Text Detection and Pose Estimation for a Reading Robot

Marius Bulacu[1], Nobuo Ezaki[2] and Lambert Schomaker[1]
[1] *Dept. of Artificial Intelligence, University of Groningen,* [2] *Information and Control Engineering Dept., Toba National College of Maritime Technology*
[1]*The Netherlands,* [2]*Japan*

1. Introduction

This chapter presents two basic processes needed in a vision system for a mobile robot which is capable of reading the textual content encountered in its environment. We describe the general system outline and we present the structure and the experimental evaluation of the system modules that we built until the present. These two modules are designed to solve two essential problems facing our robotic reading system: text detection from scene images and text-pose estimation necessary for navigation guidance. Finding text in a natural image is the first problem that must be addressed and we propose four text-detection methods based on *connected components* (CoCos). We tested the effectiveness of these methods on the ICDAR 2003 Robust Reading Competition data. After text detection, for maneuvering the robot, we must estimate the orientation of the text surface with respect the viewing axis of the camera mounted on the robot. We propose an active-vision method for estimating the pose of a planar text surface using the *edge direction distribution* (EDD) as input to a neural network. We develop and evaluate a mathematical model to analyze how the EDD changes under canonical rotations and orthographic projection. We collected a set of camera-captured images with text in front-parallel view and, by applying single-axis synthetic rotations on these images, we obtain the data necessary to train and test the neural network. Further work will be directed at integrating our text detection and pose estimation modules within a complete robotic vision system.

2. Reading Robot

Our main research effort is concentrated on developing a vision system for an autonomous robot that will be able to detect and read the text encountered in its environment. A reading robot is an interesting proof of concept and building it is challenging as it raises many essential computer-vision problems requiring real-time solutions. Provided with text reading capabilities, mobile robots can capture this information intended for human visual communication and use it for navigation and task execution.

Our robot is essentially a computer on wheels with a controllable camera on top (equipped also with sonar sensors and odometry). Camera-based text reading in 3D space is a more defiant problem then classical optical character recognition (OCR) used for processing scanned documents. Two major aspects are different and play a very important role: the text areas must be first found in the image because text may be anywhere in the scene (*text detection*) and, secondly, the orientation of the text surface with respect to the camera viewing axis needs to be inferred (*pose estimation*) as it will be different from case to case. Our aim is to solve these problems by exploiting the robot's ability to move. We explore therefore, in a robotic setting, the role of active vision in the machine recognition of text in 3D (robotic OCR).

We can identify 4 important modules that need to be integrated in the vision system of the reading robot:

- text detection
- text-pose estimation
- robot/camera motion
- character classification.

The present chapter focuses on the first two modules of the system: the text detection module that finds the location of text in a natural scene image and the pose estimation module that computes the orientation of the text surface with respect to the viewing axis of the camera mounted on the robot. Once this information is known, the robot can be maneuvered and the camera can zoom-in to obtain (if possible) a high-resolution front-parallel view of the text, which, in principle, would give the best final OCR result.

Text detection is essentially a segmentation problem and, as such, it entails a known difficulty well established in the pattern recognition community. A perfect solution for this problem is hard to find. We adopt a connected-component-based approach for text detection. Connected components (CoCos) have the advantage of offering a quick access to the objects in the image. For a given text instance in a scene, the characters are usually similar in size and placed in a horizontal string. Using rules regarding size, aspect ratio and relative positioning can reduce the indiscriminate number of CoCos extracted from an image to obtain the final candidate text areas. In this chapter, we propose and evaluate four text detection methods using CoCos.

For tackling the problem of text-pose estimation we adopt a texture-based approach. The texture feature that we shall use is the angular distribution of directions in the text region extracted from the edges. This distribution changes systematically with the rotation angle and we develop a mathematical model to describe this trend. We then show how the rotation angle of the text surface can be recovered back from the edge-direction distribution (EDD) using a feed-forward neural network.

Here we consider, for simplicity, only single-axis rotations starting from front-parallel views. We impose this severe constraint in order to obtain a basic initial working system, perfectible in the future. Because robot motion is confined to the horizontal plane, only the rotation angle β of text around the vertical axis (Y) can be used for repositioning (see fig. 7). Initial experiments will be conducted in a simplified environment to test the basic robot functionality. In the totally unconstrained case, view normalization for the other rotations, around X and Z, will have to be performed in software.

This chapter is organized as follows. In section 3, we give a general overview of the field, providing pointers to literature and commenting on prospective applications. In section 4, we propose four CoCo-based methods for text detection and analyze their strengths and weaknesses. The next section presents an overall view of general pose-estimation methods and relates our approach to the more generic problem of shape-from-texture. In section 6, we present our texture-based method for text-pose estimation: we describe the extraction of the EDD, our underlying mathematical model and the use of the neural network. In the results section, first we evaluate the performance of the CoCo-based text detection methods when used individually and in combination. We then numerically check the validity of the theoretical EDD transform model and we evaluate the performance of the proposed text-pose estimation method in terms of angular error. A discussion of the strengths and weaknesses of our approach follows and conclusions end the chapter.

As regards our general approach to the problem of reading robotic systems, the following remarks are in place here. There is no formal mathematical solution to this complex problem. Therefore, a synthesizing approach has been followed, on the basis of what is known as 'best practice' in image processing, on the basis of geometric modeling and by using empirical evaluation.

3. Text Recognition in Real Images

Text detection and recognition in still images and video receives constant research attention, the references pointing to a number of systems (Clark & Mirmehdi, 2002b; Lienhart & Wernicke, 2002; Gao et al., 2001; Yang et al., 2001; Lopresti & Zhou, 2000; Li et al., 2000; Zhong et al., 2000; Wu et al., 1999) and a recent overview of camera-based document image analysis (Doermann et al., 2003). Two major categories of text have been identified: *scene text* incidentally picked up as part of the recorded scenery and *overlay text* added on the image by post hoc editing. Overlay text appears mostly in video and is carefully directed to carry information. It is assumed to appear in front-parallel view and with clear contrast, being less problematic to detect and recognize. Robust recognition of scene text is a more difficult problem and the robot has to confront it.

Automatic text detection and reading in natural images has many potential applications. To mention only a few: Intelligent transport systems (e.g. automatic reading of traffic signs or car license plates); office space with pervasive computing (e.g. intelligent cameras might be watching over a desk and automatically respond to commands to process the captured text); image retrieval (images can extracted from large multimedia databases using their text content).

Intelligent wearable cameras for visually impaired persons represent another particularly interesting application and an important research subject (Kang & Lee, 2002; Zandifar et al., 2002). The number of visually impaired persons is increasing every year due to eye disease, traffic accidents etc (e.g. in Japan alone there are about 200,000 people with acquired blindness). Such a support system, using a portable computer, a controllable camera and a speech synthesizer, can help an unaccompanied blind person by providing auditory information about the textual content of a scene (e.g. street or shop name, restaurant menu etc). This type of application shares many points in common with our robotic research theme: the acquired scene images are complex and their textual content has high variability in pose and in character size, color, font and contrast. Text-pose estimation, which in our case is primarily used for robot navigation, might also play a role, if coupled with acoustic feedback, in helping a blind person.

Text detection methods have been broadly classified in two categories: *texture based methods* and *connected-component based methods*. The methods in the first category use text texture for detection. To the casual observer, text areas have a distinctive general appearance in natural scenes. They exhibit a significant content of high frequencies, considerable variation in the gray levels, high density of edges, oriented in multiple directions, and these attributes are uniform over a larger area. These rich textural features are exploited for text detection in combination with pattern-recognition techniques (k-means clustering, neural networks).

We opted to investigate the CoCo-based text detection methods for our robotic application because of their relative simplicity and effectiveness (Liu et al., 1998; Matsuo et al., 2002; Yamaguchi et al., 2003).

4. Text Detection Methods

In this section, we describe four connected-component based methods for detecting text in natural scene images. All four methods have the following processing steps:

- image preprocessing
- image binarization
- CoCo extraction
- CoCo selection using heuristic rules.

Only the image preprocessing step will be different from one method to another. Image binarization, CoCo extraction and selection will always be performed in similar fashion for all four proposed methods.

Using CoCos for detecting text in natural images raises a number of problems. The basic question is: "In what space are the pixels connected?" The simplest example concerns bitonal (B/W) images, where the concept of connectedness is straightforward. For gray-scale images, things already start to be complicated and binarization is needed for CoCo extraction. However, there are several ways in which a meaningful connectedness in the image can be imposed on the basis of local features such as color and texture. Assuming that such features can be transformed into a single scalar per pixel, a suitable binarization method may provide the basis for the CoCo extraction.

An inappropriate threshold might wipe away all or part of the text present in the image. A decision is also needed on whether to use a local or a global binarization method. For a defined class of images, local binarization methods can be adapted to perform significantly better than global methods (Trier & Jain, 1995). More research is needed in this direction, so we report here only on our results obtained using a global binarization method. Ideally, text characters should be extracted as individual CoCos. It is common knowledge that, unfortunately, on many occasions, this is not the case. Often, a CoCo might contain only a part of a broken character or several characters lumped together. This is an important difficulty and the different image preprocessing methods used represent an effort to confront this inherent problem. In the end, however, quite a significant number of errors still remain.

4.1 Binarization method

For binarization we use Otsu's classical method (Otsu, 1979). It is a simple, popular and quite effective global binarization method. The same threshold is used for the entire image.

Otsu's method automatically selects the binarization threshold that optimally partitions the gray-level histogram of the image into two separate subhistograms. The threshold T is selected such that the combined within-class variance σ_w^2 of the thresholded foreground and background pixels is minimized. This is also equivalent to maximizing the between-class variance σ_b^2 for the two classes of pixels:

$$
\begin{aligned}
T &= arg\,min(\sigma_w^2) = arg\,min(\omega_1\sigma_1^2 + \omega_2\sigma_2^2) \\
&= arg\,max(\sigma_b^2) = arg\,max[\omega_1(\mu_1 - \mu)^2 + \omega_2(\mu_2 - \mu)^2] \\
&= arg\,max[\omega_1\omega_2(\mu_1 - \mu_2)^2]
\end{aligned}
\tag{1}
$$

where μ is the average value for the entire gray-level histogram, ω_1, ω_2 are the integrals of the two subhistograms (i.e. the proportions of pixels in the two classes after thresholding), σ_1, σ_2 are the standard deviations of the two subhistograms and μ_1, μ_2 are their average values. In programs, the third expression is usually implemented as it allows for an elegant and fast recursive computation.

4.2 Extraction of small characters using mathematical morphology operations

The first method we propose targets the small characters and it is based on mathematical morphology operations. We use a modified top-hat processing. In general, top-hat contrast enhancement is performed by calculating the difference between the original image and the opening image (Gu et al., 1997). As a consequence, the top-hat operation is applicable when the pixels of the text characters have higher values than the background. Additionally, in (Gu et al., 1997), the difference between the closing image and the original image is also used for text detection when character pixels have lower values than the background. This method is very effective, however it becomes computationally expensive if a large filter is used in order to extract large characters.

We developed an invariant method applicable to small characters. We use a disk filter with a radius of 3 pixels and we take the difference between the closing image and the opening image. The filtered images are binarized and then CoCos are extracted.

Top-hat image processing emphasizes the thin structures present in the image (thinner than the diameter of the structural filter used). As such, this method is only applicable for small characters (less than about 30 pixels in height). Besides text characters, other thin structures present in the image will also be detected (e.g. thin window frames).

This method detects connected text areas containing several small characters. As western text consists of characters that are usually horizontally placed, we take horizontally long areas (1 < *width / height* < 25) from the output image as the final candidate text regions (see figure 1).

4.3 Three methods for extracting large characters

We propose three extraction methods for large characters (more than about 30 pixels in height). The first two are based on Sobel edge detection, an image processing technique presented in detail in section 6.1. The third text extraction method is based on RGB color information. Fig. 2 shows how the three methods act on a sample image.

Each method extracts connected components that represent candidate text areas. Decision rules based on the sizes and relative positioning of these areas are afterwards used to prune the number of possibilities and reduce the large number of false hits.

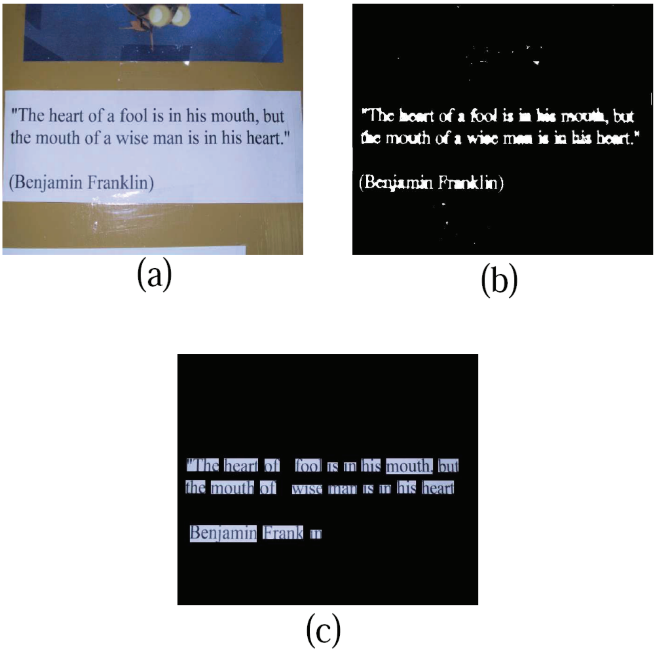

"The heart of a fool is in his mouth, but
the mouth of a wise man is in his heart."
(Benjamin Franklin)
(a)
(b)
(c)

Figure 1. Extraction of small characters using morphological operations: a) original image, b) difference between closing and opening, c) extracted area

- **Character extraction from the edge image**

In this method, Sobel edge detection is applied on each color channel of the RGB image. The three edge images are then combined into a single output image by taking the maximum of the three gradient values corresponding to each pixel. The output image is binarized using Otsu's method and then CoCos are extracted.

This method fails when the edges of several characters are lumped together into a single large CoCo that is eliminated by the selection rules. This often happens when the text characters are close to each other or when the background is not uniform (see fig. 3).

- **Character extraction from the reverse edge image**

This method is complementary to the previous one; the binary edge image is reversed before connected component extraction. It will be effective only when characters are surrounded by continuous connected edges and the inner ink area is not broken (as in the case of boldface characters).

- **Color-based character extraction**

The three methods proposed until now use morphological and edge information for text detection. However, color information is also important, because, generally, text has almost the same color for a given instance encountered in the scene. The first step is to simplify the color space and we reduce it to 8 colors by the following procedure. We apply Otsu binarization independently on the three RGB color channels. Each pixel can now have only $2^3 = 8$ possible combinations of color values. We separate the 8 binary images and then we extract and select CoCos on each one independently. This method makes evident that global thresholding is not always appropriate and text characters can be lost.

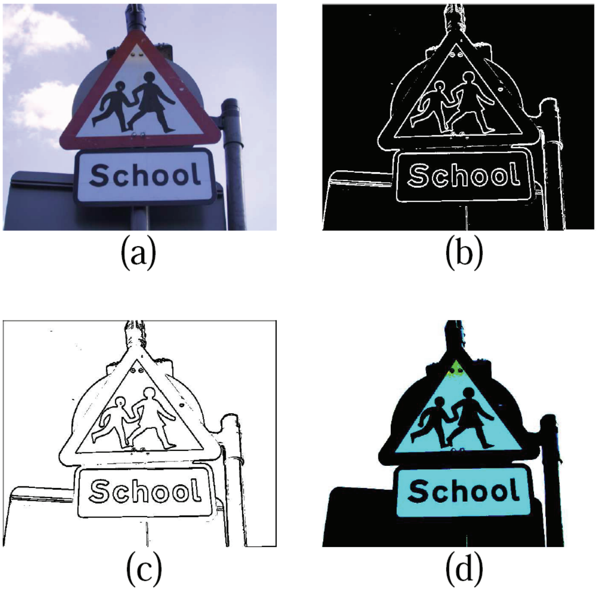


Figure 2. A "good" example: a) original image, b) edge image, c) reverse edge image, d) 8-color image

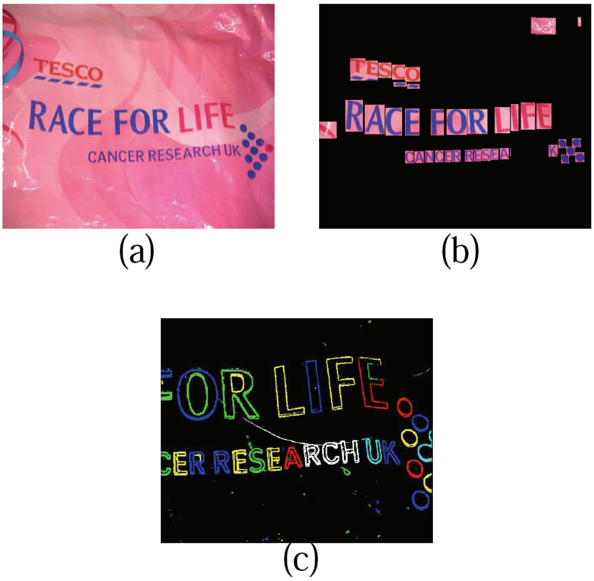


Figure 3. Several characters (namely 'R', 'C', 'H') are lumped into a single CoCo because their edges are connected by a background structure: a) original image, b) extracted area, c) close-up view of the problematic CoCo

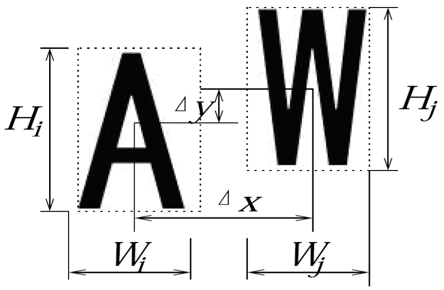


Figure 4. Size and relative positioning of extracted CoCos. W_i and H_i are the width and height of an extracted CoCo; Δx and Δy are the distances between their centers of gravity

4.4 Connected component selection rules

It can be noticed that, up to now, the proposed methods are very general in nature and not specific to text detection. As expected, many of the extracted CoCos do not actually contain text characters. At this point, rules are used to filter out the false detections (see fig. 4).

We impose constraints on the aspect ratio and area to decrease the number of non-character candidates:

$$0.1 < \frac{W_i}{H_i} < 2,\ 50 < W_i H_i \quad (2)$$

An important observation is that, generally, text characters do not appear alone, but together with other characters of similar dimensions and usually regularly placed in a horizontal string. We use the following rules to further eliminate from all the detected CoCos those that do not actually correspond to text characters:

$$0.5 < \frac{H_i}{H_j} < 2,\ \Delta y < 0.2\,max(H_i, H_j),\ \Delta x < 2\,max(W_i, W_j) \quad (3)$$

The system goes through all possible combinations of two CoCos and only those complying with all the selection rules succeed to the final proposed text region (see fig. 5).

The actual thresholds used in the CoCo selection rules can be further heuristically tuned for the individual methods. This remains an open problem. The proposed values work reasonably well on the test dataset (containing western characters). The selection rules do not completely eliminate all the erroneous non-text CoCos (see fig. 6).

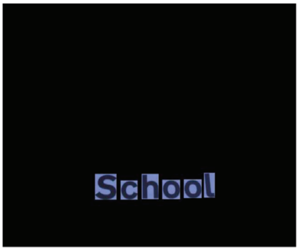


Figure 5. Final result for the example given in fig. 2

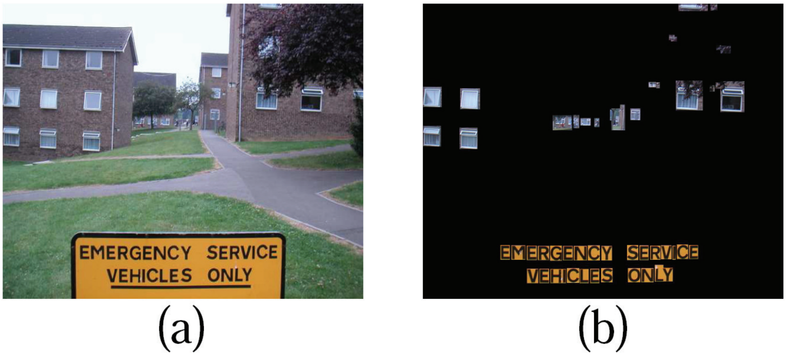


Figure 6. The windows of a building comply with CoCo selection rules and they are falsely detected as text characters

5. Approaches to Text-Pose Estimation

After text detection, the orientation of the text surface must be determined in order to provide navigation guidance to the mobile robot.

A very effective solution to text-pose estimation is based on finding vanishing points of text lines (Clark & Mirmehdi, 2002a; Myers et al., 2001). This type of knowledge-based approach has to impose restrictions on text layout (a minimum number of lines must be present, of sufficient length, with consistent paragraph justification) and the search for vanishing points is computationally expensive.

We adopt a different approach that can best be described as a simple shape-from-texture model. Determining the orientation (pose) and curvature (shape) of 3D surfaces from image texture information is a core vision problem. The proposed solutions make assumptions regarding the texture (isotropic (Garding, 1993) or homogenous (Malik & Rosenholtz, 1997)) and type of image projection (perspective (Garding, 1995; Clerc & Mallat, 1999) or orthographic (Super & Bovik, 1995)).

However, text texture does not have texels, it is homogeneous only in a stochastic sense and also, as we shall see, strongly directional, being a difficult candidate for the classical shape-from-texture algorithms.

We assume that text lies on a planar surface and we consider only single axis rotations. In this case, the general shape-from-texture problem reduces to determining the *slant angle* (the angle between the normal and the viewing axis Z) for rotations around X and Y (see fig. 7). For rotations around Z, the text surface remains parallel to the image plane and only text skew must de determined (a problem aptly addressed in the document analysis field).

General shape-from-texture algorithms rely on differential distortions in the local spatial frequency spectra of neighboring image patches. In contrast, we will use the edge-direction distribution (EDD) as a general texture signature for the entire text region and we will recover the rotation angle from it using a neural network. Realizing that a lot of information is disregarded prematurely, we consider our method presented here as a quick and helpful way of providing navigation guidance to a mobile robot, rather then a broad and generic solution.

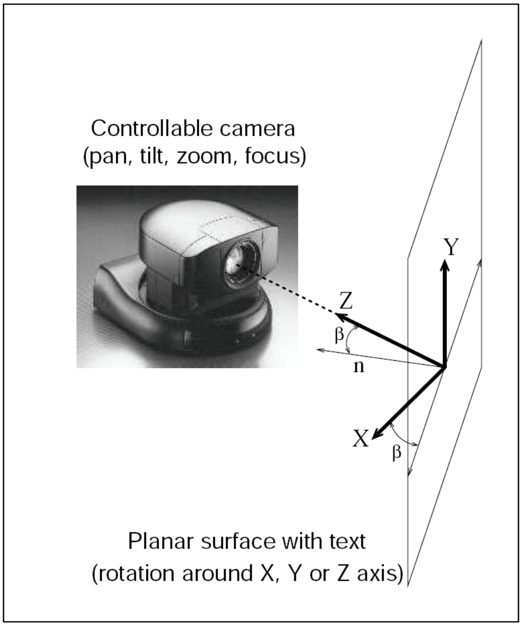


Figure 7. Experimental setup for text-pose estimation

6. Our Text-Pose Estimation Method

As demonstrated in (Clark and Mirmehdi, 2002b), the EDD and measures derived from it, such as symmetry and spread over directions, are very effective features for text detection as well. Here we will explore its use for pose estimation.

6.1 Extraction of the edge-direction distribution

As mentioned, one very important texture descriptor is the probability distribution of edge directions in the text area. EDD extraction starts with the classical Sobel edge-detection method, which is also used for two of the CoCo-based text detection methods described section 4.3 of this chapter.

Two orthogonal Sobel kernels S_x and S_y (eq. 4) are convolved with the image I (in eq. 5, $\otimes$ represents the convolution operator). The responses G_x and G_y represent the strengths of the local gradients along the x and y directions and G is their resultant total gradient (eq. 5). The orientation angle $\phi^{/}$ of the gradient vector G measured from the horizontal (gradient phase) can be computed as in eq. 6. A final correction of 90° (eq. 6) is necessary to go from gradient-direction $\phi^{/}$ to edge-direction ϕ, which is a more intuitive measure.

As the convolution runs aver the image, we build an angle histogram of the edge-directions. We count into the histogram bins only the pixels where G surpasses a chosen threshold (10% in our implementation). This makes sure that we take into consideration only the strong edge regions and not the quasi-uniform larger areas. In the end, the edge-direction histogram is normalized to a probability distribution $p(\phi)$.

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{4}$$

$$G_x = S_x \otimes I, G_y = S_y \otimes I, G = \sqrt{G_x^2 + G_y^2} \tag{5}$$

$$\phi^{/} = arctan(\frac{G_y}{G_x}), \phi = \phi^{/} + \frac{\pi}{2} \tag{6}$$

The edge-direction distribution $p(\phi)$ is the texture feature that we shall use in the sequel for pose-estimation. The EDD is built mainly on phase information, which is known to be important for vision.

6.2 Text rotation in 3D and transform model for the edge-direction distribution

In this subsection, we analyze how the edge-direction distribution changes with the rotation angle. We shall consider only single axis rotations of a planar text surface under orthographic projection (a similar analysis under perspective projection would be mathematically more unwieldy).

- **Rotation around X axis**

Consider a needle OA of length *l*₀ initially contained in the front-parallel plane XOY and oriented at angle ϕ_0 with respect to the horizontal. We rotate it by angle $\alpha \in$ (-90°, +90°) around X axis to the new position OA' and then we project it back onto the front-parallel plane to OB (see fig. 8a). The projection OB will be of length l ($l < l_0$) and oriented at angle ϕ ($\phi < \phi_0$) with respect to the horizontal. The initial needle OA and its projection OB will appear at rescaled dimensions in the image. The projection equations are:

$$l_x = l cos\phi = l_0 \, cos\phi_0 \tag{7}$$

$$l_y = l sin\phi = l_0 \, sin\phi_0 \, cos\alpha \tag{8}$$

Forward and backward relations for needle length and orientation are:

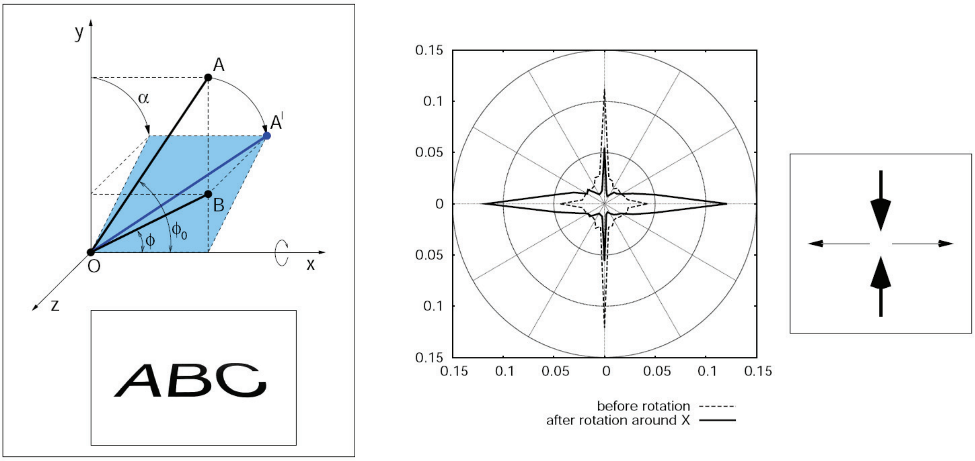
$$l = l_0\sqrt{1 - sin^2\phi_0 \, sin^2\alpha}, \; l_0 = l\frac{\sqrt{1 - cos^2\phi sin^2\alpha}}{cos\alpha} \tag{9}$$

$$\phi = arctan(tan\phi_0 \, cos\alpha), \; \phi_0 = arctan(\frac{tan\phi}{cos\alpha}) \tag{10}$$

If we consider that the needle actually stands for a small edge fragment, we can now describe how the text EDD changes from the initial $p_0(\phi_0)$ to $p_\alpha(\phi)$ after rotation. Two elements need to be taken into account: the length change $l_0 \rightarrow l$ and the angle change $\phi_0 \rightarrow \phi$. We express the new distribution as:

$$h(\phi) = p_0(\phi_0)\frac{l}{l_0}\frac{d\phi_0}{d\phi} \qquad (11)$$

where $h(\phi)$ are some intermediary values. A renormalization of these values is necessary in order to obtain a proper final probability distribution that adds up to 1.



a b

Figure 8. a) Text rotation around X axis. b) EDD change after rotation around X axis by 50°

Therefore, the EDD transform model that we propose is:

$$p_\alpha(\phi) = \frac{h_\alpha(\phi)}{\sum_\phi h_\alpha(\phi)} \text{ with } h_\alpha(\phi) = \frac{cos^2 \alpha}{(1 - cos^2 \phi sin^2 \alpha)^{3/2}} p_0(arctan(\frac{tan\phi}{cos\alpha})). \qquad (12)$$

In equation 12, the intermediary values h undergo renormalization. The expression for h is obtained from equation 11 after evaluating the lengths ratio and the angle derivative.

Unfortunately, the model cannot be formally developed beyond this point, making the numerical analysis our only option. This is the reason why we formulate equation 12 using discrete sums.

The EDD $p_\alpha(\phi)$ corresponding to rotated text cannot be expressed in closed form as a function of the rotation angle α and the base EDD $p_0(\phi_0)$ corresponding to front-parallel text.

Qualitatively, after rotation around X axis, text appears compressed vertically. This foreshortening effect is reflected in the EDD (see fig. 8b): the horizontal component of the distribution increases at the expense of the vertical one. These changes in EDD are more pronounced at larger angles and this makes possible recovering the rotation angle α.

- **Rotation around Y axis**

We apply a similar analysis considering a rotation of angle $\beta \in$ (-90°, +90°) around Y axis (see fig. 9a). The projection equations are:

$$l_x = l\cos\phi = l_0\cos\phi_0\cos\beta \tag{13}$$

$$l_y = l\sin\phi = l_0\sin\phi_0 \tag{14}$$

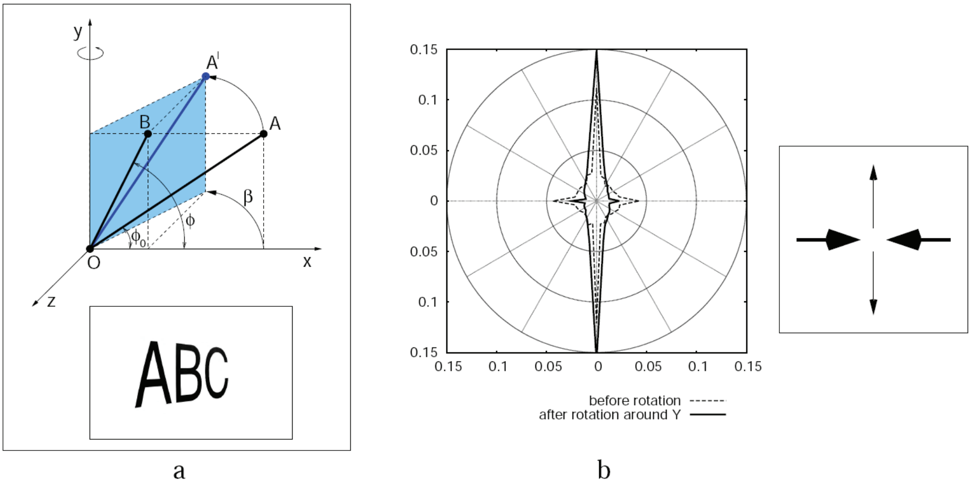


Figure 9. a) Text rotation around Y axis. b) EDD change after rotation around Y axis by 50°

Forward and backward relations for needle length and orientation are:

$$l = l_0\sqrt{1-\cos^2\phi_0\sin^2\beta},\; l_0 = l\frac{\sqrt{1-\sin^2\phi\sin^2\beta}}{\cos\beta} \tag{15}$$

$$\phi = \arctan(\frac{\tan\phi_0}{\cos\beta}),\; \phi_0 = \arctan(\tan\phi\cos\beta) \tag{16}$$

Applying equation 11, the EDD transform model becomes:

$$p_\beta(\phi) = \frac{h_\beta(\phi)}{\sum_\phi h_\beta(\phi)} \text{ with } h_\beta(\phi) = \frac{\cos^2\beta}{(1-\sin^2\phi\sin^2\beta)^{3/2}}p_0(\arctan(\tan\phi\cos\beta)) \tag{17}$$

where h are intermediary values that undergo renormalization.

Here again, $p_\beta(\phi)$ (corresponding to rotated text) cannot be expressed in closed form as a function of the rotation angle β and the base EDD $p_0(\phi_0)$ (corresponding to front-parallel text).

Qualitatively, after rotation around Y axis, text appears compressed horizontally. This foreshortening effect is reflected in the EDD (see fig. 9b): the vertical component of the distribution increases at the expense of the horizontal one. The rotation angle β can be recovered because the changes in EDD are more pronounced at larger angles.

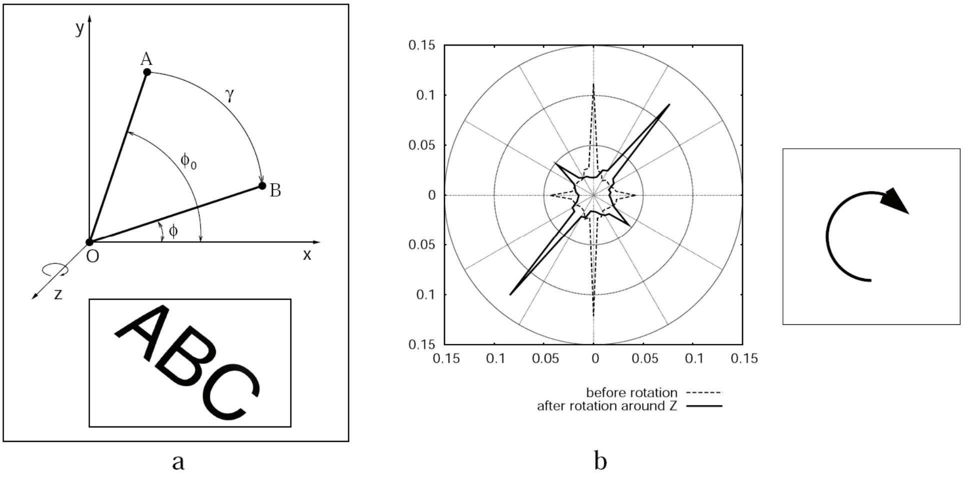


Figure 10. a) Text rotation around Z axis. b) EDD change after rotation around Z axis by 40°

- **Rotation around Z axis**

In this case, text rotation by angle $\gamma \in$ (0°, 360°) simply results in a rotation of the EDD (considered in polar form) by the same angle:

$$\phi = \phi_0 + \gamma,\ \phi_0 = \phi - \gamma \tag{18}$$

$$l = l_0,\ l_0 = l \tag{19}$$

$$p_\gamma(\phi) = p_0(\phi - \gamma) \tag{20}$$

An example showing how the EDD changes for rotations around Z axis is given in fig. 10b.

6.3 The neural network

Very early in our attempts to recover the rotation angle using multilinear regression, we obtained correlation coefficients larger than 0.85 between the cosine squared of the rotation angle and the probability values in the EDD. But an obvious and more appropriate choice is to use a neural network to extract the nonlinear inverse relationship between the EDD and the rotation angle. The ground truth data needed to train and test the neural network is obtained using synthetic rotations starting from front-parallel views.

However, in trying to recover the rotation angle directly from the EDD, two problems appear: font dependence of the base EDD and quadrant ambiguity. One of the very important underlying assumptions is that the base EDD (i.e. that corresponding to the front-parallel view) is almost the same for all machine-print text. Otherwise, a change in the EDD due to font will be wrongly interpreted as a rotation. This assumption is not true: the EDD is actually different for different fonts. This font dependence of the EDD is in fact what we, very successfully, exploited in solving the problem of identifying people based on their handwriting (Bulacu et al., 2003; Bulacu & Schomaker, 2003; Schomaker et al., 2003) (an interesting biometrics method enjoying renewed interest for its forensic applicability).

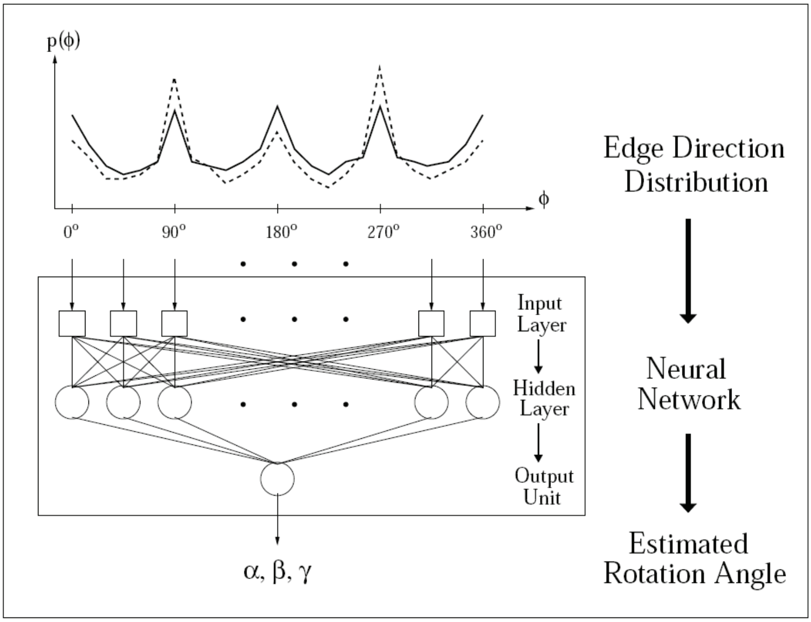


Figure 11. Text-pose estimation method. The neural network has one input unit for every EDD bin (36 in our implementation). The rotation angle is given by the output unit of the network. The difference between two successive EDDs is used as input for rotations around X and Y axes. The EDD itself is used as input for rotations around Z axis

The second problem is quadrant ambiguity for rotations around the X and Y axes: under orthographic projection, text looks the same under rotation of $+\alpha$ and $-\alpha$ ($+\beta$ and $-\beta$). The EDD cannot distinguish between the two situations and this can also be confirmed by observing that the functions depending on the rotation angle appearing in equations 12 and 17 are even.

In order to eliminate this problem, the idea is to consider in the analysis two images rather a single one, the second image being rotated at a fixed small angle δ from to the first. For a chosen δ, in one quadrant, the second image will be closer to the front-parallel view than the first. In the other quadrant, the situation will be reversed. The difference between the two EDDs extracted from the two images will clearly reflect this situation and the neural network has an easy job in inferring it from the training data. Using the difference between two EDDs diminishes also the font-dependence problem, which unfortunately cannot be completely eliminated resulting in inevitable final prediction errors.

The robot, therefore, will need - for rotations around Y axis - to make a small exploratory movement, always to the same side (say e.g. to the right) in order to alleviate the ambiguity. Two implicit assumptions are tacitly adopted here: tracking (the robot needs to look at the same text area) and satisfactory control of the rotation angle δ. We anticipate to solve these constraints using camera (auto)focus and wheel odometry.

For rotations around Z axis the quadrant ambiguity cannot be eliminated. While usually the vertical component of text is stronger than the horizontal one in machine-print, this difference is not reliable enough to obtain accurate predictions based on it. The EDD is almost symmetric to rotations of 90° around Z axis and consequently our solution can only encompass one quadrant. In this case, two images are not needed, the EDD from a single image suffices to determine the rotation angle. This problem is more effectively addressed in the document analysis field. We present our neural network solution only to have a unitary treatment throughout.

7. Experimental Results

7.1 Text detection results

For evaluating the performance of the proposed text detection methods, we used the dataset made available with the occasion of the ICDAR 2003 Robust Reading Competition (Lucas et al., 2003). The images are organized in three sections: Sample, Trial and Competition. Only the first two are publicly available, the third set of images being kept separate by the competition organizers to have a completely objective evaluation. The Trial directory has two subdirectories Trial-Train and Trial-Test. The Trial-Train images should be used to train and tune the algorithms.

As we do not use machine learning in our text detection methods, we included all the images in Trial-Test and Trial-Train for evaluation. This difficult dataset contains a total of 504 realistic images with textual content.

We used a similar evaluation method as that of the ICDAR2003 competition. It is based on the notions of precision and recall. Precision *p* is defined as the number of correct estimates *C* divided by the total number of estimates *E*:

$$p = \frac{C}{E} \tag{21}$$

Recall *r* is defined as the number of correct estimates *C* divided by the total number of targets *T*:

$$r = \frac{C}{T} \tag{22}$$

For a given image, we calculate precision and recall as the ratio between two image areas (expressed in terms of number of pixels). *E* is the area proposed by our algorithm, *T* is the manually labeled text area and *C* is their intersection. We then compute the average precision and recall aver all the images in the dataset.

There is usually a trade-off between precision and recall for a given algorithm. It is therefore necessary to combine them into a single final measure of quality *f*:

$$f = \frac{1}{\alpha / p + (1-\alpha)/r} \tag{23}$$

The parameter α was set to 0.5, giving equal weights to precision and recall in the combined measure *f*.

Our results on the ICDAR 2003 dataset are shown in table 1. The edge-based text detection method obtained top overall performance. In this context, we note that, at ICDAR 2003 (Lucas et al., 2003), the results for the winner of the competition were precision = 55%, recall = 46% and *f* = 50%.

The morphological method did not obtain good overall results because the dataset contains relative large text characters. Consequently, we selected, from the ICDAR 2003 dataset, a group of 55 images that contain only small characters. We evaluated the efficacy of the morphological method on these images and obtained precision = 38%, recall = 55% and *f* = 47%. We tested also the edge based method on these images and obtained precision = 26%, recall = 48% and *f* = 37%. The morphological method seems to be more effective for small characters.

Table 2 shows the results obtained by combining methods. Fusion is performed by ORing the results of the individual methods. By collecting all the candidate areas given by the different methods, we reduce the risk of missing a text instance. This is confirmed also by the high recall rate obtained when all methods are combined using OR. The increase in recall is outbalanced by the decrease in precision. However, for the same *f* value, the method with the highest recall rate is preferable.

In principle, it is naturally the job of the character recognizer to reject many of the false text detections based on its knowledge of character shape. The motivation for combining four text-detection methods is to have a high final recall rate.

Method	Precision	Recall	*f*
Edge (E)	60%	64%	62%
Edge reverse (R)	62%	39%	50%
8 colors (8)	56%	43%	49%
Morphology (M)	41%	16%	28%

Table 1. Results for the individual text extraction methods

Method	Precision	Recall	*f*
E + 8	54%	69%	62%
E + R	56%	70%	63%
E + M	55%	68%	62%
E + R + 8	51%	73%	62%
E + R + 8 + M	48%	76%	62%

Table 2. Results obtained after fusing methods using OR

7.2 Text-pose estimation results

For evaluating the text-pose estimation method another dataset of images was needed. We used a Sony Evi D-31 PAL controllable camera to collect 165 images containing text in front-parallel view. The images contain only text and the background is uniform. They are gray-scale (8 bits/pixel) and have a resolution of 748x556. We strived to obtain sufficient variability in the dataset: 10 different fonts, appearing at different sizes in the images, from a single word to a whole paragraph per image.

In order to test our text-pose estimation method, single-axis synthetic rotations are applied to these images using our own custom-built rotation engine. The number of bins in the EDD was set to N = 36. This was found to give a sufficiently fine description of text texture (10°/bin).

First we verify the validity of our EDD transform model and then we train a neural network to predict the rotation angle and evaluate its performance in terms of angular error.

- **Verification of the theoretical model**

From every image in the dataset, we extract the base EDD corresponding to the front-parallel view. We then randomly select a rotation angle and we theoretically compute (using equations 12, 17, 20) what the EDD should be for the rotated image (forward transform). We then apply the rotation on the image and we directly extract the EDD corresponding to the new pose. We compare the theoretically predicted EDD with the empirically extracted EDD to check the validity of our theoretical model.

An appropriate distance measure is the Bhattacharyya distance:

$$dist(f, g) = 1 - \sum_{i=1}^{N} \sqrt{f_i \, g_i} \tag{24}$$

where f and g are the two EDDs. The distance varies between 0 and 1 and we express it in percentages to have an intuitive measure. If the distance is null, the two distributions are identical.

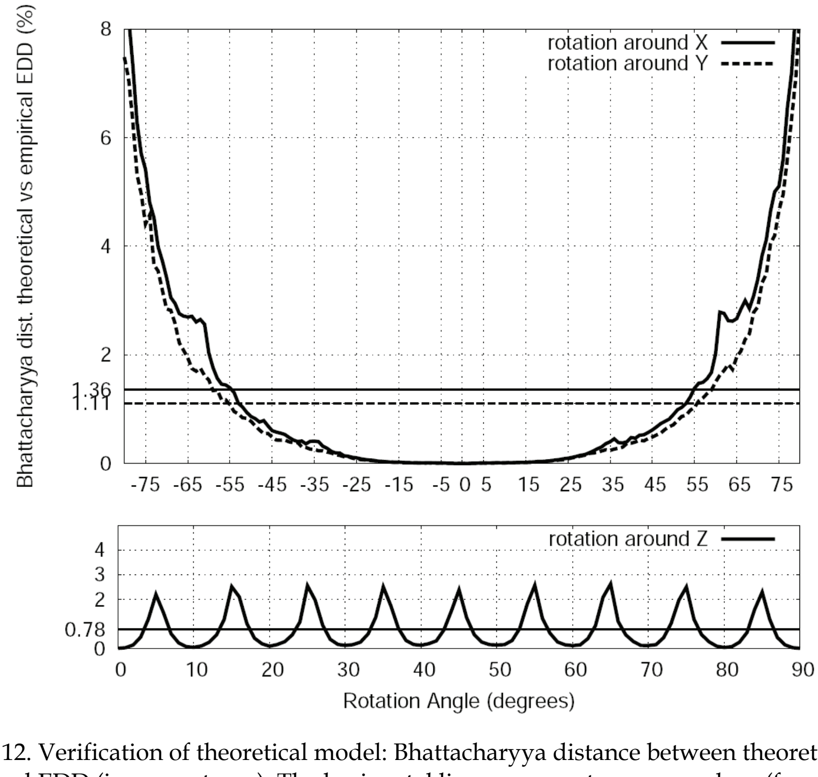


Figure 12. Verification of theoretical model: Bhattacharyya distance between theoretical and measured EDD (in percentages). The horizontal lines represent average values (from table 3 column 2)

We applied 400 random rotations on every image around each axis. The average distance is around 1% (see table 3) and in fig. 12 we show its dependence on the rotation angle.

For rotations around X and Y axes, the error increases with the rotation angle. At larger angles, text is so compressed that letters fuse together in a single lump and our mathematical model no longer correctly describes the changes in the EDD. For rotations around Z, the error is small and does not have a systematic trend, but we can observe a sampling artifact: the error shows an oscillatory behavior as the probability flows from one bin to another of the EDD.

Rotation around	Theoretical Model Error (percentages)	Angle Prediction Error (degrees)
X axis (pitch)	1.36%	3.8°
Y axis (yaw)	1.11%	6.6°
Z axis (roll)	0.78%	2.9°

Table 3. Correlation between theoretical model and empirical data (column 2). Overall angle prediction error (column 3)

- **Evaluation of the angle prediction method**

In order to predict the rotation angle from the EDD (inverse transform), we use a standard feed-forward neural network (3 layers, fully connected between layers, nonlinear transfer functions in the hidden layer). The network architecture is 36x10x1 (see fig. 11). The training method is Rprop (Riedmiller & Braun, 1993), a more effective variant of backpropagation algorithm.

From the beginning, we split the data into 100 images for training and 65 for testing. Every image is then rotated 400 times at randomly chosen angles (40000 training examples, 26000 testing examples). For rotations around X and Y axes, two rotated images are in fact generated with a slight pose difference between them δ = 10°. The network is trained to predict the rotation angle (of the second image for example) using the difference between the two EDDs. For rotations around Z axis, a single EDD is used, but rotations are limited to one quadrant.

Fig. 13 shows how the method performs on two typical examples.

On the test data, we compute the root mean square (RMS) error between the predicted and the real rotation angle. The average angular prediction error is given in table 3. The method demonstrates good performance (3°- 7°angular error).

In fig. 14 we show the dependence of the angular error on the rotation angle. As expected, it can be observed again that the error increases at larger angles for rotations around X and Y axes.

Another interesting observation is that the prediction error for rotations around Y axis is larger than that for rotations around X axis. So we performed the following simple test: we first rotated all the images by 90° around Z and subsequently we applied all the regular analysis. The angular error for rotations around X axis snaps into the range of errors for rotations around Y axis and the reverse (see fig. 14), proving to be an inherent property of the data.

The explanation is that the vertical component of text is more reliable than the horizontal one and, as it is most affected by rotations around X axis, the prediction is more accurate in this case. Unfortunately, rotations around Y axis represent the case of most interest for our robotic application.

For rotations around Z axis, an important observation is that for angles γclose to 0° and 90° the error increases as confusion appears (especially for uppercase characters) between the vertical and the horizontal components, which are the most prominent in the EDD. This is the reason why we opted for a single quadrant solution for this type of rotation.

The method becomes unreliable for small characters (less than 20 pixels in height or width) as the EDD cannot be consistently extracted. We found that the method works well if more than 10 characters are present in the image.

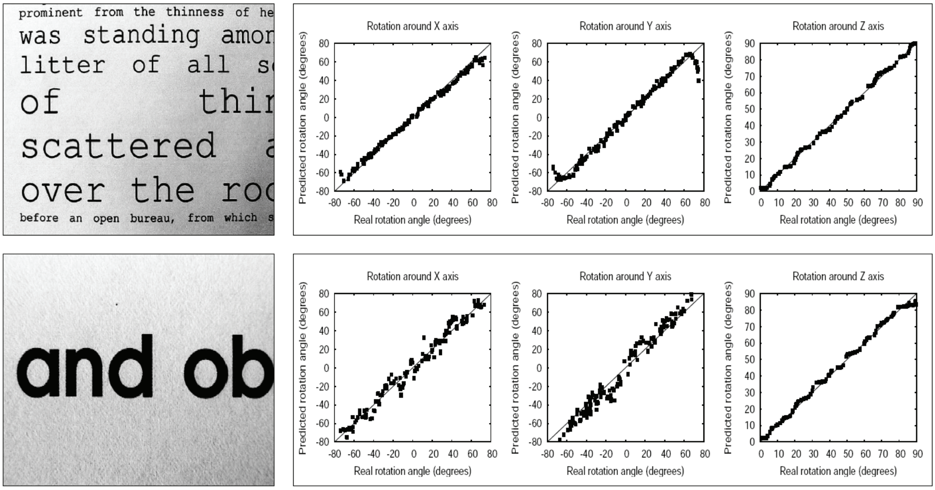


Figure 13. Typical performance: "good" example up, "bad" example down. Angular transfer functions are given for rotations around the X, Y, Z axis, from left to right panel. Ideally all the experimental points would be placed exactly on the diagonal for perfect predictions

In a qualitative evaluation, we found that the proposed method works also on-line in combination with our controllable camera. The neural network, trained and tested off-line on synthetic rotations, estimates reasonably well text-pose during on-line operation under real rotations. The errors are, nevertheless, relatively larger.

It is important to note at this point that the proposed algorithm is lightweight, on average 70 msec being necessary on a 3.0 GHz processor to extract the EDDs from 2 images and run the neural network on their difference to predict the rotation angle. Therefore, using the robot's ability to make small exploratory movements seems like an attractive idea for solving the pose-estimation problem.

8. Discussion

One very important advantage of using CoCos for text detection is that they naturally allow the analysis to take place across scales. In this approach, scale does not represent such a problematic issue because the CoCo extraction process is scale independent. CoCos give a prompt, but rather imperfect, hold to the structures present in the image and CoCo selection is an important complementary step. As the results indicate, further improvement is needed for our text detection module.

For text-pose estimation, we decided to base our analysis on orthographic projection, while most shape-from-texture methods rely on perspective effects. We consider this approach to be more robust, as perspective effects diminish if, after text detection, the camera zooms into the text area (long focal length, small field of view). The proposed texture-based method for text-pose estimation does not impose constraints on text layout. It works even when text lines are not present or they are very short, or when only a few characters are available. We found that Greek fonts can be handled surprisingly well by the same neural network.

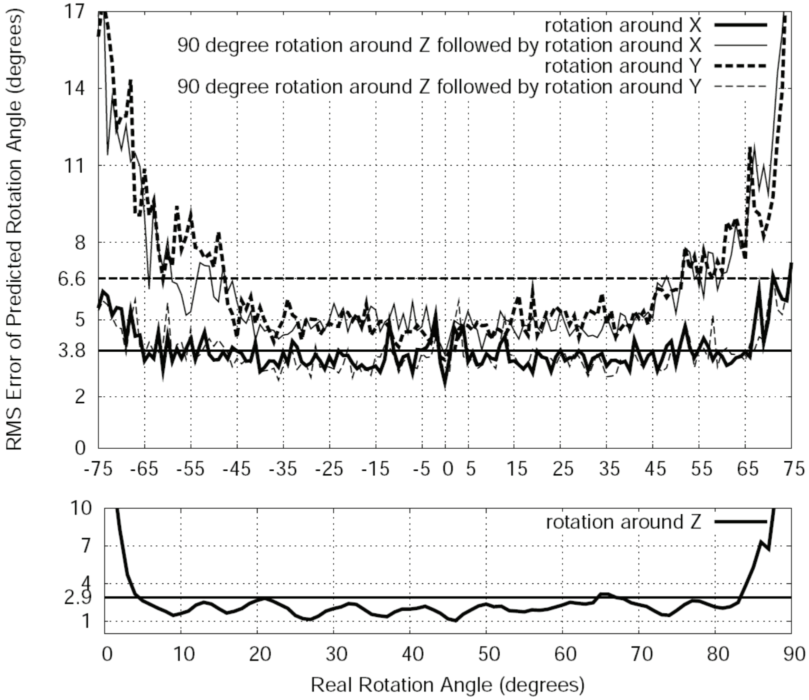


Figure 14. Prediction results: angular error (in degrees). Horizontal lines represent average values (from table 3 column 3)

We treated here only canonical rotations. The method can be directly extended to two-axis rotations, but our experiments are far from conclusive at the moment. We have not addressed free three-axis rotations. A particularly difficult instance is, for example, slanted text and text that is rotating about the surface normal. We will also consider in our future analysis second order moments of the EDD.

Our commitment to a single feature makes our approach limited in the end. However we believe that we have a promising starting point and an effective algorithm to implement on the robot for planning ballistic "text-hunting" movements.

9. Conclusions

In this chapter, we described the text detection and the pose estimation modules of a vision system for a reading robot.

Four connected-component-based methods for text detection have been implemented and evaluated. The most effective proves to be the sequence: Sobel edge detection, Otsu binarization, connected component extraction and rule-based connected component selection. A high recall rate can be achieved by collecting all the candidate text areas proposed by the four individual methods (recall = 76%, precision = 48%, *f* = 62%).

We also presented here a method for estimating the orientation of planar text surfaces using the edge-direction distribution (EDD) in combination with a neural network. We considered single-axis rotations and we developed a mathematical model to analyze how the EDD changes with the rotation angle under orthographic projection. We numerically verified the validity of our underlying mathematical model. In order to solve the quadrant ambiguity and improve performance, for rotations around X and Y axes, we consider a pair of images with a slight rotation difference between them. The change in the EDD is extracted and sent to a feed-forward neural network that predicts the rotation angle corresponding to the last image in the pair. For rotations around Z axis, a single EDD is used, the solution being applicable only to rotations in the first quadrant. The method has been tested off-line with single-axis synthetic rotations and shows good performance. In on-line operation, with real rotations, the errors are relatively larger.

Though limited in scope, the methods proposed here are elegant, quite simple and very fast. Our future work will concentrate on integrating the described modules in the complete vision system of the reading robot.

10. References

Bulacu, M. & Schomaker, L. (2003) Writer style from oriented edge fragments, *Proc. of 10th Int. Conf. on Computer Analysis of Images and Patterns (CAIP 2003): LNCS 2756*, pp. 460-469, Groningen, The Netherlands, Springer

Bulacu, M.; Schomaker, L. & Vuurpijl, L. (2003). Writer identification using edge-based directional features, *Proc. of 7th Int. Conf. on Document Analysis and Recognition (ICDAR 2003)*, Vol. II, pp. 937-941, Edinburgh, Scotland, IEEE Computer Society

Clark, P. & Mirmehdi, M. (2002)a. On the recovery of oriented documents from single images, *Proc. of Advanced Concepts for Intelligent Vision Systems (ACIVS 2002)*, pp. 190-197, Ghent, Belgium

Clark, P. & Mirmehdi, M. (2002)b. Recognizing text in real scenes, *International Journal on Document Analysis and Recognition*, Vol. 4, No. 4, pp. 243-257

Clerc, M., Mallat, S. (1999). Shape from texture and shading with wavelets, *Dynamical Systems, Control, Coding, Computer Vision, Progress in Systems and Control Theory*, Vol. 25, pp. 393-417

Doermann, D.; Liang, J. & Li, H. (2003). Progress in camera-based document image analysis, *Proc. of 7th Int. Conf. on Document Analysis and Recognition (ICDAR 2003)*, Vol. I, pp. 606-616, Edinburgh, Scotland, IEEE Press

Gao, J.; Yang, J.; Zhang, Y. & Waibel, A. (2001). Text detection and translation from natural scenes, *Technical Report CMU-CS-01-139*, Computer Science Department, Carnegie Mellon University, Pittsburgh, USA

Garding, J. (1993). Shape from texture and contour by weak isotropy, *J. of Artificial Intelligence*, Vol. 64, No. 2, pp. 243-297

Garding, J. (1995). Surface orientation and curvature from differential texture distortion, *Proc. of the IEEE Int. Conf. on Computer Vision (ICCV '95)*, pp. 733-739

Gu, L.; Tanaka, N.; Kaneko, T. & Haralick, R. (1997). The extraction of characters from cover images using mathematical morphology, *Transaction of The Institute of Electronics, Information and Communication Engineers of Japan*, Vol. J80-D-II, No. 10, pp. 2696-2704

Kang, S. & Lee, S.W. (2002). Object detection and classification for outdoor walking guidance system, *Proc. of 2nd Int. Workshop Biologically Motivated Computer Vision (BMCV 2002): LNCS 2525*, pp. 601-610, Tuebingen, Germany

Lienhart, R. & Wernicke, A. (2002). Localizing and segmenting text in images, videos and web pages, *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 12, No. 4, pp. 256-268

Li, H.; Doermann, D. & Kia, O. (2000). Automatic text detection and tracking in digital videos, *IEEE Trans. on Image Processing*, Vol. 9, No. 1, pp. 147-156

Liu, Y.; Yamamura, T.; Ohnishi, N. & Sugie, N. (1998). Extraction of character string regions from a scene image, *Transaction of The Institute of Electronics, Information and Communication Engineers of Japan*, Vol. J81-D-II, No. 4, pp. 641-650

Lopresti, D. & Zhou, J. (2000). Locating and recognizing text in www images, *Information Retrieval*, Vol. 2, No. 2/3, pp. 177-206

Lucas, S.M.; Panaretos, A.; Sosa, L.; Tang, A.; Wong, S. & Young, R. (2003). ICDAR 2003 robust reading competitions, *Proc. of 7th Int. Conf. on Document Analysis and Recognition (ICDAR 2003)*, Vol. II, pp. 682-687, Edinburgh, Scotland, IEEE Press

Malik, J. & Rosenholtz, R. (1997). Computing local surface orientation and shape from texture for curved surfaces, *Int. J. Computer Vision*, Vol. 23, No. 2, pp. 149-168

Matsuo, K.; Ueda, K. & Michio, U. (2002). Extraction of character string from scene image by binarizing local target area, *Transaction of The Institute of Electrical Engineers of Japan*, Vol. 122-C, No. 2, pp. 232-241

Myers, G.K.; Bolles, R.C.; Luong, Q.T. & Herson, J.A. (2001). Recognition of text in 3-d scenes, *Proc. of 4th Symposium on Document Image Understanding Technology*, Columbia, Maryland , USA

Otsu, N. (1979). A threshold selection method from gray-level histogram, *IEEE Trans. Systems, Man and Cybernetics*, Vol. 9, pp. 62-69

Riedmiller, M. & Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The Rprop algorithm, *Proc. of the IEEE Int. Conf. on Neural Networks (ICNN)*, pp. 586-591, San Francisco, USA

Schomaker, L.; Bulacu, M. & van Erp, M. (2003). Sparse-parametric writer identification using heterogeneous feature groups, *Proc. of Int. Conf. on Image Processing (ICIP 2003)*, Vol. I, pp. 545-548, Barcelona, Spain

Super, B.J. & Bovik, A.C. (1995). Shape from texture using local spectral moments, *IEEE Trans on PAMI*, Vol. 17, No. 4, pp. 333-343

Trier, O.D. & Jain, A.K. (1995). Goal-directed evaluation of binarization methods, *IEEE Trans on PAMI*, Vol. 17, No. 12, pp. 1191-1201

Wu, V.; Manmatha, R. & Riseman, E.M. (1999). Textfinder: An automatic system to detect and recognize text in images, *IEEE Trans. on PAMI*, Vol. 21, No. 11, pp. 1224-1229

Yamaguchi, T.; Nakano, Y.; Maruyama, M.; Miyao, H. & Hananoi, T. (2003). Digit classification on signboards for telephone number recognition, *Proc. of 7th Int. Conf. on Document Analysis and Recognition (ICDAR 2003)*, Vol. I, pp. 359-363, Edinburgh, Scotland, IEEE Press

Yang, J.; Gao, J.; Zang, Y.; Chen, X. & Waibel, A. (2001). An automatic sign recognition and translation system, *Proceedings of the Workshop on Perceptive User Interfaces (PUI'01)*

Zandifar, A.; Duraiswami, R.; Chahine, A. & Davis, L. (2002). A video based interface to textual information for the visually impaired, *Proc. of 4th Int. Conf. on Multimodal Interfaces (ICMI 2002)*, pp. 325-330, Pittsburgh, USA

Zhong, Y.; Zhang, H. & Jain, A.K. (2000). Automatic caption localization in compressed video, *IEEE Trans. on PAMI*, Vol. 22, No. 4, pp. 385-392