

Algorithms for Hardware-Based Pattern Recognition

Volker Lohweg

*Koenig & Bauer AG (KBA), Bielefeld, Westring 31, 33818 Leopoldshöhe, Germany
Email: vlohweg@kba-bielefeld.de*

Carsten Diederichs

*Koenig & Bauer AG (KBA), Bielefeld, Westring 31, 33818 Leopoldshöhe, Germany
Email: cdiederichs@kba-bielefeld.de*

Dietmar Müller

*Circuit and System Design Group, Technical University of Chemnitz, 09107 Chemnitz, Germany
Email: d.mueller@infotech.tu-chemnitz.de*

Received 27 August 2003; Revised 31 March 2004

Nonlinear spatial transforms and fuzzy pattern classification with unimodal potential functions are established in signal processing. They have proved to be excellent tools in feature extraction and classification. In this paper, we will present a hardware-accelerated image processing and classification system which is implemented on one field-programmable gate array (FPGA). Nonlinear discrete circular transforms generate a feature vector. The features are analyzed by a fuzzy classifier. This principle can be used for feature extraction, pattern recognition, and classification tasks. Implementation in radix-2 structures is possible, allowing fast calculations with a computational complexity of $O(N)$ up to $O(N \cdot \text{ld}(N))$. Furthermore, the pattern separability properties of these transforms are better than those achieved with the well-known method based on the power spectrum of the Fourier Transform, or on several other transforms. Using different signal flow structures, the transforms can be adapted to different image and signal processing applications.

Keywords and phrases: image processing, nonlinear circular transforms, feature extraction, fuzzy pattern recognition.

1. INTRODUCTION

Image retrieval, texture analysis and optical character recognition, and general inspection tasks are of main interest in the field of image processing and pattern recognition. Methods which operate automatically are of interest in the above-mentioned areas. Automation is important if the amount of data is too large to be handled manually or if the speed of the image presentation is too fast for the human inspector. Reitboeck and Brody [1] were among the first who used translation-invariant transforms for character recognition. Wagh and Kanetkar [2] presented a general class of nonlinear translation-invariant transforms, which were called circular transforms (CTs). Burkhardt et al. [3] proposed a recursive definition of the class CT, which can be used for a simple mathematical description of the transform. The well-known R(apid) and B(inary) transforms are members of the above-mentioned class of transforms. The separability properties of nonlinear transforms are generally speaking incomplete. Therefore, it is obvious to use group-theory-based methods to improve the separability properties [3, 4, 5].

For various practical image processing and pattern recognition cases in an industrial environment, it is incidental that different processes and signal distortions will occur. One prominent process factor can be described, in general, as relative movements between an object and a camera system. It is not relevant whether the object moves in front of a camera or vice versa. In any case, a feature vector can be generated by means of invariant transforms. In some applications, the process movement can be assumed as translation invariant [1, 6]. Applications like printed product pattern recognition which will be presented in this paper can be assumed as translation invariant as well. Therefore, a special class of nonlinear translation-invariant transforms proves to be appropriate for feature generation. As mentioned, it is obvious that further different distortions can also occur which in turn cannot be compensated. Accordingly, the feature vector has to be stabilized and correctly classified without exact knowledge of different stochastic processes [7]. Bocklisch and Priber [8] proposed a parametric fuzzy pattern classification (FPC) concept which was first applied for complex linear and nonlinear control systems. Also Eichhorn [9] and

others applied and modified the classification concept for various pattern recognition and classification systems. One advantage of the concept is the fact that a learning procedure is inherently given by the parametric model. Therefore, a new simplified classifier model will be presented in this paper which is well suited for industrial applications.

In this paper, we propose a combined method for pattern recognition and classification relying on a class of discrete nonlinear translation-invariant CTs [6, 10, 11, 12] and a modified FPC scheme which is based on Bocklisch and Priber [8] as well as on Eichhorn [9]. The algorithms are implemented in one field programmable gate array (FPGA), which operates at 40 MHz.

The organization of the rest of this paper is as follows. In Section 2, the properties of nonlinear CTs including a new concept on fast transforms are described, along with a modified fuzzy pattern classifier (MFPC). Section 3 provides a short survey on the features of the used FPGA, the implementation concept, and timing properties of the algorithms. Section 4 describes various experimental results regarding the separability properties of different CTs in the case of binary patterns. Section 5 discusses two possible applications, and the conclusion is presented in Section 6.

2. PROPERTIES OF NONLINEAR CIRCULAR TRANSFORMS

We now describe some properties of one-(1D) and two-dimensional (2D) discrete nonlinear CTs and FPC. In the remainder of the paper, the transforms are systematically assumed to be discrete so that their discrete nature will not be explicitly mentioned anymore.

2.1. Generalized nonlinear circular transform

Generalized nonlinear circular transforms (GNCTs) have some properties which are useful for the analysis of transient and periodic signals. The basic row vectors of the transform matrix are periodic and can have local support. This property indicates that on one hand, the basic row vectors behave like wavelets. On the other hand, the periodic row vectors structure is well suited for the periodic signal analysis. This leads to the fact that wavelet and periodic transform concepts have to be taken into account. It is well known that generally speaking, wavelets are translation *variant* if they are not redundant, but most of the power spectra of periodic transforms are translation *invariant*. Therefore, the concept of frames and biorthogonal vector bases have to be used for the CTs.

In this section, we sum up the major properties of the generalized circular transforms (GCTs). For details regarding the *generalized characteristic* and *generalized circular* matrices, we refer to other publications by the authors [6, 10, 11, 12]. Different transforms can be designed from a generalized version [12]. All transforms have in common that they use an amplitude spectrum \mathbf{G} with $\text{ld}(N) + 1$ coefficients in the 1D case and $(\text{ld}(N) + 1)^2$ coefficients in the 2D case. The coefficients are ordered in period groups similar to the power spectrum of Walsh Hadamard transform (WHT) [13, 14].

Instead of the power spectrum of the WHT, we use an absolute value determination to obtain a translation-invariant spectrum. This spectrum is much easier to implement in FPGAs than power spectra based on quadratic functions. An interesting fact is that other transforms offer this property as well [14, 15], but this fact was to our knowledge not yet explicitly referred to in the literature.

Let $\mathbf{x}^T = (x_0, x_1, \dots, x_{N-1})$, $\mathbf{x} \in \mathbb{R}^N$, be an input vector and $\mathbf{X}^T = (X_0, X_1, \dots, X_{N-1})$, $\mathbf{X} \in \mathbb{R}^N$, its transformed output vector. By \mathbf{A}_N and \mathbf{B}_N , we denote the CT matrix and its inverse, respectively. \mathbf{A}_N and \mathbf{B}_N are quadratic ($N \times N$) matrices. \mathbf{I}_N is the unity matrix and $\mathbf{diag}(\cdot, \cdot)$ defines a diagonal matrix of two submatrices:

$$\begin{aligned} \mathbf{X} &= \mathbf{A}_N \cdot \mathbf{x}, & \mathbf{x} &= \frac{1}{N} \cdot \mathbf{B}_N^T \cdot \mathbf{X}, \\ \mathbf{A}_N \cdot \mathbf{B}_N^T &= \mathbf{B}_N^T \cdot \mathbf{A}_N = \mathbf{A}_N^T \cdot \mathbf{B}_N = \mathbf{B}_N \cdot \mathbf{A}_N^T = N \cdot \mathbf{I}_N. \end{aligned} \quad (1)$$

Given a (2×2) -Hadamard matrix $\mathbf{K} = \begin{bmatrix} +1 & -1 \\ +1 & +1 \end{bmatrix}$, the transform matrices can be expressed and evaluated recursively as

$$\begin{aligned} \mathbf{A}_N &= \mathbf{diag}({}^f\mathbf{T}_{N/2}, \mathbf{A}_{N/2}) \cdot [\mathbf{K} \otimes \mathbf{I}_{N/2}], \\ \mathbf{B}_N &= \mathbf{diag}({}^r\mathbf{T}_{N/2}, \mathbf{A}_{N/2}) \cdot [\mathbf{K} \otimes \mathbf{I}_{N/2}]. \end{aligned} \quad (2)$$

The generalized characteristic matrices ${}^f\mathbf{T}_{N/2}$ and ${}^r\mathbf{T}_{N/2}$ are defined for the dimension $(N/2 \times N/2)$. Using different transform kernels ${}^f\mathbf{T}_{N/2}$ and ${}^r\mathbf{T}_{N/2}$, it is possible to assign various properties to the transforms. The spectral coefficients of all transforms \mathbf{A}_N and \mathbf{B}_N are grouped in the same way: the first $N/2$ spectral coefficients featuring a period N , followed by $N/4$ coefficients with period $N/2$. The last two coefficient vectors are the vectors with the shortest possible period 2 and the vector with the period 0.

2.1.1. Generalized characteristic matrices T

The coefficients of the matrix \mathbf{A}_N (or \mathbf{B}_N) are determined in such a way that the absolute value spectrum \mathbf{G} remains unchanged when the input vector \mathbf{x} undergoes a translation [12]. The transform matrix coefficients β_i are defined as real numbers. It has to be pointed out that complex numbers can be applied as well, but in this paper, only real numbers will be used. The definition of the generalized characteristic matrix is as follows:

$${}^f\mathbf{T}_{N/2} = \begin{bmatrix} -\beta_{N/2-1} & -\beta_{N/2-2} & \cdots & -\beta_0 \\ -\beta_0 & -\beta_{N/2-1} & \cdots & -\beta_1 \\ \vdots & \vdots & \ddots & \vdots \\ -\beta_{N/2-2} & -\beta_{N/2-3} & \cdots & -\beta_{N/2-1} \end{bmatrix}. \quad (3)$$

The coefficient matrix \mathbf{A}_N can be defined in a sparse matrix form:

$$\begin{aligned} \mathbf{A}_N &= \begin{bmatrix} {}^f\mathbf{T}_{N/2} & \cdots & 0 \\ & {}^f\mathbf{T}_{N/4} & \\ \vdots & & \ddots \\ 0 & \cdots & & 1 \end{bmatrix} \\ &\cdot \left[\prod_{i=1}^{\text{ld}(N)-1} \mathbf{diag}(\mathbf{I}_{N-2^i}, \mathbf{K} \otimes \mathbf{I}_{2^{i-1}}) \right] \cdot [\mathbf{K} \otimes \mathbf{I}_{N/2}]. \end{aligned} \quad (4)$$

The last two matrices represent the rationalized form of the modified Walsh Hadamard transform (MWHT), which was first introduced by Ahmed et al. [13, 14].

Equation (4) shows that it is possible to characterize the CTs with only one *characteristic coefficient vector*:

$$\mathbf{c}_\beta = (\beta_{N/2-1}, \beta_{N/2-2}, \dots, \beta_0, \beta_{3N/4-1}, \dots, \beta_{N/2}, \dots, \beta_{N-2}, \beta_{N-1})^T. \quad (5)$$

The following example shows the transform matrix for $N = 8$:

$$\mathbf{A}_8 = \begin{bmatrix} -\beta_3 & -\beta_2 & -\beta_1 & -\beta_0 & \beta_3 & \beta_2 & \beta_1 & \beta_0 \\ \beta_0 & -\beta_3 & -\beta_2 & -\beta_1 & -\beta_0 & \beta_3 & \beta_2 & \beta_1 \\ \beta_1 & \beta_0 & -\beta_3 & -\beta_2 & -\beta_1 & -\beta_0 & \beta_3 & \beta_2 \\ \beta_2 & \beta_1 & \beta_0 & -\beta_3 & -\beta_2 & -\beta_1 & -\beta_0 & \beta_3 \\ -\beta_5 & -\beta_4 & \beta_5 & \beta_4 & -\beta_5 & -\beta_4 & \beta_5 & \beta_4 \\ \beta_4 & -\beta_5 & -\beta_4 & \beta_5 & \beta_4 & -\beta_5 & -\beta_4 & \beta_5 \\ -\beta_6 & \beta_6 & -\beta_6 & \beta_6 & -\beta_6 & \beta_6 & -\beta_6 & \beta_6 \\ -\beta_7 & -\beta_7 & -\beta_7 & -\beta_7 & -\beta_7 & -\beta_7 & -\beta_7 & -\beta_7 \end{bmatrix}. \quad (6)$$

2.1.2. Commutative circular matrices

A subspace of all CTs (fast discrete CT) is defined by all transforms which can be generated in a radix-2 structure. We now present a strategy for the GCT sparse matrix decomposition with the help of *negacyclic circulant matrices*. This procedure leads to an approach which is much easier to calculate than the approach in [10, 12]. The computational complexity is $\mathcal{O}(N)$ up to $\mathcal{O}(N \cdot \text{ld}(N))$. The matrix topology is as follows: the coefficients in the main diagonal and in the codiagonals of each sparse matrix are expressed as a function of so-called γ -coefficients and λ -coefficients, respectively [12]. The codiagonals are equipped with the λ -coefficients. The coefficients β_i are monoms in γ and λ . The monoms of ${}^f\mathbf{T}_{N/2}$ are defined as $\beta_{N/2-1} = \gamma_0 \cdot \gamma_1 \cdots \gamma_{\text{ld}(N/2)-1}$ down to $\beta_{N/2-1} = \lambda_0 \cdot \lambda_1 \cdots \lambda_{\text{ld}(N/2)-1}$. The generalized circular matrices ${}^{f(l)}\mathbf{gC}_m$ are used to generate the generalized characteristic matrices in radix-2 structure. The *generalized circular matrix* is defined as follows:

$${}^{f(l)}\mathbf{gC}_m = \gamma_{\text{ld}(m)-1-l} \cdot \mathbf{I}_m + \lambda_{\text{ld}(m)-1-l} \cdot \boldsymbol{\eta}_m^{f(l)}, \quad 0 \leq l \leq \text{ld}(m) - 1, \gamma_{(\cdot)} \in \mathbb{R}, \lambda_{(\cdot)} \in \mathbb{R}, \quad (7)$$

\mathbf{I}_m is an $(m \times m)$ unity matrix, and $\boldsymbol{\eta}_N$ denotes a *negacyclic commutative unity matrix* of the size $(N \times N)$. Details regarding this type of matrix can be found elsewhere [16].

$$\boldsymbol{\eta}_N = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -1 & 0 & 0 & \cdots & 0 \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{I}_{N-1} \\ -\mathbf{I}_1 & 0 \end{bmatrix}. \quad (8)$$

The function $f(l)$ defines the multiplicative structure of

the characteristic matrix (cf. (9)). In general, $f(l)$ is set to $f(l) = l$ or $f(l) = 2^l$, but also other settings are possible, depending on the above-mentioned monom equations. Solutions can be found by solving the appropriate non-linear system of monom equations. The solutions are not unique, but this property provides an opportunity to select the coefficients for optimal hardware implementation. With the above-mentioned equation (7), the characteristic matrix ${}^f\mathbf{T}_{N/2}$ can be expressed as follows:

$${}^f\mathbf{T}_{N/2} = - \prod_{l=0}^{\text{ld}(N/2)-1} ({}^{f(l)}\mathbf{gC}_{N/2})^T. \quad (9)$$

The characteristic matrices ${}^f\mathbf{T}_{N/4}$, ${}^f\mathbf{T}_{N/8}$, and so on are calculated accordingly. The matrices ${}^{f(l)}\mathbf{gC}_{N/k}$ are obviously commutative. This property leads to signal flow graphs which can easily be implemented in hardware.

2.1.3. Absolute value spectrum G

We have used the well-known concept of a transform shift matrix ${}^s\mathbf{S}_N = 1/N \cdot \mathbf{A}_N \cdot {}^s\mathbf{I}_N \cdot \mathbf{B}_N^T$, $-(N-1) \leq s \leq (N-1)$. Details can be found elsewhere [14]. \mathbf{A}_N and \mathbf{B}_N are the transform matrices, whereas ${}^s\mathbf{I}_N$ is the permutation unity matrix for cyclic shifts s . The spectrum ${}^s\mathbf{X}_N$ of a shifted input vector ${}^s\mathbf{x}$ is determined as follows:

$${}^s\mathbf{X}_N = \mathbf{A}_N \cdot {}^s\mathbf{x} = {}^s\mathbf{S}_N \cdot \mathbf{A}_N \cdot \mathbf{x} = {}^s\mathbf{S}_N \cdot \mathbf{X}_N. \quad (10)$$

The symbol \mathbf{G} denotes the translation-invariant absolute value spectrum. It is defined by the above-mentioned period groups. The matrix ${}^s\mathbf{I}_N$ can be written as follows:

$${}^s\mathbf{I}_N = \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \end{bmatrix}, \quad (11)$$

$${}^s\mathbf{I}_{N/2} = {}^s\mathbf{I}_{N/2} + {}^s\mathbf{I}_{N/2}, \quad {}^s\boldsymbol{\eta}_{N/2} = {}^s\mathbf{I}_{N/2} - {}^s\mathbf{I}_{N/2}.$$

Furthermore [16],

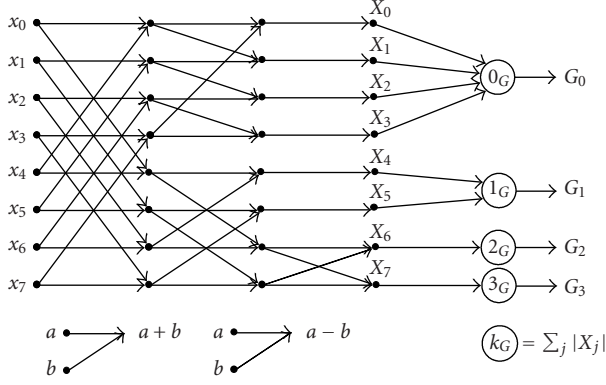
$${}^s\boldsymbol{\eta}_{N/2} = \boldsymbol{\eta}_{N/2}^s. \quad (12)$$

The shift matrix ${}^s\mathbf{S}_N$ is now determined as a diagonal matrix:

$$\begin{aligned} {}^s\mathbf{S}_N &= \frac{2}{N} \cdot \begin{bmatrix} {}^f\mathbf{T}_{N/2} \cdot {}^s\boldsymbol{\eta}_{N/2} \cdot {}^r\mathbf{T}_{N/2}^T & 0 \\ 0 & {}^f\mathbf{T}_{N/2} \cdot {}^s\mathbf{I}_{N/2} \cdot {}^r\mathbf{T}_{N/2}^T \end{bmatrix} \\ &= \cdots = \frac{2}{N} \cdot \begin{bmatrix} {}^f\mathbf{T}_{N/2} \cdot {}^s\boldsymbol{\eta}_{N/2} \cdot {}^r\mathbf{T}_{N/2}^T & 0 \\ 0 & \frac{N}{2} \cdot {}^s\mathbf{T}_{N/2} \end{bmatrix}. \end{aligned} \quad (13)$$

The product ${}^f\mathbf{T}_{N/2} \cdot {}^s\boldsymbol{\eta}_{N/2}$ is negacyclic and therefore commutative [16]. It follows that

$$\begin{aligned} {}^f\mathbf{T}_{N/2} \cdot {}^s\boldsymbol{\eta}_{N/2} \cdot {}^r\mathbf{T}_{N/2}^T &\equiv {}^s\boldsymbol{\eta}_{N/2} \cdot {}^f\mathbf{T}_{N/2} \cdot {}^r\mathbf{T}_{N/2}^T, \\ {}^f\mathbf{T}_{N/2} \cdot {}^r\mathbf{T}_{N/2}^T &= \frac{N}{2} \cdot \mathbf{I}_{N/2}. \end{aligned} \quad (14)$$

FIGURE 1: Signal flow graph of a 1D fast CT ($N = 8$).

The shift matrix is determined by

$${}^s\mathbf{S}_N = \begin{bmatrix} {}^s\eta_{N/2} & 0 \\ 0 & {}^s\mathbf{S}_{N/2} \end{bmatrix} = \begin{bmatrix} {}^s\eta_{N/2} & \cdots & 0 \\ \vdots & {}^s\eta_{N/4} & \vdots \\ 0 & \cdots & \ddots & \ddots \\ 0 & \cdots & & 1 \end{bmatrix}, \quad (15)$$

$${}^s\mathbf{S}_N = \begin{cases} [{}^1\mathbf{S}_N]^{|s|}, & s > 0, \\ \mathbf{I}_N, & s = 0, \\ [{}^1\mathbf{S}_N^T]^{|s|}, & s < 0. \end{cases}$$

The shift matrix has a block diagonal structure which correlates with the above-mentioned period groups. Therefore, it is sufficient to analyze the negacyclic unity matrix for one period group. A property of negacyclic unity matrices is that they remain negacyclic when they are raised to a power [16]. Consequently, the columns of the resulting matrix will permute, and the sign of its components will change. It follows that the columns will permute and the signs of the matrix components will change but the sums of the spectrum's absolute values will *not* change. This leads to a translation-invariant spectrum \mathbf{G} . For example, G_k is determined as follows:

$$G_0 = \sum_{j=0}^{N/2-1} |{}^sX_j| = \sum_{j=0}^{N/2-1} |X_j|, \quad (16)$$

$$G_1 = \sum_{j=N/2}^{3N/4-1} |{}^sX_j| = \sum_{j=N/2}^{3N/4-1} |X_j|,$$

and so forth. The coefficients G_k , $k \in \{2, \dots, \text{ld}(N) - 1\}$, are determined accordingly (cf. Figure 1). This spectrum contains $\text{ld}(N) + 1$ coefficients in the 1D case and $[\text{ld}(N) + 1]^2$ coefficients in the 2D case. The spectrum \mathbf{G} can be interpreted as a feature vector [9, 14].

2.1.4. Mapping strategies for two-dimensional processing

The well-known radix-2 decomposition approach is used for the 2D transform. In general, a 2D transform \mathbf{Y} of an $(N \times N)$

image \mathbf{X} is determined via $\mathbf{Y} = \mathbf{A}_N \cdot \mathbf{X} \cdot \mathbf{A}_N^T$, where \mathbf{A}_N is the 1D transform coefficient matrix. Implementation strategies for the above-mentioned 2D transform includes matrix multiplication as well as matrix transposition which is time and area consuming. However, images captured by cameras are usually processed row-wise. Accordingly, we decompose a 2D transform into a 1D transform with a data length of N^2 with the help of Roth's vec-operation [17], which is expressed as follows:

$$\mathbf{vec}(\mathbf{Y}) = (\mathbf{A}_N \otimes \mathbf{A}_N) \cdot \mathbf{vec}(\mathbf{X}). \quad (17)$$

The vec-operation is defined as a row- or column-wise organized concatenation of a matrix. Equation (17) shows that the 2D transform is calculated, operating on a 1D data stream of pixels line-wise. Furthermore, the Kronecker matrix $(\mathbf{A}_N \otimes \mathbf{A}_N)$ is decomposed into a number of $2 \cdot \text{ld}(N)$ radix-2 sparse matrices $\mathbf{A}_N^{[i]}$. The Kronecker product can be expressed as follows:

$$\begin{aligned} (\mathbf{A}_N \otimes \mathbf{A}_N) &= (\mathbf{A}_N^{[\text{ld}(N)-1]} \otimes \mathbf{A}_N^{[\text{ld}(N)-1]}) \cdots (\mathbf{A}_N^{[0]} \otimes \mathbf{A}_N^{[0]}) \\ &= \cdots = (\mathbf{I}_N \otimes \mathbf{A}_N^{[\text{ld}(N)-1]}) \cdot (\mathbf{A}_N^{[\text{ld}(N)-1]} \otimes \mathbf{I}_N) \\ &\quad \cdots (\mathbf{I}_N \otimes \mathbf{A}_N^{[0]}) \cdot (\mathbf{A}_N^{[0]} \otimes \mathbf{I}_N), \\ (\mathbf{A}_N \otimes \mathbf{A}_N) &= \prod_{i=0}^{\text{ld}(N)-1} (\mathbf{I}_N \otimes \mathbf{A}_N^{[i]}) \cdot (\mathbf{A}_N^{[i]} \otimes \mathbf{I}_N). \end{aligned} \quad (18)$$

The 2D spectrum \mathbf{G}_{2D} is calculated as follows: the absolute value vector $\mathbf{vec}(\tilde{\mathbf{Y}})$ is generated by means of the absolute value components $|Y_i|$. It is defined as $\mathbf{vec}(\tilde{\mathbf{Y}}) = \mathbf{vec}(|Y_i|)$. The spectrum is determined by

$$\mathbf{vec}(\mathbf{G}_{2D}) = (\mathbf{S}_N \otimes \mathbf{S}_N) \cdot \mathbf{vec}(\tilde{\mathbf{Y}}), \quad (19)$$

with

$$\mathbf{S}_G = \begin{bmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 1 & \cdots & 1 & 0 & \cdots & 0 \\ \vdots & & & \vdots & & \ddots & \vdots & \vdots \\ 0 & & & \cdots & & & 1 & 0 \\ 0 & & & \cdots & & & 0 & 1 \end{bmatrix}. \quad (20)$$

\mathbf{S}_G is a sum matrix with $N/2$ ones in the first row, $N/4$ ones in the second row, and so on. The Kronecker product will also be decomposed into $2 \cdot \text{ld}(N)$ radix-2 matrices. Each radix-2 matrix $(\mathbf{I}_N \otimes \mathbf{A}_N^{[i]})$ and $(\mathbf{A}_N^{[i]} \otimes \mathbf{I}_N)$, as well as the radix-2 matrices of the spectrum \mathbf{G}_{2D} are mapped into linear systolic arrays (LSAs).

2.2. Fuzzy pattern classifier

The FPC is a useful approach for modeling complex systems and classifying data [8]. It is based on a concept which allows the simultaneous calculation and aggregation of distance measures. FPC is based on membership functions $\mu(m; \mathbf{p})$. They are modeled as unimodal potential functions [8]. The behaviour of the feature m is described with the appropriate parameter vector \mathbf{p} .

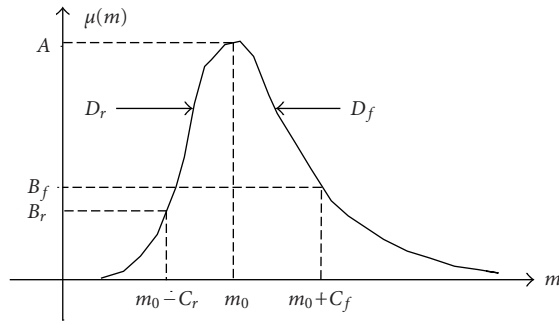


FIGURE 2: Prototype of a unimodal potential function.

A feature vector \mathbf{m} is generated by a preprocessing unit, which in our case computes a nonlinear CT, the translation invariant spectrum $\text{vec}(\mathbf{G}_{2D})$ derived thereof being interpreted as a feature vector \mathbf{m} . For each feature, a membership function is determined. The membership function can be described with 8 parameters which will be defined below. The parameters are determined in a learning phase, or by an expert, mixed strategies being also possible, finally resulting in a time-invariant classifier. Also a time-variant classifier can be constructed [8]. In the *working phase*, a level of affinity is calculated for every incoming set of data and used for the classification. The prototype of a 1D potential function $\mu(m; \mathbf{p})$ can be expressed as follows (Figure 2):

$$\mu(m; \mathbf{p}) = A \cdot [1 + d(m; \mathbf{p})]^{-1}, \quad (21)$$

with the difference measure

$$d(m; \mathbf{p}) = \begin{cases} \left(\frac{1}{B_r} - 1 \right) \cdot \left(\frac{|m - m_0|}{C_r} \right)^{D_r} & \forall m < m_0, \\ \left(\frac{1}{B_f} - 1 \right) \cdot \left(\frac{|m - m_0|}{C_f} \right)^{D_f} & \forall m \geq m_0. \end{cases} \quad (22)$$

This difference can be interpreted as a generalized Minkowski distance. The potential function gets comprehensively determined by the parameter vector $\mathbf{p} = (m_0, B_r, B_f, C_r, C_f, D_r, D_f)^T$. Referring to Figure 2, the elementary parameters belonging to the vector \mathbf{p} are defined as follows.

The parameter m_0 corresponds to the average value of a 1D signal or feature, or the center of gravity in case of an M -dimensional feature space. The value A denotes the maximum value of the function. In the hardware design described in this paper, $A = 1$. The elements m_0 and A are interrelated by the formula $A = \mu(m_0; \mathbf{p})$.

The parameters B_r and B_f determine in turn the value of the membership function on the boundaries $m_0 - C_r$ and $m_0 + C_f$. The membership values for the rising and falling edges are given by the expressions $\mu(m_0 - C_r; \mathbf{p}) = B_r$ and $\mu(m_0 + C_f; \mathbf{p}) = B_f$. The parameters C_r and C_f define the maximum distance from the center of gravity. This value is calculated from the maximum and minimum of the signal amplitude of each feature.

The parameters D_r and D_f are determined from each feature's amplitude distribution. They model the decrease in membership with the increase of the distance from the center of gravity. A detailed description of the parameters and their calculations can be found in [8].

In an M -dimensional feature space, the membership functions (equation (21)) are connected together in a conjunctive way. All feature representatives m_k , $k \in \{0, 1, \dots, M - 1\}$, exhibit their specific parameters ${}^k \mathbf{p} = (m_{0k}, B_{rk}, B_{fk}, C_{rk}, C_{fk}, D_{rk}, D_{fk})^T$. The scalar function $\mu(\mathbf{m}; {}^k \mathbf{p})$ for M features is described as follows:

$$\mu(\mathbf{m}; {}^k \mathbf{p}) = A \cdot \left[1 + \sum_{i=0}^{M-1} d(m_i; {}^i \mathbf{p}) \right]^{-1}. \quad (23)$$

All distance measures are summed up. The result is one membership function for one class. Membership functions for K classes are also constructed in the same way. The classification is generated with a disjunction/conjunction network and $\text{argmin}(\cdot)$ and $\text{argmax}(\cdot)$ operations. The potential function is again mapped into an LSA. Furthermore, the above-mentioned definition of a membership function is not the only possible one. Different potential functions can be defined [9]. The membership function which is used for the hardware implementation is determined as follows:

$$\mu(\mathbf{m}; {}^k \mathbf{p}) = 2^{-\sum_{i=0}^{M-1} d(m_i; {}^i \mathbf{p})}, \quad B_{fk} = B_{rk} = \frac{1}{2}. \quad (24)$$

This concept has some advantages in implementation. The membership function is calculated with logical shifts and one multiplication is saved without loss of classification accuracy, considering that (cf. (22), (24)) $(1/B_r - 1) = 1$ and $(1/B_f - 1) = 1$.

Basically translation-invariant output spectra are sufficient in order to describe image contents. In real-image scenes and applications however, a simple comparison of invariant spectra is not easy to achieve. In practice, situations can occur which prevent the performing of simple comparisons due to, for example, object shifts under the camera system, noncyclic shifts, aliasing effects during the digitization and further effects induced by different backgrounds, and so forth [7]. Therefore, nonlinear CTs should be used in conjunction with postprocessing units such as FPC.

In our approach, the MFPC performs this task. In the learning phase, a certain number of image samples is used to create a minimum and maximum master spectrum. The minimum and maximum of each feature are determined for the creation of the distance C_k measured along the dimension k :

$$C_k = \frac{1}{2} \cdot [\max(m_k) - \min(m_k)]. \quad (25)$$

For each feature, a potential function is defined. All outputs of the functions are aggregated with a fuzzy AND function network (cf. (24)), resulting in a single membership value $\mu(\mathbf{m}; {}^k \mathbf{p})$ per image. This value is then compared with

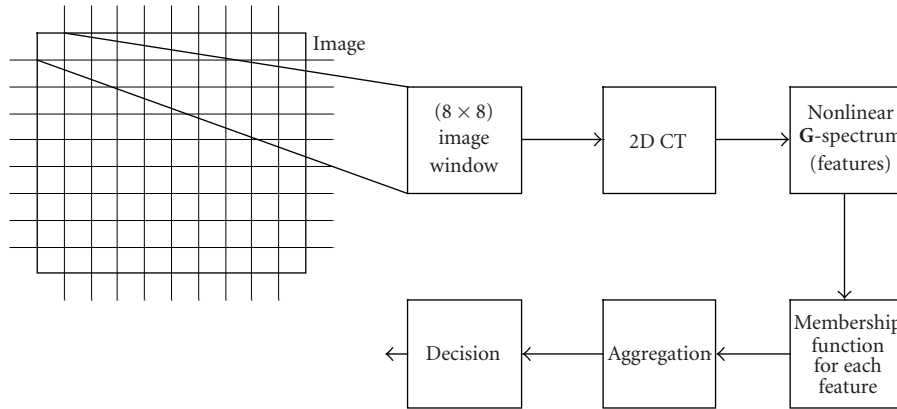


FIGURE 3: Signal flow.

a threshold μ_t to produce the decision c defined as follows:

$$c = \begin{cases} 1, & \mu(\mathbf{m};^k\mathbf{p}) \geq \mu_t, \\ 0, & \mu(\mathbf{m};^k\mathbf{p}) < \mu_t, \end{cases} \quad (26)$$

with acceptance value $c = 1$. The threshold is adjusted manually by an operator at system installation and exploitation time.

3. IMPLEMENTATION ON FPGA

The 2D CT and the FPC are implemented on a single FPGA [18]. Figure 3 shows the signal flow of the processing unit. The signal flow indicates how a complete image is analyzed by the system. The data input and output accesses are designed for monochrome images of a size of (2048×2048) pixels. The features are calculated and classified within $(N \times N)$ windows of the typical size of (8×8) pixels whereas other window sizes can be used as well.

We use an Altera Apex EP20K600E FPGA device [18], counting 24 320 logic elements (LEs). The decision for selecting the above-mentioned FPGA was motivated by the internal structure of the FPGA. In this paper, we propose a concept to circumvent the drawbacks affecting the clock skew in application specific integrated circuits in the case of systolic array implementation.

The main unit is the LE. One LE consists mainly of a register, a 4-input lookup table, preset and reset logic, and a clock distribution unit. Indeed, 10 LEs get grouped to compose so-called logic array blocks (LABs) benefiting from local interconnections. Furthermore, LABs are in turn grouped by a number of 16 to form so-called MegaLABs. Accordingly, the latter operates with up to 160 LEs interconnected by short data and clock signals.

One general challenge we have to cope with is the clock distribution scheme. It is a known fact that the design of clock trees in application specific integrated circuits is not an easy task [18]. One main problem is the fact that clock lines have different lengths and therefore the clock signals will be

differently delayed. This effect is called clock skew. Insertion of a so-called balanced clock tree has to take account of layout so that the tree is balanced not only in terms of the number of flip-flops attached, but also of clock drivers fan-out and especially of the wire lengths. The clock delay and clock skew parameters account for a significant portion of the total setup time and clock-to-output delay in larger devices.

As clock skew depends heavily on the placement of the macrofunctions in gate arrays, special care has to be taken in the placement of these elements. Therefore, macros, such as local systolic arrays, have to be placed on the chip while making sure that the clock distribution is perfectly designed. In general, using phase-locked loops (PLLs) during the clock synthesis helps improve the clock jitter and clock phase performance [18]. Indeed, PLLs can be tuned to produce output clock signals performing at different low jitter levels and predefined phase. The PLLs are able to perform different low jitter and defined phase clock output. Assuming the use of some PLLs for different macrofunctions, which are controlled for proper clock skew, opens one opportunity to increase the system performance. One drawback of gate array design is that a major part (up to 50%) of the development time has to be reserved for the clock tree and PLL design.

Systolic arrays are usually stretched over several thousand LEs, so clock skew can become a major issue. Taking into account the clock scheme principles, it is obvious that the implementation of systolic arrays on application specific integrated circuits is not easy to achieve.

The used FPGA is equipped with 4 programmable PLLs and a clock network which is connected to all MegaLAB structures. The integrated analogue PLL circuit enables a chip design with phase alignment capability. Phase shifting is used to minimize the clock skew between different system clock domains. The clock network consists of 4 global clock-buffers with very high fan-out count. The clock distribution networks inside the MegaLAB structure guarantee low clock skew distribution to each LE. All clock lines feature equal lengths. Compared to a gate array implementation, there is no need to generate a complex clock tree.

TABLE 1: Number of FPGA-LEs used for the implementation of the functional blocks. All local interconnections related to the systolic arrays are included in the listed number of LEs. External input and output data busses are counted separately.

Implementation	LEs	Utilization (%)
GCT	4 149	17.1
Translation-invariant spectrum \mathbf{G}	1 465	6.0
FPC	9 944	41.0
Min/max network	268	1.1
Control and glue logic, data busses	4 964	20.4
Total	20 790	85.6

Referring to the above-mentioned remarks, a positive coincidence appears when implementing the systolic arrays in FPGAs with the above-mentioned clock network properties. Each PLL is used for one of the systolic arrays (cf. Table 1) and adjusted for minimum clock skew. An increase of performance in maximum clock frequency is achievable. Minimizing the clock skew with 4 PLLs, the clock frequency performance increases from approximately 34 MHz to 40 MHz (> 17%). The phase shift is implemented within a step resolution lower than 1 nanosecond.

The transform and the invariant spectrum \mathbf{G}_{2D} as well as the FPC and the min/max determination are based on LSAs. Most of the processing elements are designed as 16 bit inner product step processors, which correspond to a multiplier and accumulator cell (MAC). In general, one cell is designed with one MegaLAB structure. Of course, the divider and potential networks, which were both designed for 32 bit data, operates with up to 6 MegaLAB structures but in a *straight-forward* design. Therefore, it is possible to operate each systolic array with a serial clock distribution scheme. Care has to be taken at the interconnections between the arrays. It is absolutely necessary to synchronize the data flow and the clock with a set of registers. A proper cut-set retiming was used to achieve the processing times which are mentioned in the following section.

Approximately 20% of all LEs are foreseen for the control and glue logic as well as for the input and output data busses. The control unit is equipped with RAM controllers and a VMEbus interface. Table 1 shows the percentage of utilized LEs for the 2D transform, the translation-invariant spectrum \mathbf{G}_{2D} , the FPC, the min/max unit, and the RAM controllers, glue logic, and timing control. It has to be pointed out that all necessary local connections within the units are *included* in the LE count.

Table 1 shows a total amount of 20 790 LEs, which is equivalent to a factor of approximately 85.6% chip utilization. The overall latency time (defined as the time interval between the application of specific input data and delivery of corresponding results at the output) per block is calculated with 249 clock cycles before the first result leaves the classifier. The FPGA operates with a maximum clock frequency of 40 MHz. Therefore, a processing time of 6.23 microseconds per block is achieved, if an (8×8) window is used. Table 2

TABLE 2: Latency times.

	Spectrum \mathbf{G}_{2D}	Fuzzy pattern classifier
Clock cycles	132	117
Time (μ s)	3.3	2.93

presents an overview of different latency times of the components.

4. EXPERIMENTAL RESULTS

In this section, we present some experimental results using the transforms for pattern separability tests. The results were previously published in [12]. However, applying the new monoms resolution strategy based on (9), it is possible to find fast transforms for all mentioned CTs. We used binary test patterns as input vectors. Binary numbers can be interpreted as patterns under cyclic permutation. Thus, if a left (or right) shift is used with a particular number, a new number in the class will be generated. We compared our results with the results given by the well-known Fourier transform power spectrum and the rapid transform spectrum. Three CTs were defined (example for $N = 16$):

- (1) CT1: $\mathbf{c}_{\beta 1} = (2^7, 2^6, \dots, 2^0, 2^3, 2^2, \dots, 2^0, 0, -1, -1, -1)^T$. This CT has a computational complexity of $N \cdot \text{ld}(N)$. All computations can be processed with integers;
- (2) CT2: $\mathbf{c}_{\beta 2} = -k \cdot \cos(\pi \cdot (i + 1/2)/N)$ with $i = 1, 2, \dots, N - 1$. A radix-2 structure with real numbers is possible. The factor k is chosen such that the last spectral coefficient represents the average value of the input vector;
- (3) CT3: $\mathbf{c}_{\beta 3} = (r_0, r_1, \dots, r_{N-1})^T$. The CT3 coefficients are defined as a Gaussian noise signal with variance $\sigma = 1$ and average = 0. Calculations in radix-2 structure are also possible with a proper radix-2 decomposition.

Table 3 shows the results of the separability test. It is obvious that the proposed CTs are superior in comparison to the Fourier power spectrum and the rapid transform for $N > 4$.

5. APPLICATIONS

5.1. Printed image inspection and image retrieval

Our approach is effective for inspection of printed or hard-copy images, especially in areas with high contrast differences, for example, edges. It is well known that concepts of iconic image processing are weak in these areas. The above-mentioned concepts remain at level of pixel-based algorithms like pixel differences, pixel thresholds, min/max operations, and so on. The algorithms tend to generate areas of massive deviations from an average area gray value when applied to printed contrast differences. Because of the local movement provoked naturally by various printing processes which are in most practical cases translative, the spectrum \mathbf{G} of the CT is able to stabilize the unknown local dynamics. As the printed format (sheet) moves under a camera, the

TABLE 3: Separability properties of binary pattern. The amount of separable patterns is processed by Polyá's counting theory (cf. [3]). The number in column 3 indicates the maximum translation-invariant patterns which is achievable for binary patterns. All data in the columns 4 to 10 indicate the number of separable patterns under different transforms.

N	2^N	Amount of separable patterns	Rapid transform	Fourier power spectrum	CT1	CT2	CT3	RMWHT [6]	SWT [15]
2	4	3	3	3	3	3	3	3	3
4	16	6	6	6	6	6	6	6	6
8	256	36	21	31	31	31	33	21	29
16	65536	4116	225	1876	3245	3496	3527	208	668

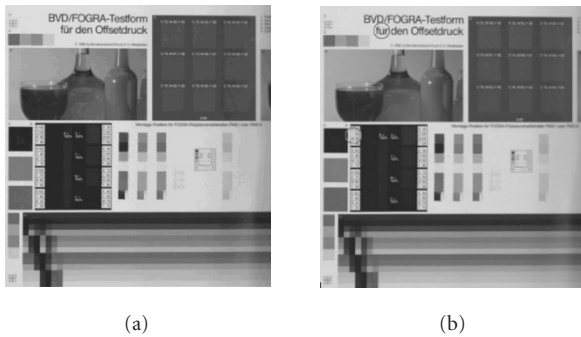


FIGURE 4: (a) Cutout of a reference image. (b) Cutout of an error image; the errors are marked with circles.

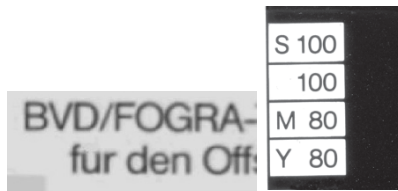


FIGURE 5: Zoomed error cutouts.

image has to be triggered for a stable image representation. Under practical considerations, slight object movements will always occur, which causes slight changes in the image representation. These changes are detectable as amplitude noise in the spectral amplitude of each coefficient. The spectrum G_{2D} has to be characterized as translation *tolerant*. Therefore, the FPC has to cope with these spectral dynamics. Furthermore, a dichotomic decision such as “good/bad” and so forth is in most cases sufficient. A further advantage is that the system operates in real time because of the above-mentioned latency times. As an illustrative example, we present an analysis of test prints with typical printing flaws. Figure 4 shows a (740×780) -pixel cutout of a (2048×1536) -pixel image as a reference and an error image. Approximately 100 reference sheets are used for the classifier training. Following, different sheets, which were not trained, were inspected. Typical errors which were detected are shown in Figure 5. The first error consists of two missing dots above the letter “ü” and the second error of a missing letter “C”.

The approach can be used for image retrieval as well. It has to be pointed out that the window size depends on the application. In the case of image retrieval, the windows

are placed over the image in form of a grid pattern. Within each window, the calculated spectral coefficients can be considered as local. Different CTs and potential functions were examined. Transforms with good separation characteristics work favorably with (16×16) windows. For (8×8) windows, too many details were mapped. For (8×8) windows, transforms with low separation characteristics are optimal (e.g., RMWHT [6], SWT [15]).

5.2. Character recognition

The algorithms can also be used in the area of handwriting or printed character recognition. The procedure is sketched as follows: on the basis of scanned characters (A, B, \dots, Z) , which are stored for example as (8×8) - or (16×16) -pixel data fields, prototypes of handwritten characters are trained. A parameter field, consisting of $M \cdot p$ items, is then determined for each character. This data matrix represents the trained parameters. In the test phase, a learned CT feature matrix of a test character is compared with the trained data set in the classifier system. For each character, a membership value is generated regarding the test character. This means that a membership value vector $Z = (\mu_A, \mu_B, \dots, \mu_Z)^T$ can be checked for the value with the maximum membership amplitude (max of height). The position of the maximum value is defined as a certain character $c = \arg \max(Z_i)$.

6. CONCLUSION

We have presented algorithms and a corresponding FPGA implementation, which are suitable for image processing applications based on an FPGA Altera Apex EP20K600E. The FPGA operates with a clock frequency of 40 MHz. Different nonlinear CTs and an FPC are implemented as feature generators and classifier, respectively. The combination of both modules leads to a flexible pattern recognition approach, which is adaptable to the application tasks. Typical applications are image retrieval, texture and image analysis, similarity detection, and character recognition.

REFERENCES

- [1] H. Reitboeck and T. P. Brody, “A transformation with invariance under cyclic permutation for applications in pattern recognition,” *Information and Control*, vol. 15, no. 2, pp. 130–154, 1969.
- [2] M. D. Wagh and S. V. Kanetkar, “A class of translation invariant transforms,” *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 25, no. 2, pp. 203–205, 1977.

- [3] H. Burkhardt, A. Fenske, and H. Schulz-Mirbach, "Invariants for the recognition of planar contour and gray-scale images," *Technisches Messen tm*, vol. 59, no. 10, pp. 398–407, 1992.
- [4] M. Fang and G. Häusler, "Modified rapid transform," *Applied Optics*, vol. 28, no. 6, pp. 1257–1262, 1989.
- [5] J. Turan and K. Althöfer, "A novel system for 3D acoustic object recognition based on the modified rapid transform," *Journal of Electrical Engineering*, vol. 46, no. 8, pp. 265–269, 1995.
- [6] V. Lohweg and D. Müller, "Anwendung schneller diskreter Spektraltransformationen zur translationsinvarianten Merkmalgewinnung [Application of fast discrete spectral transforms for translation invariant feature extraction]," in *Mustererkennung 1999, 21. DAGM-Symposium*, Informatik Aktuell, pp. 266–275, Springer, Bonn, Germany, September 1999.
- [7] S. Siggelkow and H. Burkhardt, "Image retrieval based on local invariant features," in *Proc. IASTED International Conference on Signal and Image Processing*, pp. 369–373, Las Vegas, Nev, USA, October 1998.
- [8] S. F. Bocklisch and U. Priber, "A parametric fuzzy classification concept," in *Proc. International Workshop on Fuzzy Sets Applications*, pp. 147–156, Akademie-Verlag, Eisenach, Germany, March 1986.
- [9] K. Eichhorn, *Entwurf und Anwendung von ASICs für musterbasierte Fuzzy-Klassifikationsverfahren*, Ph.D. thesis, Circuit and System Design, Technical University of Chemnitz, Chemnitz, Germany, 2000.
- [10] V. Lohweg and D. Müller, "Ein generalisiertes Verfahren zur Berechnung von translationsinvarianten Zirkulartransformationen für die Anwendung in der Signal- und Bildverarbeitung [A generalized method for circular transforms translation invariance determination with applications in signal and image processing]," in *Mustererkennung 2000, 22. DAGM-Symposium*, Informatik Aktuell, pp. 213–220, Springer, Kiel, Germany, September 2000.
- [11] V. Lohweg and D. Müller, "A complete set of translation invariants based on the cyclic correlation property of the generalized circular transforms," in *Proc. 6th Digital Image Computing Techniques and Applications (DICTA '02)*, pp. 134–138, Australian Pattern Recognition Society, Melbourne, Australia, January 2002.
- [12] V. Lohweg and D. Müller, "Nonlinear generalized circular transforms for signal processing and pattern recognition," in *IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP '01)*, Baltimore, Md, USA, June 2001.
- [13] N. Ahmed, K. R. Rao, and A. Abdussattar, "BIFORE or Hadamard transform," *IEEE Transactions on Audio and Electroacoustics*, vol. 19, no. 3, pp. 225–234, 1971.
- [14] N. Ahmed and K. R. Rao, *Orthogonal Transforms for Digital Signal Processing*, Springer, New York, NY, USA, 1975.
- [15] D. Covey and J. Pender, "New square wave transform for digital signal processing," *IEEE Trans. Signal Processing*, vol. 40, no. 8, pp. 2095–2097, 1992.
- [16] P. J. Davis, *Circulant Matrices*, John Wiley & Sons, New York, NY, USA, 1979.
- [17] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1994.
- [18] Altera, *Digital Library of FPGAs*, San Jose, Calif, USA, March 2002, <http://www.altera.com>.

Volker Lohweg is Head of Koenig & Bauer AG, Bielefeld branch (optical systems). His research interests include image processing and pattern recognition for banknote printing, as well as VLSI design. Volker Lohweg has a Ph.D. degree in electrical engineering from Chemnitz University of Technology. He is appointed Professor of digital systems at Lippe and Hoexter University of Applied Science. Volker Lohweg is a Member of the German Association for Pattern Recognition (DAGM) and the Institute of Electrical and Electronic Engineers (IEEE).



Carsten Diederichs is the Head of the Hardware Design Group at Koenig & Bauer AG, Bielefeld branch. His interests include field-programmable logic design and efficient hardware implementation of computer arithmetic algorithms. Carsten Diederichs has a Dipl.-Ing. (FH) degree in electrical engineering from the Lippe and Hoexter University of Applied Science.



Dietmar Müller is a Professor of electrical engineering and Head of the Circuit and System Design Group at Chemnitz University of Technology. His research interests include VLSI design and field-programmable logic. Dietmar Müller has a Ph.D. degree in electrical engineering from both the University of Dresden and Chemnitz University of Technology. He is a Member of the Association for Electrical, Electronic, and Information Technologies (VDE) and the Information Technology Society (ITG).

