

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Rotation Angle Estimation Algorithms for Textures and Their Implementations on Real Time Systems

Cihan Ulas, Onur Toker and Kemal Fidanboyulu
*Fatih University,
Turkey*

1. Introduction

In this chapter, rotation angle estimation algorithms for textures and their real time implementations on a custom smart camera called FU-SmartCam is introduced (Ulas et al., 2007) and improved. In the textile industry, weft-straightening is a fundamental problem which is closely related to the rotation angle estimation. Earlier weft-straightening machines used simple sensors and hardware; however, with the increased complexity of fabric types and demand for faster and more accurate machines, the whole industry started to switch to smart camera systems. Three basic methods, which are based on FGT constellation, polar transformation, and statistical features, are proposed and their performances are evaluated. As an improvement to statistical based method, we introduce a neural network based approach to choose optimum weights for the statistical features. Moreover, a comparison between FU-SmartCam and a commercial one called Tattile Smart Camera is given. Experimental results show that the introduced algorithms provide satisfactory performance, and can be used in real time systems.

Weft-straightening operation is a well-known problem in the textile industry. After the fabric is washed, before it goes to the drying process, weft-straightening must be done. Namely, rotation and deformations in the fabric must be measured and corrective action must be taken. In principle, this can be done by a human operator at relatively low speeds. An experienced operator can both detect the rotation and/or deformation in the fabric with naked eye and take corrective action by sending the proper commands to the motor drivers. Primitive weft-straightening machines used relatively simpler optical sensors and hardware. That is, instead of using CCD cameras and embedded systems to analyze the captured images in real-time, earlier systems and their sensors were based on interference and other optical/physical properties of light. Seiren Electronics' DENSIMATIC is one of such example (Seiren Electronics). However, speed and accuracy can be improved considerably by using machine vision systems. With the increased complexity of fabric types and demand for faster and more accurate processing, use of advanced machine vision algorithms with CCD cameras and embedded systems started to appear in commercial products. ElStraight manufactured by Erhardt + Leimer Company (Erhardt + Leimer) is a well-known example that uses four cameras (Tattile Smart Cameras) for weft-straightening as shown in Fig. 1.



Fig. 1. Weft-straightening machine from by Erhardt+Leimer company (Erhardt+Leimer).

There are several known pattern recognition algorithms to identify different patterns in an image under the presence of translation and rotation. Some of the relevant research papers are (Tuceryan and Jain, 1998), (Loh and Zisserman, 2005), (Josso et al., 2005), and (Araiza et al., 2006), and the references therein. However, in the weft-straightening problem, we have a known texture which is subject to translation and rotation, and the problem is to estimate the rotation angle only. In a typical industrial setup, the width of a fabric equals to a couple of meters, and there are four to six equally spaced sensors, each measuring the local rotation angle. By interpolating these rotation angle measurements, it is possible to estimate mild deformations and curvatures in the fabric. Basically, we have a known 2-D periodic or almost periodic signals if the textile irregularities are taken into account. The problem is to estimate the rotation angle from discretized and windowed version of the rotated texture under the presence of camera noise and quantization errors.

Rotation angle estimation is not a new subject in computer vision. LI et al. proposed a robust rotation angles estimation algorithm from image sequences using annealing m-estimator (Li et al, 1998). They call the method robust since the proposed method can deal with the outliers. Their aim of proposing a rotation angle estimation algorithm was to solve the motion estimation problem. In (Kim Yul and Kim Sung, 1999), another method based on Zernike moments is proposed to estimate rotation angles of the circular symmetric patterns. Since circular symmetric objects have similar eigenvalues in both directions, the principal axes cannot be used for rotation angle estimation. Therefore, they introduce a robust method which uses the phase information of Zernike moments. Recently, a rotation angle estimation algorithm based on wavelet analysis is proposed for textures (Lefebvre et al., 2011). The key point is to find the rotation angle that best concentrates the energy in a given direction of a wavelet decomposition of the original image.

A typical texture image and its 10 degree rotated version are given in Fig. 2. In Fig. 3, a hypothetical deformation of the fabric is shown, which is exaggerated for a better illustration. Measurement of four local rotation angles can be interpolated to estimate the actual deformation.

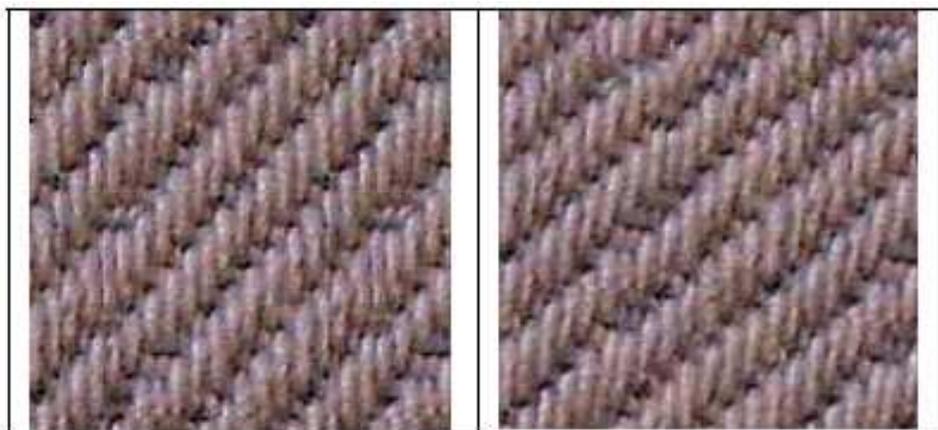


Fig. 2. A typical fabric image (Left) and its rotated and translated version (Right) are shown.

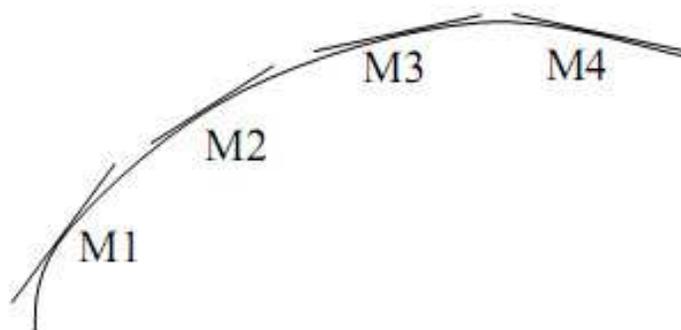


Fig. 3. Measurement of four local rotation angles is used to estimate the fabric deformation.

In real applications, deformations are much smaller than the exaggerated deformation curve shown above.

In this study, in order to solve the rotation angle estimation problem, three algorithms, which are based on “FGT Constellation”, “Polar Transformation”, and “Statistical Parameters” are proposed. In addition, neural networks based approach is used to choose optimum weights for statistical parameters. All of these methods are dedicated to solve weft-strengthening problem in the textile industry.

In Section 2, the proposed methods, FGT constellation, Polar Transformation, and Statistical Parameters and its extension to neural networks are discussed. Their performance analysis is given in Section 3. Finally, some concluding remarks are made in Section 4.

2. Rotation angle estimation algorithms

2.1 Polar transform approach

Polar transformation approach is related to the computation of the autocorrelation, $R_t(x, y)$, of the texture, $t(x, y)$. For an $M \times N$ image, the autocorrelation function is also an image and can be written as;

$$R_t(x, y) = \sum_{i=1}^M \sum_{j=1}^N t(i, j) t(i+x, i+y) \quad (1)$$

where i and j are the dummy variables for summation. This eliminates translation dependence. Rotation and translation of the texture, $t(x,y)$ only rotates $R_t(x,y)$. To estimate pure rotation in $R_t(x,y)$, one can look at its polar representation, $R_{t,polar}(r,\theta)$. It is easy to see that pure rotation around the origin in the Cartesian space corresponds to translation in the θ direction in the polar space. Therefore, now the problem is reduced to the estimation of the shift in the y direction of $R_{t,polar}(r,\theta)$. A simple correlation analysis can be used to find the value of d for which the correlation between $R_{t,polar}(x,y+d)$ and $R_{t-rotated,polar}(x,y)$ is maximum:

$$\arg \max_d \langle R_{t,polar}(x,y+d), R_{t-rotated,polar}(x,y) \rangle \quad (2)$$

This requires a search over d , and for each d value, the computation of the inner product requires $O(n^2)$ floating point operations. As an alternative, it can be considered taking 2D Fourier Transform of $R_{t,polar}(x,y)$, which converts translation in the second coordinate to a linear phase shift in the second coordinate. A simple graphical approach can be used to estimate the proportional constant in the linear phase shift, and hence estimate the rotation angle.

Preliminary tests indicate that both variations of this approach are computationally demanding, but give accurate angle estimates. For more information about the polar transform approach based rotation angle estimation one can look at (Sumeyra, 2007).

2.2 FGT Constellation approach

The FGT-Constellation approach also involves computation of the autocorrelation, $R_t(x,y)$, of the texture, $t(x,y)$. However, following this a thresholding is done, and a “constellation” like image is obtained. Basically, peaks of $R_t(x,y)$, will appear as bright spots in the thresholded image. If the texture is rotated, as shown in Fig. 4, one can see that the bright points also rotate in the same way and the same amount as shown in Fig. 5. Then the problem turns into finding the brightest point position on the thresholded image by searching in the first quadrant of the coordinate axis (see Fig. 6). An illustrative video which shows the operation of this algorithm can be found in the following link; www.fatih.edu.tr/~culas/rotationestimation/video1.avi.

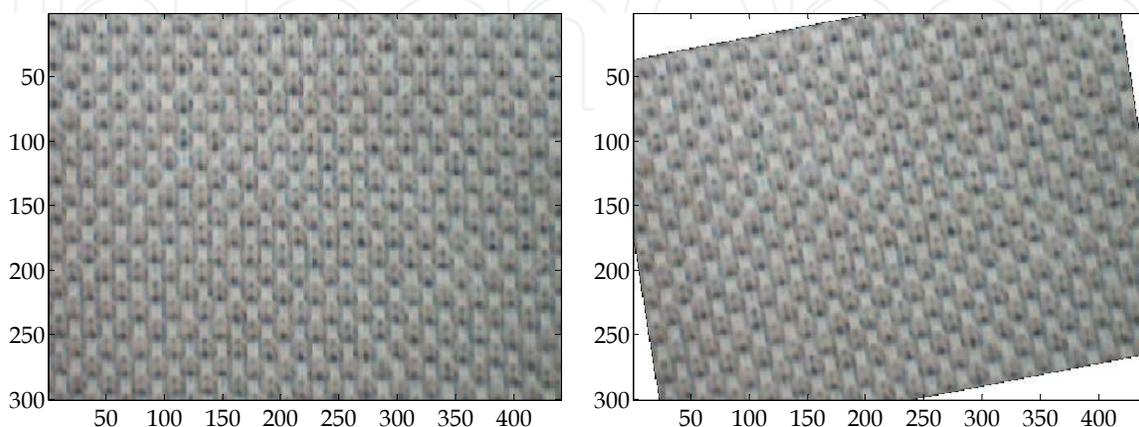


Fig. 4. Left: A picture of texture is having a size of 300 x 450 pixels. Right: The texture is rotated about 10 degrees in the counter clockwise direction.

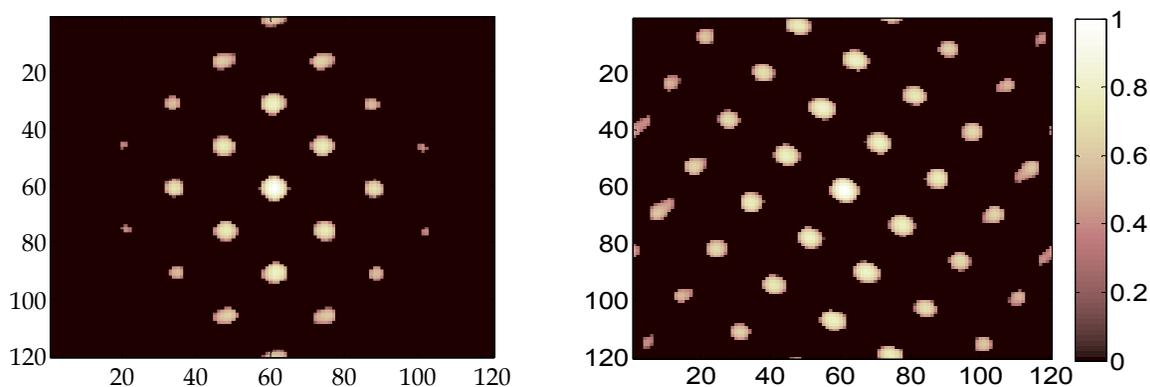


Fig. 5. Left: FGT constellation of the original image texture. Right: The FGT constellation of the rotated image. When the picture is rotated, the bright spots also rotate by the same amount and in the same direction; however, new small spots may appear in the constellation.

Also, preliminary tests on the .NET platform using the DirectX framework showed the feasibility of this approach both computational-wise, and performance-wise. This algorithm is later implemented on the FU-SmartCam, and for 64x64 image size, we were able to get a couple of estimates per second with about 1 degree or better accuracy. However, significant improvement can be achieved if larger images are used. To overcome the computationally demanding autocorrelation computation, which is done by floating point FFT, we tested 2-D autocorrelation computation via $GF(p)$ transformation (GT), where p is a large prime satisfying

$$p > 2N^2 2^{2m} \tag{3}$$

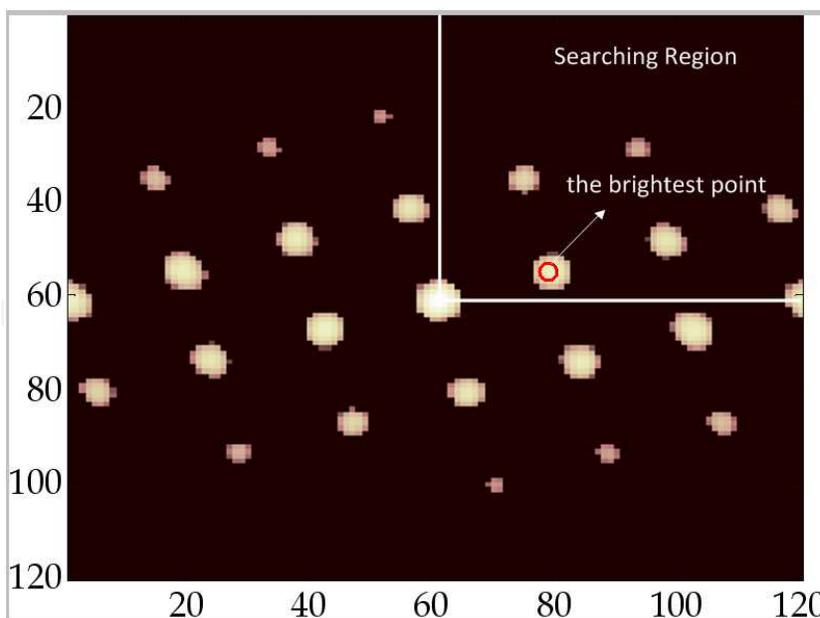


Fig. 6. Brightest point search region.

where N is the image size, and m is the number of bits used to store the intensity information. Note that, to be able to perform fast GT by using the well-known “divide and conquer” type approach of FFT, we need the prime to be of the form,

$$p = 2^{u+1}k + 1 \quad (4)$$

where $N = 2^u$ is the image size. We have successfully tested this approach in .NET platform using the DirectX framework: We have selected, $p = 1073741953$ and used $g = 10$ instead of $e^{i2\pi/128}$. Preliminary tests indicate significant speedup, because instead of floating point operations, only 64 bit integer operations are needed, and the structure of the code is very similar to FFT code structure. The theoretical FGT approach, in principle can also be applied for the computation of the autocorrelation in the Polar Transform method. However, there seems to be no simple and fast way of doing Cartesian to polar transformation without using computationally expensive floating point operations.

2.3 Extraction of statistical features

In order to avoid the computational difficulty of autocorrelation computation, and polar transformation, we use a completely different method based on the computation of several statistical features from the fabric. Parameters varying significantly with rotation are considered as suitable for the rotation estimation. In addition, the parameter changes should be preferably linear or almost linear with the rotation, and for real-time applications, they should be easy to compute. For this purpose, five statistical features are proposed for a texture image. These are given as;

- 1-D linear model
- 2-D linear model
- Means of the standard deviations parallel to the x - axis
- Means of the standard deviations parallel to the y - axis
- Means of the standard deviations along the diagonal axes.

These features affect the overall system performance. In performance evaluation tests, we show the benefit of using large number of statistical parameters instead of using small set of features. Statistical parameters based approach is computationally more attractive. The reason is that a look up table is generated based on the reference fabric and this is stored in the memory. Then for a rotated image, these statistical features are computed, and then using the nearest neighborhood method or weighted 1-norm distance metric, best matching rotation angle is estimated.

Computation of the statistical features is explained starting from following subsection. Then two scenarios for rotation estimation based on statistical features are analyzed,

- By using only 2-D model parameters,
- By using all features.

2.3.1 2-D modeling

We model each pixel value with the following equation.

$$\hat{t}(x, y) = \alpha t(x, y - d_y) + \beta t(x - d_x, y) \quad (5)$$

where α and β are the model parameters. Apart from these parameters, there are also two variables which are d_x and d_y . These variables, d_x and d_y , correspond to shifts/periods in x and y directions respectively. These parameters can be determined by using a trial and

error depending on the fabric type. Hence, there are a total of 4 parameters. One possible method to determine α and β given the values of d_x and d_y is the minimization of the following cost function.

$$f = \sum_{x=1}^M \sum_{y=1}^N [-\hat{t}(x,y) + \alpha t(x,y-d_y) + \beta t(x-d_x,y)]^2 \tag{6}$$

To minimize this cost function, derivatives with respect to α and β are computed, and set to zero;

$$\frac{\partial f}{\partial \alpha} = 0 \quad \frac{\partial f}{\partial \beta} = 0 \tag{7}$$

After the solving these two equations, one can get α and β as,

$$\alpha = \frac{A_{22}B_1 - A_{12}B_2}{A_{11}A_{22} - A_{21}A_{12}}, \quad \beta = \frac{A_{12}B_1 - A_{11}B_2}{A_{12}A_{21} - A_{11}A_{22}}, \tag{8}$$

where A_{11} , A_{12} , A_{21} , A_{22} , B_1 , and B_2 and are:

$$A_{11} = \sum_{x=1}^M \sum_{y=1}^N t^2(x-d_x,y), \quad A_{12} = \sum_{x=1}^M \sum_{y=1}^N t(x-d_x,y) \cdot t(x,y-d_y), \tag{9}$$

$$A_{21} = \sum_{x=1}^M \sum_{y=1}^N t^2(x-d_x,y), \quad A_{22} = \sum_{x=1}^M \sum_{y=1}^N t(x-d_x,y) \times t(x,y-d_y), \tag{10}$$

$$B_1 = \sum_{x=1}^M \sum_{y=1}^N t(x,y-d_y) \cdot t(x,y) \quad B_2 = \sum_{x=1}^M \sum_{y=1}^N t(x,y) \times t(x-d_x,y). \tag{11}$$

Variation of the model parameters with respect to rotation angle is shown in Fig. 7. The texture image is rotated from -30 to +30 with 0.5 degree steps and its corresponding 2-D model parameters are computed. We observe that these parameter variations are almost linear when the distance values are $d_x = 2$ and $d_y = 1$.

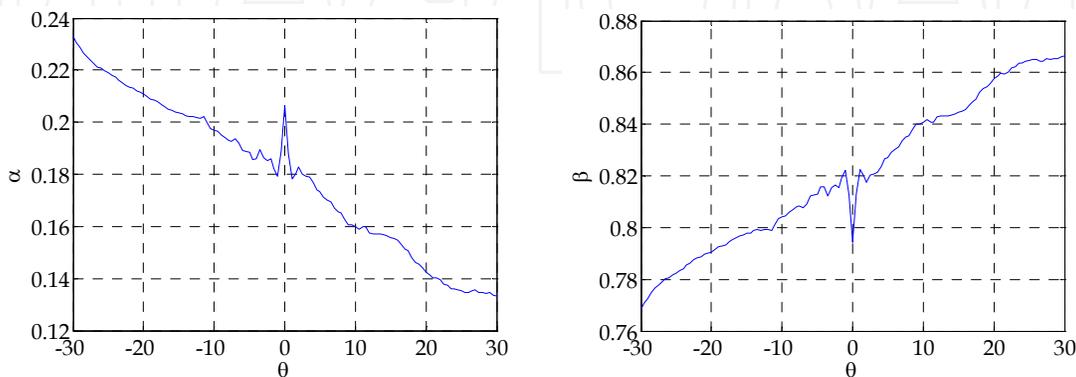


Fig. 7. 2-D model parameters versus rotation angle. The rotation angle is incremented from -30 to +30 with 0.5 degree steps.

2.3.2 1-D modeling

The 1-D model parameter approximation is similar to 2-D model parameter approximation. The following 1-D equation can be used for this type of modeling,

$$\hat{t}(x, y) = \gamma t(x - d_x, y - d_y) \quad (12)$$

In this case, γ has the following equality,

$$\gamma = \frac{\sum_{x=1}^M \sum_{y=1}^N t(x - d_x, y - d_y) t(x, y)}{\sum_{x=1}^M \sum_{y=1}^N t(x - d_x, y - d_y) t(x - d_x, y)} \quad (13)$$

Variation of the 1-D model parameter, γ , with respect to rotation angle is shown in Fig. 8. The texture image is rotated from -30 degrees to +30 degrees with 0.5 degree steps and its corresponding 2-D model parameter is plotted versus the rotation angle. As it is seen, variation is not linear for the distance values $d_x = 2$ and $d_y = 1$ and this is a high undesirable feature.

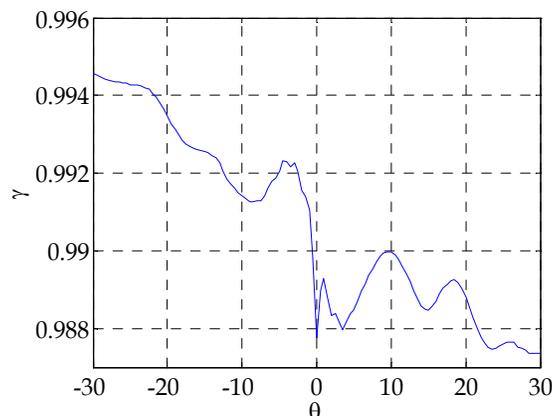


Fig. 8. 1-D model parameter, γ variation versus rotation angle. The rotation angle is incremented from -30 degrees to +30 degrees with 0.5 degree steps.

2.3.3 Mean of the standard deviations along the X-Axis

The mean of the standard deviations along the x axis can be expressed as follows;

$$I = \frac{\sum_{i=1}^M \sigma_{x_i}}{M} \quad \Phi_x = \frac{I}{\Psi} \quad (14)$$

where I is the mean of standard deviation along the x axis of the texture image. We divide I to Ψ , which is the mean of the gray level image pixels, in order to eliminate the ambient illumination effects of the environment. σ_{x_i} is the standard deviations of the i^{th} row, and M , represents the width of the image. The variation of the mean of standard deviations along the x-axis versus rotation is shown in Fig. 9.

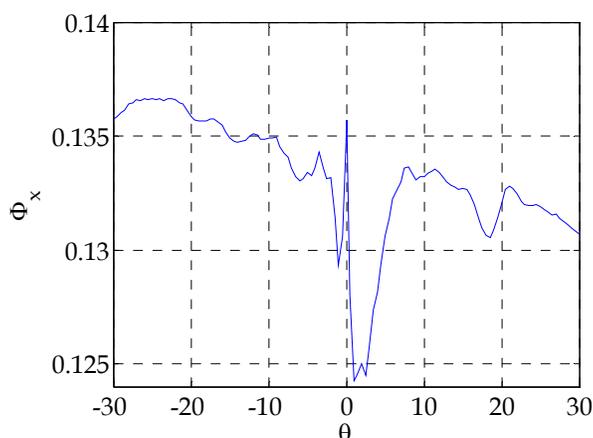


Fig. 9. Mean of standard deviations along x-axis versus rotation angle.

2.3.4 Mean of the standard deviations along Y axis

Similarly, the mean of the standard deviations along the y axis can be expressed as follows;

$$J = \frac{\sum_{i=1}^N \sigma_{y_i}}{M} \quad \Phi_y = \frac{J}{\Psi} \quad (15)$$

J is the mean of standard deviations along the y-axis of the texture image. Similarly, it is divided to mean intensity level of the image to find Φ_y . The standard deviation of the i^{th} column is denoted by σ_{y_i} , and N represents the height of the image. The variation of Φ_y with the rotation angle is shown in Fig. 10.

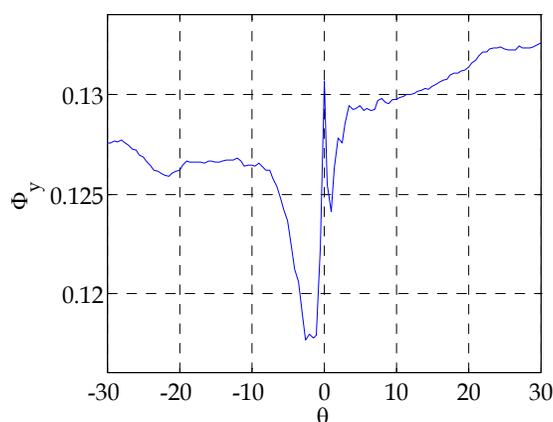


Fig. 10. Mean of standard deviations along y-axis.

2.3.5 Statistical feature based on mean of the standard deviations along diagonal axes

$$K_1 = \frac{\sum_{i=1}^D \sigma_{d_{ii}}}{D} \quad K_2 = \frac{\sum_{i=1}^D \sigma_{d_{i(D-i)}}}{D} \quad \Phi_d = \frac{K_1 + K_2}{2\Psi} \quad (16)$$

K_1 and K_2 are the means of standard deviations on the diagonal and off-diagonal axes of the texture image. D stands for the number of the diagonal elements. Actually, for all study, M , N , and D are the same size since we work with the square images. Fig. 11 shows the variation of this new parameter with respect to the rotation angle.

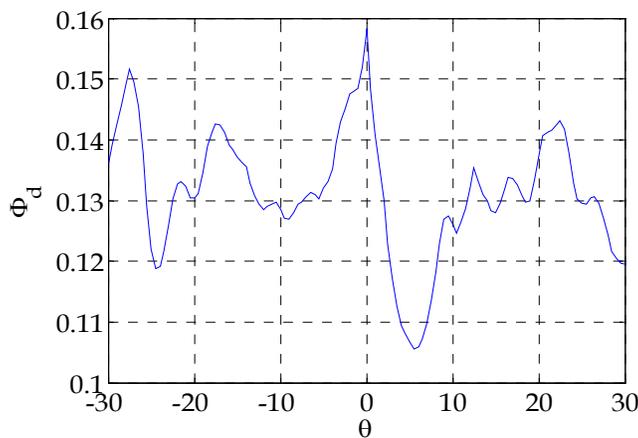


Fig. 11. Mean of standard deviations along diagonal axes.

2.4 Rotation angle estimation by using statistical features and model parameters

The main problem in rotation angle estimation is to find the useful statistical parameters which change significantly and linearly with rotation. To be able to find the best candidates, variations of each parameter are drawn as in the previous section and looked for the linear and important changes. Then, we decide if these parameters might be used or not. After determining the statistical features, a look up table is generated by rotating the reference image in a range and its statistical parameters are stored. In the estimation process, for the rotated image, the statistical features are computed, and the parameters are searched through the look up table by using a nearest neighborhood search (NNS) method. The closest parameter combination is accepted as rotation estimation. However, this idea best works by assuming that the features change linearly with the rotation, and all of them have the same importance. In fact, this is not true for many cases. Because, neither the parameters change linearly nor have the same importance. For this reason, we append the artificial neural networks to the proposed method to overcome this problem.

Another issue is to use sufficient number of statistical features. To show the importance of the number of parameters, the experiments are divided into two parts. In the first section, only 2-D model parameter is chosen as feature parameters, which are the most effective ones. In the second section, all statistical parameters are exploited to show the improvements.

2.4.1 Using only 2D model parameters

In this subsection, the 2-D model parameter is used as the statistical feature. A look-up table is generated by rotating the reference image in a desired region and calculating the 2D model parameters for each rotation.

$$\Gamma(\theta_i) = [\alpha_i \ \beta_i] \quad 1 \leq i \leq M \quad (17)$$

where, θ , denotes the amount of rotation, and i is the index of each rotation starting from the first to last rotation, M . After the look-up table is built, the system performance is tested in the same region with higher resolution. For example, one can generate the look-up table in the region of -30 degrees to +30 degrees with 0.5 degree steps, and then the method is tested in the same region with 0.1 degree steps.

Another important point is the problem of choosing distance parameters, d_x and d_y . These parameters have to be chosen properly because they significantly affect the linearity of the variations. To decide which distance values are suitable for the reference texture image, we followed two ways. The first one is to draw the parameter-rotation graph for each, d_x and d_y combinations and look at the linearity and amount of change. The second and more professional one is to calculate the sum of least square errors between actual and estimated rotations for d_x and d_y combinations and accept the combinations which give the smallest error.

After we decide the distance parameters, d_x and d_y , the measured model parameters, α and β , are searched through the look-up table and the closest variations are used for rotation estimation. In general case, if we have foreknowledge about the weights of the parameters, we can use the weighted nearest neighborhood search as,

$$e = \sum_{i=1}^N w^T \{\Gamma(\theta_i) = [\alpha_i \beta_i]\} \quad (18)$$

where $\Gamma(\theta_i)$ is the i^{th} row of the look-up table if the parameters are put on the columns. The weights are represented as w which emphasizes some statistical parameters over others. In this section, w is chosen as unity vector.

2.4.2 Using all statistical parameters

To get better results in rotation estimation algorithm, all statistical features explained in Section 2.3 are used. The idea is very similar to the two model rotation estimation algorithm; the first difference is that we generate the look-up table with all these statistical parameters. The other difference is the computational cost. In this method, the processing and searching time increases due to the number of the parameters that are used. The look-up table has the following structure;

$$\Gamma(\theta_i) = [\alpha_i \beta_i \gamma_i \Phi_{x_i} \Phi_{y_i} \Phi_{d_i}] \quad 1 \leq i \leq M \quad (19)$$

2.4.3 Drawbacks of nearest neighborhood search (NNS)

The NNS is used to solve one norm distance problem. However, the NNS algorithm may fail if the parameters do not have the equal weights and linear changes with the rotation. We observe that the statistical parameters, explained in Section 2.3, are neither exactly linear nor have same importance. The second problem with the NNS is the searching time. If the look-up table is generated with high resolution, the table becomes so large and takes long time to estimate the amount of rotation. The method can be accelerated from $O(N)$ to $O(\log N)$ by using kd-tree space quantization methods (Bentley, 1980). Another and better solution is to use artificial neural networks in order to find the weighting factor and speed up the

algorithm significantly. In neural network based solution, the method becomes much faster since it does not need any searching method, and it is sufficient to train look up table.

2.5 Neural network improvement to statistical parameters based approach

Statistical parameters do not have equal importance on estimating the rotation angle; therefore, artificial neural networks are used to choose optimum weights for these parameters. To do this, Fletcher-Reeves updates etc., as shown in Fig. 12, we formed a global user interface in Matlab to test the performance of the various neural network training methods such as Levenberg-Marquardt (LM), BFGS quasi-Newton back propagation, conjugate gradient back propagation with Fletcher-Reeves updates etc.

Statistical features based method is computationally very attractive. Experimental results show quite affirmative results: An accuracy of less than 0.2 degree can easily be achieved with relatively little computational effort. In our tests, we observed that LM training method provides better performance over others.

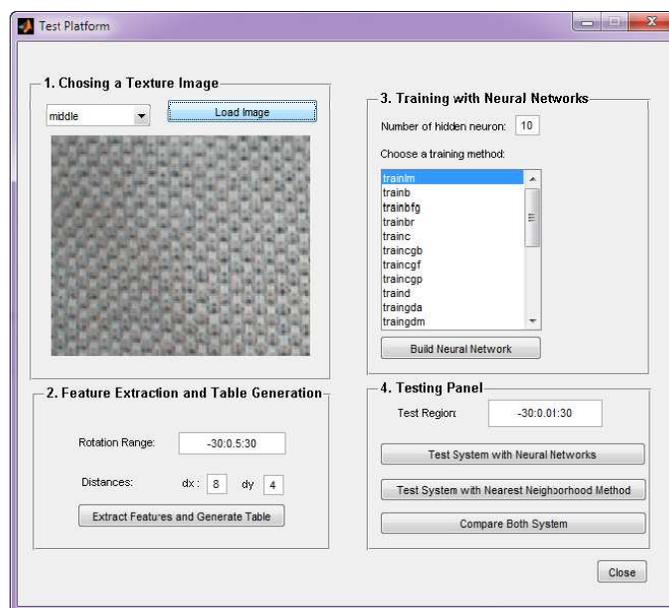


Fig. 12. A Test Platform formed in Matlab to test Neural Network with various training methods and compare the results with Nearest Neighborhood method.

3. Experimental results

3.1 Experimental setup

Tattile Smart Camera has a StrongARM SA-1100 processor based board, and Lattice iM4A3 programmable logic device. The StrongARM SA-1100 board is a low power embedded system running a Linux port. The architecture of the board is heavily based on the LART project done at Delft University (LART project). Tattile Smart Camera also has a progressive CCD sensor, and image acquisition is done by the high speed logic circuit implemented on the Lattice iM4A3 programmable logic device. In principle, it is possible to use a desktop PC, cross compile the code for StrongARM and then upload (ftp) it to the board via network connection. However, one still has to work with cross compilers, and simulators for prototyping experiments.

On the other hand, the FU-SmartCam embedded system which does not use that low power, but has more processing power, memory, and flash storage. It has Vortex86 processor (Vortex system) in it, and runs the Intel x86 port of Linux, i.e. the Linux port which runs on regular desktop PCs. Because of this, there is no need for cross compilers, and simulators. Complete prototyping experiments can be performed on the desktop PC, and the generated code will run without any modification on the target board, which is slow as the Vortex86 but not as powerful as a Pentium 4. The image acquisition was performed using a low cost interlaced scan camera with a Conexant Bt848 chip. The FU-SmartCam is shown in Fig. 13.

The FU-SmartCam has VGA, Keyboard, Ethernet, and RS-232 connections, is extremely flexible, and easily reconfigurable. Currently, we are also developing a relay board with a small 8-bit microcontroller interfaced to the Vortex over RS-232. This relay board will enable direct connection of the FU-SmartCam to several industrial equipment.



Fig. 13. FU-SmartCam : On the top, we have a low cost interlaced scan camera, in the middle we have the Vortex system, and in the bottom we have a dual output power supply. The Vortex system itself consists of two boards. The overall system is quite small.

3.2 FGT constellation performance analysis

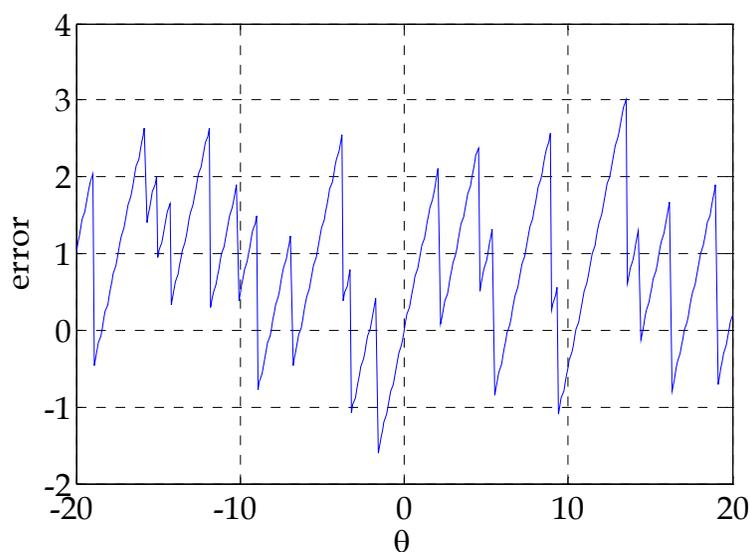


Fig. 14. Error versus rotation in the range of -20 degrees to 20 degrees with 0.1 degree steps.

In this part, the performance of the FGT constellation approach is investigated, as seen from the Fig. 14, the maximum absolute error is about 3 degree and the average of error is about 1 degree. Due to periodicity, although we expect to see algorithm to work in the range of -45 to +45, we observe that the method works fine in the range of -30 to +30 (see Fig. 15.) However, for weft-strengthening problem, this range can be acceptable for rotation angle estimation.

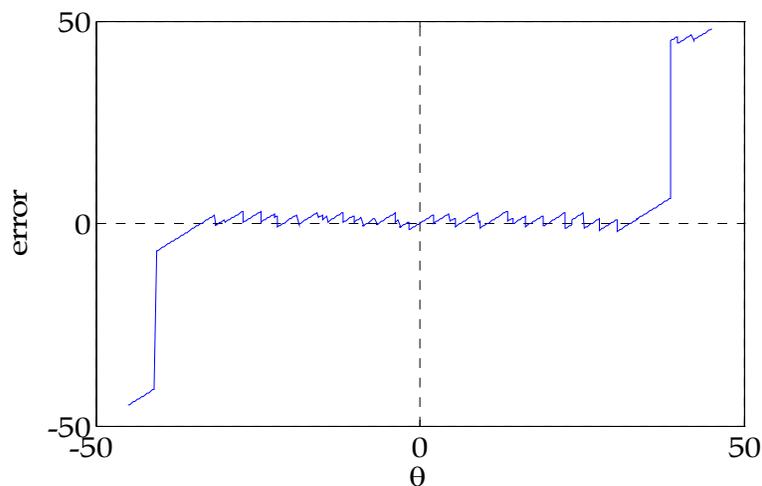


Fig. 15. The drawback of the FGT constellation based approach. The algorithm cannot estimate rotation angle if there is a 30 degrees rotation in both directions.

3.3 Using only 2-D model parameters

In order to test the performance of the statistical parameters based approach, we plot the errors with respect to rotations. The look-up table is generated in the region of -30 degrees to 30 degrees with 0.5 degree steps and it is tested in the same region with 0.1 degree resolution.

First of all, to show the effect of the distance parameters, the distance values are taken randomly as $d_x = 1$ and $d_y = 1$ and the error variations versus rotation is given in Fig.16. From this figure it can be seen that, the error becomes considerably high in some rotations and cannot be acceptable as acceptable.

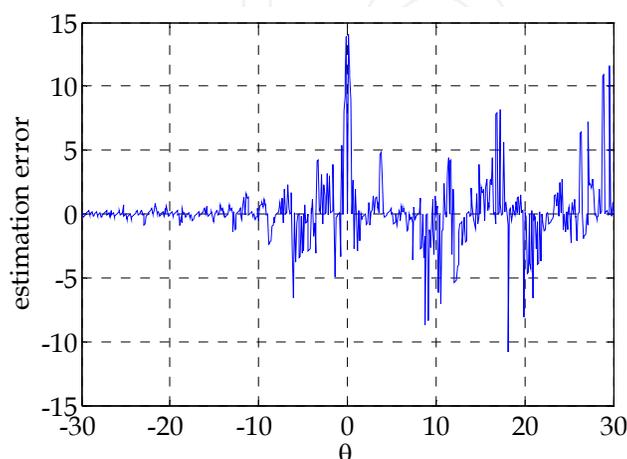


Fig. 16. The effect of the distance values. Randomly, they are taken as $d_x = 1$ and $d_y = 1$.

However, as explained in Section 2.4, if the proper distance parameters are chosen as $d_x = 9$ and $d_y = 8$, the error variation versus rotation is shown in Fig. 17, and can be considered as acceptable. The average absolute error of the estimation is less than 0.5 degree.

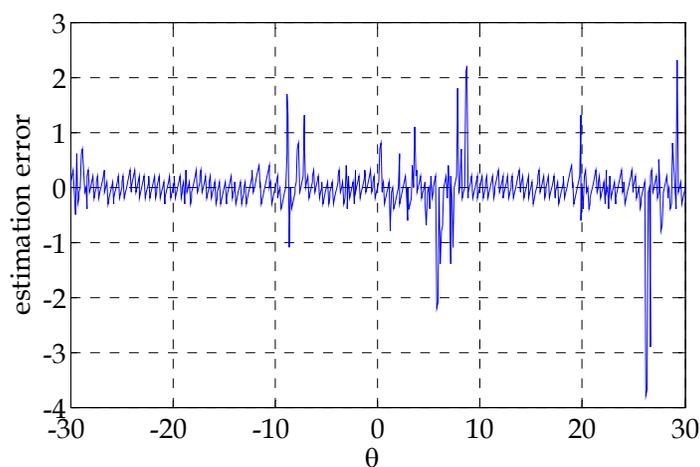


Fig. 17. The error variation versus rotation for the optimum distance, $d_x = 9$ and $d_y = 8$.

3.4 Using all statistical parameters with nearest neighborhood method

In this section, all statistical feature parameters are used for rotation angle estimation. The testing region is again from -30 degrees to +30 degrees with 0.1 degree steps. In this case the results are very attractive and the absolute error is about 0.2 degree. The distance parameter were chosen as $d_x = 9$ and $d_y = 8$. The variation of estimation error versus rotation angle is shown in Fig. 18.

3.5 Using all statistical parameters with artificial neural networks

To compare the results of neural networks and nearest neighborhood based methods, we used the same texture image and same optimum distance values in the same testing region. In Fig. 19, the error variation for only 2-D model parameters is shown.

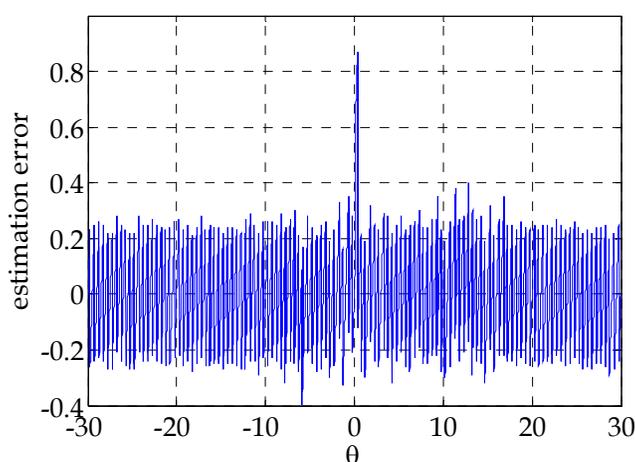


Fig. 18. Rotation estimation error versus rotation angle. The estimation error is about 0.2 degree.

In order to show the power of using all statistical parameters with neural networks, we increased the testing resolution from 0.1 degree to 0.01 degree and plot the result in Fig. 20. The estimation error is decreased up to 0.1 degree, which was about 0.2 degree in nearest neighborhood method. Therefore, the performance is increased almost two times. If we compare the computation time, neural network is much faster than nearest neighborhood based method.

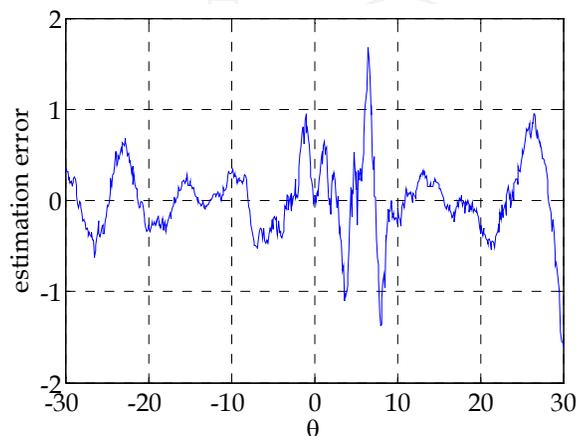


Fig. 19. Rotation estimation error variation versus rotation angle. Here only 2-D model parameters are trained in neural networks.

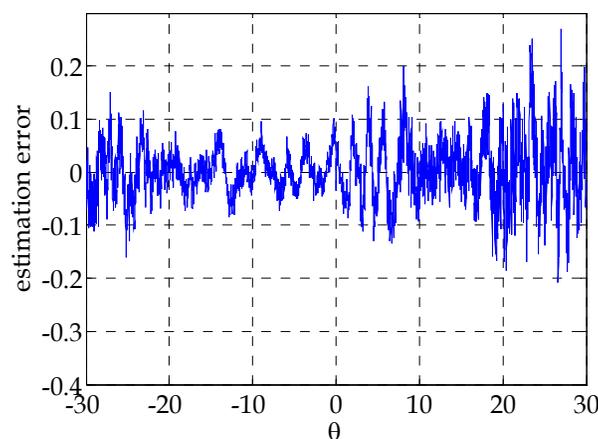


Fig. 20. Rotation estimation error variation versus rotation angle. Here only all model parameters are trained in neural networks. The testing resolution is increased to 0.01 degree to show robustness of the algorithm.

4. Conclusion

In this chapter, the weft-straightening problem encountered in the textile industry is described. As a solution of rotation angle estimation which is the fundamental part of the weft-straightening problem, three different algorithms are introduced. The first algorithm is based on the Polar Transform which is applied to auto-correlated images; therefore, the translation in the θ direction gives the rotation angle. The second one is based on FGT constellation approach, and it depends on the autocorrelation of the thresholded image. FGT constellation consists of regularly distributed bright spots and the rotation angle is estimated by finding the brightest point in the first quadrant of the coordinate axis. The third algorithm is based on

statistical parameters. These parameters, firstly, are computed for the reference image and a look up table is generated for its artificially rotated images. The statistical parameters of the input image are searched in the look up table and the closest one is found as rotation angle. Finally, in order to improve the statistical parameters based approach, neural networks are used to choose optimum weight factors since not all parameters have the same importance. Various neural network training methods are tested to find the best performance. The results show that the proposed methods can be successfully implemented in the real-time systems.

5. Acknowledgements

This work is supported by the Scientific Research Fund of Fatih University under the project number P50061001_2.

6. References

- Araiza, R., M. G. Averill, G. R. Keller, S. A. Starks, and C. Bajaj, "3-D Image Registration Using Fast Fourier Transform, with Potential Applications to Geoinformatics and Bioinformatics," Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU'06, Paris, France, July 2-7, 2006, pp. 817-824.
- Bentley, L. Multidimensional divide and conquer, Communications of the ACM, vol. 22, no. 4, pp. 214-229, 1980.
- Josso, B., D. R. Burton, M. J. Lalor, "Texture orientation and anisotropy calculation by Fourier transform and Principal Component Analysis", Mechanical Systems and Signal Processing 19 (2005) 1152-1161.
- Kim, W.Y., Kim, Y.S. (1999). Robust Rotation Angle Estimator, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 8, pp. 768-773.
- Lefebvre, A. Corpetti, T. Hubert-Moy, L. (2011). Estimation of the orientation of textured patterns via wavelet analysis. *Pattern Recognition Letters*, Vol. 32, pp. 190-196.
- Li, B.; Xu, Y. & Choi, J. (1996). Applying Machine Learning Techniques, *Proceedings of ASME 2010 4th International Conference on Energy Sustainability*, pp. 14-17, ISBN 842-6508-23-3, Phoenix, Arizona, USA, May 17-22, 2010
- Li, S. Z., Wang, H., and Soh, W.Y.C. (1998). Robust Estimation of Rotation Angles from Image Sequences Using the Annealing M-Estimator, *Journal of Mathematical Imaging and Vision* Vol. 8, pp. 181-192.
- Loh, A. M., A. Zisserman, "Estimating the affine transformation between textures," Proceedings of the Digital Imaging Computing: Techniques and Applications (DICTA 2005).
- Tuceryan, M., and A. K. Jain (1998), "Texture Analysis," The Handbook of Pattern Recognition and Computer Vision 2nd Edition), by C. H. Chen, L. F. Pau, P. S. P. Wang (eds.), pp. 207-248, World Scientific Publishing Co.
- Ulas, C., Demir, S., Toker, O., Fidanboylyu, K. (2007). Rotation Angle Estimation Algorithms for Textures and Their Real-Time Implementation on the FU-SmartCam, *Proceedings of the 5th International Symposium on image and Signal Processing and Analysis*, pp. 469-475, Istanbul, Turkey.
- LART project page at Delft University

<http://www.lartmaker.nl/>

Vortex system

http://www.icop.com.tw/products_detail.asp?ProductID=119

Seiren Electronics' DENSIMATIC-SVW II

<http://textileinfo.com/en/it/seiren-e/p01-01.html>

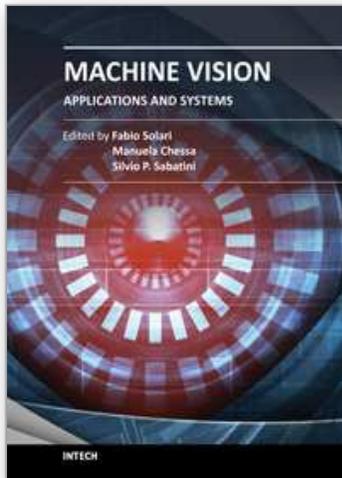
Erhart+Leimer's ElStraight

http://www.erhardt-leimer.it/prodotti/raddrizzatrama_eng.html

Tattile Smart Camera

<http://www.tattile.com/new.site/start.htm>

IntechOpen



Machine Vision - Applications and Systems

Edited by Dr. Fabio Solari

ISBN 978-953-51-0373-8

Hard cover, 272 pages

Publisher InTech

Published online 23, March, 2012

Published in print edition March, 2012

Vision plays a fundamental role for living beings by allowing them to interact with the environment in an effective and efficient way. The ultimate goal of Machine Vision is to endow artificial systems with adequate capabilities to cope with not a priori predetermined situations. To this end, we have to take into account the computing constraints of the hosting architectures and the specifications of the tasks to be accomplished, to continuously adapt and optimize the visual processing techniques. Nevertheless, by exploiting the low-cost computational power of off-the-shelf computing devices, Machine Vision is not limited any more to industrial environments, where situations and tasks are simplified and very specific, but it is now pervasive to support system solutions of everyday life problems.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Cihan Ulas, Onur Toker and Kemal Fidanboyly (2012). Rotation Angle Estimation Algorithms for Textures and Their Implementations on Real Time Systems, Machine Vision - Applications and Systems, Dr. Fabio Solari (Ed.), ISBN: 978-953-51-0373-8, InTech, Available from: <http://www.intechopen.com/books/machine-vision-applications-and-systems/rotation-angle-estimation-algorithms-for-textures-and-their-real-time-implementation->

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen