

Introduction Into the Novel Two-Dimensional Discrete Orthogonal Transforms Based on Rotation Angles

P. Misans, U. Derums

Faculty of Electronics and Telecommunications of Riga Technical University,

Azenes iela 12, LV-1048, Riga, Latvia, phone: +371-9-135-489, fax: +371-7-08-92-92, e-mail: name.family@rtu.lv

crossref <http://dx.doi.org/10.5755/j01.eee.112.6.456>

Introduction

In the IEEE Member Digital Library [1] we can find about a dozen of papers concerning the signal processing based on the rotation angles. We suppose that the cornerstones for the description of discrete orthogonal (orthonormal) transforms are rotation angles of planes in Euclidian space [2]. The angular approach is very well known in linear algebra but it has been used not so often by signal processing people. It seems that H. C. Andrews [3] is a pioneer in the area of parameterization of fast orthogonal (orthonormal) transforms by rotation angles of planes. P. Rieder with colleagues (for example, [4]) use rotation angles in the context of CORDIC-based implementation of orthogonal wavelet transforms and DCT. The pioneer in introducing and using of rotation angle based orthogonal filters is P. P. Vaidyanathan [5]. We avoid here a detailed overview of all available works on rotation angle approach because of the limited space of paper. Such overview is available in [6].

In [2] we introduced several classes of real discrete Rotation Angle Based Orthogonal Transforms (RABOT). These transforms can be interpreted also as a single parametrical transform with infinite number of shapes of basis functions (BFs). On the other hand, we can use rotation angles as the base for different classification schemes of orthogonal transforms [2].

We expect that the use of RABOT transforms could be very promising in signal analysis and synthesis. We are working on that. Particularly, preliminary results show that one of the subclasses of RABOT (in [2] called as CRAIMOT) is very useful for compression of speech. We need a relatively small number of orthogonal functions (sometimes only a few BFs) to represent, for example, the vowels of Latvian speech [7]. Our team works on the development of FPGA-based "angular" devices. Experimental generators [8], analyzers [9] and orthogonal filters [10] are our results in the previous year. We expect an appearance of devices for more wide range applications in the nearest future. Simultaneously with the practical

programming of FPGAs we work on the "theory" of angle-based functions (Phi-functions). One of the recent papers deals with a possible generalization of Haar functions [11].

Although our team has issued more than a twenty of papers about Phi-functions during the recent 6 years, this introductory paper is only the first step in the range of "theoretical" works dealing with two-dimensional (2-D) transforms. We do not aim for mathematical perfection and absolute novelty. Some things are well known but some of them appear for the first time.

Basics of two-dimensional orthogonal transforms

The real discrete 2-D Rotation Angle Based Orthogonal Transform (RABOT2D) is closely related to the other real discrete 2-D transforms (Haar, Hartley, Hadamard etc. [12]). It is a separable linear transformation - that is, the 2-D is equivalent to a one-dimensional (1-D) RABOT performed along a single dimension followed by a 1-D RABOT in the other dimension. The definition of the RABOT2D for an input image (signal) \mathbf{X} (size of $N \times N$) and output image (spectrum) \mathbf{Y} (also size of $N \times N$) is:

$$\begin{cases} y(r, k) = \sum_{m=1}^N \sum_{n=1}^N \eta(r, m) \cdot \eta(k, n) \cdot x(m, n), \\ r \in [1, N], k \in [1, N], N = 2^n, \end{cases} \quad (1)$$

where $x(r, k)$ and $y(r, k)$ are the value of the element of signal (pixel) and spectrum, respectively. r and k are row and column indexes. $\eta(p, q)$ represents the element of 1-D transform matrix \mathbf{H} in p -th row and q -th column. We reduce here and below orthogonal transforms to orthonormal transforms by ignoring the scaling factors that are usual for such transforms. Such approach simplifies manipulations with formulas.

We can also represent a direct 2-D transform in the matrix form

$$\mathbf{Y} = (\mathbf{H} \cdot (\mathbf{H} \cdot \mathbf{X})^T)^T, \quad (2)$$

where \mathbf{H} is a 1-D orthonormal matrix but $(\dots)^T$ means the transposition of matrix.

The inverse transform can be presented similarly

$$\mathbf{X} = (\mathbf{H}^T \cdot (\mathbf{H}^T \cdot \mathbf{Y})^T)^T. \quad (3)$$

It is easily to find the equivalent of (1) for the inverse transform, but this is not so significant because for the calculation of orthonormal transforms the fast algorithms can be used.

The inverse transform (3) can be used to calculate the chosen 2-D BF $\mathbf{F}(r,k)$, if we assign the $\mathbf{1}$ to the element $y_0(r,k)$ of the matrix of all zeros \mathbf{Y}_0

$$\mathbf{F}(r,k) = (\mathbf{H}^T \cdot (\mathbf{H}^T \cdot \mathbf{Y}_0(r,k))^T)^T, \quad (4)$$

where

$$\mathbf{Y}_0(r,k) = \begin{matrix} & & & & k \\ & & & & 0 & 0 & \dots & 0 \\ & & & & 0 & 0 & 0 & \dots & 0 \\ & & & & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ & & & & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ r & & & & 0 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ & & & & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ & & & & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ & & & & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{matrix}. \quad (5)$$

This assignment means that here has been used only one 2-D spectrum coefficient which corresponds to chosen 2-D BF. If we take into account (5), we can simplify (4)

$$\mathbf{F}(r,k) = (\mathbf{H}^T \cdot \mathbf{F}_0(r,k))^T = \mathbf{F}_0^T(r,k) \cdot \mathbf{H}, \quad (6)$$

where $\mathbf{F}_0(r,k)$ is the matrix with the r -th 1-D BF instead of k -th row and all zeros in the other rows

$$\mathbf{F}_0(r,k) = k \begin{bmatrix} 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \eta(r,1) & \eta(r,2) & \dots & \eta(r,N) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}. \quad (7)$$

Finally, the 2-D BF can be expressed as

$$\mathbf{F}(r,k) = \begin{bmatrix} \eta_{r,1} \cdot \eta_{k,1} & \eta_{r,1} \cdot \eta_{k,2} & \dots & \eta_{r,1} \cdot \eta_{k,N} \\ \eta_{r,2} \cdot \eta_{k,1} & \eta_{r,2} \cdot \eta_{k,2} & \dots & \eta_{r,2} \cdot \eta_{k,N} \\ \vdots & \vdots & \ddots & \vdots \\ \eta_{r,N} \cdot \eta_{k,1} & \eta_{r,N} \cdot \eta_{k,2} & \dots & \eta_{r,N} \cdot \eta_{k,N} \end{bmatrix}, \quad (8)$$

where $\eta_{p,q} = \eta(p,q)$.

Fast Algorithms

The 1-D RABOT transform has a fast algorithm (RABFOT) defined as the product of sparse orthogonal (orthonormal) matrices [2]

$$\mathbf{H} = \mathbf{B}_l \cdot \dots \cdot \mathbf{B}_2 \cdot \mathbf{B}_1 = \mathbf{B}(\varphi_l) \cdot \dots \cdot \mathbf{B}(\varphi_2) \cdot \mathbf{B}(\varphi_1), \quad (9)$$

$$l \leq n = \log_2(N),$$

where \mathbf{B}_p – sparse orthonormal matrix (the Stairs-like Orthonormal Generalized Rotation Matrix (SOGRM) [2])

$$\mathbf{B}_j = \begin{bmatrix} \tau_{1,j}^1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \tau_{2,j}^1 & \dots & 0 & 0 \\ \dots & \dots & \ddots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \tau_{N/2,j}^1 & \dots \\ \tau_{1,j}^2 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \tau_{1,j}^2 & \dots & 0 & 0 \\ \dots & \dots & \ddots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \tau_{N/2,j}^2 & \dots \end{bmatrix}, \quad (10)$$

where $\tau_{i,j}^1$ and $\tau_{i,j}^2$ are 2-element row-vectors which originate from the elementary four-element rotation matrix

$$T_{i,j} = \begin{bmatrix} \tau_{i,j}^1 \\ \tau_{i,j}^2 \end{bmatrix} = \begin{bmatrix} f^{11}(\phi_{ij}) & f^{12}(\phi_{ij}) \\ f^{21}(\phi_{ij}) & f^{22}(\phi_{ij}) \end{bmatrix} = \begin{bmatrix} c_{ij} & s_{ij} \\ -s_{ij} & c_{ij} \end{bmatrix}. \quad (11)$$

Elements $f_{i,j}^{mn}$ in (11) present the cosine and sine values of angle ϕ_{ij} . We use shortcuts for the cosine (c) and sine (s) values

$$\begin{cases} G^\pm: f_{ij}^{11} = c_{ij}, f_{ij}^{12} = \mp s_{ij}, f_{ij}^{21} = \pm s_{ij}, f_{ij}^{22} = c_{ij}, \\ R^\pm: f_{ij}^{11} = \pm s_{ij}, f_{ij}^{12} = c_{ij}, f_{ij}^{21} = c_{ij}, f_{ij}^{22} = \mp s_{ij}, \end{cases} \quad (12)$$

where \mathbf{G} and \mathbf{R} geometrically represent the kind of rotation of planes (see [2]). In such a way we can create an infinite number of orthonormal (also orthogonal) transforms by using the rectangular matrix of rotation angles ([2])

$$\boldsymbol{\varphi} = \begin{bmatrix} \phi_{1,1} & \phi_{1,2} & \phi_{1,3} & \dots & \phi_{1,l} \\ \phi_{2,1} & \phi_{2,2} & \phi_{2,3} & \dots & \phi_{2,l} \\ \phi_{3,1} & \phi_{3,2} & \phi_{3,3} & \dots & \phi_{3,l} \\ \dots & \dots & \dots & \dots & \dots \\ \phi_{\frac{N}{2},1} & \phi_{\frac{N}{2},2} & \phi_{\frac{N}{2},3} & \dots & \phi_{\frac{N}{2},l} \end{bmatrix}. \quad (13)$$

In formula (9) φ_p represents the p -th column-vector of angle matrix $\boldsymbol{\varphi}$

$$\varphi_p = [\phi_{1,p}, \phi_{2,p}, \dots, \phi_{N/2,p}]^T. \quad (14)$$

We can define some relations between the elements of angle matrix and get some classes of orthogonal

transforms (OT) (see the section about the classification of RABOTs).

We can replace the transform matrix \mathbf{H} in (2) and (3) by the product of SOGRM matrices (10) to get the fast version of RABOT2 – RABFOT2. For the direct transform we obtain

$$\mathbf{Y} = (\mathbf{B}_1 \cdot \dots \cdot \mathbf{B}_2 \cdot \mathbf{B}_1 \cdot (\mathbf{B}_1 \cdot \dots \cdot \mathbf{B}_2 \cdot \mathbf{B}_1 \cdot \mathbf{X})^T)^T. \quad (15)$$

But, in the case of inverse transform we have

$$\mathbf{X} = ((\mathbf{B}_1 \cdot \dots \cdot \mathbf{B}_2 \cdot \mathbf{B}_1)^T \cdot ((\mathbf{B}_1 \cdot \dots \cdot \mathbf{B}_2 \cdot \mathbf{B}_1)^T \cdot \mathbf{Y})^T)^T. \quad (16)$$

Sometimes for the implementation of \mathbf{H}^T it is more convenient to use the transposed form of SOGRM matrices

$$\mathbf{H}^T = (\mathbf{B}_1 \cdot \dots \cdot \mathbf{B}_2 \cdot \mathbf{B}_1)^T = \mathbf{B}_1^T \cdot \mathbf{B}_2^T \cdot \dots \cdot \mathbf{B}_1^T. \quad (17)$$

In the case of the generation of single 2-D BF ((6)) it is also possible to get the benefit of the number of operations used for the generation of BF by the allocation of SOGRMs

$$\mathbf{F}(r, k) = \mathbf{F}_0^T(r, k) \cdot \mathbf{B}_1 \cdot \dots \cdot \mathbf{B}_2 \cdot \mathbf{B}_1. \quad (18)$$

Classes of RABOT2 transforms

Each reader can create his own novel class of transform by choosing restrictions on the angle matrix (13). It is possible also to define 2-D RABOT (RABOT2) similar to the 1-D RABOT [2]. We will limit our efforts to a preliminary short description of RABOT2 only (see Table 1).

Table 1. Description of RABOT2

Transform		Comments
RABOT – Rotation Angle Based OT	$\phi_{ij} = \phi_{ij}$	angles in (13) are different
CRAOT – Constant Rotation Angle OT	$\phi_{ij} = \phi$	all angles are equal
CRAIMOT – Constant Rotation Angle Inside Matrix OT	$\phi_{ij} = \phi_i$	all rows in (13) are equal (one constant angle per \mathbf{B})
CRMOT – Constant Rotation Matrix OT	$\phi_{ij} = \phi_j$	all columns in (13) are equal (all \mathbf{B} matrices are equal)
RAHT – Rotation Angle-based Haar-like Transforms	see [11]	angle matrix has a triangle structure with constant value angle in the right lower corner

Shapes of BFs

We can get an infinite number of shapes of BFs by a simple change of angles in matrix (13). In such a way, for example, the angle matrix (for $N=2^3$) with a triangle structure and two different values ($\pi/6, \pi/4$)

$$\boldsymbol{\varphi} = \begin{vmatrix} \pi/6 & \pi/6 & \pi/6 \\ \pi/6 & \pi/6 & \pi/4 \\ \pi/6 & \pi/4 & \pi/4 \\ \pi/6 & \pi/4 & \pi/4 \end{vmatrix} \quad (19)$$

gives the set of BFs which are shown in Fig. 2.

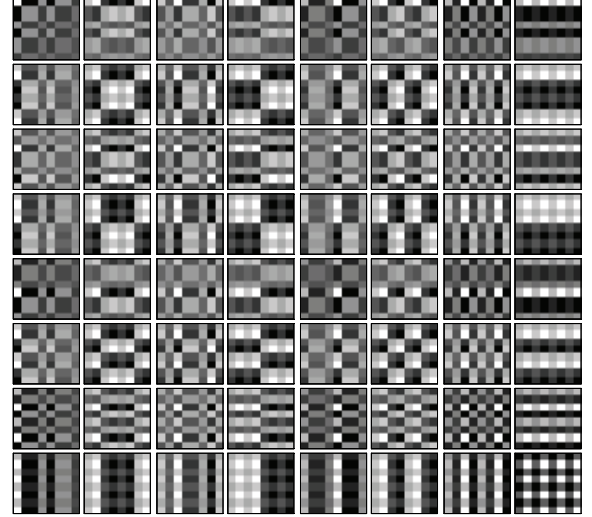


Fig. 1 The full set of 2-D basis functions for rotation angles defined by (19)

It is possible also to define the a 2-D version of generalized Haar-like functions [11] for the angle matrix with zeros in the lower right corner

$$\boldsymbol{\varphi} = \begin{vmatrix} \pi/6 & \pi/6 & \pi/6 \\ \pi/6 & \pi/6 & 0 \\ \pi/6 & 0 & 0 \\ \pi/6 & 0 & 0 \end{vmatrix}. \quad (20)$$

Corresponding BFs are shown in Fig. 3

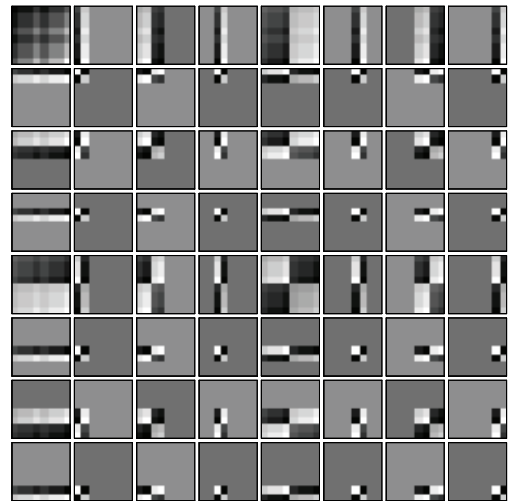


Fig. 2 The full set of 2-D basis functions for rotation angles defined by (20)

Number of Operations for the Calculation of RABOT2

In the case of practical implementation of RABOT2 the estimation of the number of arithmetical operations is of primary importance.

1. Number of Operations for the Direct Calculation.

The formula (1) or (2) can be used to estimate the number of multiplications and summations. We have $N \times N = N^2$ 2-D BFs or the same number of spectrum coefficients. For the calculation of spectrum we need:

$$\begin{cases} n_{mult\ 2D} = 2 \cdot N \cdot N^2 = 2 \cdot N^3, \\ n_{sum\ 2D} = 2 \cdot N \cdot N \cdot (N-1) = 2 \cdot N^2 \cdot (N-1), \\ n_{op\ 2D} = n_{mult\ 2D} + n_{sum\ 2D} = 4 \cdot N^3 - 2 \cdot N^2, \end{cases} \quad (21)$$

where $n_{mult\ 2D}$ – the number of multiplications, $n_{sum\ 2D}$ – the number of additions, but $n_{op\ 2D}$ – the total number of operations. It is evident that such a number of operations is practically senseless (e.g. [12]). We need close to 8 millions (8355840) operations for a gray scale picture with resolution of 128×128 pixels.

2. Number of Operations for the Fast Calculation.

As mentioned above, the sparseness of \mathbf{B}_j in (10) ensures a fast algorithm for the calculation of RABOT. The total number of mathematical operations for the calculation of fast transform (FT) by using (15) or (16) may be easily found. Matrices \mathbf{B}_j contain mainly zeros. There are only two elements per column and per row which are not equal to zero. A single elementary rotation takes four multiplications and two summations in general case. We have $N/2$ rotations per each SOGRM and we must perform these rotations on each of N columns of the transformed matrix (in the first step - on the signal/spectrum \mathbf{X}/\mathbf{Y} matrix). The mentioned operations must be repeated 2×1 times:

$$\begin{cases} n_{mult\ 2DFT} = 2 \cdot l \cdot 4 \cdot N \cdot (N/2) = 4 \cdot l \cdot N^2 \leq 4 \cdot n \cdot N^2, \\ n_{sum\ 2DFT} = 2 \cdot l \cdot 2 \cdot N \cdot (N/2) = 2 \cdot l \cdot N^2 \leq 2 \cdot n \cdot N^2, \\ n_{op\ 2DFT} = 6 \cdot l \cdot N^2 \leq 6 \cdot \log_2(N) \cdot N^2, \end{cases} \quad (22)$$

where $n_{mult\ 2DFT}$ – the number of multiplications for FT, $n_{sum\ 2DFT}$ – the number of additions for FT, but $n_{op\ 2DFT}$ – the total number of operations for FT. The improvement of performance is almost 10000 times for the example from the previous subsection (a picture with resolution of 128×128 pixels).

In the case of CORDIC implementation of rotations it is possible to reduce the number of operations to

$$n_{oprot\ 2DFT} = l \cdot N^2 \leq \log_2(N) \cdot N^2. \quad (23)$$

3. Number of Operations for the Generation of BF.

Our experience with 1-D signals shows that sometimes for the synthesis of signals are necessary only a few BFs. In

this case FT can be slower as the direct calculation. We expect that a similar situation could occur with 2-D signals too. In such a context the estimation of the number of operations needed for the calculation of standalone 2-D BF could be important. From (18) we see that $N \times N$ multiplications are necessary in the final stage (multiplication by \mathbf{F}_0). The number of multiplications increases in a geometrical progression for the sequence of terms \mathbf{B}_j when going leftward through the factorized matrices

$$\begin{aligned} n_{op\ 2DBF} &= N^2 + 4 \cdot N \sum_{k=1}^{l-1} 2^{k-1} = N \cdot (N + 2^{l+1} - 4) \leq \\ &\leq N \cdot (3 \cdot N - 4), \quad \text{for } N \geq 4. \end{aligned} \quad (24)$$

4. Comparison of the Number of Operations. The next figure shows the results obtained by formulas (22)-(24).

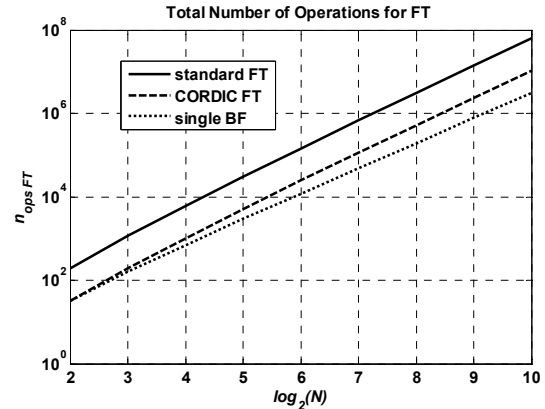


Fig. 3 The total number of operations for the fast calculation of transform and single BF

This result indicates that for the calculation of single BF we need a relatively large number of operations when comparing it with the number of operations needed for FT.

Analyzer-Synthesizer-Generator

The main practical result of this work is a MATLAB based tool called as "Analyzer-Synthesizer-Generator" (ASG). This initial tool is very useful for 2-D Phi-Function generation (BFS). ASG is also very handy for image analysis and synthesis and can be used for educational and research goals. ASG allows the import of images from different sources and displays them in the left window of GUI desktop (Fig. 4). In dependence of the state of control this window shows also spectrum image, set of BFs, synthesized image and 1D transform matrix. The right part of desktop is used for setting the angles, but the lower part is allocated for the tool control. In the recent version (look for next papers) the tool is equipped with the automatic search of optimal transform (i.e. angle matrix) for a given image, and it is targeted on the development of optimal image compression algorithms.

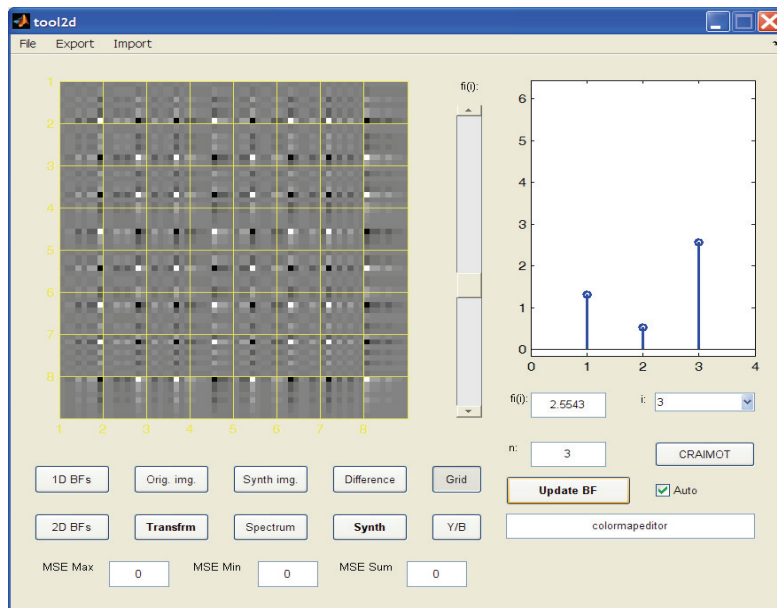


Fig. 4. The main desktop of ASG. Example of BFs for $N=8$ and some set of angles

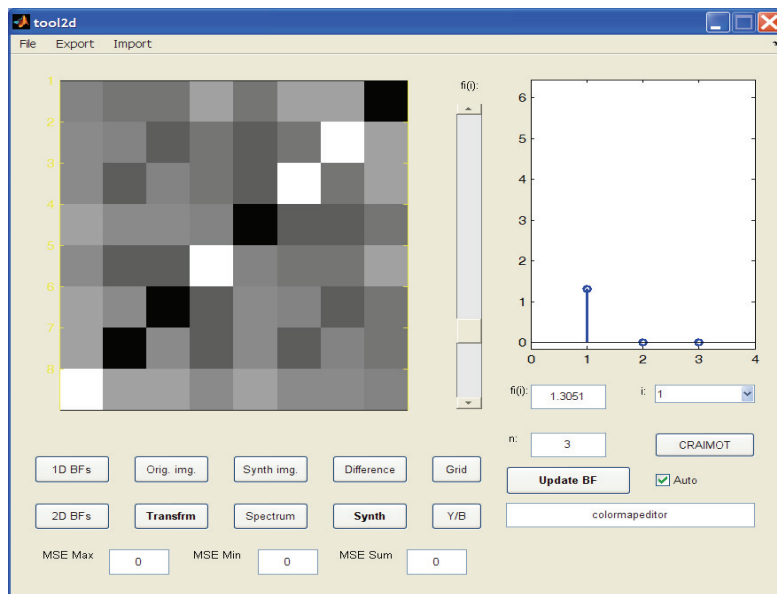


Fig. 5. The main desktop of ASG. Example of 1-D RABOT matrix (matrix values correspond to the intensity of pixels) for $N=8$ and some set of angles

Conclusions

This is an introductory work offering a compact, on rotation angles based description of existing and new classes of fast 2-D orthogonal transforms.

In our opinion the angle-based approach provides:

- ◆ unified fast algorithm for the synthesis of novel 2-D parametrical orthogonal transforms,
- ◆ new schemes of classification and comparison of existing and novel 2-D orthogonal transforms,
- ◆ new techniques for analysis and synthesis of 2-D signals – to a much wider extent the image processing possibilities than in the case of wavelets,
- ◆ promising 2-D signal compression algorithms,
- ◆ promising 2-D signal filtering algorithms.

Acknowledgments

This work has been partly supported by the project 09.1552 (the Council of Science of Latvia), by the European Social Fund doctoral scholarship support project (EU) and by the National Research Program “Innovative signal processing technologies for development of smart and efficient electronic systems” (the Ministry of Education and Science). We are also very thankful to Andra Martinsone for collaboration and help.

References

1. **Aboltins A.** Comparison of Orthogonal Transforms for OFDM Communication System // Electronics and Electrical Engineering. – Kaunas: Technologija, 2011. – No. 5(111). –

- P. 77–80.
2. **Misans P., Terauds M.** Introduction into the fast orthogonal transforms based on rotation angles – part I: A new methodical approach only or a gateway to novel DSP algorithms? // Proc. of the 5th Electronic Circuit and Systems Conference (ECS'05). – Slovakia, Bratislava, 2005. – P. 85–94.
 3. **Andrews H. C.** Multidimensional rotations in feature selection // IEEE Transactions On Computers, 1971. – Vol. 18. – No. 5. – P. 1045–1051.
 4. **Rieder P., Goetze J., Nossek J. A., Burrus C. S.** Parameterization of orthogonal wavelet transforms and their implementation // IEEE Transactions On Circuits And Systems—II: Analog And Digital Signal Processing, 1988. – Vol. 45. – No. 2. – P. 217–226.
 5. **Vaidyanathan P. P.** A unified approach to orthogonal digital filters and wave digital filters, based on LBR two–pair extraction // IEEE Transactions On Circuits And Systems, 1985. – Vol. CAS–32. – No. 7. – P. 673–686.
 6. **Terauds M.** Generation of discrete fast orthogonal transforms, Ph.D. dissertation. – Faculty of Electronics and Telecommunications, Riga Technical University, 2009.
 7. **Valters G., Misans P.** Initial version of FPGA–based CRAIMOT basis functions generator // Proc. of the 14th IEEE International Conference Mixed Design of Integrated Circuits and Systems (MIXDES 2007). – Ciechocinek, Poland, 2007. – P. 632–637.
 8. **Misans P., Terauds M., Valters G., Derums U., Vasilevskis N.** FPGA–based CRAIMOT basis function generator // Proc. of the 25th IEEE Norchip Conference. – Aalborg, Denmark, 2007.
 9. **Misans P., Valters G.** Initial version of FPGA–based CRAIMOT spectrum analyzer // Proc. of the 6th Electronic Circuits and Systems Conference (ECS'07). – Bratislava, Slovakia, 2007. – P. 159–164.
 10. **Misans P., Valters G.** Introduction into the parametrical decomposition–reconstruction filters based on Haar–like orthonormal transforms // Proc. of the 6th Electronic Circuits and Systems Conference (ECS'07). – Bratislava, Slovakia, 2007. – P. 107–112.
 11. **Misans P.** Introduction into the Haar like transforms based on rotation angles // Scientific Proc. of Riga Technical University, Telecommunications and Electronics. – Riga, RTU, 2007. – Vol. 7. – P. 6–13.
 12. **Jain A. K.** Fundamentals of digital image processing. – Englewood Cliffs, NJ, Prentice Hall, 1989. – 153 p.

Received 2011 02 15

P. Misans, U. Derums . Introduction Into the Novel Two-Dimensional Discrete Orthogonal Transforms Based on Rotation Angles // Electronics and Electrical Engineering. – Kaunas: Technologija, 2011. – No. 6(112). – P. 95–100.

A brief description of the novel two dimensional discrete orthogonal transform based on rotation angles is presented. A unified approach for the definition of novel classes of orthogonal transforms is developed. The fast algorithms for the presented transforms are given. Five different subclasses of transforms are provided. A comparison of the number of operations for the calculation of transforms is given. Some examples of sets of basis functions are presented. An initial version of a tool for image analysis and synthesis and generation of basis functions is developed. Ill. 5, bibl. 12, tabl. 1 (in English; abstracts in English and Lithuanian).

P. Misans, U. Derums. Dvimačių ortogonalinių transformacijų, sąlygojamų sukimosi kampo, tyrimas // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2011. – Nr. 6(112). – P. 95–100.

Pateikiama trumpa informacija apie dvimates ortogonalias transformacijas, sąlygojamas sukimosi kampo. Pristatytas tyrimas paremtas naujų klasių nustatymu ortogonaliosiose transformacijose. Pateiktas algoritmas. Skaičiuojant transformacijas, atliktas operacijų palyginimas. Pateikta pavyzdžių. Il. 5, bibl. 12, lent. 1 (anglų kalba; santraukos anglų ir lietuvių k.).