Erlangen-Nürnberg
Department Informatik

Martensstr. 3 · 91058 Erlangen · Germany

www.informatik.uni-erlangen.de

Definition and Evaluation of the NEOCR Dataset for Natural-Image Text Recognition

Robert Nagy, Anders Dicker, Klaus Meyer-Wegener
Chair for Computer Science 6 (Data Management)
Dept. of Computer Science, University of Erlangen, Germany
robert.nagy@cs.fau.de

***Abstract*—Recently growing attention has been paid to recognizing text in natural images. Natural image text OCR is far more complex than OCR in scanned documents. Text in real world environments appears in arbitrary colors, font sizes and typefaces, often affected by perspective distortion, lighting effects, textures or occlusion. Currently there is no dataset publicly available that covers all aspects of natural image OCR. A comprehensive well-annotated configurable dataset for optical character recognition in natural images is defined and created for the evaluation and comparison of approaches tackling with natural-image text OCR. Furthermore, current open source and commercial OCR tools have been analyzed in various test scenarios using the proposed NEOCR dataset. Based on the results further steps to be addressed by the OCR community are concluded towards all-embracing natural-image text recognition.**

***Index Terms*—optical character recognition; OCR; dataset; text detection; scene text recognition; natural image text recognition; latin characters**

I. Introduction

Optical character recognition (OCR) for machine-printed documents and handwriting has a long history in computer science. For clean documents, current state-of-the-art methods achieve over 99% character recognition rates [1].

With the prevalence of digital cameras and mobile phones, an ever-growing amount of digital images are created. Many of these natural images contain text. The recognition of text in natural images opens a field of widespread applications, such as:

- help for visually impaired or blind (e.g., reading text not transcribed in braille in [2]),
- mobile applications (e.g., translating photographed text for tourists and foreigners as shown in [3, 4] or using the knfbReader [5] and Word Lens [6]),
- object classification (e.g., multimodal fusion of text and visual information as described in [7]),
- image annotation (e.g., for web search in [8])
- vision-based navigation and driving assistant systems as suggested in [9].

Recently growing attention has been paid to recognizing text in real world images, also referred to as natural-image text OCR [4] or scene text recognition (STR) [1]. Natural images are far more complex in contrast to machine-printed documents. Problems arise not only from background variations, but from the depicted text too, which usually takes on a great variety of appearances. In some cases even for humans it is very hard to draw the line between text and architectural forms. In addition to the survey of [2], which compared the capturing devices (scanners and digital cameras), we summarized main characteristics of scanned document OCR and scene text recognition in table I.

[10] identified image binarization and segmentation as one of the crucial points for high OCR accuracy in general. Because of the high variations of background, text and surrounding objects in natural images, detecting text and recognizing words becomes significantly harder for real world scenes. For the development, evaluation and comparison of techniques developed specifically for natural-image text OCR, a publicly available well annotated dataset is required.

All current datasets (see section II) annotate only the words and bounding boxes in images. Also most text appears in horizontal character arrangement, while in natural scenes humans are often confronted with text, where its characters are arranged vertically or circularly (text following a curved, wavy or circular line). Currently there is no well-annotated dataset publicly available that covers all aspects distinguish-

Criteria	Scanned documents	Natural-image text
background	homogeneous, usually white or light paper	any color, even dark or textured
blurredness	sharp (depending on scanner)	possibly motion blur, blur because of depth of field
camera position	fixed, document lies on scanner's glass plate	variable, geometric and perspective distortions almost always present
character arrangement	clear horizontal lines	horizontal and vertical lines, rounded, wavy
colors	mostly black text on white background	high variability of colors, also light text on dark background (e.g. illuminated text) or only minor differences between tones
contrast	good (black/dark text on white/light background)	depends on colors, shadows, lighting, illumination, texture
font size	limited number of font sizes	high diversity in font sizes
font type (diversity in document)	usually 1-2 (limited) types of fonts	high diversity of fonts
font type (in general)	machine-print, handwriting	machine-print, handwriting, special (e.g. textured such as light bulbs)
noise	limited / negligible	shadows, lighting, texture, flash light, reflections, objects in the image
number of lines	usually several lines of text	often only one single line or word
occlusion	none	both horizontally, vertically or arbitrary possible
rotation (line arrangement)	horizontally aligned text lines or rotated by ± 90 degrees	arbitrary rotations
surface	text "attached" to plain paper	text freestanding (detached) or attached to objects with arbitrary nonplaner surfaces, high variability of distortions

TABLE I
Differences between OCR on scanned documents and natural-image text.

ing scene text recognition from scanned document OCR.

We propose the new NEOCR (Natural Environment OCR) dataset consisting of real world images extensively enriched with additional metadata. Based on this metadata several subdatasets can be created to identify and overcome weaknesses of OCR approaches on natural images. Main benefits of the proposed dataset compared to other related datasets are:

- annotation of all text visible in images,
- additional distortion quadrangles for a more precise ground truth representation of text regions,
- rich metadata for simple configuration of subdatasets with special characteristics for more detailed identification of shortcomings in OCR approaches.

Based on the defined NEOCR dataset leading current open source and commercial OCR applications have been evaluated. Thanks to the additional metadata the recognition performance could be analyzed for special characteristics of natural-image texts on a much deeper level of detail than ever before. The resulting conclusions give an extensive overview of the current state and further steps for scene text recognition.

The report is organized as follows: In the next section we give a short overview of currently available datasets for OCR in natural images and compare them to our proposed NEOCR dataset. We describe the construction of our new dataset and the annotated metadata in section III. In section IV we give an overview of distance functions for string comparison. Experiments using open source and commercial OCR software are presented in section V. We discuss the results of the experiments in section VI and conclude further steps for natural-image text recognition in section VII.

Dataset	#images	#boxes	avg. #char/box
ICDAR 2003	509	2263	6.15
Chars74K	312	2112	6.47
Microsoft Text DB	307	1729	10.76
Street View Text	350	904	6.83
NEOCR	**659**	**5238**	**17.62**

TABLE II
Comparison of natural-image text recognition datasets.

II. Datasets for Natural-Image Text Recognition

Unfortunately, publicly available OCR datasets for scene text recognition are very scarce. The ICDAR 2003 dataset [11] of [12, 13] is the most widely used in the community. In the ICDAR 2003 Robust Reading dataset 258 training and 251 test images have been annotated with bounding boxes and the caption of the contained text. Although the images in the dataset show a considerable diversity in font types, the images are mostly focused on the depicted text and the dataset contains largely indoor scenes depicting book covers or closeups of device names. The dataset doesn't contain any vertically or circularly arranged text at all. The high diversity of natural images, such as shadows, light changes, illumination, character arrangement (e.g. vertical text) is not covered in the dataset.

Recent progress in natural-image OCR resulted in several new datasets. The Chars74K dataset [14] introduced by [15] focuses on the recognition of Latin and Kannada characters in natural images. The dataset contains 1922 images mostly depicting sign boards, hoardings and advertisements from a frontal viewpoint. About 900 images have been annotated with bounding boxes for characters and words, of which only 312 images contain latin word annotations. Unfortunately, not all words visible in the images have been annotated and images with occlusion, low resolution or noise have been excluded.

[4] proposed the Street View Text dataset [16], which is based on images harvested from Google Street View [17]. The dataset contains 350 outdoor images depicting mostly business signs. At total 904 rectangular textfields were annotated. Unfortunately, bounding boxes are parallel to the axes, which is insufficient for marking text in natural scenes. Another deficit is that not all words depicted in the image have been annotated.

In [18] a new stroke width based method was introduced for text recognition in natural scenes. The algorithm was evaluated using the ICDAR 2003 dataset and additionally on a newly proposed dataset (Microsoft Text DB [19]). The 307 annotated images cover the characteristics of natural images more comprehensively as in the ICDAR dataset. Unfortunately, not all text visible in the images has been annotated and the bounding boxes are parallel to the axes.

Additionally some special datasets of license plates, book covers or digits have been used in different publications. Still sorely missed is a well-annotated dataset covering the aspects of natural images comprehensively, which could be applied for comparing different approaches and identifying gaps in natural-image text recognition.

Ground truth annotations in the related datasets presented above are limited to bounding box coordinates and text transcriptions. Therefore, our comparison of current datasets is limited to statistics on the number of annotated images, the number of annotated textfields (bounding boxes) and the average number of characters per textfield. The Chars74K dataset is a special case, because it contains word annotations and redundantly its characters are also marked. For this reason, only annotated words with a length bigger than 1 and consisting of latin characters or digits only were included in the statistics in table II.

Compared to other datasets dedicated to natural-image OCR the NEOCR dataset contains much more annotated bounding boxes. Because not only words, but also phrases have been annotated in the NEOCR dataset, the average text length per bounding box is also higher. None of the related datasets includes additional metadata information for the annotated bounding boxes. NEOCR surpasses all other natural-

image OCR datasets with its rich additional metadata, which enables more detailed evaluations and more specific conclusions on weaknesses of OCR approaches.

III. The NEOCR Dataset

A comprehensive dataset with rich annotation for OCR in natural images is introduced. The images cover a broad range of characteristics that distinguish real world scenes from scanned documents. Example images from the dataset are shown in figure 1.

The dataset contains a total of 659 images with 5238 bounding boxes (textfields). Images were captured by the authors and members of the lab using various digital cameras with diverse camera settings to achieve a natural variation of image characteristics. Afterwards, images containing text were hand-selected with particular attention to achieving a high diversity in depicted text regions. The first release of the NEOCR dataset covers the dimensions discussed in this section each by at least 100 textfields. Figure 2 shows examples from the NEOCR dataset for typical problems in natural-image OCR.

Based on the rich annotation of optical, geometrical and typographical characteristics of bounding boxes, the NEOCR dataset can also be tailored into specific datasets to test new approaches for specialized scenarios. Additionally to bounding boxes, distortion quadrangles were marked in the images too for a more accurate ground truth annotation of text regions and automatic derivation of rotation, scaling, translation and shearing values. These distortion quadrangles also enable a more precise representation of slanted text areas close to each other, which usually overlap when using bounding boxes only with their sides parallel to the axes.

Fig. 1. Example images from the NEOCR dataset. Note that the dataset also includes images with text in different languages, text with vertical character arrangement, light text on dark and dark text on light background, occlusion, normal and poor contrast.

For image annotation, the web-based annotation tool of [20] for the LabelMe dataset [21] was used. Due to the simple browser interface of LabelMe the NEOCR dataset can be extended continuously. Annotations are provided in XML for each image, separately describing global image features, bounding boxes of text and its special characteristics. The XML-schema of LabelMe has been adapted and extended by tags for additional metadata. The annotation metadata is discussed in more detail in the following sections.

A. Global Image Metadata

General image metadata contains the filename, folder, source information and image properties. For each whole image its width, height, depth, brightness and contrast are annotated. Brightness values are obtained by extracting the luma channel (Y-channel) of the images and computing the mean value. The standard deviation of the luma channel is annotated as the contrast value. Both brightness and contrast values are obtained using ImageMagick [22].

B. Textfield Metadata

All words and coherent text passages appearing in the images of the NEOCR dataset are marked by bounding boxes. Coherent text passages are several lines of text in same font size and typeface, color, texture and background (e.g., as they usually appear on memorial plaques or signs). All bounding boxes are rectangular and parallel to the axes. Additionally annotated distortion quadrangles inside the bounding boxes give a more accurate representation of text regions. The metadata is enriched by optical, geometrical and typographical characteristics.

1) Optical Characteristics: Optical characteristics contain information about the blurredness, brightness, contrast, inversion (dark text on light background or vice versa), noise and texture of a textfield.

a) Texture: Texture is very hard to measure automatically, because texture differences can form the text and text itself can be a texture too. Following three categories have been defined:

(a) emboss, engrave

(b) lens blur

(c) perspective distortion

(d) crop, rotate, occlusion, circular

(e) textured background

(f) textured text

Fig. 2. Example images from the NEOCR dataset depicting typical characteristics of natural-image text recognition.

- low: single color text with single color background,
- mid: multi-colored text or multi-color background,
- high: multi-colored text and multi-colored background, or text without a continuous surface (e.g., luminous advertising built from light bulbs).

b) Brightness and contrast: Brightness and contrast values for bounding boxes are obtained the same

way as for the whole image (see section III-A). As an attribute of the contrast characteristic we additionally annotate whether dark text is represented on light background or vice versa (inverted).

c) Resolution: In contrast to 1000dpi and more in high resolution scanners, images taken by digital cameras only achieve resolutions up to 300dpi. The lower the focal length, the bigger the area captured by the lens. Depending on the pixel density and the size of the camera sensor small text can turn unrecognizable. As a measure we define text resolution as the number of pixels in the bounding box divided by the number of annotated characters.

d) Noise: Image noise can originate from the noise sensitivity of camera sensors or from image compression artifacts (e.g., in JPEG images). Usually, the higher the ISO values or the higher the compression rates, the bigger the noise in the images. Because noise and texture are difficult to distinguish we classify the bounding boxes into low, mid and high noise judged by eye.

e) Blurredness: Image blur can be divided into lens and motion blur. Lens blur can result from depth of field effects when using large aperture depending on the focal length and focus point. Similar blurring effects can result from image compression too. Motion blur can originate either from moving objects in the scene or camera shakes by the photographer. [23] and [24] give overviews on different approaches for measuring image blur. As a measure for blurredness we annotated the kurtosis value to the bounding boxes. First edges are detected using a Laplacian-of-Gaussian Filter (LoG). Afterwards the edge image is Fourier transformed and the steepness (kurtosis) of the spectral analysis is computed. The higher the kurtosis, the more blurred the image.

2) Geometrical Characteristics: Character arrangement, distortion, occlusion and rotation are subsumed under geometrical characteristics.

a) Distortion: Because the camera sensor plane is almost never parallel to the photographed text's plane, text in natural images usually appears perspectively distorted. Several methods can be applied to represent distortion. In our annotations we used 8 floating point values as described in [25]. The 8 values can be represented as a matrix, where s_x and s_y describes scaling, r_x and r_y rotation, t_x and t_y translation, and p_x and p_y shearing:

$$\begin{pmatrix} s_x & r_x & p_x \\ r_y & s_y & p_y \\ t_x & t_y & 1 \end{pmatrix} \tag{1}$$

The equations in [25] are defined for unit length bounding boxes only. We adapted the equations for arbitrary sized bounding boxes. The derivation of the equations is discussed in detail in appendix A. Based on the matrix and the original coordinates of the bounding box, the coordinates of the distorted quadrangle can be computed using the following two equations:

$$x' = \frac{s_x x + r_y y + t_x}{p_x x + p_y y + 1} \tag{2}$$

$$y' = \frac{r_x x + s_y y + t_y}{p_x x + p_y y + 1} \tag{3}$$

b) Rotation: Because of arbitrary camera directions and free positioning in the real world, text can appear diversely rotated in natural images. The rotation values are given in degrees as the offset measured from the horizontal axis given by the image itself. The text rotation is computed automatically based on the distortion parameters. Further details and equations are discussed in appendix B.

c) Arrangement: In natural images characters of a text can be arranged vertically too (e.g., some hotel signs). Also, some text follows curved, wavy or circular baselines. In the annotations we distinguish between horizontally, vertically and circularly arranged text. Single characters were categorized as horizontally arranged.

d) Occlusion: Depending on the chosen image detail by the photographer or objects present in the image, text can appear occluded in natural images. Because missing characters (vertical cover) and horizontal occlusion need to be treated separately by OCR methods, we distinguish between both in our annotations. The amount of cover is annotated as percentage value.

3) Typographical Characteristics: Typographical characteristics contain information about font typefaces and languages.

a) Typefaces: Typefaces of bounding boxes are categorized into standard (print), handwriting and special categories. The annotated text is case-sensitive, the font size can be derived from the proportion of the resolution value and the size of the bounding box. Font thickness is not included in the annotation metadata.

b) Language: Languages can be a very important information when using vocabularies for correcting errors in recognized text. Because the images were taken in several countries, 15 different languages are present in the NEOCR dataset, though the visible text is limited to latin characters only. In some cases, text cannot be clearly assigned to any language. For these special cases we introduced categories for numbers, abbreviations, persons and business names.

C. Summary

Figure 3 shows statistics on selected dimensions for the NEOCR dataset. The graphs prove the high diversity of the images in the dataset. The more accurate and rich annotation allows more detailed inspection and comparison of approaches for natural-image text OCR.

Figure 4 shows a screenshot of the adapted LabelMe annotation tool with an example image. The corresponding annotation for the example image and the range of values for each metadata dimension are listed in table III. Further details for the annotations, the XML-schema and the dataset itself can be found on the NEOCR dataset website [26]. Some OCR algorithms rely on training data. For these approaches a disjoint split of the images in training and testing data is provided on the NEOCR dataset website.

Fig. 4. Example image from the NEOCR dataset. The annotated metadata is shown in table III.

IV. Distance Functions for String Comparison

In the evaluation based on the NEOCR dataset in section V the manually annotated ground truth is compared to the text recognized by OCR tools. When applying OCR on natural images, in most cases regions without any characters are detected as text too. Also single characters inside a word are sometimes misclassified by OCR methods.

Several distance functions and algorithms have been proposed to compare sequences. [27] and [28] give comprehensive overviews on the topic of sequence and string comparison. In this section the focus is limited to the most popular string comparison distance functions.

A. Definitions

We define $d(x_n, y_m)$ as the distance between strings x_n and y_m, which is the minimal cost of operations transforming string x_n into y_m. $\emptyset$ is the empty string and n is the length of string x_n. Transforming operations are defined as $\delta(a, b) = t$, where a and b are different characters and t is the assigned cost for the given transformation operation. From the mathematical perspective distance functions need to satisfy the four metric axioms:

- nonnegative property: $d(x_n, y_m) \geq 0$
- zero property: $d(x_n, y_m) = 0$ if and only if $x_n = y_m$
- triangle inequality: $d(x_n, y_m) + d(y_m, z_o) \geq d(x_n, z_o)$
- symmetry: $d(x_n, y_m) = d(y_m, x_n)$

The first three properties are always valid for all strings x_n and y_m. If the symmetry holds for transforming operations too ($\delta(a, b) = \delta(b, a)$), then the space of strings forms a metric space.

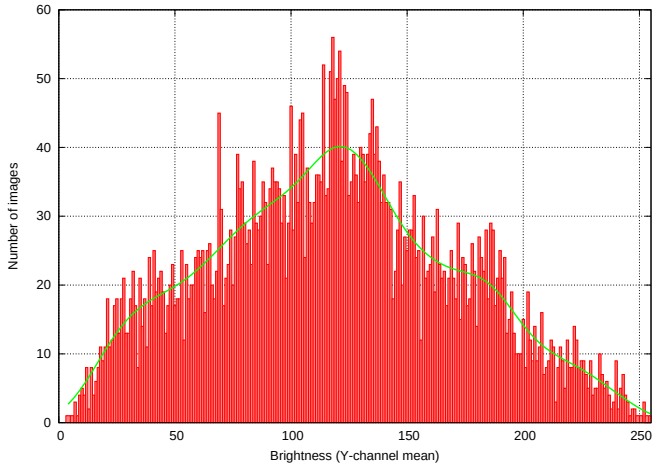
For strings, transforming operations are limited in [27] and [28] to following:

- insertion: $\delta(\emptyset, a)$, inserting character a
- deletion: $\delta(a, \emptyset)$, removing character a
- substitution (replacement): $\delta(a, b)$ for $a \neq b$, replacing character a by b
- transposition (swap): $\delta(ab, ba)$ for $a \neq b$, swapping adjacent characters a and b

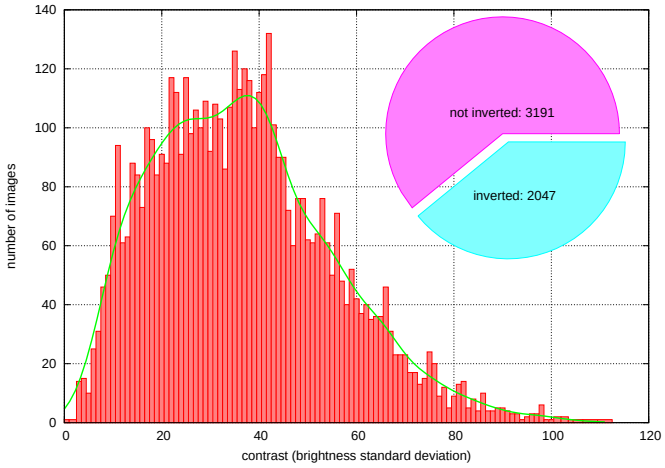
[27] also mentions compression of two or more characters into one character, and its reverse operation, expansion as transforming operations for sequence comparison. These are less relevant for string comparison and therefore they are not considered here.

B. Distance Functions

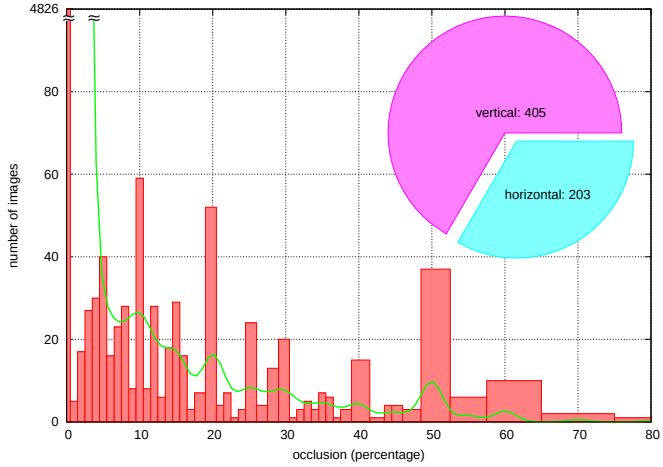
In this section the most popular string comparison distance functions are introduced in detail based on the notation presented in section IV-A.



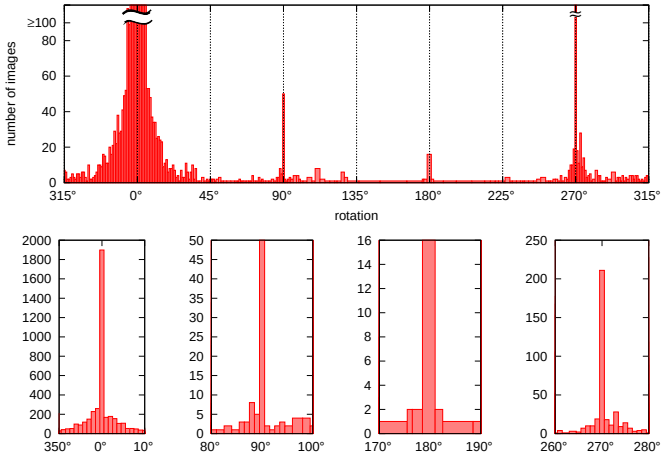
(a) brightness



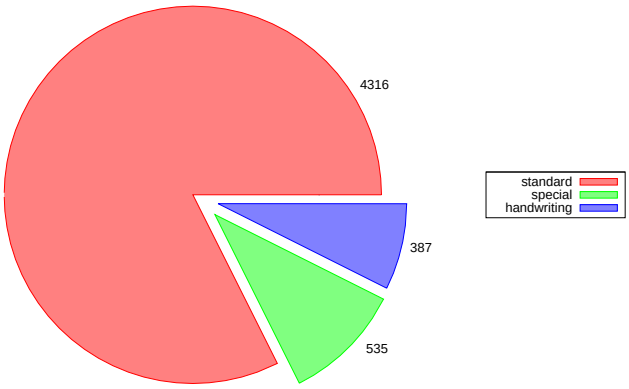
(b) contrast



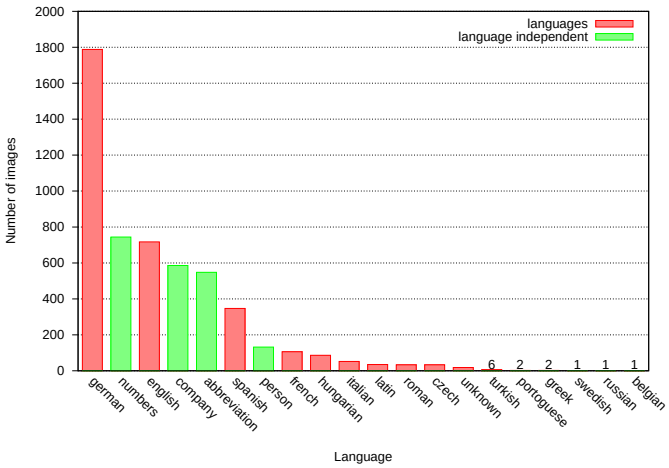
(c) occlusion



(d) rotation



(e) font



(f) language

Fig. 3. Brightness, contrast, rotation, occlusion, font and language statistics proving the diversity of the proposed NEOCR dataset. Graphs 3(a) and 3(b) also show the usual value of a scanned text document taken from a computer science book. The number of images refers to the number of textfields marked by bounding boxes.

Category	Datatype	Values range	Example value
texture	string	low, mid, high	mid
brightness	float	[0;255]	164.493
contrast	float	[0;123]	36.6992
inversion	boolean	true, false	false
resolution	float	[1;1000000]	49810
noise	string	low, mid, high	low
blurredness	float	[1;100000]	231.787
distortion	8 float values	sx: [-1;5], sy: [-1;1.5], rx: [-15;22], ry: [-23;4], tx: [0;1505], ty: [0;1419], px: [-0.03;0.07], py: [-0.02;0.02]	sx: 0.92, sy:0.67, rx: -0.04, ry: 0, tx: 0, ty: 92, px:-3.28-05, py: 0
rotation	float	[0;360]	2.00934289847729
character arrangement	string	horizontal, vertical, circular	horizontal
occlusion	integer	[0;100]	5
occlusion direction	string	horizontal, vertical	vertical
typeface	string	standard, special, handwriting	standard
language	string	german, english, spanish, hungarian, italian, latin, french, belgian, russian, turkish, greek, swedish, czech, portoguese, numbers, roman date, abbreviation, company, person, unknown	german
difficult	boolean	true, false	false

TABLE III
Range of values for each metadata dimension and annotations for the example image depicted in figure 4.

1) Hamming Distance: The simplest distance function for string comparison is the exact match or Hamming distance [29]. It is considered as string matching with k mismatches, where each mismatch is assigned with cost 1. For computing the Hamming distance, simply the number of positions is counted in which the corresponding characters of the strings are different:

$$d(x_n, y_m) = \sum_{x_i \neq y_i} 1 \tag{4}$$

, where x_i is the character at position i in string x_n.

2) Levenshtein Distance: The Levenshtein [30] or edit distance is the most common string comparison method. It is considered as string matching with k differences, allowing insertions, deletions and replacements. If all transformation operations are assigned with cost 1, then the Levenshtein distance can be also considered as the minimum number of transformation operations required to change string x_n into y_m. Formally the Levenshtein distance is defined as:

$$d(\emptyset, \emptyset) = 0 \tag{5}$$

$$d(x_n, \emptyset) = d(\emptyset, x_n) = n \tag{6}$$

$$d(x_n, y_m) = \min \begin{cases} d(x_{n-1}, y_{m-1}) + 0 & \text{if } \; x_i = y_j, \\ d(x_{n-1}, y_{m-1}) + 1 & \text{(replacement)}, \\ d(x_n, y_{m-1}) + 1 & \text{(insertion)}, \\ d(x_{n-1}, y_m) + 1 & \text{(deletion)} \end{cases} \tag{7}$$

, where x_{n-1} resembles the string x_n shortened by 1 character.

The Damerau-Levenshtein distance [31] extends the Levenshtein distance by adding the transposition transformation operation with cost c. For this, equation 7 needs to be adapted as follows:

$$d(x_n, y_m) = \min \begin{cases} d(x_{n-1}, y_{m-1}) + 0 & \text{if } \; x_i = y_j, \\ d(x_{n-1}, y_{m-1}) + 1 & \text{(replacement)}, \\ d(x_n, y_{m-1}) + 1 & \text{(insertion)}, \\ d(x_{n-1}, y_m) + 1 & \text{(deletion)}, \\ d(x_{n-2}, y_{m-2}) + c & \text{(swap) if} \\ \qquad x_i = y_{i-1} \text{ or } x_{i-1} = y_i. \end{cases} \tag{8}$$

3) Longest Common Substring Distance: The longest common substring as defined in [32] and [33] is the longest pairing of characters in strings x_n and y_m, so that the characters of the substring appear in the same order in the strings. The length of the longest common substring can be computed by following recursive definition:

$$lcs(\emptyset, \emptyset) = lcs(x_n, \emptyset) = lcs(\emptyset, x_n) = 0 \tag{9}$$

$$lcs(x_n, y_m) = \begin{cases} lcs(x_{n-1}, y_{m-1}) + 1 & \text{if } x_i = y_j, \\ \max(lcs(x_{n-1}, y_m), lcs(x_n, y_{m-1})) & \\ & \text{if } x_i \neq y_j \end{cases} \tag{10}$$

, where deletions can only be made at the beginning or the end of the compared strings. The distance is defined in [34] as the number of unpaired characters, formally:

$$d(x_n, y_m) = n + m - 2lcs(x_n, y_m) \tag{11}$$

, for strings x_n and y_m with length n and m respectively. Efficient algorithms for computing longest common substrings are discussed in [35], [34] and [36].

4) Jaro Distance: The Jaro distance is defined in [37] and [38] as the weighted sum of common characters and the number of transpositions, formally:

$$d(x_n, y_m) = \left(\frac{w_x c}{n} + \frac{w_y c}{m} + \frac{w_\tau (c - \tau)}{c} \right) \tag{12}$$

, where w_x and w_y are weights associated with the strings x_n and y_m and w_τ is the weight associated with transpositions. Two characters are considered in common only if they are no further apart than $c = \frac{\max(n,m)}{2} - 1$. The number of transpositions τ is computed by comparing the common characters positionwise. The number of mismatched characters divided by two yields the number of transpositions.

The Jaro-Winkler distance modifies the basic Jaro distance according to whether the first few characters in the strings x_n and y_m are the same. Formally the Jaro-Winkler distance is defined in [38] as:

$$d(x_n, y_m) = d_J(x_n, y_m) + \jmath \cdot 0.1(1 - d_J(x_n, y_m)) \tag{13}$$

, where $d_J(x_n, y_m)$ is the Jaro distance between strings x_n and y_m (see equation 12) and $\jmath$ is the length of the common prefix at the start of the strings up to a maximum of 4. For this distance function [38] uses 1/3 for the weights w_x, w_y and w_τ, which we applied in our experiments too.

C. Normalized String Similarity

The Hamming, Levenshtein, Damerau-Levenshtein and Longest Common Substring distance functions count the number of character operations. Because the distance depends on the string length too – comparing two long strings naturally yields a higher distance value on average due to the higher number of compared characters –, the distance has to be normalized. Therefore, we chose to divide the resulting distance by the length of the longer string, which leads to distance values between 0 and 1.

Similarity measures can be easily derived from distance functions by subtracting the normalized distance from 1. As a result, we propose the following normalized similarity function for comparing the recognition accuracy of different OCR approaches:

$$s(x_n, y_m) = 1 - \frac{d(x_n, y_m)}{\max(n, m)}. \tag{14}$$

D. Summary

In this section we gave an overview of the most common and popular distance functions for string comparison.

Because OCR tools are usually applied on scanned text documents, they are trained to recognize text with horizontal character arrangement. Horizontally arranged strings recognized by OCR from natural images contain in most cases the characters in correct order, so usually there are no transpositions required. Though, some characters can be missing or misclassified as other characters and also some characters might be recognized which don't exist in the image. Because of these properties the longest common substring and the classic Levenshtein distance measures are most applicable for comparing horizontally arranged OCR text from natural images with the ground truth annotation.

In contrast to scanned documents, natural images contain text with vertical or circular character arrangement too. Because OCR tools in general don't find any correspondences between the letters of a vertically or circularly arranged word, it is possible that characters are recognized in mixed order. For the comparison of these types of text in natural images transposition operations might be useful. Therefore the Damerau-Levenshtein and Jaro distances should

be a better choice for comparing the recognized text with the ground truth.

In the following section the distance functions are applied for comparing the strings recognized by OCR applications and the annotated ground truth in the NEOCR dataset.

V. Evaluation of OCR Applications

To form an impression what the current state of natural-image text recognition is, popular open source and leading commercial OCR applications have been compared based on the NEOCR dataset. Due to differences to scanned documents, first optimal configurations were identified for natural images for each OCR tool based on the 9 sample images depicted in figure 1. OCR tools that couldn't be configured to recognize any useful text have been discarded. Altogether 9 of 23 inspected OCR tools were left for the test scenarios below. For all test scenarios the optimal configuration determined manually for the 9 sample images was applied.

Because we don't want to compare the tools themselves but rather get an overall impression of the current state of OCR in natural images, the names of the tools have been made anonymous. Tools `A` and `B` are the same but using different vocabularies (`A` English, `B` German). Tools `H` and `I` are different versions of the same software, where `H` refers to the older version. We kept both versions because the results were significantly different. In the test scenarios, where the recognized text is compared to the annotated ground truth, the normalized string similarity from equation 14 is applied using the Levenshtein distance.

A. Text Detection

In the first scenario the text detection quality of the OCR applications was evaluated on whole images. Images not containing any text should be rejected by OCR tools, that is the output of the tools should be empty or a certain error message should be issued. Whether the text itself was recognized correctly or not is disregarded in this test. Because all images in the NEOCR dataset contain text, true negatives have been selected manually from the MIRFLICKR-25000 dataset [39, 40]. All the 659 images from the NEOCR dataset and the same number of true negative images were used for testing. The results for text images are shown in figure 5(a) and for non-text images in figure 5(b).

OCR applications `D`, `H` and `I` always detected text in the images, regardless of whether there was text visible in the images at all. Some applications are more cautious, especially `F`, which only rarely detected text. Because in this scenario only the text detection and rejection was considered, in the following evaluations the correctness of the recognized text is analyzed in detail.

B. Text Recognition in Textfields

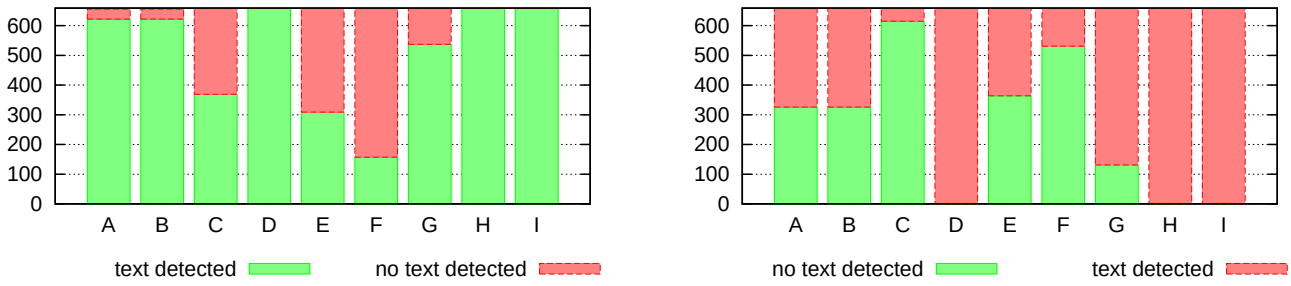
All evaluations in this section consider only the annotated bounding boxes (textfields). Because the text itself doesn't has to be located first inside the image, considering textfields only allows to limit the analysis to the text recognition performance. For each experiment the number of textfields belonging to the current characteristic is attached.

Both the results for distorted and straightened textfields are presented for each test scenario. Straightening was implemented based on the annotated distortion quadrangles and transformation operations of ImageMagick. Figure 6(a) shows example textfields from the NEOCR dataset with perspective distortion. In figure 6(b) the according straightened textfields are depicted. Problems with straightening distorted textfields arise for images with low resolution, strings not completely contained in their bounding boxes and texts with circular character arrangement. Overall, the resulting straightened textfields are largely satisfying.

1) Overall Performance: Before examining the recognition performance of the selected OCR tools for different characteristics, the overall text recognition performance was evaluated. In contrast to subsequent tests, all bounding boxes of the NEOCR dataset were used in this test. The average similarities are presented in figure 7.

The OCR applications were evaluated with 4 different versions of the same textfields: original color version, grayscale image and the inverted versions of both. For each OCR tool the best recognition rate is displayed along with the difference to the worst and mean rate of the 4 versions.

The overall performance is poor, although only textfields cut from the whole images were used in this test. Even the best OCR tool recognizes only every third string correctly. To answer the question which characteristics of natural images cause the most severe problems, text recognition is evaluated based



(a) Images containing text (NEOCR Dataset).

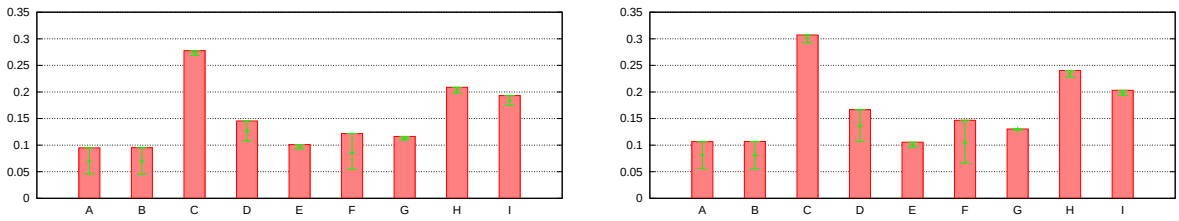
(b) Images without text (MIRFlickr-25000).

Fig. 5. Figure 5(a) showing the ratio of images where the OCR tools found text in the true positive images of the NEOCR dataset (disregarding the correct recognition of the detected text). The green part represents images where text was detected (true positives), the red part are rejected images (false positives). Figure 5(b) showing the ratio of images where the OCR tools found text in the true negative images selected from the MIRFLICKR-25000 dataset. The green part represents rejected images (true negatives), the red part are images where text was detected (false negatives).

(a) Examples for distorted textfields.

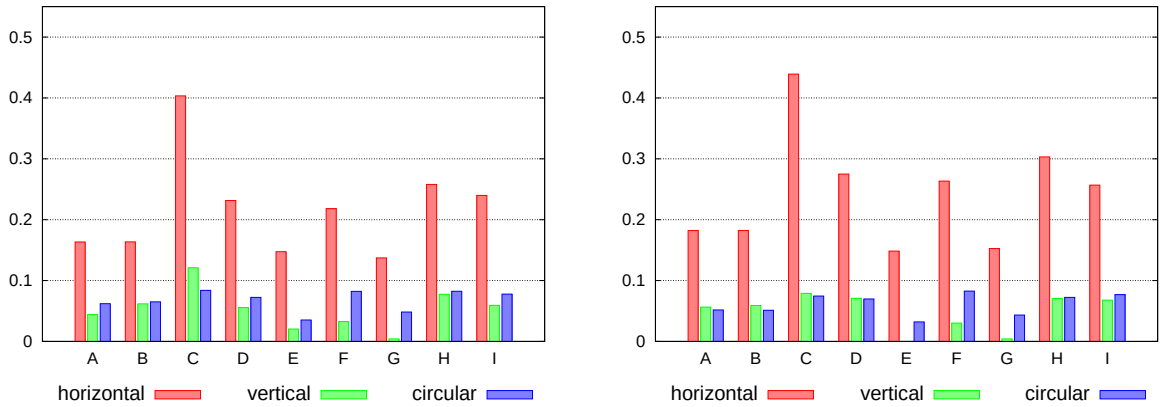
(b) Straightened textfields.

Fig. 6. Examples of textfields with perspective distortion and their straightened versions.



(a) distorted (b) straightened

Fig. 7. Overall recognition correctness of the OCR tools for all bounding boxes (5238).



(a) distorted (b) straightened

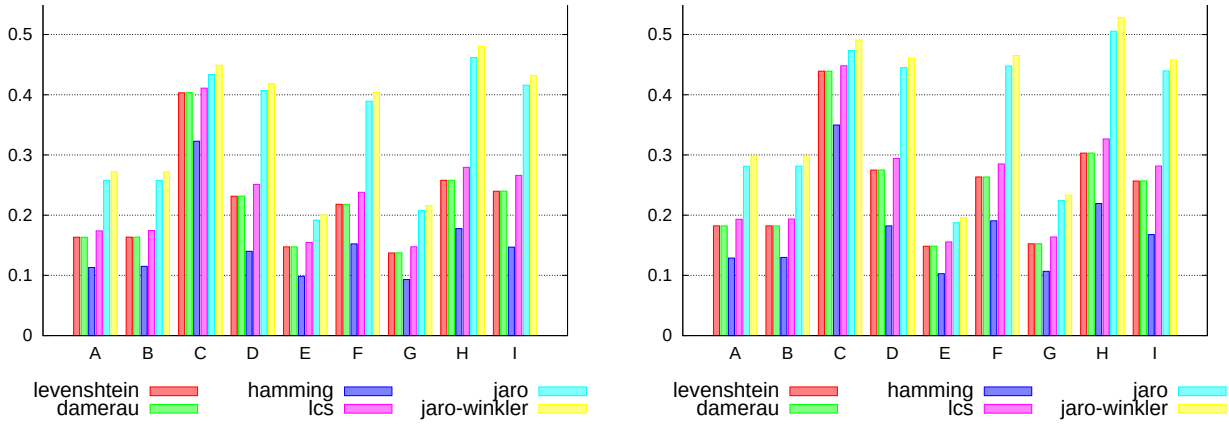
Fig. 8. Text recognition performance for textfields with different character arrangement: horizontal (2598), vertical (24) and circular (97).

on different metadata dimensions in the following sections.

2) Character Arrangement: Text in scanned documents usually consists of horizontally arranged characters, while in natural images texts appear frequently in vertical or circular character arrangement. To reduce the effect of other text characteristics, only images annotated as not difficult, without occlusion and not inverted (dark text on light background) were used in this test. The values for average similarities between recognized text and ground truth annotations are presented in figure 8. It is clearly observable that the recognition of text with vertical and circular character arrangement is significantly worse. Also straightening the textfields doesn't help much, because it doesn't change the arrangement itself.

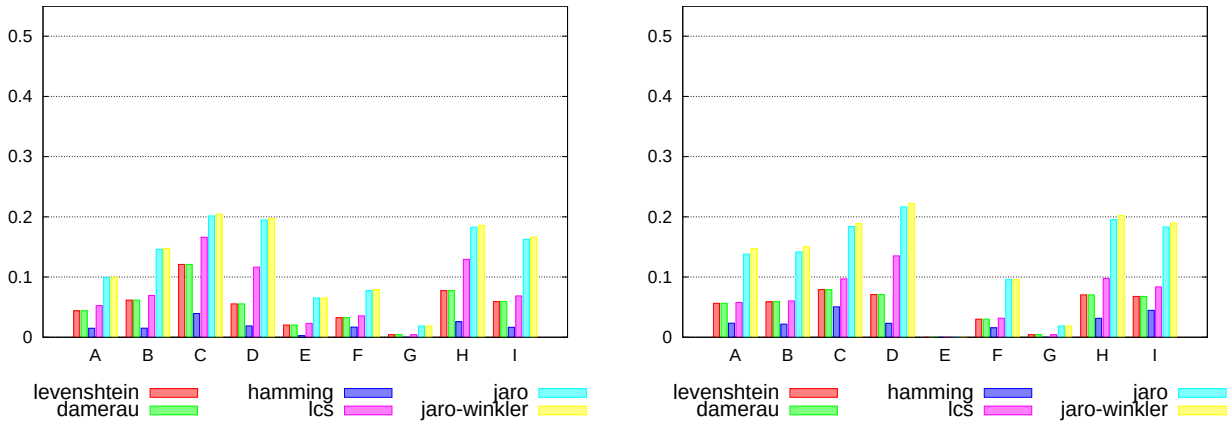
In section IV-D we suggested to choose a different distance function for vertical or circular text. The results for different distance functions are depicted in figures 9, 10 and 11 for different character arrangements. Although the similarity value is higher when applying the Jaro- or Jaro-Winkler distances compared to the Levenshtein distance, there is no clear benefit derivable from the possibility of character swapping in the distance functions. The similarity values for horizontal character arrangement are high too and there is no difference observable from the ratio of the similarity values between different character arrangements. Because there is obviously no benefit discoverable for using other distance functions, all further experiments were conducted using the Levenshtein distance.

3) Inversion: Text in scanned documents usually appears in black or dark color on white or light background. Natural images contain text in much higher variations and therefore light text on dark backgrounds appears quite often too. To reduce the effect of other text characteristics, only images annotated as not difficult, without occlusion and with horizontal character arrangement were considered in this test scenario. Figure 12 shows that some OCR tools indeed assume text to be dark on light background and therefore have severe problems recognizing light text on dark background.



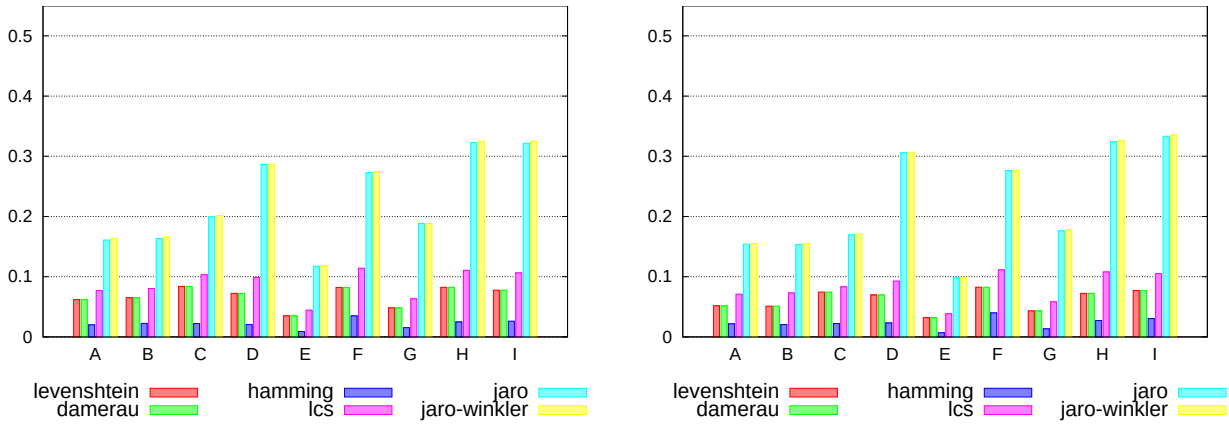
(a) distorted (b) straightened

Fig. 9. Comparison of different distance functions for horizontal character arrangement (2598).



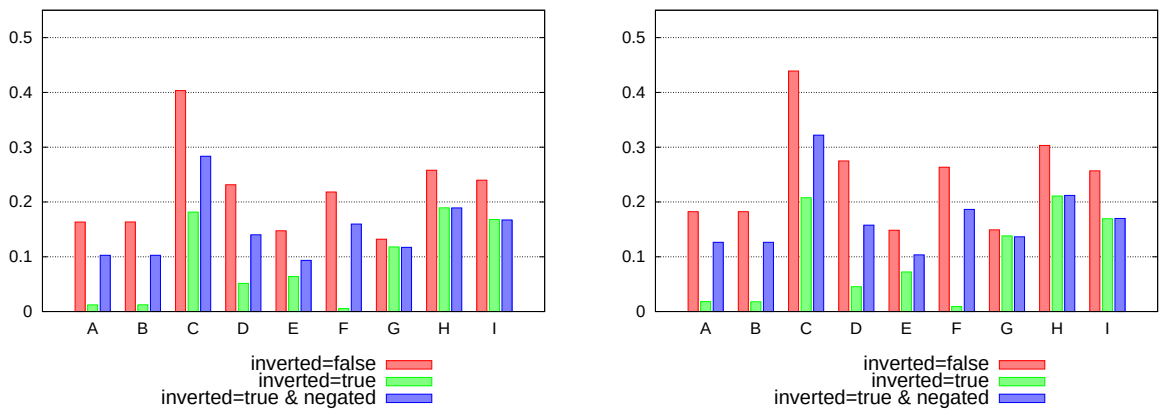
(a) distorted (b) straightened

Fig. 10. Comparison of different distance functions for vertical character arrangement (24).



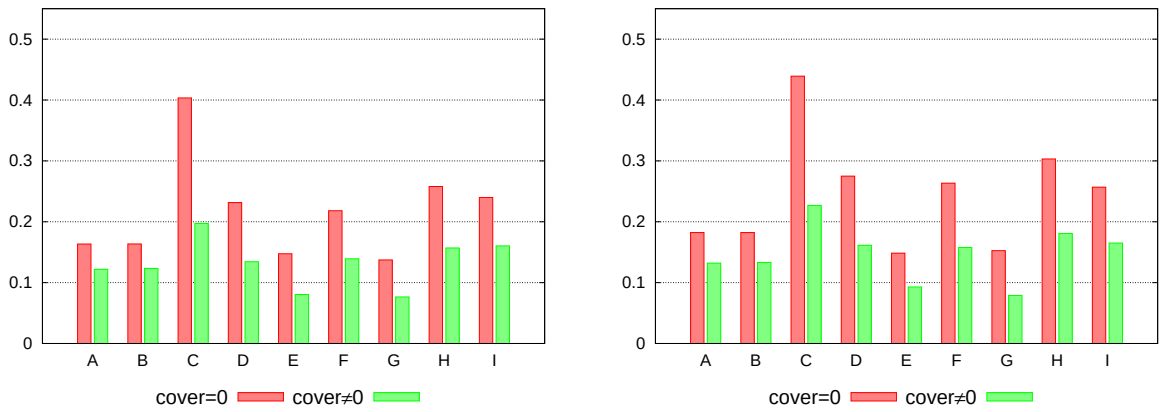
(a) distorted (b) straightened

Fig. 11. Comparison of different distance functions for circular character arrangement(97).



(a) distorted (b) straightened

Fig. 12. Text recognition performance for textfields grouped by inversion: dark text on light background (*inverted=false*, 2598), light text on dark background (*inverted=true*, 1641) and light text on dark background with inverted textfield (*inverted=true & negated*, 1641)



(a) distorted (b) straightened

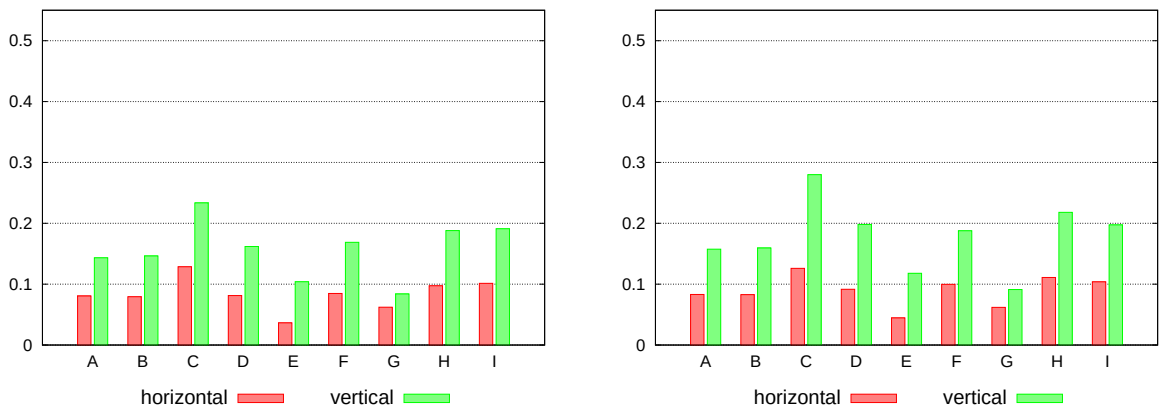
Fig. 13. Text recognition performance for textfields without (*cover=0*, 2598) and with occlusion (*cover≠0*, 336).

As confirmation the negated (inverted) versions of the same inverted (light text on dark background) textfields were evaluated too, which resulted in better recognition performance for all OCR tools. It is quite obvious that dark text on light background is recognized more precisely.

4) Occlusion: Occlusion in natural images can result from objects covering parts of the image or the choice of image detail by the photographer. Depending on whether the occlusion is horizontal or vertical, parts from all characters or only single characters might be missing. While missing parts from all characters can lead to not recognizing any text at all, single missing letters might be corrected using vocabularies.

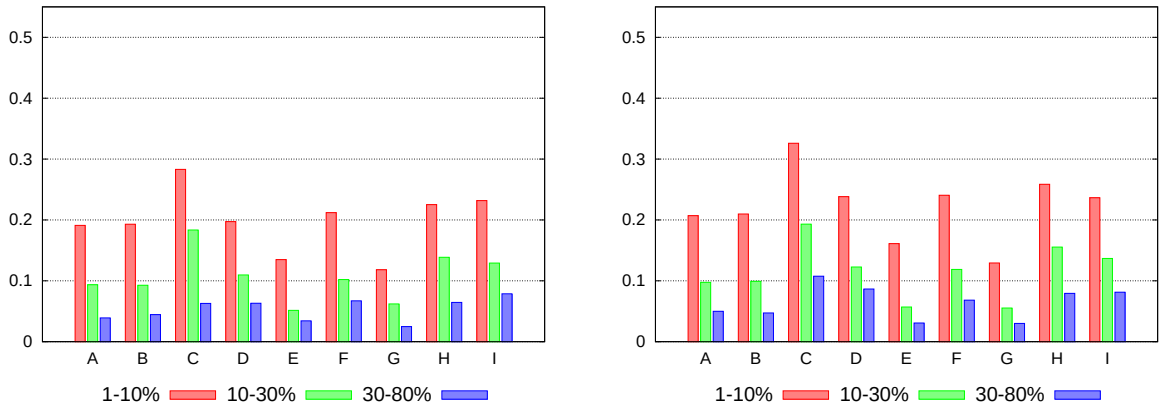
Figure 13 depicts the recognition rate for text with and without occlusion. To reduce the effect of other text characteristics, only images annotated as not difficult, with horizontal character arrangement and without inversion were used in this test scenario. The recognition rate for occluded text is, quite obviously, much worse than for text without occlusion. Figure 14 compares horizontal and vertical occlusion and confirms the assumption that missing characters affect text recognition less than horizontal occlusion.

Figure 15 confirms the assumption that increasing occlusion deteriorates the recognition performance. The same decrease can be observed in figure 16, but some OCR tools seem to benefit from an internal vocabulary which might correct missing or falsely



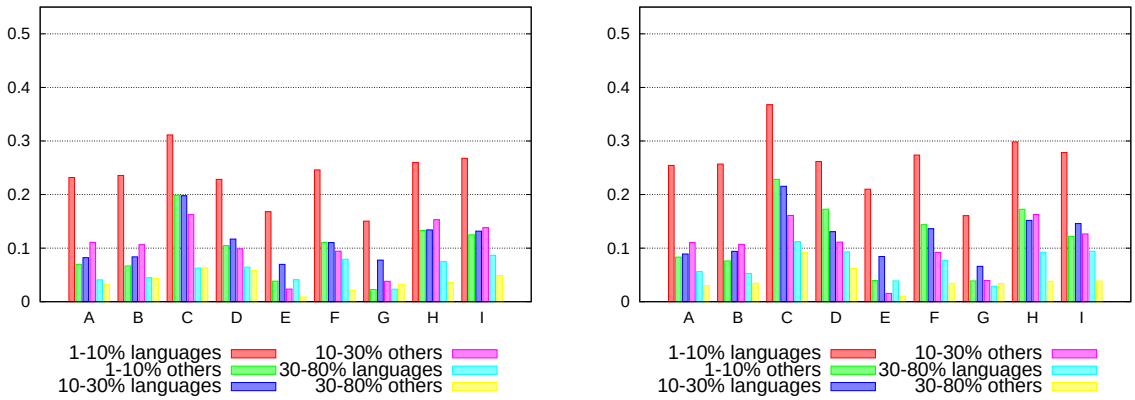
(a) distorted (b) straightened

Fig. 14. Text recognition performance for textfields with horizontal (115) and vertical (221) occlusion.



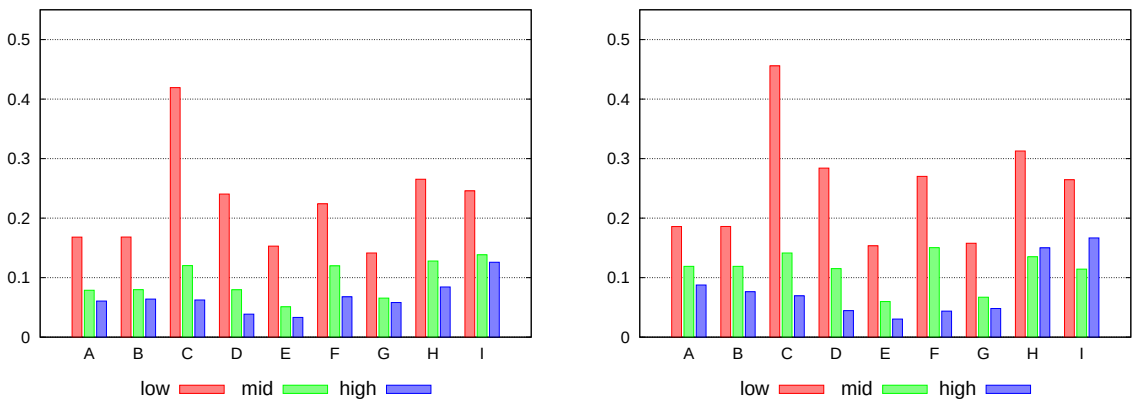
(a) distorted (b) straightened

Fig. 15. Text recognition performance for textfields based on their occlusion: 1-10 % (135), 10-30 % (134), 30-80 % (67).



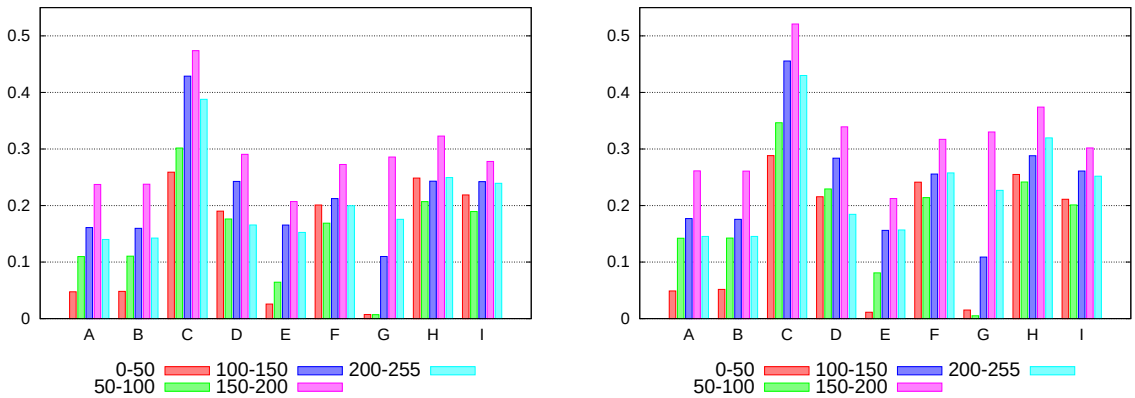
(a) distorted (b) straightened

Fig. 16. Text recognition performance for textfields based on their occlusion as a percentage and grouped by their language type: 1-10 %: languages (101), others (34); 10-30 %: languages (81), others (53); 30-80 %: languages (53), others (14).



(a) distorted (b) straightened

Fig. 17. Text recognition performance for horizontally arranged text with low (2463), mid (123) and high texture (12). Note that the low number for highly textured text comes from the limitation to images tagged as not difficult.



(a) distorted (b) straightened

Fig. 18. Text recognition performance for textfields grouped by brightness values: 0-50 (135), 50-100 (494), 100-150 (1031), 150-200 (687) and 200-255 (251).

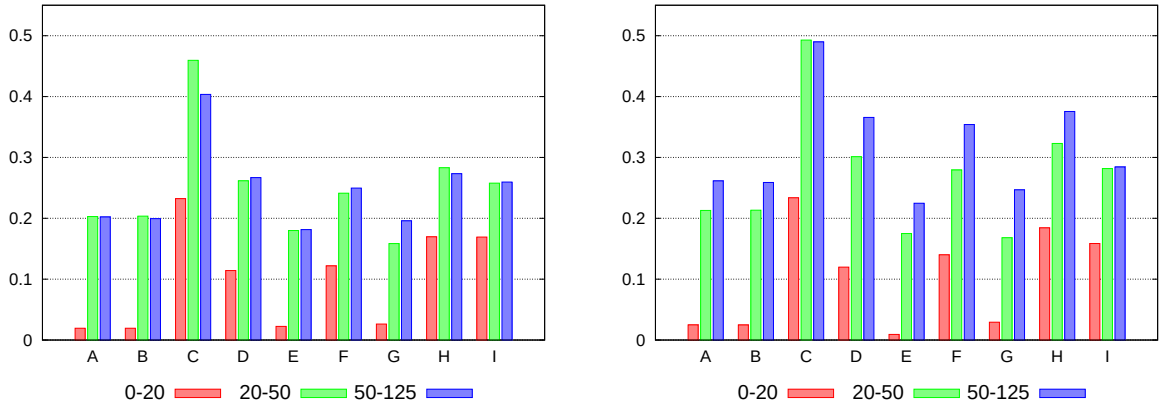
recognized characters. With the help of vocabularies, obviously the recognition performance for language dependent words is much better compared to abbreviations or numbers.

5) Texture: For the analysis of texture effects and all subsequent test scenarios only textfields annotated as not difficult, not inverted, with horizontal character arrangement and without occlusion were considered. The results in figure 17 confirm the assumption that the recognition performance decreases with higher texture. The reason for the low number of highly textured images is that textfields in this category are usually annotated as difficult and these images were excluded from the test.

6) Brightness: For the analysis of text recognition regarding brightness the values range was split into 5 groups. The best results in figure 18 are consistent with the brightness range of most scanned documents.

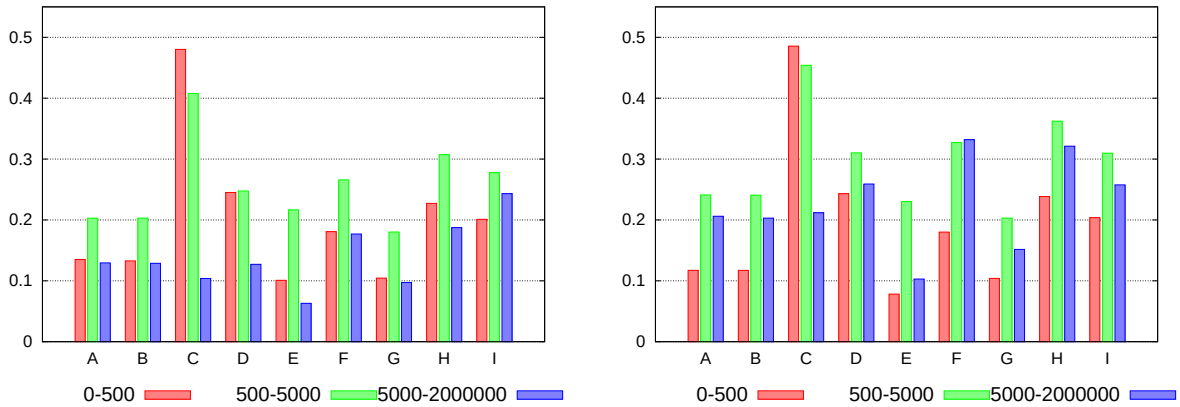
7) Contrast: The values range for contrast was divided based on the distribution in figure 3(b) into 3 groups. The results in figure 19 clearly show, that some OCR tools fail by dealing with low contrast textfields (values between 0 and 20). Similarly to the evaluation of brightness effects, best recognition rates are obtained for textfields in the range of scanned documents.

8) Resolution: The values range for resolution was split into 3 groups based on the usual values for pixels per characters in scanned documents. Based on the results depicted in figure 20, text with very low and very high resolution is recognized more imprecisely. This observation can be explained by the quality of



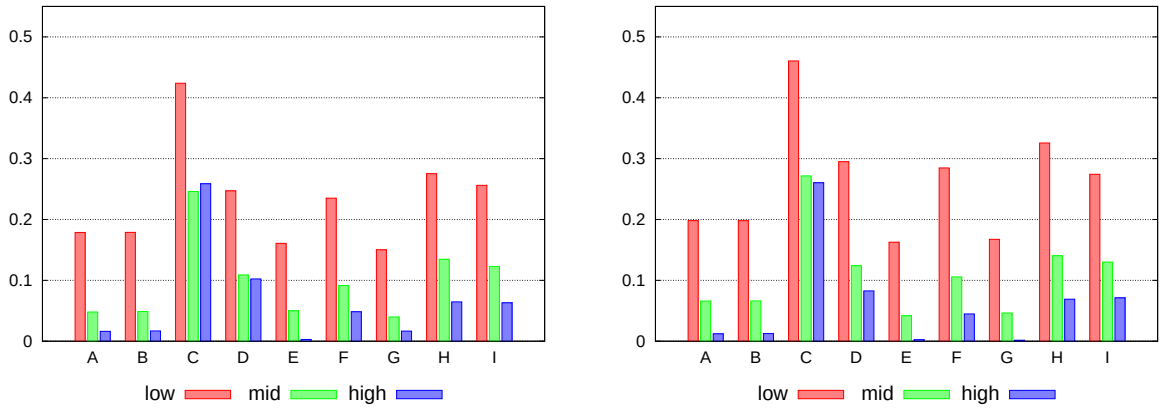
(a) distorted (b) straightened

Fig. 19. Text recognition performance for textfields grouped by contrast values: 0-20 (533), 20-50 (1625) and 50-125 (440).



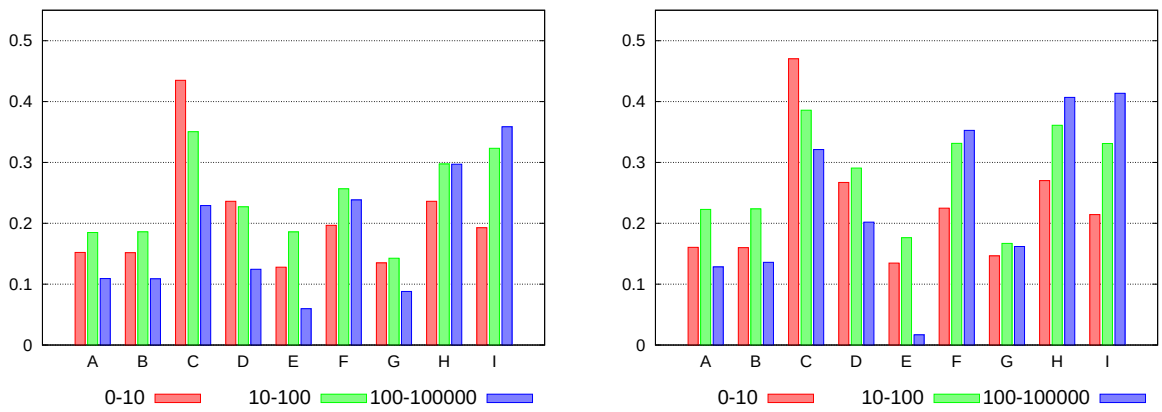
(a) distorted (b) straightened

Fig. 20. Text recognition performance for textfields grouped by their resolution (average number of pixels per character): 0-500 (1134), 500-5000 (1157) and 5000-2000000 (307).



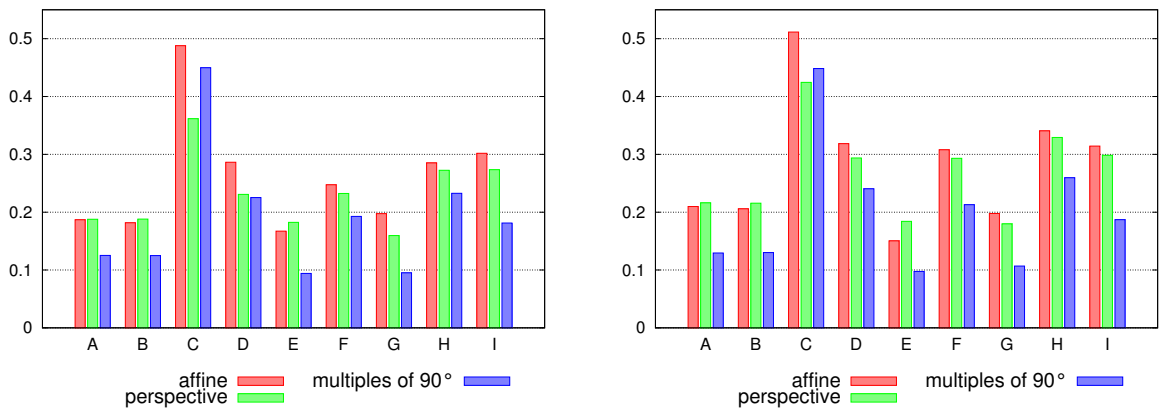
(a) distorted (b) straightened

Fig. 21. Text recognition performance for horizontally arranged text with low (2307), mid (241) and high noise (50).



(a) distorted (b) straightened

Fig. 22. Text recognition performance for horizontally arranged text with low (1-10, 1660), mid (10-100, 918) and high blurredness (100-100000, 20).



(a) distorted (b) straightened

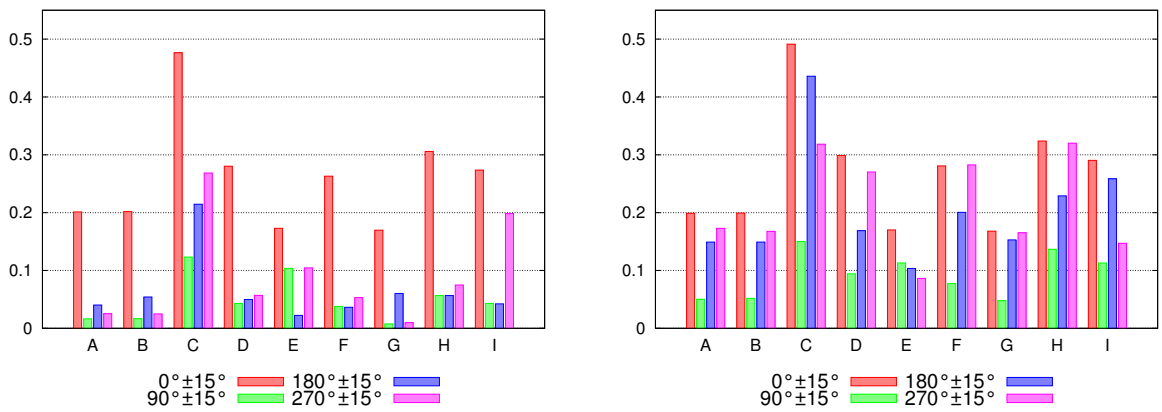
Fig. 23. Text recognition performance for horizontally arranged text according to its distortion: no perspective (164), multiples of 90°(994) and others (1440).

scanned documents, where one single character is represented on average by 2500 pixels and higher resolutions are quite uncommon. OCR tool `C` is an exception, it seems to deal with low-resolution text better than with other resolutions.

9) Noise: Similarly to the effects of texture, increasing noise degrades the recognition performance. The results in figure 21 clearly show the negative effect of increasing noise. OCR tools `A`, `B`, `E` and `G` seem to have serious problems with high-noise images.

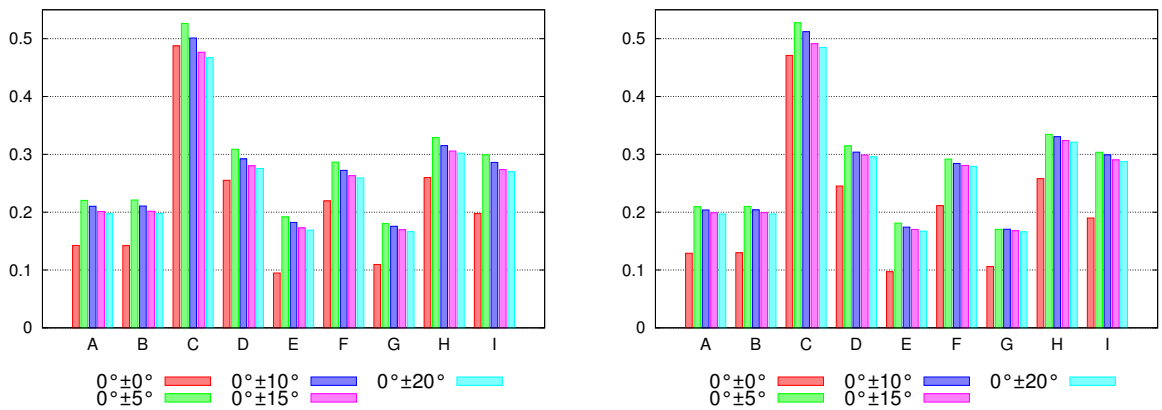
10) Blurredness: The values range for blurredness was split into 3 groups. The results of the experiments in figure 22 indicate the best recognition rates for images with slight unsharpness (except for OCR tool `C`). These results coincide with the observations for resolution in section V-B8, which confirms the direct relationship between the resolution and our defined measurand for blurredness in section III-B.

11) Distortion: For a more detailed evaluation of distortion effects on text recognition, textfields without any perspective distortion or containing only affine distortions were compared to images with perspective distortion and rotations of multiples of 90°. The results in figure 23 show that most OCR tools have serious problems with rotated text. Straightening the textfields based on the annotated distortion quadrangles clearly improves the recognition performance. For example, the recognition rate both for affine and perspective distortions is improved significantly.



(a) distorted (b) straightened

Fig. 24. Text recognition performance for horizontally arranged text with rotation values around 0°(345°to 15°) (2036), 90°(75°to 105°) (67), 180°(165°to 195°) (8) and 270°(255°to 285°) (227).



(a) distorted (b) straightened

Fig. 25. Text recognition performance for horizontally arranged text with rotation values around 0°: 0°±0°(861), 0°±5°(1681), 0°±10°(1898), 0°±15°(2036), 0°±20°(2087).

12) Rotation: Two seperate scenarios were analyzed for the effects of rotation. Figure 24 shows the results for text rotations of multiples of 90°. Most OCR tools struggle with severe rotations, although some applications seem to have a built-in rotation of 180°.

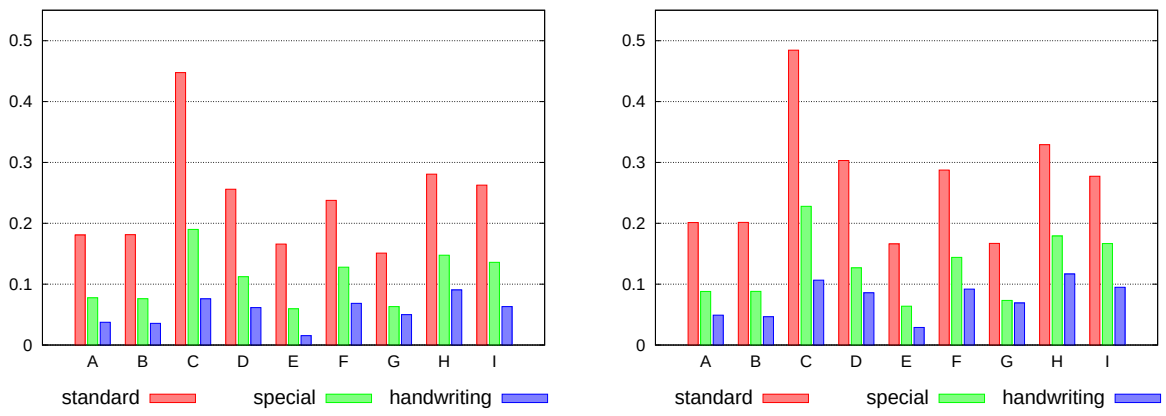
In the second scenario the rotation tolerancy around 0° was analyzed in more detail. Surprisingly, figure 25 shows best results for all OCR tools around ±5°. This indicates that obviously all applications already assume small rotations for scanned documents.

13) Font: The evaluation of different font typefaces in figure 26 confirmed the assumption that the best recognition performance is achieved for standard fonts. Possibly, the recognition correctness for handwritings would be much higher using a specially-tailored OCR application.

14) Language: Depending on whether an OCR application is correcting recognized text using a built-in vocabulary, the recognition performance can be improved significantly. Hints for the utilization of vocabularies are depicted in figure 27, which shows better recognition rates for language-dependent texts for all OCR tools.

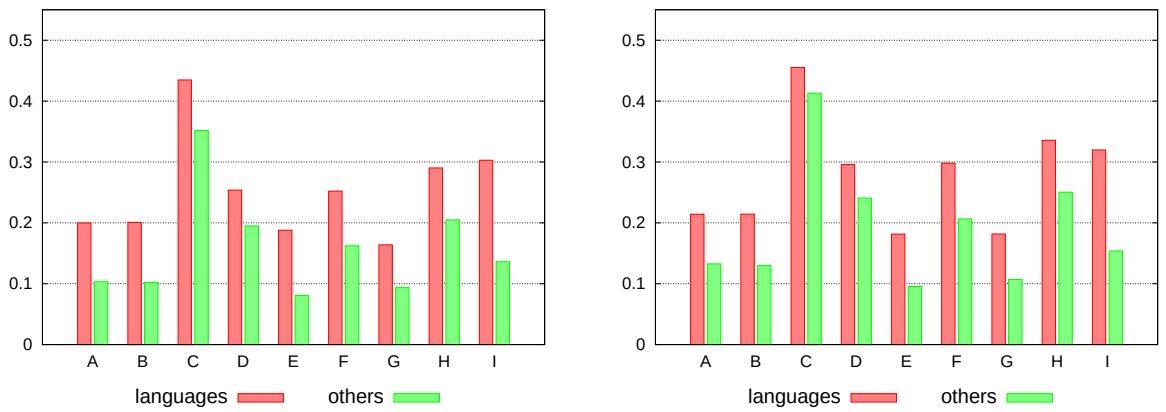
A more detailed inspection of languages is presented in figure 28.

The results for language-independent texts are presented in figure 29. For most OCR applications the recognition performance is poor for numbers and abbreviations. Names of persons or companies are



(a) distorted (b) straightened

Fig. 26. Text recognition performance for horizontally arranged text with different font types: standard (2215), handwriting (142) and special (241).



(a) distorted (b) straightened

Fig. 27. Text recognition performance for textfields grouped by language dependency: languages (1616) and others (982).

better recognized, which can be explained by their higher average string length and possible inclusion in vocabularies.

C. Summary

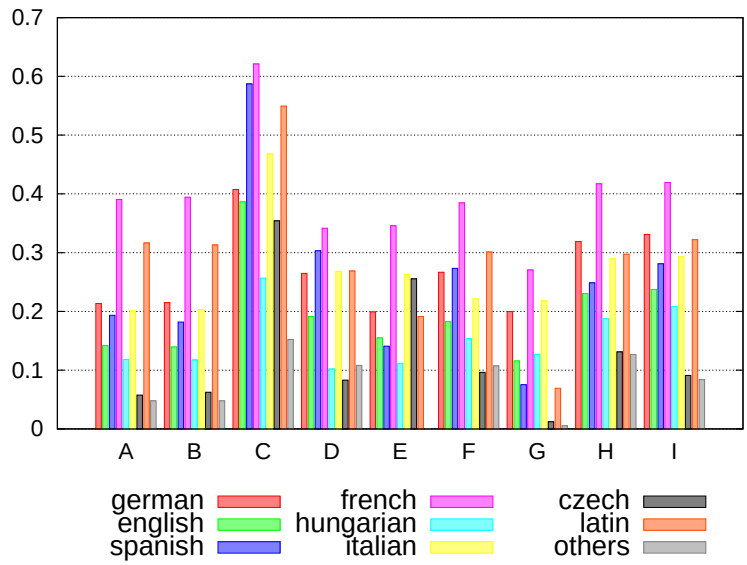
In this section the current state of text recognition in natural images was evaluated based on the proposed NEOCR dataset and using 9 open source and leading commercial OCR applications. Thanks to the rich metadata of the NEOCR dataset, the recogition performance could be analyzed from different perspectives and several weaknesses of natural-image OCR were identified. The results are discussed in the next section in detail.

VI. Discussion

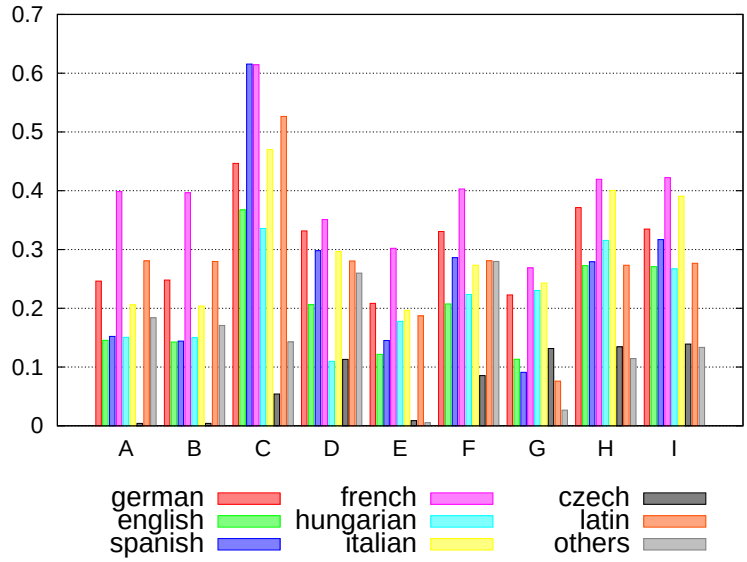
After defining a new dataset for OCR in natural images in section III and analyzing current open source and commercial OCR applications in detail in section V, we discuss the results of our evaluation.

The results show a poor overall performance both for detecting and filtering real text from whole images and for recognizing characters and words. Obviously, OCR applications `D`, `H` and `I` always found text in images, whether the images contained text or not, while other tools were more cautious with detecting text. Poor filtering of image contents erroneously recognized as text is still a major problem in natural-image text OCR.

Although the subsequent tests were limited to bounding boxes only, the average correctness of the



(a) distorted

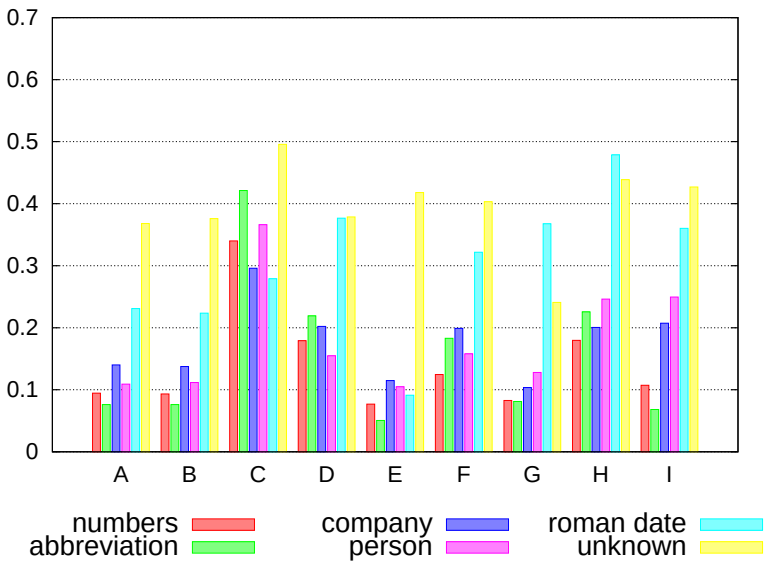


(b) straightened

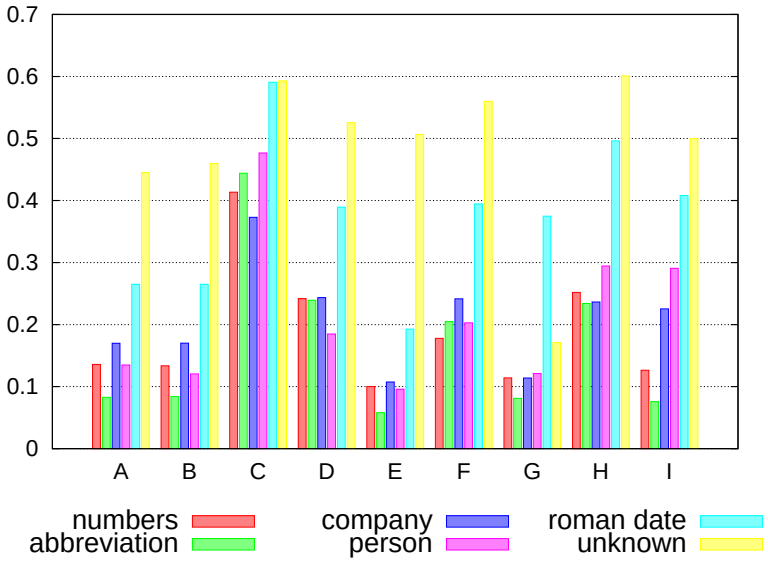
Fig. 28. Text recognition performance for textfields grouped by languages: german (908), english (286), spanish (240), french (81), hungarian (30), italian (29), czech (27), latin (8) and others (turkish, portoguese, greek, swedish, russian, belgian, 7).

text recognition in natural images is unsatisfying compared to scanned documents. The low performance results both from shorter (no text detected) and longer text annotations (additional text was detected) as well as incorrectly recognized characters. For the best OCR application, the overall recognition correctness considering textfields only resulted in around 30% match between recognized text and ground truth annotation.

Based on the rich annotation of the NEOCR dataset the performance of current open source and commercial OCR tools was evaluated for different characteristics of natural images in detail. As assumed, vertically and circularly aligned characters are poorly detected. For inverted text (light text on dark background) a simple negation of the bounding box results in a much better performance for several OCR tools, though the average similarity still falls behind not inverted text (dark text on light background). As



(a) distorted



(b) straightened

Fig. 29. Text recognition performance for language independent texfields: numbers (431), abbreviations (236), companies (225), persons (72), roman dates (8) and uncategorizable texts (10).

presumed, the performance for horizontally occluded text is much worse than for text with vertical occlusion (missing characters).

Based on the additionally annotated distortion quadrangles in the NEOCR dataset the effects of perspective distortions could be analyzed in more detail. Straightened text had on average 2 to 5% better recognition performance. In special cases – e.g. rotations of multiples of 90° – a 20% improvement in recognition performance was observed. The use of vocabularies significantly improves text recognition, which was clearly observable by comparing the recognition performance of language-dependent and language-independent strings.

In table IV values for each metadata characteristic are collected based on the evaluation results in section V. These values are no surprise, most of them represent the values range of scanned documents. Although optimal configurations were explored for natural images for each analyzed OCR application,

Category	Values Range	Optimal Value
texture	low, mid, high	low
birghtness	0-255	150-200
contrast	0-127	50-127
inversion	true, false	false
resolution	1-2000000	500-5000
noise	low, mid, high	low
blurredness	0-100000	10-100
distortion	sx: [-1;5], sy: [-1;1.5], rx: [-15;22], ry: [-23;4], tx: [0;1505], ty: [0;1419], px: [-0.03;0.07], py: [-0.02;0.02]	px: 0, py: 0
rotation	0-360	0°±5°
character arrangement	horizontal, vertical, circular	horizontal
occlusion	0-100	0
occlusion direction	horizontal, vertical	vertical
typeface	standard, special, handwriting	standard
language	german, english, spanish, hungarian, italian, latin, french, belgian, russian, turkish, greek, swedish, czech, portoguese, numbers, roman date, abbreviation, company, person, unknown	french
difficult	true, false	false

TABLE IV
Overview of evaluated characteristics and their corresponding optimal values.

the optimal values listed in table IV indicate a very strong tailoring of the tools for scanned documents.

VII. Conclusion

In this report the NEOCR dataset has been presented for natural-image text recognition. Besides the bounding box annotations, the dataset is enriched with additional metadata like rotation, occlusion or inversion. For a more accurate ground truth representation distortion quadrangles have been annotated too. Due to the rich annotation, several subdatasets can be derived from the NEOCR dataset for testing new approaches in different situations. We evaluated 9 popular open source and commercial OCR tools to their applicability on natural-image text OCR. Due to the rich annotation, we conclude that the configurable NEOCR dataset can be applied for evaluation and comparison of natural-image text OCR approaches. By using the dataset, differences among OCR methods can be emphasized on a more detailed level and deficits can be identified more accurately.

Based on the comprehensive evaluation of OCR tools using the NEOCR dataset we conclude that text recognition in natural images is currently poor. Due to the severe shortcomings, unfortunately, text recognition in its current state cannot be applied for enhancing classification and automatic annotation of natural images. Possibilities for further improvement include the use of vocabularies [1] or the context (e.g., other text or objects) inside the whole image [7]. Recently, popular features from object recognition based on histograms of gradients were explored for character recognition with very promising results [4]. Because current tools are obviously tailored to scanned documents, the major challenge is to tackle the problem of high diversity in natural-image text. The proposed NEOCR dataset enables the very detailed evaluation of new methods and assists the

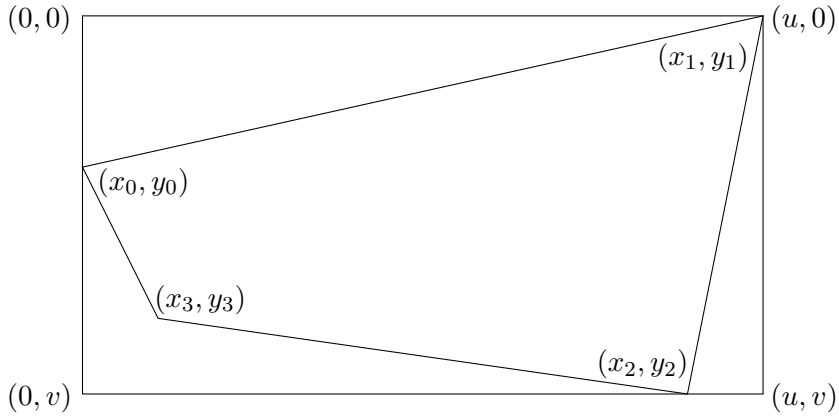


Fig. 30. Example bounding box and distortion quadrangle.

development of new techniques for OCR in natural images. In future we plan to increase the number of annotated images by opening access to our adapted version of the LabelMe annotation tool.

Appendix A
Distortion Quadrangle Coordinate Equations

[25] defined the distortion matrix and the corresponding equations for boxes of unit length. For bounding boxes with arbitrary length the equations need to be adjusted.

The general representation of a perspective transformation is defined in [25] as follows:

$$[x', y', w'] = [u, v, w] \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{15}$$

By normalizing the matrix according to the scaling parameter ($a_{33} = 1$), we get following equations for the conversion of the original coordinates (u_k, v_k) for $k = 0, 1, 2, 3$:

$$x = a_{11}u + a_{21}v + a_{31} - a_{13}ux - a_{23}vx \tag{16a}$$

$$y = a_{12}u + a_{22}v + a_{32} - a_{13}uy - a_{23}vy \tag{16b}$$

For the four coordinate pairs we get following linear system of equations:

$$\begin{bmatrix} u_0 & v_0 & 1 & 0 & 0 & 0 & -u_0x_0 & -v_0x_0 \\ u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1x_1 & -v_1x_1 \\ u_2 & v_2 & 1 & 0 & 0 & 0 & -u_2x_2 & -v_2x_2 \\ u_3 & v_3 & 1 & 0 & 0 & 0 & -u_3x_3 & -v_3x_3 \\ 0 & 0 & 0 & u_0 & v_0 & 1 & -u_0y_0 & -v_0y_0 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -u_1y_1 & -v_1y_1 \\ 0 & 0 & 0 & u_2 & v_2 & 1 & -u_2y_2 & -v_2y_2 \\ 0 & 0 & 0 & u_3 & v_3 & 1 & -u_3y_3 & -v_3y_3 \end{bmatrix} A = X \tag{17}$$

In the presented case a rectangle is transformed into an arbitrary inlying quadrangle as depicted in figure 30:

This limitation simplifies the system of equations to:

$$(0,0) \rightarrow (x_0, y_0)$$
$$(u,0) \rightarrow (x_1, y_1)$$
$$(u,v) \rightarrow (x_2, y_2)$$
$$(0,v) \rightarrow (x_3, y_3)$$

By applying this simplification on the linear system of equations 17 we get following equations for the transformation parameters:

$$a_{31} = x_0 \tag{18a}$$
$$a_{32} = y_0 \tag{18b}$$
$$a_{11}u + a_{31} - a_{13}ux_1 = x_1 \tag{18c}$$
$$a_{12}u + a_{32} - a_{13}uy_1 = y_1 \tag{18d}$$
$$a_{11}u + a_{21}v + a_{31} - a_{13}ux_2 - a_{23}vx_2 = x_2 \tag{18e}$$
$$a_{12}u + a_{22}v + a_{32} - a_{13}uy_2 - a_{23}vy_2 = y_2 \tag{18f}$$
$$a_{21}v + a_{31} - a_{23}vx_3 = x_3 \tag{18g}$$
$$a_{22}v + a_{32} - a_{23}vy_3 = y_3 \tag{18h}$$

Inserting 18a in 18c:

$$a_{11}u + x_0 - a_{13}ux_1 = x_1$$
$$\Rightarrow \quad a_{11} = (-x_0 + x_1 + a_{13}ux_1)/u \tag{19}$$

Inserting 18b in 18d:

$$a_{12}u + y_0 - a_{13}uy_1 = y_1$$
$$\Rightarrow \quad a_{12} = (-y_0 + y_1 + a_{13}uy_1)/u \tag{20}$$

Inserting 18a in 18g:

$$a_{21}v + x_0 - a_{23}vx_3 = x_3$$
$$\Rightarrow \quad a_{21} = (-x_0 + x_3 + a_{23}vx_3)/v \tag{21}$$

Inserting 18b in 18h:

$$a_{22}v + y_0 - a_{23}vy_3 = y_3$$
$$\Rightarrow \quad a_{22} = (-y_0 + y_3 + a_{23}vy_3)/v \tag{22}$$

Inserting 18a, 19, 21 in 18e:

$$(-x_0 + x_1 + a_{13}ux_1) + (-x_0 + x_3 + a_{23}vx_3) + x_0 - a_{13}ux_2 - a_{23}vx_2 = x_2$$

$$-x_0 + x_1 - x_2 + x_3 + a_{23}v(x_3 - x_2) = a_{13}u(x_2 - x_1)$$

$$\Rightarrow \quad a_{13} = \frac{-x_0 + x_1 - x_2 + x_3 + a_{23}v(x_3 - x_2)}{u(x_2 - x_1)} \tag{23}$$

Inserting 18b, 20, 22, 23 in 18f:

$$(-y_0 + y_1 + a_{13}uy_1) + (-y_0 + y_3 + a_{23}vy_3) + y_0 - a_{13}uy_2 - a_{23}vy_2 = y_2$$

$$-y_0 + y_1 - y_2 + y_3 + a_{13}u(y_1 - y_2) = a_{23}v(y_2 - y_3)$$

$$(-y_0 + y_1 - y_2 + y_3)(x_2 - x_1) + (-x_0 + x_1 - x_2 + x_3)(y_1 - y_2) = a_{23}v((y_2 - y_3)(x_2 - x_1) - (x_3 - x_2)(y_1 - y_2))$$

$$a_{23} = \frac{(-y_0 + y_1 - y_2 + y_3)(x_2 - x_1)}{v((x_2 - x_1)(y_2 - y_3) - (x_3 - x_2)(y_1 - y_2))} + \frac{(-x_0 + x_1 - x_2 + x_3)(y_1 - y_2)}{v((x_2 - x_1)(y_2 - y_3) - (x_3 - x_2)(y_1 - y_2))} \tag{24}$$

Inserting 18b, 20, 22 in 18f:

$$(-y_0 + y_1 + a_{13}uy_1) + (-y_0 + y_3 + a_{23}vy_3) + y_0 - a_{13}uy_2 - a_{23}vy_2 = y_2$$

$$-y_0 + y_1 - y_2 + y_3 + a_{13}u(y_1 - y_2) = a_{23}v(y_2 - y_3)$$

$$\Rightarrow \quad a_{23} = \frac{-y_0 + y_1 - y_2 + y_3 + a_{13}u(y_1 - y_2)}{v(y_2 - y_3)} \tag{25}$$

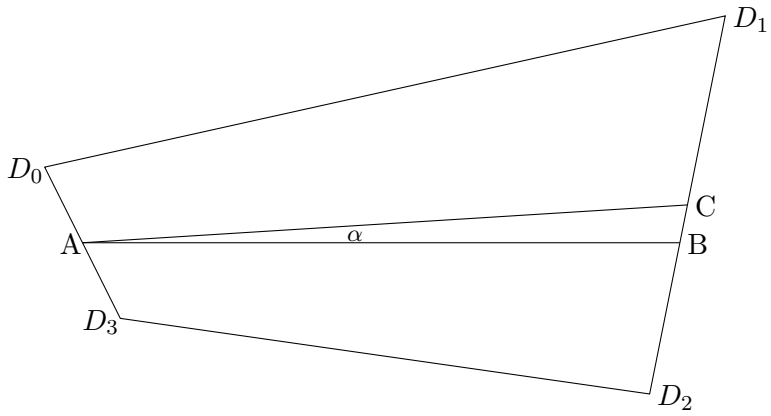


Fig. 31. Example distortion quadrangle and its rotation angle.

Inserting 18a, 19, 21, 25 in 18e:

$$(-x_0 + x_1 + a_{13}ux_1) + (-x_0 + x_3 + a_{23}vx_3) + x_0 - a_{13}ux_2 - a_{23}vx_2 = x_2$$

$$-x_0 + x_1 - x_2 + x_3 + a_{23}v(x_3 - x_2) = a_{13}u(x_2 - x_1)$$

$$(-x_0 + x_1 - x_2 + x_3)(y_2 - y_3) + (-y_0 + y_1 - y_2 + y_3)(x_3 - x_2) = a_{13}u((x_2 - x_1)(y_2 - y_3) - (y_1 - y_2)(x_3 - x_2))$$

$$a_{13} = \frac{(-x_0 + x_1 - x_2 + x_3)(y_2 - y_3)}{u((x_2 - x_1)(y_2 - y_3) - (x_3 - x_2)(y_1 - y_2))} + \frac{(-y_0 + y_1 - y_2 + y_3)(x_3 - x_2)}{u((x_2 - x_1)(y_2 - y_3) - (x_3 - x_2)(y_1 - y_2))} \tag{26}$$

For the example in figure 30 we get:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \approx \begin{bmatrix} 0.36 & -0.22 & -0.07 \\ 0.24 & 0.56 & 0.04 \\ 0 & 2 & 1 \end{bmatrix}$$

, which consists of following individual components:

- translation matrix: $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_u & T_v & 1 \end{bmatrix}$
- rotation matrix: $\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- scaling matrix: $\begin{bmatrix} S_u & 0 & 0 \\ 0 & S_v & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- shearing (x-axis): $\begin{bmatrix} 1 & 0 & 0 \\ H_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- shearing (y-axis): $\begin{bmatrix} 1 & H_u & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

The translation values (a_{31} and a_{32}) can be derived directly from the figure 30, because they are equivalent to the coordinates of the first point of the distorted quadrangle. Instead of computing θ, the type of rotation can be derived alternatively from the shearing values (a_{12} and a_{21}), because they describe the gradients of the corresponding lines. For the example above $\arctan a_{12}$ results in an angle $\approx -12°$ because the deflection goes upwards, and for a_{21} we get $\arctan a_{21} \approx 13°$. The scaling parameters (a_{11} and a_{22}) are necessary, because the image size might change due to rotation or shearing. The negative value of the parameter a_{13} implies an enlargement of the image on the x-axis. Accordingly, the positive value of a_{23} implies a reduction for the image width.

Appendix B
Rotation Angle Equations

Rotation angles are derived from the coordinates of the bounding box rectangle and the coordinates of the inlying distortion quadrangle (see figure 31).

Based on the four coordinates of the distortion quadrangle $D_k(x_k, y_k)$ for $k = 0, 1, 2, 3$ and the three points $A(x_4, y_4)$, $B(x_5, y_5)$ and $C(x_6, y_6)$ the goal is to compute $\alpha = \sphericalangle BAC$:

$$\overline{D_0D_3} = \sqrt{(|x_0 - x_3|)^2 + (|y_0 - y_3|)^2} \approx 2.2$$
$$\overline{D_1D_2} = \sqrt{(|x_1 - x_2|)^2 + (|y_1 - y_2|)^2} \approx 5.1$$
$$\overline{D_0D_1} = \sqrt{(|x_0 - x_1|)^2 + (|y_0 - y_1|)^2} \approx 9.2$$
$$\overline{D_2D_3} = \sqrt{(|x_2 - x_3|)^2 + (|y_2 - y_3|)^2} \approx 7.1$$

If $\overline{D_0D_3} = \overline{D_1D_2}$ and $y_0 = y_1$: $\alpha = 0$ (rectangle) or if $\overline{D_0D_3} = \overline{D_1D_2}$ and $\overline{D_0D_1} = \overline{D_2D_3}$ (parallelogram), we get: $\tan\alpha = \frac{|y_0 - y_1|}{|x_0 - x_1|}$

For all other cases (trapeze, arbitrary quadrangle):

$$x_4 = (|x_3 - x_0|)/2 + \min(x_0, x_3)$$
$$y_4 = (|y_3 - y_0|)/2 + \min(y_0, y_3)$$
$$x_6 = (|x_2 - x_1|)/2 + \min(x_1, x_2)$$
$$y_6 = (|y_2 - y_1|)/2 + \min(y_1, y_2)$$
$$y_5 = y_4 \qquad x_5 = ?$$

Point of intersection of lines $\overline{D_1D_2}$: $y = y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$ and AB:

$$\Rightarrow x_5 = \frac{(y_4 - y_1)(x_2 - x_1)}{y_2 - y_1} + x_1$$

With the application of the law of cosines we get:

$$\overline{AB} = x_5 - x_4$$
$$\overline{AC} = \sqrt{(|x_6 - x_4|)^2 + (|y_6 - y_4|)^2}$$
$$\overline{BC} = \sqrt{(|x_6 - x_5|)^2 + (|y_6 - y_5|)^2}$$

$$\cos\alpha = \frac{\overline{AB}^2 + \overline{AC}^2 - \overline{BC}^2}{2\overline{AB}\,\overline{AC}}$$
$$\text{if } y_6 > y_5\text{:} \quad \alpha = 360 - \alpha$$
$$\Rightarrow \alpha \approx 5.5^\circ \quad \text{for the example in figure 31.}$$

References

[1] J. J. Weinman, E. Learned-Miller, and A. R. Hanson, "Scene Text Recognition Using Similarity and a Lexicon with Sparse Belief Propagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1733–1746, October 2009.

[2] J. Liang, D. Doermann, and H. Li, "Camera-based Analysis of Text and Documents: a Survey," *International Journal on Document Analysis and Recognition*, vol. 7, pp. 84–104, 2005, 10.1007/s10032-004-0138-z.

[3] L. Z. Chang and S. Z. ZhiYing, "Robust Preprocessing Techniques for OCR Applications on Mobile Devices," in *International Conference on Mobile Technology, Application & Systems*, September 2009.

[4] K. Wang and S. Belongie, "Word Spotting in the Wild," in *European Conference of Computer Vision*, 2010, pp. 591–604.

[5] "knfbReader." [Online]. Available: http://www.knfbreader.com

[6] "Word Lens." [Online]. Available: http://questvisual.com/

[7] Q. Zhu, M.-C. Yeh, and K.-T. Cheng, "Multimodal Fusion using Learned Text Concepts for Image Categorization," in *ACM International Conference on Multimedia*, 2006, pp. 211–220.

[8] D. Lopresti and J. Zhou, "Locating and Recognizing Text in WWW Images," *Information Retrieval*, vol. 2, no. 2-3, pp. 177–206, May 2000.

[9] W. Wu, X. Chen, and J. Yang, "Incremental Detection of Text on Road Signs from Video with Application to a Driving Assistant System," in *ACM International Conference on Multimedia*, 2004, pp. 852–859.

[10] R. G. Casey and E. Lecolinet, "A Survey of Methods and Strategies in Character Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 690–706, July 1996.

[11] "ICDAR Robust Reading Dataset," 2003. [Online]. Available: http://algoval.essex.ac.uk/icdar/Datasets.html

[12] S. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "ICDAR 2003 Robust Reading Competitions," in *International Conference on Document Analysis and Recognition*, August 2003, pp. 682–687.

[13] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, K. Ashida, H. Nagai, M. Okamoto, H. Yamamoto, H. M. Miyao, J. Zhu, W. Ou, C. Wolf, J.-M. Jolion, L. Todoran, M. Worring, and X. Lin, "ICDAR 2003 Robust Reading Competitions: Entries, Results, and Future Directions," *International Journal on Document Analysis and Recognition*, vol. 7, no. 2-3, pp. 105–122, 2005.

[14] "Chars74K Dataset," 2009. [Online]. Available: http://www.ee.surrey.ac.uk/CVSSP/

demos/chars74k/

[15] T. E. de Campos, M. R. Babu, and M. Varma, "Character Recognition in Natural Images," in *International Conference on Computer Vision Theory and Applications*, 2009, pp. 273–280.

[16] "Street View Text Dataset," 2010. [Online]. Available: http://vision.ucsd.edu/~kai/svt/

[17] "Google Street View." [Online]. Available: http://maps.google.com

[18] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting Text in Natural Scenes with Stroke Width Transform," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2963–2970.

[19] "Microsoft Text Detection Database," 2010. [Online]. Available: http://research.microsoft.com/en-us/um/people/eyalofek/text_detection_database.zip

[20] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "LabelMe: A Database and Web-Based Tool for Image Annotation," *International Journal of Computer Vision*, vol. 77, pp. 157–173, May 2008.

[21] "LabelMe Dataset." [Online]. Available: http://labelme.csail.mit.edu/

[22] "ImageMagick." [Online]. Available: http://www.imagemagick.org

[23] C. F. Batten, "Autofocusing and Astigmatism Correction in the Scanning Electron Microscope," Master's thesis, University of Cambridge, August 2000.

[24] R. Ferzli and L. J. Karam, "A No-Reference Objective Image Sharpness Metric Based on the Notion of Just Noticeable Blur (JNB)," *IEEE Transactions on Image Processing*, vol. 18, no. 4, pp. 717–728, 2009.

[25] G. Wolberg, *Digital Image Warping*. Los Alamitos, CA, USA: IEEE Computer Society Press, 1994.

[26] "NEOCR Dataset." [Online]. Available: http://www6.cs.fau.de/research/projects/pixtract/neocr

[27] J. B. Kruskal, *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley, 1983, ch. An Overview of Sequence Comparison, pp. 1–44.

[28] G. Navarro, "A Guided Tour to Approximate String Matching," *ACM Computing Surveys*, vol. 33, no. 1, pp. 31–88, March 2001.

[29] R. W. Hamming, "Error Detecting and Error Correcting Codes," *The Bell System Technical Journal*, vol. 29, no. 2, pp. 147–160, April 1950.

[30] V. I. Levenshtein, "Binary Codes Capable of Correcting Deletions, Insertions, and Reversals," *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, February 1966.

[31] F. J. Damerau, "A Technique for Computer Detection and Correction of Spelling Errors," *Communications of the ACM*, vol. 7, no. 3, pp. 171–176, March 1964.

[32] S. B. Needleman and C. D. Wunsch, "A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins," *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443–453, March 1970.

[33] C. A. Ratanamahatana, J. Lin, D. Gunopulos, E. Keogh, M. Vlachos, and G. Das, *The Data Mining and Knowledge Discovery Handbook*. Springer, 2005, ch. Mining Time Series Data, pp. 1069–1103.

[34] L. Bergroth, H. Hakonen, and T. Raita, "A Survey of Longest Common Subsequence Algorithms," in *International Symposium on String Processing and Information Retrieval*, September 2000, pp. 39–48.

[35] A. Apostolico and C. Guerra, "The Longest Common Subsequence Problem Revisited," *Algorithmica*, vol. 2, no. 1-4, pp. 315–336, 1987.

[36] D. S. Hirschberg, "A Linear Space Algorithm for Computing Maximal Common Subsequences," *Communications of the ACM*, vol. 18, no. 6, pp. 341–343, June 1975.

[37] M. A. Jaro, "Advances in Record Linkage Methodology as Applied to the 1985 Census of Tampa Florida," *Journal of the American Statistical Association*, vol. 84, no. 406, pp. 414–420, June 1989.

[38] W. E. Winkler, "String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage," in *Section on Survey Research Methods*, 1990, pp. 354–359.

[39] M. J. Huiskes and M. S. Lew, "The MIR Flickr Retrieval Evaluation," in *ACM International Conference on Multimedia Information Retrieval*, 2008, pp. 39–43.

[40] "The MIRFLICKR-25000 Image Collection," 10. 12. 2010. [Online]. Available: http://press.liacs.nl/mirflickr/