Text-Pose Estimation in 3D Using Edge-Direction Distributions

Marius Bulacu and Lambert Schomaker

AI Institute, Groningen University, The Netherlands
{bulacu, schomaker}@ai.rug.nl

Abstract. This paper presents a method for estimating the orientation of planar text surfaces using the edge-direction distribution (EDD) extracted from the image as input to a neural network. We consider canonical rotations and we developed a mathematical model to analyze how the EDD changes with the rotation angle under orthographic projection. In order to improve performance and solve quadrant ambiguities, we adopt an active-vision approach by considering a pair of images (instead of only one) with a slight rotation difference between them. We then use the difference between the two EDDs as input to the network. Starting with camera-captured front-parallel images with text, we apply single-axis synthetic rotations to verify the validity of the EDD transform model and to train and test the network. The presented text-pose estimation method is intended to provide navigation guidance to a mobile robot capable of reading the textual content encountered in its environment.

1 Introduction

Our main research effort is concentrated on developing a vision system for an autonomous robot that will be able to find and read text. This paper focuses on the problem of text-pose estimation and we propose a method to compute 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 to obtain a front-parallel view of the text, which, in principle, would give the best final OCR result.

Camera-based text reading in 3D space is a more defiant problem than 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.

We built a connected-component-based text-detector that exploits edge, color and morphological information to find candidate text regions from scene images [1]. Though far from perfect, we assume, in the rest of the paper, that text detection is solved.

M. Kamel and A. Campilho (Eds.): ICIAR 2005, LNCS 3656, pp. 625–634, 2005.

After text detection, the orientation of the text surface must be determined. A very effective solution to text-pose estimation is based on finding vanishing points of text lines [2, 3]. This type of knowledge-based approach has to impose restrictions on text layout and the search for vanishing points is computationally expensive.

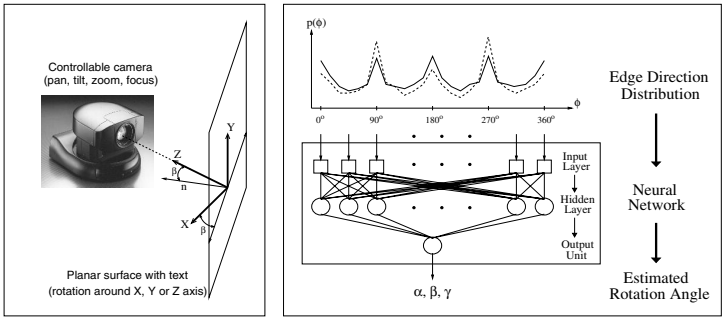


Fig. 1. a) Experimental setup. b) Text-pose estimation method. The neural network has one input unit for every EDD bin. The rotation angle is given by the output unit.

In contrast, we assume 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 [4] or homogeneous [5]) and type of image projection (perspective [6] or orthographic [7]). These general shape-from-texture algorithms rely on differential distortions in the local spatial frequency spectra of neighboring image patches. 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 adopt a simplified, but more robust, feature-based method to solve the problem of text-pose estimation. The 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. 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 the X and Y axes. We impose these constraints in order to obtain a basic module working on the robot in real-time, rather than a broad and generic solution. 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. 1a).

2 Extraction of the Edge-Direction Distribution

The probability distribution of edge directions in the text area is extracted following a classical edge-detection method. Two orthogonal Sobel kernels S_x and S_y are convolved with the image I (in eq. 1, $\otimes$ represents the convolution operator). The responses G_x and G_y represent the strengths of the local gradients along the x and y directions. We compute the orientation angle $\phi^{/}$ of the gradient vector measured from the horizontal (gradient phase). A correction of 90 degrees is then applied to go from gradient-direction ($\phi^{/}$) to edge-direction (ϕ), which is a more intuitive measure.

$$G_x = S_x \otimes I,\ G_y = S_y \otimes I,\ \phi^{/} = arctan(\frac{G_y}{G_x}),\ \phi = \phi^{/} + \frac{\pi}{2} \tag{1}$$

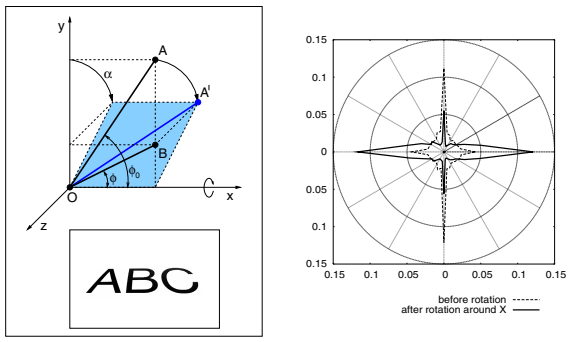
As the convolution runs over the image, we build an angle histogram of the edge-directions by counting the pixels where the gradient surpasses a chosen threshold. In the end, the edge-direction histogram is normalized to a probability distribution $p(\phi)$.

3 Text Rotation in 3D and Transform Model for the EDD

In this section, we analyze how the EDD changes with the rotation angle. We shall consider canonical rotations of a planar text surface under orthographic projection.

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 from 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. 2a). The projection OB will



a b

Fig. 2. a) Text rotation around X axis, b) EDD change after rotation around X by 50°

be of length l ($l < l_0$) and oriented at angle ϕ ($\phi < \phi_0$) from the horizontal. The projection equations are:

$$l_x = l\,cos\phi = l_0\,cos\phi_0,\ l_y = l\,sin\phi = l_0\,sin\phi_0\,cos\alpha \tag{2}$$

Forward and backward relations for needle length and orientation are:

$$l = l_0\,\sqrt{1 - sin^2\phi_0\,sin^2\alpha},\ l_0 = l\,\frac{\sqrt{1 - cos^2\phi\,sin^2\alpha}}{cos\alpha} \tag{3}$$

$$\phi = arctan(tan\phi_0\,cos\alpha),\ \phi_0 = arctan(\frac{tan\phi}{cos\alpha}) \tag{4}$$

The initial needle OA and its projection OB will appear at rescaled dimensions in the image. 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:

$$h(\phi) = p_0(\phi_0)\,\frac{l}{l_0}\,\frac{d\phi_0}{d\phi} \tag{5}$$

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.

Therefore, the EDD transform model that we propose is:

$$p_\alpha(\phi) = \frac{h_\alpha(\phi)}{\sum_\phi h_\alpha(\phi)},\ h_\alpha(\phi) = \frac{cos^2\alpha}{(1 - cos^2\phi\,sin^2\alpha)^{\frac{3}{2}}}\,p_0(arctan(\frac{tan\phi}{cos\alpha})) \tag{6}$$

In eq. 6, the intermediary values h undergo renormalization. The expression for h is obtained from eq. 5 after evaluating the lengths ratio and the angle derivative.

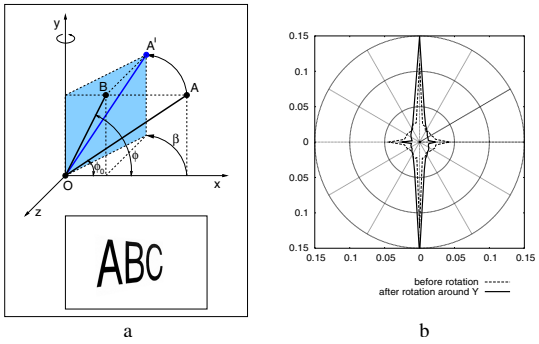
Unfortunately, the model cannot be formally developed beyond this point, making the numerical analysis our only option. This is the reason why we formulate eq. 6 using discrete sums. The EDD $p_\alpha(\phi)$ corresponding to rotated text cannot be expressed in closed form as a function of the rotation 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 (fig. 2b): the horizontal component of the distribution increases at the expense of the vertical one. The 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. 3a). The projection equations are:

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



a b

Fig. 3. a) Text rotation around Y axis, b) EDD change after rotation around Y by 50°

Forward and backward relations for needle length and orientation are:

$$l = l_0 \sqrt{1 - cos^2\phi_0 \, sin^2\beta}, \;\; l_0 = l \, \frac{\sqrt{1 - sin^2\phi \, sin^2\beta}}{cos\beta} \tag{8}$$

$$\phi = arctan(\frac{tan\phi_0}{cos\beta}), \;\; \phi_0 = arctan(tan\phi \, cos\beta) \tag{9}$$

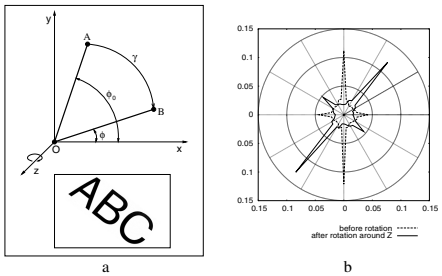
Applying eq. 5, the EDD transform model becomes:

$$p_\beta(\phi) = \frac{h_\beta(\phi)}{\sum_\phi h_\beta(\phi)}, \;\; h_\beta(\phi) = \frac{cos^2\beta}{(1 - sin^2\phi \, sin^2\beta)^{\frac{3}{2}}} \, p_0(arctan(tan\phi \, cos\beta)) \tag{10}$$

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 (fig. 3b): 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.



a b

Fig. 4. a) Text rotation around Z axis, b) EDD change after rotation around Z 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 (see fig. 4):

$$\phi = \phi_0 + \gamma,\ l = l_0,\ p_\gamma(\phi) = p_0(\phi - \gamma) \tag{11}$$

4 Text-Pose Estimation Method

First we attempted to recover the rotation angle using multilinear regression and 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 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.

An important underlying assumptions is that the base EDD 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. We very successfully exploited this fact in solving the problem of identifying people based on their handwriting [8].

The second problem is quadrant ambiguity for rotations around X and Y: 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 6 and 10 are even. For eliminating 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. 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. This will be clearly reflected in the difference between the EDDs extracted from the two images and the neural network will learn it from the training data. Using the difference between two EDDs diminishes also the font-dependence problem. The robot, therefore, will need - for rotations around Y axis - to make a small exploratory movement, always to the same side (e.g. to the right) in order to alleviate the ambiguity.

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.

5 Results

We used a Sony Evi D-31 PAL controllable camera to collect 165 images containing text in front-parallel view (gray-scale, 8 bits/pixel, 748x556 resolution). 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. 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.

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 6, 10, 11) 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 formal model. An appropriate similarity measure between the two EDDs is Bhattacharyya distance: 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.

We applied 400 random rotations on every image around each axis. The average distance is around 1% (see table 1) and in fig. 5a 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

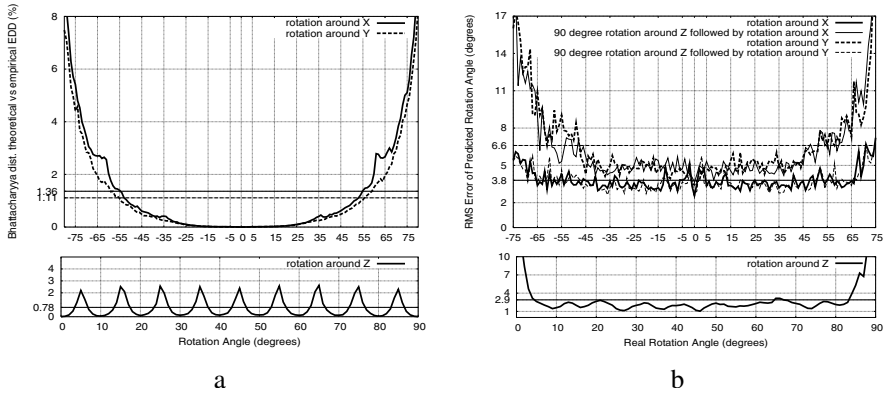


Fig. 5. a) Verification of theoretical model: Bhattacharyya distance between theoretical and measured EDDs (in percentages). b) Prediction results: angular error (in degrees). Horizontal lines represent average values (from table 1).

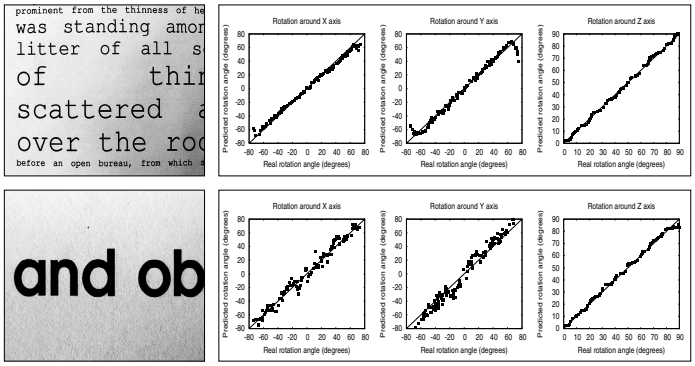


Fig. 6. Typical performance: "good" example up, "bad" example down. Angular predictions are given for rotations around X, Y, Z from left to right panel. Ideally all the experimental points would be placed exactly on the diagonal for perfect predictions.

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.

Evaluation of the Angle Prediction Method

For predicting the rotation angle from the EDD (inverse transform), we use a standard feed-forward neural network (3 layers, fully connected, nonlinear transfer functions in the hidden layer). The network architecture is 36x10x1 (see fig. 1b).

From the start, we split the data into 100 images for training and 65 for testing. Every image is then rotated 400 times at random angles (40000 training examples, 26000 testing examples). For rotations around X and Y, two rotated images are in fact generated with a slight pose difference between them $\delta = 10°$. The network is trained to predict the rotation angle (e.g. of the second image) using the difference between the two EDDs. For rotations around Z, a single EDD is used with rotations limited to one quadrant. Fig. 6 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 1. The method demonstrates good performance (3° - 7° angular error). In fig. 5b 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

Table 1. Correlation between theoretical model and empirical data (column 2). Overall angle prediction error (column 3).

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°

range of errors for rotations around Y axis and the reverse (see fig.5b), 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, we can observe that for angles γ close 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 (see fig. 6). 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. We found that Greek fonts can be handled too by the same neural network. 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. We treated here only canonical rotations. The method can be directly extended to two-axis rotations. We have not addressed free three-axis rotations. 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.

6 Conclusions

We presented 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 text rotation angle. The method has been tested off-line with single-axis synthetic rotations and shows good performance. Though limited in scope, the text-pose estimation method proposed here is elegant, quite simple and very fast. Further work will be directed at integrating this pose estimation method within a complete robotic reading system.

References

1. Ezaki, N., Bulacu, M., Schomaker, L.: Text detection from natural scene images: Towards a system for visually impaired persons. In: Proc. of 17th Int. Conf. on Pattern Recognition (ICPR 2004), Cambridge, UK, IEEE CS (2004) 683–686
2. Clark, P., Mirmehdi, M.: On the recovery of oriented documents from single images. In: Proc. of ACIVS 2002, Ghent, Belgium (2002) 190–197
3. Myers, G.K., Bolles, R.C., Luong, Q.T., Herson, J.A.: Recognition of text in 3-d scenes. In: Proc. of 4th Symposium on Document Image Understanding Technology, Columbia, Maryland , USA (2001)
4. Garding, J.: Shape from texture and contour by weak isotropy. J. of Artificial Intelligence **64** (1993) 243–297
5. Malik, J., Rosenholtz, R.: Computing local surface orientation and shape from texture for curved surfaces. Int. J. Computer Vision **23** (1997) 149–168
6. Clerc, M., Mallat, S.: Shape from texture and shading with wavelets. Dynamical Systems, Control, Coding, Computer Vision, Progress in Systems and Control Theory **25** (1999) 393–417
7. Super, B.J., Bovik, A.C.: Shape from texture using local spectral moments. IEEE Trans on PAMI **17** (1995) 333–343
8. Schomaker, L., Bulacu, M.: Automatic writer identification using connected-component contours and edge-based features of uppercase western script. IEEE Trans on PAMI **26** (2004) 787–798