Design of Pipelined Signal Adaptive Architectures for Processing and Filtering of 2D Signals

Veselin N. Ivanović[1], Nevena R. Brnović[1], Srdjan Jovanovski[2]
[1] Department of Electrical Engineering
University of Montenegro
81000 Podgorica, Montenegro
{very, nevenar}@ac.me
[2] Faculty of Information Technology
Mediterranean University
81000 Podgorica, Montenegro
srdjaj@t-com.me

***Abstract*— Signal adaptive multiple-clock-cycle hardware design of the space/spatial-frequency optimal (Wiener) filter with all implementation and verification details, as well as the extensive comparative analysis has been already developed, [1]. The design allows the implemented filter to take variable (signal adaptive) number of clock cycles per a space/spatial-frequency point during the execution, resulting in the optimization of the complexity of the implemented system and in minimization of its execution time. But, completely pipelined solution could additionally improve the execution time (for a clock cycle per each space/spatial-frequency point performed within the execution). Development of the completely pipelined filtering 1D and 2D systems represents our topic in progress.**

Keywords—Overlaping in execution, Pipelining, Execution time

I. Introduction

Space/spatial-frequency (S/SF) filter that avoids distortion of an estimated nonstationary 2D FM signal is defined as [1]

$$(Hx)(\vec{n}) = \sum_{\vec{k}=-N/2+1}^{N/2} L_H(\vec{n},\vec{k}) STFT_x(\vec{n},\vec{k}) \qquad (1)$$

where $STFT_x(\vec{n},\vec{k}) = FT_{\vec{m}}[w(\vec{m})x(\vec{n}+\vec{m})]$ is the linear 2D short-time Fourier transform (2D STFT) which contains information about the estimated q-component noisy 2D signal $x(\vec{n}) = \text{sum}_{i=1,..,q}(f_i(\vec{n})) + \varepsilon(\vec{n})$, $L_H(\vec{n},\vec{k})$ is the filter's region of support (FRS), $N\times N$ is the duration of the windowed signal $w(\vec{m})x(\vec{n}+\vec{m})$, and $\vec{n} = (n_1, n_2)$. In the case of 2D FM signals $f_i(n_1,n_2)$, $i=1,...,q$, highly concentrated (in the S/SF space) around their local frequencies (LFs), and of a widely spread white noise, FRS of the optimal filter corresponds to the combination of LFs of signals $f_i(n_1,n_2)$, [1]. Then, in a single realization of noisy signal and the optimal filter case, as well as the S/SF framework, the FRS can be estimated by determining frequency-frequency (FF) points $(k_1,k_2)_i$, $i=1,...,q$, where S/SF distribution of the noisy signal has local maximum, [1, 2],

$$LF_i(\vec{n}) = \arg[\max_{\vec{k}\in Q_{\vec{k}_i}} CTFWD_x(\vec{n},\vec{k})]. \qquad (2)$$

$Q_{\vec{k}_i}$ is the basic FF region around $f_i(n_1,n_2)$, the LF of which is $LF_i(\vec{n})$, whereas the 2D cross-terms-free Wigner distribution (2D CTFWD), used in (2), has the best LF estimation characteristics among all S/SF tools [2] and is defined based on the same 2D STFTs used in (1), [3].

II. Signal Adaptive Design

The LF estimation-based optimal S/SF filter, completely developed in [1], is given in Figs. 1, 2, for a predefined maximum convolution window width L_m (used in the 2D CTFWD definition [3]) and the sliding matrix size L (determined by the LF estimation procedure [1]). Input memory is used to enable the design to import the 2D STFT input data owing to each double clock cycle (CLK), but to propagate them through the system by the signal adaptive *STFTLoad/CTFWDStore* cycle. Convolution window register block determines address order of the STFT-to-CTFWD gateway's inputs, necessary for the 2D CTFWD calculation [3] in the S/SF point corresponding to the central register block element. Convolution window operation file provides sliding over input 2D STFTs to enable calculation in different S/SF points. Sliding matrix register block creates basic FF region from (2), whereas the sliding matrix operation file provides sliding of this region over the calculated 2D CTFWDs. Implementing the LF estimation procedure [1], the COMP block (set of comparators combined with the basic logic gates) tests an LF existence in the S/SF point corresponding to the central sliding matrix register block element lying outside bordering positions, inv(*Left_Border_2+Bottom_Border_2*)=1. Based on (1) and the LF existence, it allows the corresponding 2D STFT element to participate in the $(Hx)(\vec{n})$ creation.

To respectively provide the 2D CTFWD calculation [3] and the LF estimation (in the last–Estimation–CLK), the design takes multiple, but variable (signal adaptive) number of CLKs in different S/SF points within the execution. In each S/SF point, only the first execution and Estimation CLKs have to be performed to provide the LF estimation-based S/SF filtering

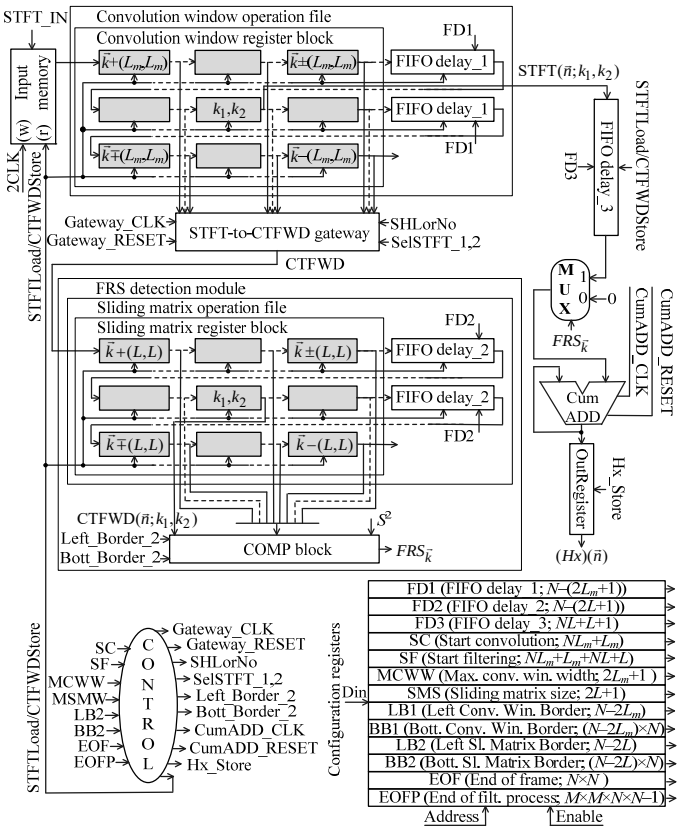

Fig. 1 *Developed design. In configuration registers, the parameters are expressed by the number of STFTLoad cycles (M×M – 2D signal size).*

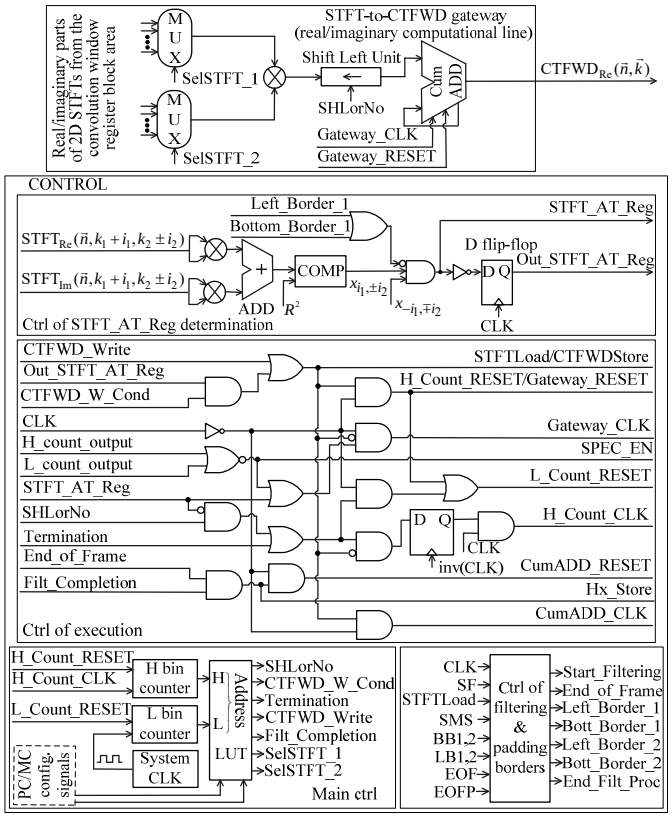

Fig. 2 *STFT-to-CTFWD gateway and CONTROL block from Fig.* 1.

related to the 2D STFT energetic form. Residual CLKs are used in the 2D CTFWD calculation, [3], and to improve quality of the LF estimation-based S/SF filtering up to the 2D CTFWD-based one, but they can be performed only in S/SF points lying inside the 2D STFT auto-terms' domains, Fig. 4. The *CTFWD_Write_Cond* signal (set in the Look-up-Table (LUT) that manages the execution, Table 1) and the *STFT_AT_Reg* one (generated, as a basic signal adaptive control signal, depending on the estimated signal shape and the noise distribution, Fig. 2, [3], to define the 2D STFT auto-terms' domains) control the CLKs execution per an S/SF point.

III. Completely Pipelined Design

To provide participation in the other control signals creation, [1], the *STFT_AT_Reg* signal is generated (through a multiplier, an adder, and a comparator, Fig. 2) in a half of a CLK. Its generation defines both the longest path of the design and the fastest CLK time T_c, $T_c/2=T_m+T_a+T_{comp}$ (T_m, T_a, T_{comp} are multiplication, addition, and comparisson times, resp.). But, operations that participate in execution within the first execution and Estimation CLKs in each S/SF point (storing and propagation through the convolution window and sliding matrix operation files, gateway reset and summation into the output CumADD) require significantly smaller time for their executions, as well as operations that create final–Completion–CLK in a signal point (n_1,n_2) (storing $(Hx)(n_1,n_2)$ into the OutRegister and the output CumADD reset). Following these principles, the further design could provide a completely pipelined execution, making essential improvement in comparison to the signal adaptive design [1]. It would provide overlaping in execution of unconditional–the Estimation and the first execution–CLKs respectively performing in adjacent S/SF points (n_1,n_2,k_1,k_2), (n_1,n_2,k_1,k_2+1), $n_1,n_2=-M/2+1,...,M/2$, $k_1,k_2=-N/2+1,...,N/2$, including bordering S/SF points $((n_1,n_2, k_1,N/2)$, $(n_1,n_2,k_1+1,-N/2+1)$; $(n_1,n_2,N/2,N/2)$, $(n_1,n_2+1,-N/2+1, -N/2+1)$; and $(n_1,M/2,N/2,N/2)$, $(n_1+1,-M/2+1,-N/2+1, -N/2+1))$. Also, the unconditional Completion CLK of a signal point (n_1,n_2), $n_1,n_2=-M/2+1,...,M/2$, would be overlapped in execution with the Estimation CLK of the bordering S/SF point $(n_1,n_2,N/2,N/2)$, but also with the first execution CLK of the next bordering S/SF point $(n_1,n_2+1,-N/2+1,-N/2+1)$. Residual CLKs could not be included in overlaping in execution, since they are not neccessarily performed in each S/SF point.

Acknowledgement. This work is supported by the Ministery of Science of Montenegro through the national scientific project "Design of the pipelined signal adaptive architectures for processing and estimation of the highly non-stationary one-dimensional and two-dimensional signals".

References

[1] V.N. Ivanović, N. Radović: "Signal adaptive hardware implementation of a System for Highly Nonstationary Two-Dimensional FM Signal Estimation," *AEUE – International Journal of Electronics and Communications*, vol. 69 (2015), pp.1854-1867.

[2] V.N. Ivanović, M. Daković, LJ. Stanković: "Performances of quadratic time-frequency distributions as instantaneous frequency estimators," *IEEE Trans. on SP*, vol.51 (1), Jan.2003, pp.77-89.

[3] V.N. Ivanović, S. Jovanovski, "Signal adaptive system for space/ spatial-frequency analysis," 2009, *EURASIP J. Advances in Signal Process.*, pp.1-15.

An Efficient Hardware Implementation for CAVLC Encoder in H.264/AVC

Milica Orlandić
Department of Electronics and Telecommunications
Norwegian University of Science and Technology, NTNU
Trondheim, Norway
orlandic@iet.ntnu.no

Abstract—**This paper presents an efficient FPGA implementation of the context-based adaptive variable length coding (CAVLC) in H.264/AVC. The proposed entropy encoder architecture includes three stages: scan stage, coding stage and bitstream packing. Its performance is optimized by eliminating or weakening data dependencies such as memory accessing and context based data, by computing codewords on-the-fly and by pipelining stages in the encoding process. The design has been synthetized and implemented on Kintex 7 FPGA board.**

Keywords - H.264/AVC, Entropy Encoding, CAVLC, FPGA

I. INTRODUCTION

Video compression is an intensive computational application involving several complex stages such as transform coding, prediction algorithms and entropy coding. Entropy encoders are characterized by serial nature, thus their parallel system implementation represents a challenge. This limits entropy encoders in existing video players to provide support for data rates of high resolution videos. In addition, real-time constraint for compression of high definition (HD) video sequences requires high-performance processors or dedicated hardware implementations, in particular in video standards characterized by large computational complexity such as H.264/AVC. The information produced in the encoder stages, such as transform coding residuals, are referred in entropy encoder as syntax elements. In H.264/AVC standard a number of entropy encoding techniques are defined for different types of syntax elements and video standard profiles. Depending on profile of H.264/AVC standard, the choice for encoding residual data consists of CAVLC and CABAC encoding modules. Entropy encoding is recognized as the bottleneck due to the inherent dependence among the syntax elements and, therefore, a number of hardware designs for CAVLC have been proposed in order to meet the high throughput requirement. The recent works on CAVLC implementations are reported in literature [1], [2], [3], [4].

In this paper, an efficient hardware implementation of the CAVLC encoder for H.264/AVC video coding standard is presented. The architecture proposes a number of optimization techniques in order to speed up the encoding process and to save the hardware area.

The paper is organized as follows. Details of the CAVLC algorithm are briefly reviewed in Section II. Section III presents the proposed CAVLC encoder architecture. In Section IV the achieved performance in terms of hardware resources is discussed. Finally, conclusions are presented in Section V.

II. CAVLC ALGORITHM

A CAVLC encoding process can be partitioned into three phases: pre-processing block scan, syntax element encoding and bitstream formation. The scan ordering tends to group significant coefficients by the proposed zig-zag order in the manner that non-zero elements are clustered around the DC coefficient. The syntax element encoding produces five syntax elements. Each syntax element contains several look-up tables for storing VLC tables. *Coeff_token* syntax element presents a pair of numbers: number of non-zero coefficients and number of trailing ones. Trailing ones parameter indicates the number of high frequency coefficients, and its range is [0-3]. Depending on the number of elements in the neighboring left and upper blocks, one among defined VLC tables for *Coeff_token* is chosen. String of trailing ones signs is the second syntax element. *Levels* is the vector of remaining non-zero coefficients within 4x4 block and its syntax element code is given by [0 . . . 0 1 *x* . . . *x s*], where the string of zeros followed by stop bit '1' is *Prefix level*, the sequence of bits after the stop bit is *Suffix Level* and *s* is the sign of the *Levels* coefficient. The level encoding is context adaptive since successive level coding depends on the magnitude of the previously coded level. The *Total_Zeros* syntax element counts the number of embedded zeros in between non-zero coefficients. The codeword *Total_Zeros* is encoded by using look-up table depending on total number of non-zero coefficients and total number of zeros. The final codeword of the encoding sequence is *Run_before*, encoded by a look-up table which depends on two derived parameters, the vector *Run* and the parameter *Zeros_par*. The vector *Run* counts zeros between each non-zero coefficients, whereas *Zeros_par* is the number of embedded zeros yet to be encoded until the last non-zero element.

III. THE PROPOSED CAVLC ARCHITECTURE

The design organization is given in Figure 1. The preprocessing stage reorganizes the coefficients within 4x4 block in the zig-zag order. In this stage the flags needed for encoding syntax elements, such as Sign flag, Non-zero coefficient flag and Ones flag, are computed. *Coeff_token*, *Total_zero* and *Run_before* codewords are extracted from look-up tables containing VLC tables. In *Level* encoding stage, the technique Arithmetic Table Expression (ATE) is employed for reducing memory utilization. It consists of detecting relations between codewords in a table and using arithmetic operation for codeword reconstruction. A packing stage concatenates

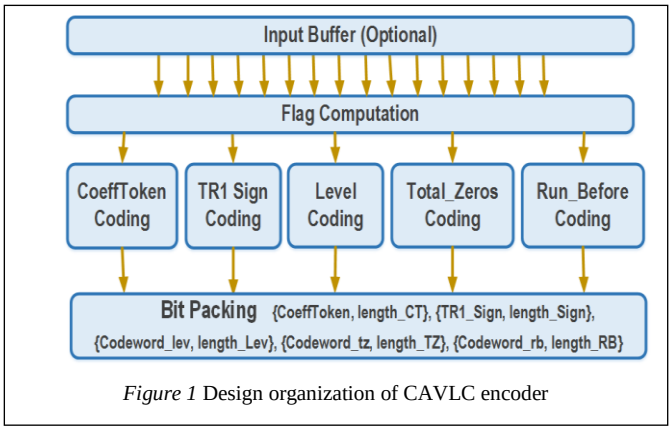

Figure 1 Design organization of CAVLC encoder

syntax element codewords within a block into a partial bitstream. Memory buffer for storing the partial bistream consists of four 32-bit words. The storing data operation includes also 32-bit alignment with unaligned data from the previous blocks. If partial bitstream for current block is larger than 32-bit chunks, these chunks are sent out. The time diagram for each phase of the proposed implementation is given in Figure 2. As presented, the critical path is lowered by performing the syntax encoding and the bit packing stages in parallel when data dependency allows.

IV. RESULTS

The proposed architecture has been modeled in VHDL and synthetized and implemented in Xilinx ISE 14.7 on Kintex 7 FPGA board. Preliminary synthesis results in terms of hardware resources utilization are reported in Table I. Operating frequency for the proposed design is 110 MHz. The data from reference syntax element encoder in MATLAB are compared with the simulation data of the proposed implementation produced in Active HDL.

TABLE I. UTILIZATION OF HARDWARE RESOURCES

Technology	FPGA, KINTEX 705-XC7K325T
LUTS	6210 (3%)
FPGA Slices	2442 (4%)
FF Utilization	1485 (1%)
RAM (kbits)	38.2

V. CONCLUSIONS

In this paper a hardware implementation of entropy encoding algorithm CAVLC in H.264/AVC video standard has been presented. The development process was focused on achieving a high throughput performance in syntax element encoding and bit packing. Future work includes implementation of the output FIFO and improved pipeline alignment among blocks in bit packing phase, further testing of bit packing, operating frequency improvement, statistical analysis of bitsream in HD video sequences and adaptation of the proposed architecture as hardware accelerator within larger SoC which includes components such as external video sources, processor and large memory for storing video data.

REFERENCES

[1] G. D. Licciardo and L. F. Albanese, "Design of a context-adaptive variable length encoder for real-time video compression on reconfigurable platforms," *Image Processing, IET*, vol. 6, pp. 301-308, 2012.

[2] N.-M. Nguyen, X.-T. Tran, P. Vivet, and S. Lesecq, "An efficient Context Adaptive Variable Length coding architecture for H. 264/AVC video encoders," in *Advanced Technologies for Communications (ATC), 2012 International Conference on*, 2012, pp. 158-164.

[3] C.-W. Chang, W.-H. Lin, H.-C. Yu, and C.-P. Fan, "A high throughput CAVLC architecture design with two-path parallel coefficients procedure for digital cinema 4K resolution H. 264/AVC encoding," in *Circuits and Systems (ISCAS), 2014 IEEE International Symposium on*, 2014, pp. 2616-2619.

[4] M. P. Hoffman, E. J. Balster, and W. F. Turri, "High-throughput CAVLC architecture for real-time H. 264 coding using reconfigurable devices," *Journal of Real-Time Image Processing*, vol. 11, pp. 75-82, 2016.

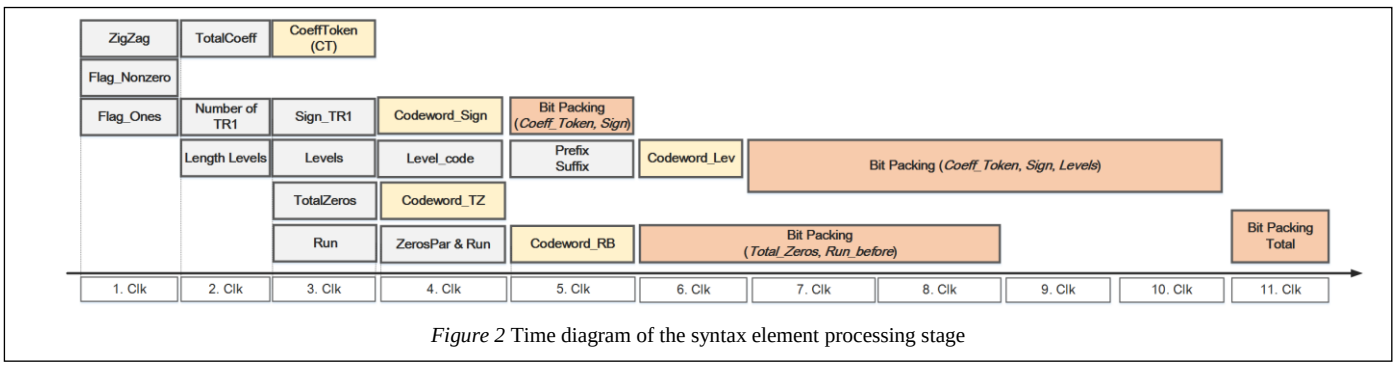

Figure 2 Time diagram of the syntax element processing stage

Discrete Wavelet Transforms of the Finite Spectrum Digital Images

Viktor P. Dvorkovich
Multimedia Technology and Telecom. Department
Moscow Institute of Physics and Technology
Moscow, Russian Federation
dvp@niircom.ru

Alexander V. Dvorkovich
Multimedia Systems and Technology Lab.
Moscow Institute of Physics and Technology
Moscow, Russian Federation
a_dvork@niircom.ru

Abstract—**A new approach for calculation of wavelet basis for multiscale wavelet decomposition of digital images is proposed. Two, three, four and five sub-band filter banks for decomposition and restoration of images are presented.**

Keywords-multiscale analysis, wavelet decomposition, FIR filters, sub-band image coding

I. INTRODUCTION

The concept of digital image multiscale analysis with finite spatial spectrum consists in a representation of signal waveform as a sequence of approximations from a coarse one to more fines on different slots of base domain [1-3].

Previously two-channel frequency decomposition has been usually implemented in signal coding – low frequency (LF) and high frequency (HF) as shown on the Fig. 1.

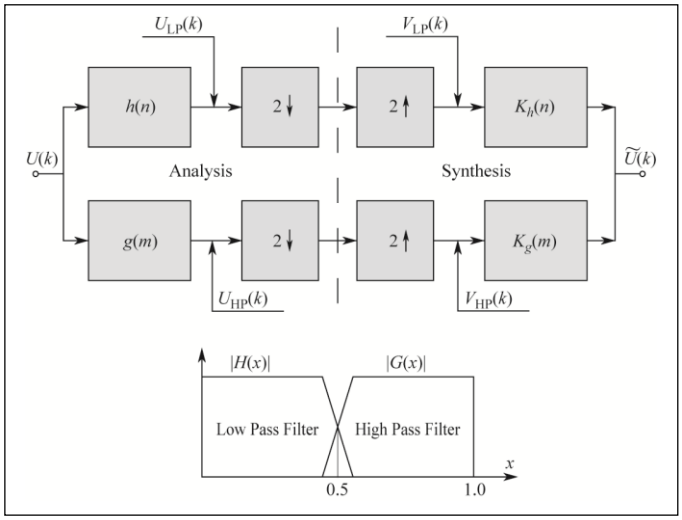

Figure 1. Block diagram of a two-band system of sub-band transform and coding.

II. TWO-BAND FILTER BANK

A new method for calculation of wavelet filter banks useful for image decomposition in presented here. Orthogonal finite impulse response (FIR) filters with either an odd number of tuples and linear (zero) phase frequency response (PFR) (Fig. 2), or with an even number of tuples (Fig. 3) are commonly applied [4].

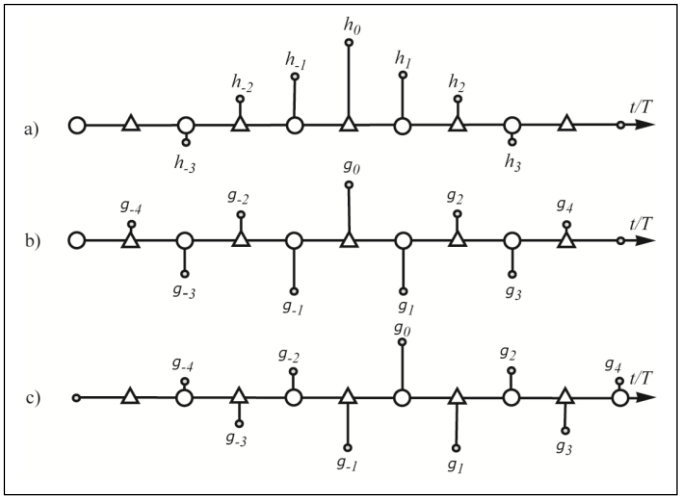

Figure 2. Pulse response of LF (a) and HF (b, c) odd tuple wavelet transform filters.

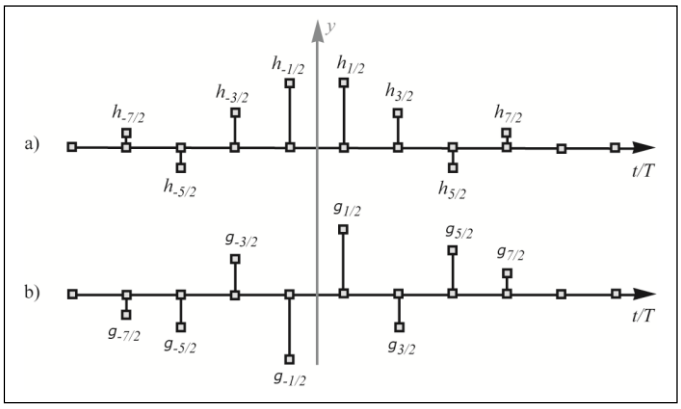

Figure 3. Pulse response of LF (a) and HF (b) even tuple wavelet transform filters.

In the first case if LF and HF wavelet filters may be represented as follows:

This work was supported by Russian Ministry of Education and Science under Grant ID RFMEFI58115X0015.

$$\begin{aligned} H(x) &= h_0 + 2\sum_{n=1}^{N} h_n \cos(\pi n x), \\ G(x) &= g_0 + 2\sum_{m=1}^{M} g_m \cos(\pi m x), \\ \bar{H}(x) &= h_0 + 2\sum_{n=1}^{N} (-1)^n h_n \cos(\pi n x), \\ \bar{G}(x) &= g_0 + 2\sum_{m=1}^{M} (-1)^m g_m \cos(\pi m x), \end{aligned} \quad 0 \le x \le 1 \qquad (1)$$

and the reverse filter forms may be derived as:

$$\begin{cases} \bar{H}(z)\cdot Kh(z) \pm \bar{G}(z)\cdot Kg(z) = 0, \\ H(z)\cdot Kh(z) + G(z)\cdot Kg(z) = 2, \end{cases} \qquad (2)$$

where using of the minus sign corresponds to co-phase placement of LF and HF filter tuples as in Fig. 2 (a, b) and using the plus sign corresponds to shifted placement of LF and HF filter tuples as in Fig. 2 (a, c).

The determinant of the equation system equals to the following constant:

$$A_0 = 2\left(h_0 g_0 + 2 \sum_{n=1}^{\min\{N,M\}} (-1)^n h_n g_n \right) \qquad (3)$$

and in this case

$$\begin{cases} H(0) = h_0 + 2\sum_{n=1}^{N} h_n = \sqrt{2}, \\ H(1) = h_0 + 2\sum_{n=1}^{N} (-1)^n h_n = 0, \\ G(0) = g_0 + 2\sum_{n=1}^{M} g_m = 0, \\ G(1) = g_0 + 2\sum_{n=1}^{M} (-1)^m g_m = \sqrt{2}. \end{cases} \qquad (4)$$

Accounting for the need for $N + M + 2$ equations to solve the system by parameters h_n, $0 \le n \le N$, and g_m, $0 \le m \le N$, it is worth to equate to zero the even moments of functions $H(x)$ and $G(x)$ – $H^{(2r)}(x)\big|_{x=0 \text{ or } x=1}$, $r = 1, 2, 3, \ldots$ and $G^{(2p)}(x)\big|_{x=0 \text{ or } x=1}$, $p = 1, 2, 3, \ldots$

Amplitude frequency responses (AFR) of some filters calculated according to the algorithm described above are illustrated on Fig. 4. These filters are identical to the filters used in JPEG2000 standard [5, 6].

In the second case (Fig. 3) frequency characteristics of respective filters may be obtained as:

$$\begin{aligned} H(x) &= 2\sum_{n=1}^{N} h_{(2n-1)/2} \cos(\pi \frac{2n-1}{2} x), \\ G(x) &= -2j\sum_{m=1}^{M} g_{(2n-1)/2} \sin(\pi \frac{2m-1}{2} x). \end{aligned} \qquad (5)$$

And the filter calculation defined as:

$$\begin{cases} \bar{H}(x)\cdot Kh(x) + \bar{G}(x)\cdot Kg(x) = 0, \\ H(x)\cdot Kh(x) + G(x)\cdot Kg(x) = 2, \end{cases} \qquad (6)$$

where $\bar{H}(x) = 2j\sum_{n=1}^{N} (-1)^n h_{(2n-1)/2} \sin(\pi \frac{2n-1}{2} x)$,

$\bar{G}(x) = -2\sum_{m=1}^{M} (-1)^n g_{(2m-1)/2} \cos(\pi \frac{2m-1}{2} x)$, $Kh(x) = \bar{G}(x)$,

$Kg(x) = \bar{H}(x)$.

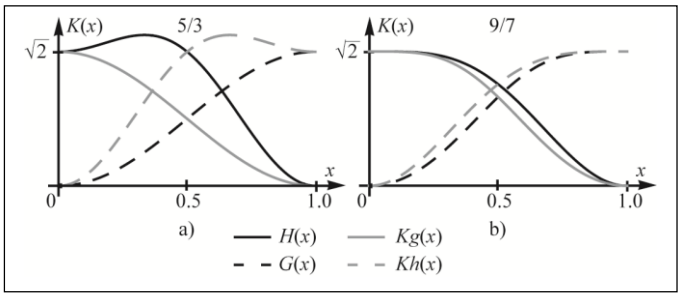

Figure 4. AFR of analysis and synthesis wavelet filters 5/3 (a) and 9/7 (b).

AFR of some filters calculated according to the second algorithm are illustrated on Fig. 5.

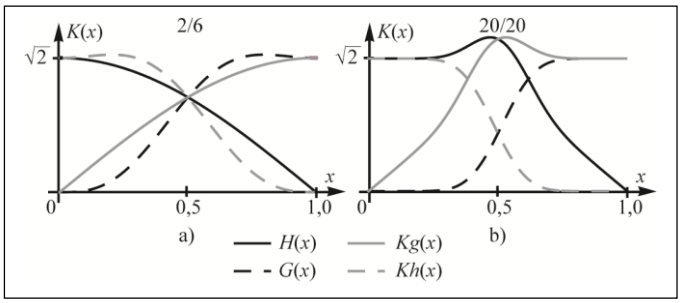

Figure 5. AFR of analysis and synthesis wavelet filters 2/6 (a) and 20/20 (b).

III. THREE-, FOUR- AND FIVE-BAND FILTER BANKS

The approach presented above may be used for calculation of multiple-band wavelet filters banks.

On the Fig. 6 a block diagram of a three-channel system of sub-band coding and decoding is shown, and Fig. 7 presents pulse responses of LF (a), MF (middle frequency) (b) and HF (c) filters.

LF, MF and HF wavelet filters may be represented as follows:

$$H(x) = h_0 + 2\sum_{n=1}^{N} h_n \cos(\pi n x),$$
$$B(x) = 2j\sum_{m=1}^{M} b_m \sin(\pi m x), \qquad (7)$$
$$G(x) = g_0 + 2\sum_{k=1}^{K} g_k \cos(\pi k x).$$

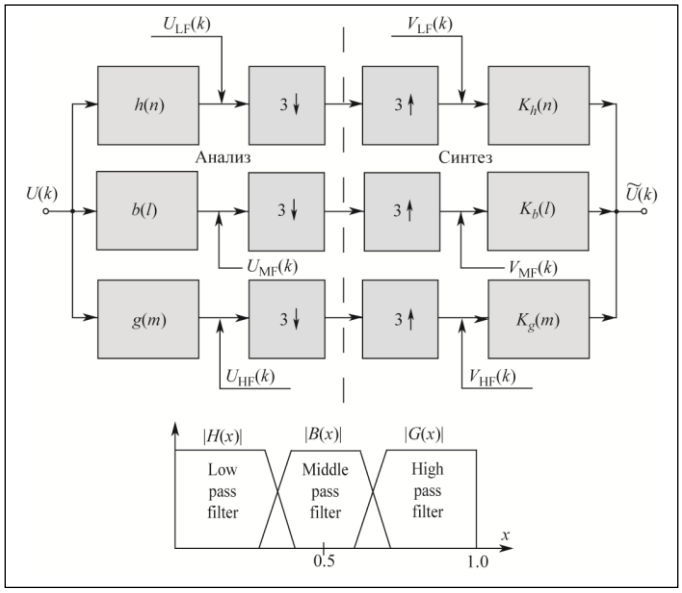

Figure 6. Block diagram of a three-band system.

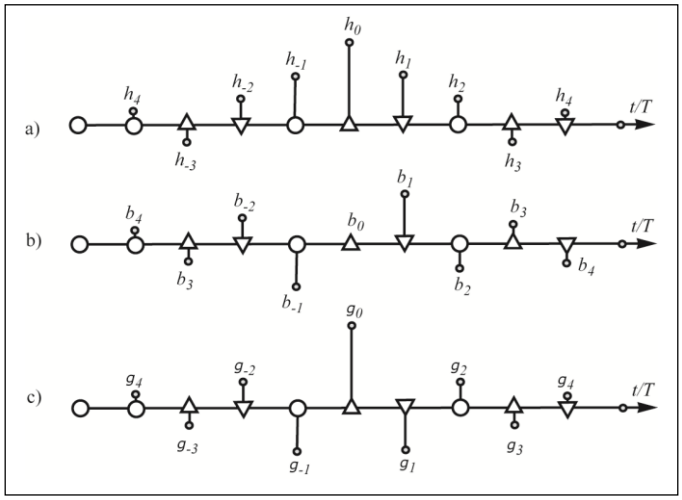

Figure 7. Pulse response of LF (a), MF (b) and HF (c) wavelet transform filters.

Reverse filters are calculated as follows:

$$\begin{cases} H_{\circ}(x)Kh(x) + B_{\circ}(x)Kb(x) + G_{\circ}(x)Kg(x) = 1, \\ H_{\nabla}(x)Kh(x) + B_{\nabla}(x)Kb(x) + G_{\nabla}(z)Kg(z) = 1, \\ H_{\Delta}(x)Kh(x) + B_{\Delta}(x)Kb(x) + G_{\Delta}(z)Kg(z) = 1. \end{cases} \qquad (8)$$

Fig. 8 shows some examples of three-band filter characteristics.

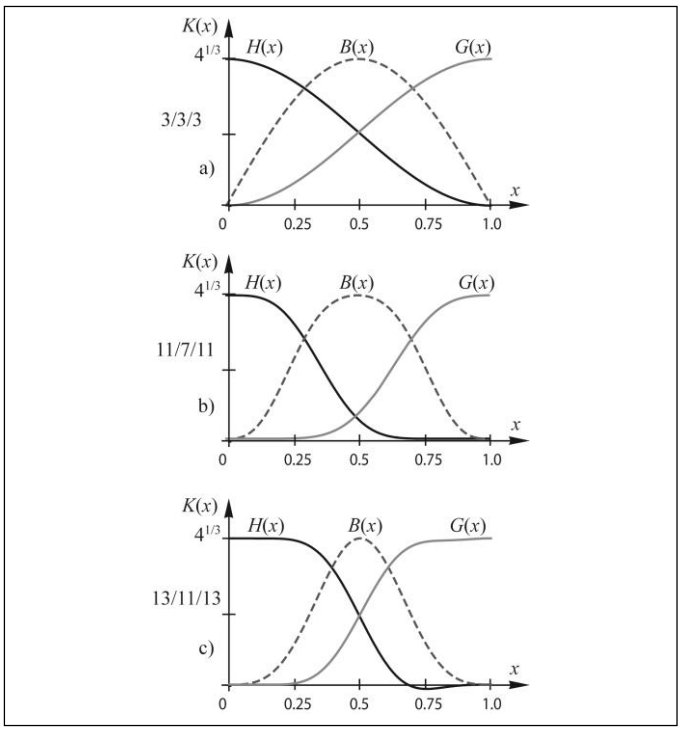

Figure 8. AFR of three-band analysis wavelet filters 3/3/3 (a), 11/7/11 (b) and 13/11/13 (c).

On the Fig. 9 a block diagram of a four-band system of sub-band coding and decoding is shown, and Fig. 10 depicts the pulse response of LF pass filter (a), lower middle frequency (LMF) pass filter (b), higher middle frequency (HMF) pass filter (c) and HF pass filter (d).

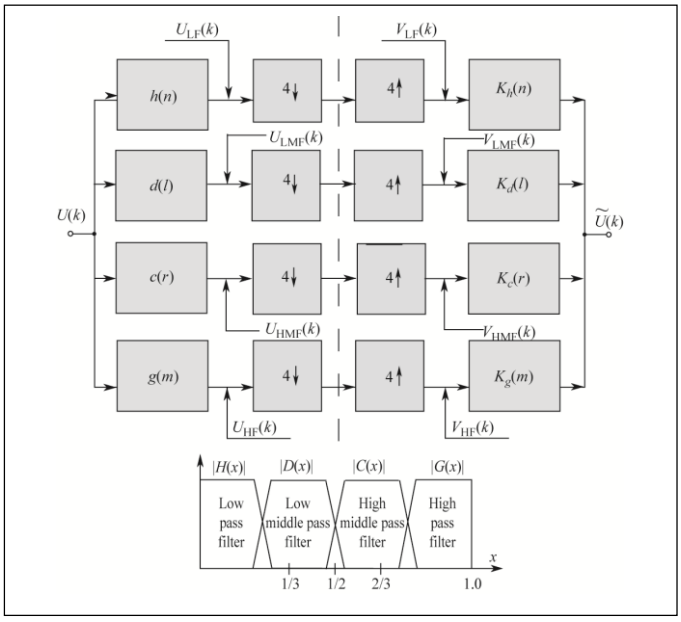

Figure 9. Four sub-band system.

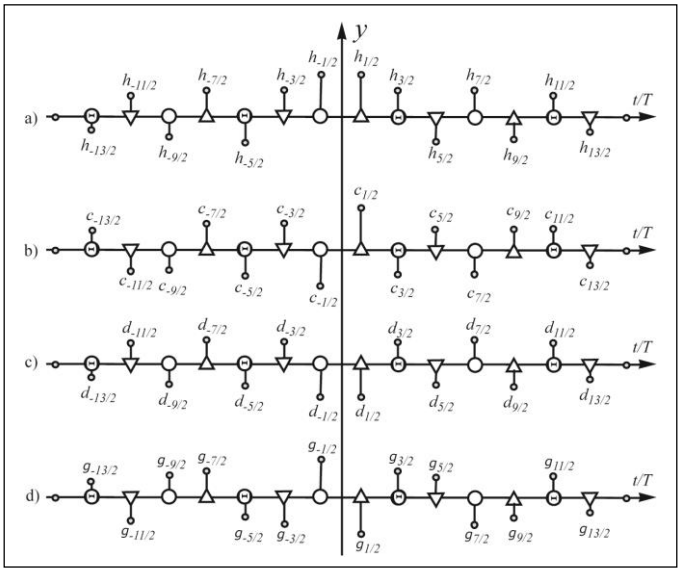

Figure 10. Pulse response of LF (a), LMF (b), HMF (c) and HF (d) wavelet transform filters.

Here we have:

$$
\begin{aligned}
H(x) &= 2\sum_{n=1}^{N} h_{n-1/2}\cos\left(\frac{2n-}{2}\pi x\right),\\
C(x) &= 2\sum_{m=1}^{M} c_{m-1/2}\cos\left(\frac{2m-}{2}\pi x\right),\\
D(x) &= 2\sum_{l=1}^{L} d_{l-1/2}\sin\left(\frac{2l-}{2}\pi x\right),\\
G(x) &= 2\sum_{k=1}^{K} g_{k-1/2}\sin\left(\frac{2k-}{2}\pi x\right).
\end{aligned}
\qquad (9)
$$

Calculation of the reverse wavelet filter system characteristics is performed using the following system:

$$
\begin{cases}
H_{\mathrm{O}}(x)Kh(x) + D_{\mathrm{O}}(x)Kd(x)\\
\quad C_{\mathrm{O}}(x)Kc(x) + G_{\mathrm{O}}(x)Kg(x) \quad 1,\\
H_{\nabla}(x)Kh(x) \quad D_{\nabla}(x)Kd(x) +\\
\quad C_{\nabla}(x)Kc(x) \quad G_{\nabla}(z)Kg(z) = 1,\\
H_{\Delta}(x)Kh(x) \quad D_{\Delta}(x)Kd(x) +\\
\quad C_{\Delta}(x)Kc(x) \quad G_{\Delta}(z)Kg(z) = 1,\\
H_{\Theta}(x)Kh(x) + D_{\Theta}(x)Kd(x)\\
\quad C_{\Theta}(x)Kc(x) + G_{\Theta}(z)Kg(z) \quad 1.
\end{cases}
\qquad (10)
$$

Fig. 11 gives examples of four-band filter characteristics.

A block diagram of a five-band system of sub-band coding and decoding is shown on the Fig. 12. Fig. 13 illustrates the pulse response of LF (a), LMF (b), MF (c), HMF (d) and HF (e) pass filters.

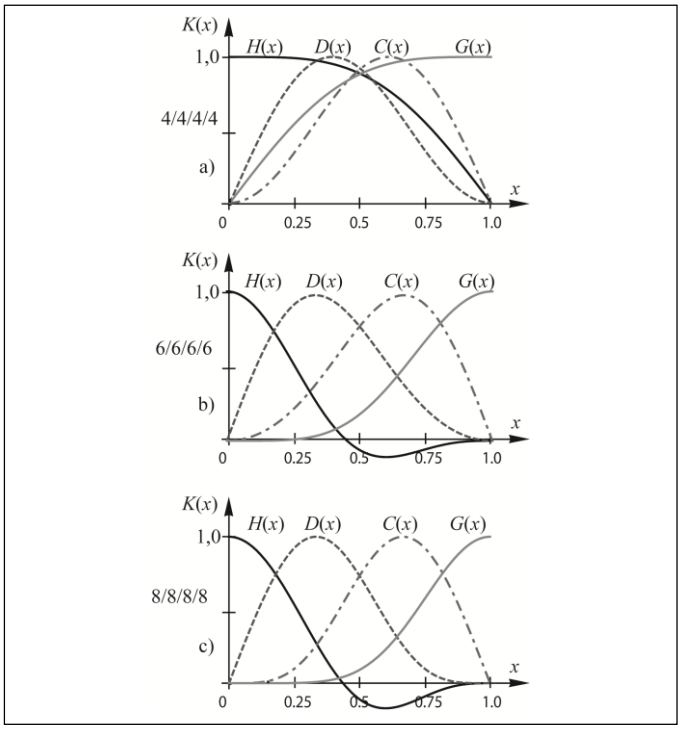

Figure 11. AFR of four-band analysis wavelet filters 4/4/4/4 (a), 6/6/6/6 (b) and 8/8/8/8 (c).

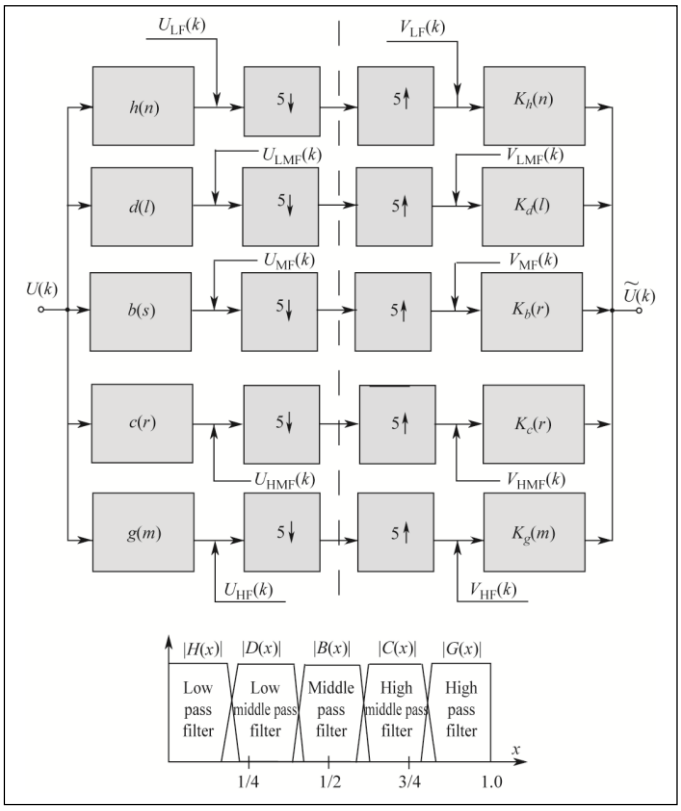

Figure 12. Five sub-band system.

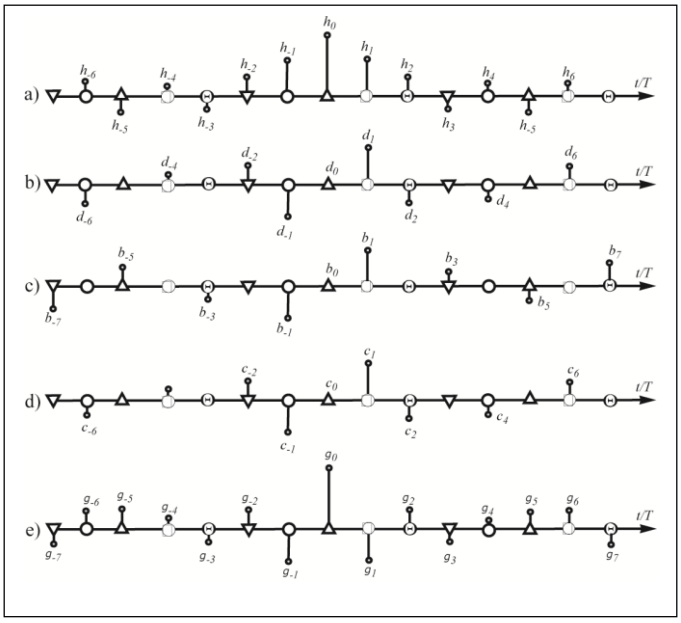

Figure 13. Pulse response of LF (a), LMF (b), MF (c), HMF (d) and HF (e) wavelet transform filters.

In this case we have:

$$
\begin{aligned}
H(x) &= h_0 + 2\sum_{n=1}^{N} h_n \cos(n\pi x),\\
D(x) &= 2j\sum_{l=1}^{L} d_l \sin(l\pi x), \quad d_0 \equiv 0,\\
B(x) &= 2j\sum_{r=1}^{R} b_r \sin(r\pi x), \quad b_0 \quad 0, \quad \text{or}\\
B(x) &= b_0 + 2\sum_{r=1}^{R} b_r \cos(r\pi x),\\
C(x) &= 2j\sum_{m=1}^{M} c_m \sin(m\pi x), \quad c_0 \quad 0,\\
G(x) &= g_0 + 2\sum_{k=1}^{K} g_k \cos(k\pi x).
\end{aligned}
\tag{11}
$$

Calculation of the reverse wavelet filter system characteristics is performed using the following system:

$$
\begin{cases}
H_{\mathrm{O}}(x)Kh(x) + D_{\mathrm{O}}(x)Kd(x) + B_{\mathrm{O}}(x)Kb(x) + \\
\quad C\ (x)Kc(x) \quad G_{\mathrm{O}}(x)Kg(x) = 1,\\
H_{\nabla}(x)Kh(x) + D_{\nabla}(x)Kd(x) + B_{\nabla}(x)Kb(x) + \\
\quad C\ (x)Kc(x) \quad G_{\nabla}(z)Kg(z) = 1,\\
H_{\Delta}(x)Kh(x) + D_{\Delta}(x)Kd(x) + B_{\Delta}(x)Kb(x) + \\
\quad C\ (x)Kc(x) \quad G_{\Delta}(z)Kg(z) = 1,\\
H_{\Theta}(x)Kh(x) + D_{\Theta}(x)Kd(x) + B_{\Theta}(x)Kb(x) + \\
\quad C\ (x)Kc(x) \quad G_{\Theta}(z)Kg(z) = 1,\\
H_{\oplus}(x)Kh(x) + D_{\oplus}(x)Kd(x) + B_{\oplus}(x)Kb(x) + \\
\quad C\ (x)Kc(x) \quad G_{\oplus}(z)Kg(z) = 1.
\end{cases}
\tag{12}
$$

Fig. 14 shows some five-band filter characteristics.

Figure 14. AFR of five-band analysis wavelet filters 5/5/3/5/5 (a), 7/5/7/5/7 (b) and 7/7/11/7/7 (c).

IV. CONCLUSION

A new method for multi-band wavelet filter bank calculation was presented. This method allows to create both standard filters (for example, filters used in JPEG2000 standard) and filters with characteristics needed for special purposes, particularly for multi-band wavelet decomposition of images.

REFERENCES

[1] I. Daubechies, “Ten Lectures on Wavelets”, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1992.

[2] Ch. Blatter, “Wavelets: A Primer”, A K Peters/CRC Press, 1999.

[3] H.-G. Stark, “Wavelets and Signal Processing: An Application-Based Introduction”, Springer, 2005.

[4] V.P. Dvorkovich, A.V. Dvorkovich, “Window Functions for Harmonic Analysis of Signals”, Moscow: Technosfera, 2014.

[5] ITU-T Recommendation T.800, “Information Technology - JPEG 2000 Image Coding System: Core Coding System”, ITU, 2015.

[6] ITU-T Recommendation T.801, “Information Technology - JPEG 2000 Image Coding System: Extensions”, ITU, 2002.

Modelling and Simulation of Accidental Air Pollutant Dispersion in Urban Areas – an Approach Suitable for Developing Countries

Živorad Kovačević, Radovan Stojanović, Gojko Nikolić
Faculty of Electrical Engineering in Podgorica
University of Montenegro,
Podgorica, Montenegro
stox@ac.me, Zika12@gmail.com,

Abstract— **In this paper an idea of using commercial, low cost software for simulation of air pollution dispersion over complex terrain is presented. The modified Gaussian equation is used as well as custom design software pre-processor, needed to prepare elevation image. The case study with terrain map of city Podgorica, Montenegro, with elevation and embedded urban area is presented. The simulation system, named Z-plume is very suitable for developing countries since request low cost and widely available, GIS software as Global Mapper and mathematical tool as MATLAB. The approach is mainly based on the research and technical results from NATO GEPSUS project [1].**

Keywords – air pollution, hazardous gases, software simulation, Gaussian equation, GEPSUS

I. INTRODUCTION

When hazardous gases are released into the atmosphere, whether accidentally or due to terrorist attacks, emergency response authorities and other responsible bodies require quick and relevant information about affected populations and infrastructure. The process is time-critical, especially in urban areas, because of population density and consequences of a delayed response.

The Great Smog of ’52, The Seveso Disaster’76, Island Nuclear Explosion’79, The Bhopal Disaster’84, The Three Mile, The Kuwait Oil Fires’91, The Chernobyl Nuclear Explosion’86, Phillips Disaster’89 are only few of the examples of man-caused disasters that resulted in uncontrolled emission of pollutants and losses in lives and properties [2]. Hence, there is a pressing need from emergency responders and other civil protection stakeholders to have access, inter alia, to support systems for modeling, simulation and visualization of hazardous gas releases through space and time. Such systems should include all available emerging technologies including various hardware and software tools.

Recently, numerous air pollutant modeling software have been developed such as ALOHA, MEMPLEX, Breeze, SAFER, SAM, and TRACE that can integrate in themselves different dispersion models like SCREEN3, AERSCREEN, AERMOD, ISC3, CALPUFF, ROADS, HISPLIT, DEGADIS, SLAB and others. However, these pollutant modeling software provide only a partial solution. They predominantly model off-line gas dispersion over simple terrain (2D space). The graphical presentation of calculated threat zones (plumes) are mainly static and do not consider real-time changes of atmosphere (weather) and source (strength, type etc.). Also, the complex terrain, elevation and urban infrastructure are not taken in consideration.

The idea of modeling gas dispersion is almost 100 years old. The main objective was to assess the spread of toxic chemicals discharged on battlefields. The purpose was extended to dispersion of hazardous gases in industrial areas. In the beginning, the calculation was done manually, using simple tables and graphs. Today, computer packages and powerful processors are employed. There are a number of different types of models for modeling gas dispersion. The selection of the appropriate one depends on the specific application, space and atmospheric conditions and problem dimension as well of the available input and output parameters and calculation speed.

Generally, the dispersion models can be divided into two groups: physical models and mathematical models. Physical models simulate real phenomena in significantly reduced conditions like in the laboratory (such as models of wind tunnels, etc). They discover mechanisms of dispersion and provide validation data obtained by mathematical models. Mathematical models are a set of analytical numerical algorithms that describe the physical and chemical aspects of the problem and can be further divided into deterministic models and statistical models. Diffusion of gas pollutants can be numerically simulated, within deterministic approach, using Eulerian, Langrangian and Gaussian models. Eulerian and Langrangian models track the movement of a large number of particles of pollutants from their initial location [3]. Eulerian reference system is fixed (relative to the ground) while Langrangian tracks the movement of the wind. Gaussian models are a combination of Eulerian and Langangian. The Gaussian model is perhaps the oldest (circa 1936) and perhaps the most commonly used model type [4]. It assumes that the air pollutant dispersion has a Gaussian distribution, meaning that the pollutant distribution has a normal probability distribution. Gaussian models are most often used for predicting the dispersion of continuous, buoyant air pollution plumes

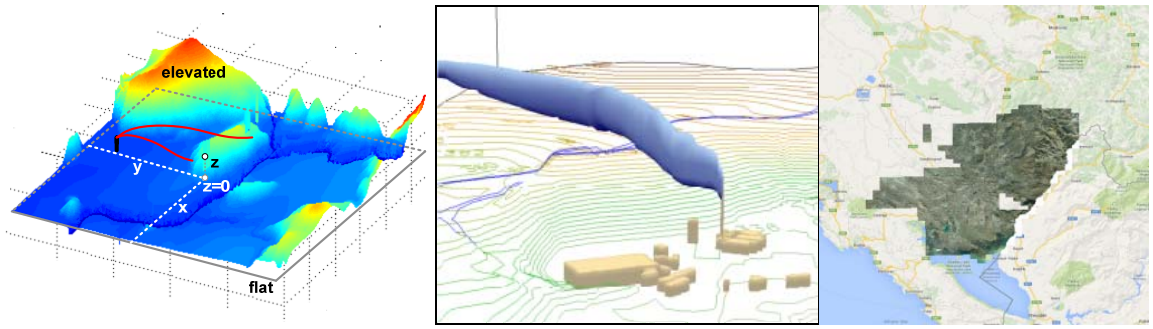

Figure 1. Complex terrain, elevation map, elevation map + urban architecture, LIDAR data

originating from ground-level or elevated sources. Gaussian models may also be used for predicting the dispersion of non-continuous air pollution plumes (called puff models).

II. Z –PLUME APPROACH

Although the derivation of the Gaussian Plume model assumes ideal conditions such as an infinite, flat, homogeneous area, the Gaussian plume model can be used to predict concentrations at receptors in complex, elevated terrain, Fig. 1 (left). Complexity should include elevation or urban infrastructure, Fig. 1 (middle panel), or both Fig. 1 (right). There are several ways to account for these effects. The models SCREEN3 and ISC3 are both similar in their approach to terrain modelling. The approach used in AERMOD is more sophisticated. In all these models, the vertical distribution function is affected. The CALPUFF model allows the mechanism for lateral deflection of the plume due to terrain. GEPSUS modified this approach from ISC3.

Generally seen the Gaussian dispersion equation can be expressed as:

$$C(x,y,z)=A\cdot V\cdot H \quad (1)$$

Where:

$$A=(Q*CORR*K)/(2\pi u\delta_y\delta_z) \quad (2)$$

$$H=e^{-\frac{y^2}{2\delta_y^2}} \quad (3)$$

$$V=\left(e^{-\frac{(z-H)^2}{2\delta_z^2}}+e^{-\frac{(z+H)^2}{2\delta_z^2}}\right) \quad (4)$$

$$CORR=(400-(z-Hef))/400 \quad (5)$$

For $z>H_{ef}$; CORR=1 for $z<=H_{ef}$;

$$Hef=z_0-(1-f_t)*(z-z_s)=H+d_h-(1-f_t)*(z-z_s) \quad (6)$$

For f_t=0.5, for stability categories A-D and f_t=0, for stability categories E, F.

and:

- Q - pollutant emission rate,
- K - units scaling coefficient,
- X - downwind distance from source centre to receptor measured along plume axe,
- Y - lateral distance from plume axe to receptor,
- Z - attitude in receptor position,
- z_s - attitude of the stack base,
- δ_y, δ_z - dispersion coefficients along x and z axis,
- H - stack high,
- d_h - plume rise,
- f_t - terrain adjustment factor.

The above terms are derived from a modified Gaussian equation taking in account complex terrain adjustments according to the ISC3 model [5,6]. As seen, V and CORR are the terms which depend on z.

III. RESULTS - EXAMPLE OF Z-PLUME APPROACH

In order to improve the Z-plume model the precise elevation of the terrain including urban area features (buildings, streets etc.) has been considered (Fig. 2), detail of city Podgorica in Montenegro. In addition to the characteristics of source, the following GIS attributes are used as a system input: Elevations for each point (Z), Projections, North-West corner of terrain area (Long, Lat), North-West corner of considered urban area (Long, Lat), Pixel Spacing-X (meters), Pixel Spacing-Y (meters), X size of terrain area (meters), Y size of terrain area (meters), X size of urban area (meters), Y size of urban area (meters).

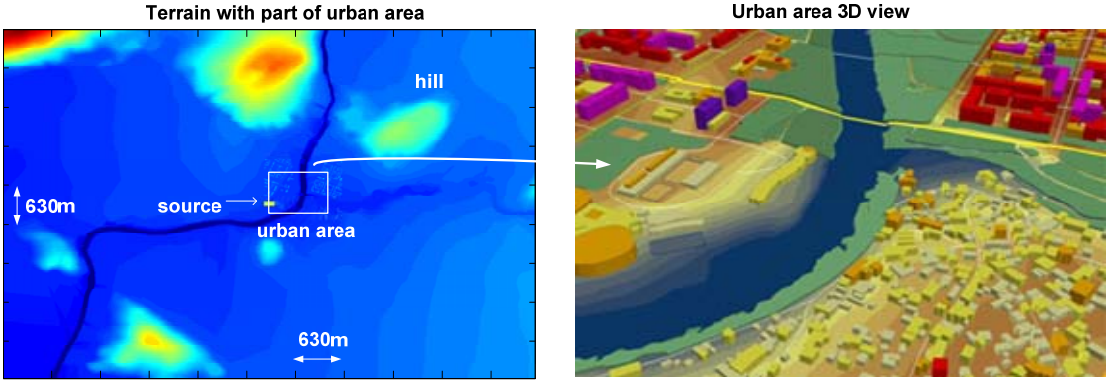

Figure 2. Terrain map of Podgorica with elevation (left) and embedded urban area (right)

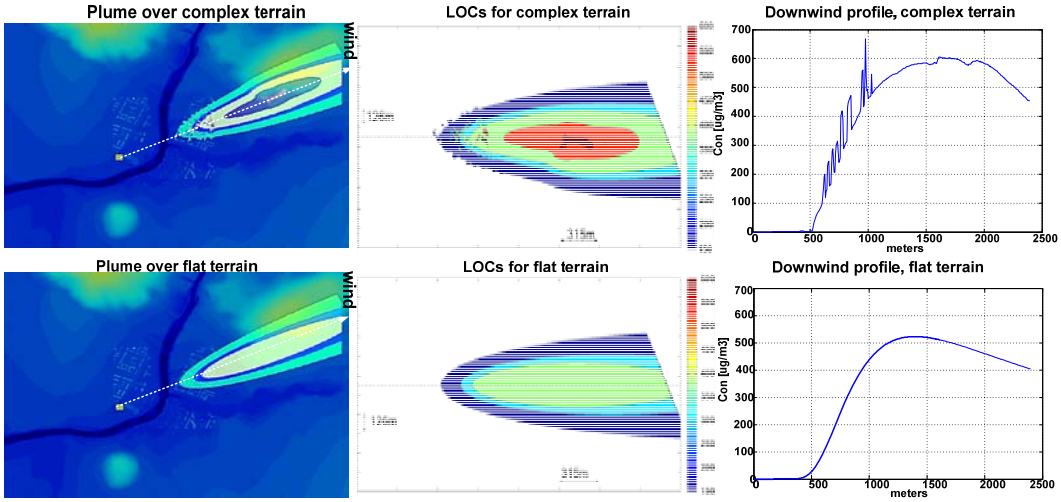

Figure 3. Comparison of air pollution dispersion in case of complex and flat terrains. Effects obtained by improved Z-Plume software version

The analysis of dispersion effects over complex and flat terrain have been performed using the Equation 1, of course, after implementing a test software in MATLAB. Fig. 2 shows some of the results. Sub-figures in the left column present different geometrical shapes of Level of Concerns (LOCs) for flat and elevated terrain. This is more clearly represented in the sub-figures, middle column. When complex terrain is considered, five levels of LOCs are detected. The LOCs for flat terrain shows only LOC1 to LOC3 (blue, light blue and light green colours). Sub-figures in the right column represent downwind profiles for complex and flat terrain.

TABLE I. NUMERICAL OUTPUT OF GEPSUS SYSTEM AS AREA FOR PARTICULAR LOC AREA

LOC	**Flat terrain, area affected** [m^2]	**Complex terrain, area affected** [m^2]
LOC1	0	11790
LOC2	0	164320
LOC3	361620	309700
LOC4	169640	141260
LOC5	354310	324550

Then the overlapped terrain is examined in term of areas in square matters affected by denoted levels of concern. The differences can be seen in Table 1 and Table 2 which show numerical output of Z-plume system for LOC1-LOC5 where the affected area in m2 is reported for flat terrain as well as complex terrain. When, in the model, flat terrain is considered, the area affected for LOC1 and LOC2 is 0 m2 while when the developed model for complex terrain is used, this area is rather significant for LOC1, 11,790 m2 and LOC2 164,320 m2. The LOC3 area when the model for flat terrain is used is 361,620 m2 and is larger than in the case where complex terrain model is used, 309,700 m2, due to undetected LOC1 and LOC2 areas. LOC4 (red) and LOC5 (cafe) also differ mainly due to different accuracy of the applied models.

TABLE II. NUMERICAL OUTPUT OF GEPSUS SYSTEM AS AREA FOR LOC WHEN ONLY URBAN AREA IS CONSIDERED

LOC	**Flat terrain, urban area affected** [m^2]	**Complex terrain, urban area affected** [m^2]
LOC1	0	516
LOC2	0	1310
LOC3	11748	20559
LOC4	19805	20321
LOC5	59217	57471

Table 2 shows the size of urban area affected. Here the urban area territory 1000 m x 1000 m is considered. In the urban area, which is of the greatest concern due to the exposed population, the area of LOC1 and LOC2 is reported as 0 m^2. When the complex terrain model is used, the area of LOC1 is given as 516 m^2 and the area of LOC2 as 1310 m^2. The area of LOC3 is also given as larger in the case of complex terrain model indicating a larger area exposed to the LOC3 levels. LOC4 and LOC5 in this case also differ mainly due to the different accuracy of the applied models.

The results of Z-plume model (LOCs) could be exported in KML format and integrated in any GIS browser, like Google Earth, using an approach described in [7]. Fig. 4 shows the example of dispersion in urban area and influence of a hill.

Figure 4. Z-plume model exported in KML and its visualization over Google Earth

IV. SOFTWARE PREPROCESSOR FOR Z-PLUME MODEL

In order to adopt Z-plume for stand-alone application, several modifications should be performed. They are related to importing images of elevation terrain using the designed software preprocessor. Fig. 5 shows the preprocessor tasks needed to prepare elevation image. The final aim was to compose elevation matrix of terrain including urban architecture in the form XYZ, where Z is elevation in meters and XY projections. For pre-processing a MATLAB based GUI is used.

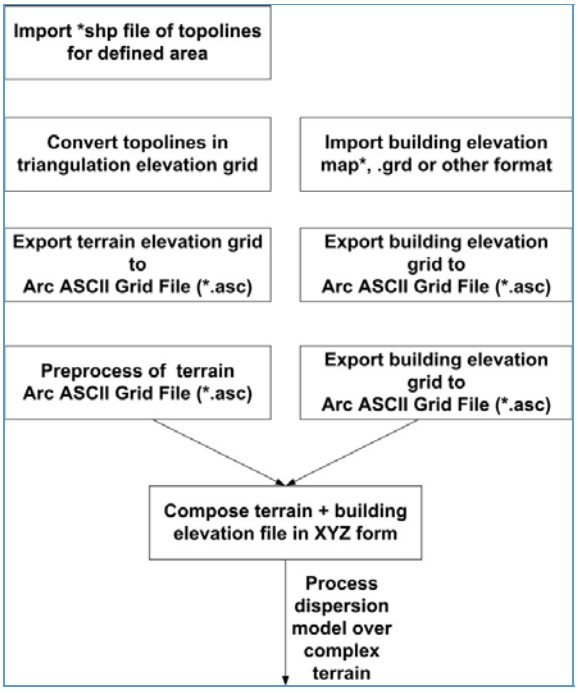

Figure 5. Preprocessing algorithm for composing complex terrain in XYZ form

V. CONCLUSIONS

The modified Gaussian equation can be used for air pollution dispersion over complex terrain are presented. Software preprocessor based on widely available low costs software can be composed to prepare elevation image. The case study with terrain map of city Podgorica, Montenegro, with elevation and embedded urban area is presented. It is obviously that such system is suitable for developing countries, like Western Balkan Region, especially Bosnia and Herzegovina with many examples of complex terrains.

REFERENCES

[1] NATO GEPSUS SPS 983510 project, accessible at www.gepsus.ac.me

[2] M. J. Assael and K. E. Kakosimos, "Fires, Explosions and Toxic Gas Dispersion –Effects Calculation and Risk Analysis", CRC Press, 2012, ISBN : 978-1-4398-2675-1.

[3] A. Daly, P. Zannetti, "Air Pollution Modeling – An Overview". Chapter 2 of Ambient Air Pollution (P. Zannetti, D. Al-Ajmi, and S. Al-Rashied, Editors). Published by The Arab School for Science and Technology (ASST) and The EnviroComp Institute, 2007

[4] M. R. Beychok, "Fundamentals of Stack Gas Dispersion", 4th Edition ed., 2005, ISBN 0-9644588-0-2.

[5] User's Guide for the Industrial Source Complex (Isc3) Dispersion Models Volume I - User Instructions, 1995, accessible at http://www.epa.gov/scram001/userg/regmod/isc3v1.pdf

[6] User's Guide for the Industrial Source Complex (Isc3) Dispersion Models Volume Ii - Description of Model Algorithms, 1995, accessible at http://www.epa.gov/scram001/userg/regmod/isc3v2.pdf

[7] R. Stojanović, A. Škraba, S. Berkowicz, R. D. Amicis, D. Elhanani, G. Conti, D. Kofjač, M. Dragović, N. Lekić and G. Nikolić, GEPSUS: Simulation-Based Decision Making System for Air Pollution Accidents, Organization - Organizacija, Journal of Management, Information Systems and Human Resources, Verista Publishing, London, Vol. 45, No. 5, September-October 2012

3D Routing and Localization for IoT Enable Medical Sensor Network

Vakhatng Mosidze
TTU, Department of Informational Technology
Tallinn, Estonia
mosidze@gmail.com

Abstract— **IoT in general terms, enables network connectivity, between smart devices at all times, everywhere, and about everything. In this context, Wireless Sensor Networks (WSNs) play an important role in increasing the ubiquity of networks with smart devices that are low-cost and easy to deploy. However, sensor nodes are restricted in terms of energy, processing and memory. Additionally, low-power radios are very sensitive to noise, interference and multipath distortions. In this context, this article proposes a routing protocol, based on 3D localization and addressing.**

Keywords – Mesh Network; 3D routing; Geographic routing; sensor network, special relativity theory,IoT routing;IQRF.

I. INTRODUCTION

Energy efficiency and communication algorithm in IoT/WSN networks needs a major shift toward to more scalable and more affordable solutions. For example, in case of applications for healthcare, patients can carry medical sensors to monitor key parameters, such as body temperature, blood pressure, ECG (electrocardiogram) and so on. Furthermore, medical centers are able to perform advanced remote monitoring to access a patient's condition. However, in this particular case, we need to use some gateway device, to allow long-range communication. This completely negates all advantages of a sensor network and leads us to use regular mobile or radio communication, as a middleware between end node and some monitoring center.

According to above, we have low powered, radio-emitting, effective devices, which are not able to provide any data in the emergency.

Problem is, that is for now, all of available IoT solution's are more internet oriented, and the is no backup connectivity in case of ER-situations.

For instance, in case of earthquake, IoT devices, can provide data stream, which can be used by researcher's to predict future disease, and also can help to emergency services to find and save trapped under the rubble people
It's also important to consider that's IoT able to monitor not only medical data or be a part of smart homes, they can provide data acquisition for an air quality, detect hazardous gas pollution and, according to [17] can be used in environment, particular landslide monitoring. However, due to a desire of simplify, and due to limitations in memory, processor and so on, IoT devices are often connected to existing infrastructures like smart phones or a Wi-Fi network.
If we will take in account that's IoT solution in most cases are originated from Sensor Networks, [7] we can to resume, that's they also have same nature and behavior, at least in radio communication domain e.q they can be interconnected (or just connected) by using 3 available topologies

- Point to Point or a bus connection
- Pont to multipoint - as a star topology
- MESH topology

The last one is the best from the perspective of creating a self-organized network and implement full fault-tolerance network coverage.

Moreover, the mesh network, is most energy effective, since only in Radio transmitting mesh, we can adaptively adjust power consumption by choosing next hope neighborhood according to Received Signal Strength Indicator (RSSI) or Link Quality Indicator (LQI) [13],[14],[15]. As an IoT is a part of Software Defined Radio (SDR), and in some cases and additional changes in hardware can represent the model of future Cognitive Radio System

Considering the many different options for routing algorithm, we can see one regularity- all the basic models are built completely true and accurate, except for the fact that they were all originally built for two-dimensional basis [3] and are adapting protocols for wired, stationary networks
For nowadays, there is no dedicated algorithm that takes into account two features: -three dimensionality and dynamic sensor network as a mobile radio system. Before moving to the question of basic model of three-dimensional network, we must to understand the existing models and identify some features of radio, which has been omitted, such as radio wave propagation [11] and wave-particle duality. During propagation, digital radio (mesh network also belong to this class) just as well will be influenced by the environment as

well as analog radio. To begin with, it should be understood that the space routing requires some coordinates of nodes. This is one of the important issues when planning the network, getting the device coordinates related primarily to its localization, and building a coordinate system.

II. STATE OF ART

As an integral part of Sensor network, IoT nodes are not aware from challenges, which are common for WSN, e.q [6]

A. Node Deployment

Unlike conventional networks where network topologies are determined in the beginning of network construction Node, deployment in WSN's is randomized. In such randomized deployment, sensor nodes are randomly scattered creating an unknown and unstable network topology. Data routing in this type of node deployment inherently possesses no prior knowledge of network topology and thus requires processing more routing data. [11]

B. Energy consumption :

Routing protocols are required to maximize the energy-conserving form of communications and computations to prolong the battery lifetime. However, these types of communications and computations still provide needed accuracy of routing protocols. The second aspect of energy concern in WSNs is to maintain the accuracy of routing protocols in a presence of low power sensor nodes.

C. Network dynamic:

Most of WSNs consist of non-stationary sensor nodes. Routing messages in this type of dynamic networks are more challenging due to quickly changing routing path. In a dynamic network, strategy for routing protocols is to simply generate routing path on demand. Due to the unsuitability of the network, pre-calculating of routing path is almost impossible.

Routing protocols for IoT can be classified in many ways, depending on different criteria.

Flat routing protocols are mainly used for networks with flat structure with a large amount of sensor nodes. Each sensor node plays equal role in the network and neighboring nodes can collaborate to gather information or perform sensing task. The large number of IoT nodes results in the impossibility of assigning global unique identifier for each node. This has led to data centric routing mechanism where the receiver node sends queries to a certain group of sensor nodes and waits for reply from the intended sensors. An example of flat routing protocols is SPIN (Sensor Protocols for Information via Negotiation) where each node considers every other nodes as potential receivers. The protocol utilizes the similar data in the neighboring nodes to avoid sending redundant data throughout the network.

Hierarchical routing protocols are designed for networks with a hierarchical structure like Internet. The idea is to divide the network into clusters and select from each cluster a cluster head. Usually, the higher energy nodes are used to process information, send data while the lower energy nodes used to sense in the proximity of the target. This type of routing protocols offers the advantages of scalability and efficient communication at the expense of the overhead of cluster formation and cluster head selection in the beginning.

Location-based routing protocols are protocols that take into consideration the specific location of sensor nodes. The location [3] can be addressed by the signal strength if nodes are close to each other. In the case of distant nodes, a relative coordinate of nodes can be extracted through information exchanged between neighboring nodes.

Currently, there are several solutions of location-based routing in three-dimensional space- the most practically and usable are:

Robotic Routing Protocol (RRP) is a Grid based recovery technique. Network is logically partitioned into cells / grids with edge length dGdR / p2 where dR is the transmission radius. In other words, all nodes in a cell are one-hop neighbors and can communicate directly. Each node in the network maintains the routing information about its neighbor cells as well as its one-hop neighbor nodes. Simple Right hand rule is performed between the cells to walk around the obstacle / empty cell. [10]

Greedy Distributed Spanning Tree (GDSTR) uses minimal path spanning trees called hull trees for void handling. Beacon-based solutions select some nodes in the network as landmark node switch help in routing around the void. [10]

Beacon Vector Routing (BVR). BVR assigns a set of randomly chosen nodes as beacon nodes. Every node in the network, learns its distance in terms of hop count to all the beacon nodes and forms a vector of these distances called the beacon vector. [10]

As we can see from above, this systems are working pseudo 3D space, by using only 2 real, depended on radio propagation service. Mostly they not were dealing with localization of each node, as they are using a principle of simple, hop count routing- respectively to algorithm, and are concentrated on void handling, which, is mandatory.

However, as clearly seen, all these technologies or engaged in reference to the coordinates of the Earth's surface (Geo-

Logical Routing (GLR) or use an algorithm to find the shortest path on a sphere with a pseudo coordinate (GDSTR). We, for our part in this work, will provide the way of creating virtual coordinates for networks and show the ability to use the key mechanisms of radio signal propagation to build an independent, networks-oriented coordinate system.
However, as this task is not easy, we need to split it into two parts, and only part one will be discussed in this paper.

- *Localization of nodes*
- *Use radio propagation values for creation of coordinate system*

III. CURRNT LOCALISATION TECHNICS

They are not so much localization methods for the moment [4]:

RSSI, ToA (Time of Arrive) and AoA (Angle of Arrive).They are related directly to the transceiver, and all change their characteristics dynamically, depending on the position in space and time. All existing localization and routing system currently uses these three parameters (as standalone or as combination of) to determine the distance and shortest path

IV. TOA OR TIME OF ARRIVAL

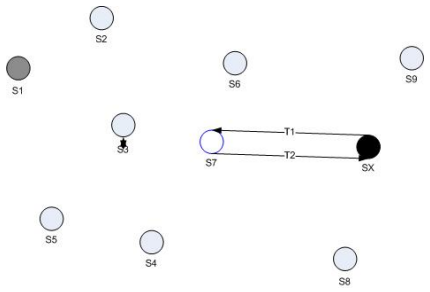

Figure 1

The very first thing that the easiest way to use- the time required to get the return signal. S_x transmitter sends a test signal, note the time and waits for the response. As soon as it receives it, according to the formula

$$D_s=c\ (T_2-T_1) \qquad (1)$$

Where D- distance, c = speed of propagation of the radio signal coincides with the speed of light, T2 and T1 time sensor 1 and 2, respectively. Thus, it is possible up to a certain point, which depends on the accuracy of real-time clock to figure out the distance from S_7 to S_x in our case.

However, if we are dealing with the real world, after checking and calculating the distance from each point to other - we get an picture.

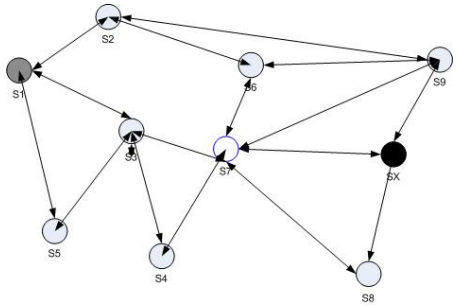

Figure 2

Table 1

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	Sx
S_1	0,0	1,1	2,2	3,3	4,4	5,5	6,6	7,7	8,8	9,9
S_2	1,1	0,0								8,8
S_3	2,2		0,0							7,7
S_4	3,3			0,0						6,6
S_5	4,4				0,0					5,5
S_6	5,5					0,0				4,4
S_7	6,6						0,0			3,3
S_8	7,7							0,0		2,2
S_9	8,8								0,0	1,1
Sx	9,9									0,0

That is, as a result of signal propagation, we have a huge amount of data (Table 1), which gives us time periods, and the distance between nodes (figure 2). It would seem that this is enough, however, to construct a route with two coordinates trivial task, since the distance to the sensor- good, time to return the signal- also good, the question in the direction of the source and destination remains open. Because we have no estimation to send and from were expected signal-, we have no information about a direction of a signal. After all, the closest and strongest signal does not mean the right direction to send data. Nevertheless, there is a simple way to get direction-trilateration.

Generally, in the field of positioning with a minimum of data, there are more than capable multilateration, which is used by all GPS system, but for a more or less accurate data requires, at least, four satellites in the existing coordinate system, fixed emitters, with very precise synchronization of time. In the future, we will look at these issues in more detail.

V. RECEIVER SIGNAL STRENGTH

The second most important characteristics and signal - is RSSI *In* IEEE 802.11 system, RSSI is the relative received signal strength in a wireless environment, in arbitrary units. RSSI is an indication of the power level being received by the antenna.

Therefore, the higher RSSI number, then stronger the signal. Means that the stronger is the signal, the closer the location of the receiver to the transmitter. Respective to the inverse-square law:

$$P = \frac{1}{R^2} \quad (2)$$

Where P-power and distance R. Multiple authors [4], [5] claim that they have achieved acceptable results in the determination of the location of node. However, in accordance with the principle of Huygens, we have a Fresnel zone, according to which each point of the medium to which comes disturbance itself becomes a source of secondary waves and the radiation field will be considered as superposition of all the secondary waves. Based on this principle can be shown that objects lying within concentric circles held around the line of sight of two transceivers, can affect the quality, both positively and negatively. All the obstacles that get inside the first circle, the first Fresnel zone, have the most negative impact.

Consider a point located on the direct path between the transmitter and the receiver, wherein the distance from the transmitter is equal to S, and the distance from the receiver equal to D, that the distance between the transmitter and the receiver S + D.-figure 3

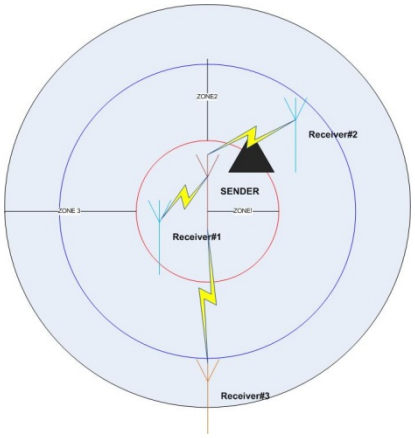

Figure 3

$$R_m = 17{,}3\left(\frac{\sqrt{1}}{f_{GHz}}\frac{S_{km}D_{km}}{S_{km}+D_{km}}\right) \quad (3)$$

Where R_m is the distance to the first Fresnel zone.
In addition, in telecommunications, a formula explains the transmission power depending on the radius a.k.a Friis formula.

$$\frac{P_r}{P_t} = G_t G_r \left(\frac{\lambda}{4\pi R}\right)^2$$

Where:
Gt - The gain of the transmitting antenna
Gr - The gain of the receiving antenna
Pt - power of the transmitting antenna (W) (without loss)
Pr - power received by the antenna (W) (without loss)
R - Distance between antennas in meters
λ - wavelength in meters corresponding to the transmission frequency
With take into account the fact that we have identical transceivers, we can calculate the radius of the transmission as:

$$R = 4\sqrt{\frac{g_1 P_1}{\pi^2 g_1 P_2 l^2}} \quad (4)$$

Knowing the radius, we can in the future, based on the coordinate data, calculate the maximum effective transmission range of each transmitter, depend to power, frequency, attitude and distance that in the future will give us the opportunity in the calculation of the optimal path. Thus, demonstrable that the method for obtaining data RSSI is the correct, nevertheless, to obtain the exact distance is impossible. However, since this characteristic allows obtaining data- it should be considered in further developments

VI. ANGE OF ARRIVAL

Angle-of-Arrival (AoA): AoA estimates the angle at which signals are received, and use simple geometric relationships to calculate node positions. Generally, AoA techniques provide more accurate localization result than RSSI based techniques but the cost of hardware of very high.

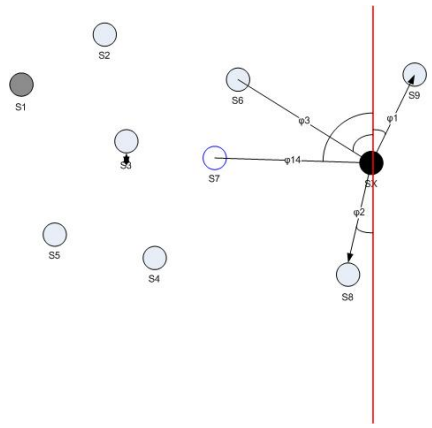

Figure 4

These method- one of the best in terms of positioning, but this method requires a fixed coordinates and sect oral antenna array, which is expensive firstly, and secondly poorly suited for a sensor network, according to the power consumption and portability.

Figure 4 demonstrates AoA work principle, as more mowers we have at directional antenna then is more probability of correct direction, and combining this with RSSI data gives us away to calculate correct location. However, without using a some coordinate basement, is useable, but using with combination with RSSI can build own coordinate structure. In real world application is often used for Radio triangulation schemes

Summarizing all of the above, we getting the following results

- *RSSI- measures stronger signal, possibility to estimate position by triangulation and directional antenna.*
- *TODA- measures distance to sensor, with trilateration and beacons can detect position of sensor node, with 3 beacons at least.*
- *AoA- measures angle and direction. Calculation to signal is possible with directional antennas, need to be bounded to some coordinates greed*

VII. BUILD ON INDEPENDED COORDINATE SYSTEM

In the construction of any route in space, you need to know at least 4variables - source point, destination point, the distance between them and direction. Itself a mathematical model of finding a particular point coordinates known since the mid-10th century, as well as the of mapping. However, in our system, several problems do not allow us to use real geographic coordinates- for example power efficiency

Taking in to account above description, received values from all existing techniques and, if we take as a basis a Cartesian coordinate system, we can represent the following characteristics for axes as:

RSSI values as =Z;
ToA values as =Y;
And
Distance (Calculated byToA) as =X

Respectively, we can build the following representative model. For now, we have a model of sensors network placed in 3-dimension space. However, if RSSI gives us values in raw mode, ToA and distance need to be calculated. Also, as we have virtual Cartesian coordinates, we have to prove that's this coordinates are not based on some material base-only in this case we can continue to build routing path in future. Because, building network for some particular scenario is not good approach. By taking a look to this network from the point of view of the postulates of the Special Theory of Relativity, namely

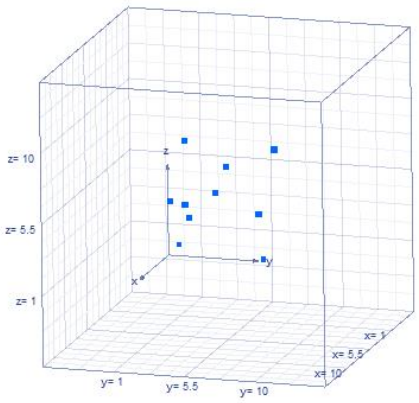

Figure 5

Reference system is a material body, chosen as the origin of this system, a method for determining the position of objects with respect to the reference system and method for measuring time. Usually distinguish between reference systems and coordinate systems. Adding measurement procedure time to the coordinate system "converts" it into a the frame of reference.

Reference systems (RS) - is a system where the object is not subject to external influences, moving uniformly in a straight line. It is postulated that there are RS and any frame of reference moving with respect to the inertial system of uniform and rectilinear, is also belong to RS.

In STR postulated the possibility of defining a single time in the framework of the inertial reference system. For this, the procedure of synchronizing two clocks at different points in IRS. Suppose the first hours from the time **t1** to the second time signal is sent (not necessarily light) at a constant speed **u**. Immediately after reaching the second clock (in their readings at time T) signal is sent back to the same constant velocity u and reaches the first clock at the time **t2**. Watches are considered *synchronized if the relation*

$$T = \frac{(t_2 + t_2)}{2} \qquad (5)$$

Time synchronization is high demand for our system, as we are going to calculate the distance between two nodes by sending and receiving signal, deviation in nanosecond is impossible, because speed of radio propagation, which is almost equal to speed of light 1 ns, means 1 m of distance.
Based on SRT time synchronization principle we can offer a different approach to building networks are as follows:

We draw attention to the basic problems of localization-accuracy; we can say that they cannot achieve acceptable performance, as it cannot accurately position a node in space. It should be noted that the system requires precise positioning of time, even in such as modern GPS system atomic clocks provide insufficient accuracy. However, in our system we do not plan to rely on the route survey points, the question arises why do we then bind your system to the standard time intervals?

Consider in more detail our system- we have the coordinates of each of the sensors, and can put them on the axes in a certain space. From the point of view of laws of geometry, physics and mathematics, we get completely coordinate system, the axes of which will be set aside, in this case, such as the magnitude and distance RSSI, ToA and distance in same way and we have a frame of reference.

As we have established a system of coordinates, and since it is not based on any geographical coordinates, we can introduce a so-called zero time calculated from as start of an initiating sensor with coordinates 0.0.0. In other words, we have to adjust to the time interval for private system, which already allows us to calculate the time intervals comfortable for us, in such manner we getting the invariant coordinate system. Let us prove that the system satisfies the postulates of SRT

1) The initiating object is at the origin.
2) All the calculated values of the distance and relative to initiator by time reference.
3) The initiator provides the starting point of time and in space..
4) Moving initiator in space does not occur changes of his coordinates and does not affect to cell units.
5) Any node, leaving the limits of the cell, and once in the range of another will be programmed in accordance with the initiator of the cell, as he is a slave and not being able to create a cell around him, if it was not provided with the topology.

Chosen the coordinates are correct and the corresponding inertial system because:

RSSI, is logarithmic value expressing strength of the received signal, thus, when moving the source around, we have values changing over time, and, given that the measurement can be carried out in a conventional timing system, and in the interior, it is possible to assume that this parameter is independent of any coordinate system, or from time to time, the value is only the distance and the measured signal in a given time interval. Thus, the coordinate is the right choice

ToA - time pinpoint from starting the transmission and to receive feedback. Does not depend on the coordinates, using only time period, which pinpoints the beginning of the transmission. Obtained that a system can operate with a timer, as well as with ordinary clock but, however, is more independent timer in our analysis. Mathematical, distance has been associated with the speed of light and ToA (1), that is a valid value in this context,

Thus, we can consider these coordinates as satisfying the conditions for the creation of an inertial IRS

Based on these data, we can create a table of coordinates of each sensor in our network

S_0=0; 0; 100
S1=0, 5; 2; 50
S3=1; 3, 3; 10

Where:

X=distance=c (t2-t1), (cm)
Y=ToA=t=t2-t1 (ns)
Z=RSSI dB (%)

By using formulas (3) and (4), and substitute values we can find that's

$$R = 4\sqrt{\left(7 * \frac{0.4}{pi^2 * 7 * (0.2) * (0.16)^2}\right)} = 7{,}95\text{m}$$

It turns out that our system is effective at a distance 7.9 m in this condition

Since we are dealing with three-dimensional model, the calculation of the distance between two points is not problematic

$$|a| = \sqrt{X^2 + Y^2 + Z^2} \quad (6)$$

If we want to calculate, the distance between two points is

$$|ab| = \sqrt{(X_2^2 - X_1^2) + (Y_2^2 - Y_2^2) + (Z_2^2 - Z_2^2)} \quad (7)$$

That is, we getting an a scalar value, which give us a possibility to calculate shortest path by using simple geometry

VIII. FUTURE WORK

Thus, the problem of finding the shortest path reduced to finding the shortest distance between two points located in the same space. Knowing all three coordinates, consider finding direction is trivial solutions of the triangle, and this work will not be considered

However, this method is optimal in the presence of only a few sensors - in the case of calculating a route in a network consisting of more than 10 sensors, the problem becomes more difficult.

Introduction to the theory of relativity in our work allows us to use the entire mathematical formalism used in this theory, and

although the use of Hamiltonian systems and entropy we produce is not required at this stage, however, the use of tensor mathematics and probability theory in ISO gives us a theoretical possibility to calculate the optimal route without resorting to energy-intensive operations

REFERENCES

[1] A. T. Parameswaran, M. I. Husain, and S. Upadhyaya, "Is RSSI a Reliable Parameter in Sensor Localization Algorithms – An Experimental Study.

[2] Q. Du, X. Sun, R. Ding, Z. Qian, and S. Wang, "TOA-Based Location Estimation Accuracy for 3D Wireless Sensor Networks," pp. 1–4, 2009.

[3] L. Mazurek, "Localization in wireless sensor networks," … Sens. Networks, 2007. IPSN 2007 …, 2007.

[4] J. Zhou, Y. Chen, B. Leong, and B. Feng, "Practical virtual coordinates for large wireless sensor networks," in Proceedings - International Conference on Network Protocols, ICNP, 2010, pp. 41–51.

[5] H. C. Chen, T. H. Lin, H. T. Kung, C. K. Lin, and Y. Gwon, "Determining RF angle of arrival using COTS antenna arrays: A field evaluation," Proc. - IEEE Mil. Commun. Conf. MILCOM, 2012.

[6] D. Project, "Localization in Wireless Sensor Networks," 2005.

[7] G. S. Kuruoglu, M. Erol, and S. Oktug, "Localization in Wireless Sensor Networks with Range Measurement Errors," 2009 Fifth Adv. Int. Conf. Telecommun., pp. 261–266, 2009.

[8] M. K. Watfa and S. Commuri, "Optimality measures for coverage in 3D wireless sensor networks," 2006 1st Int. Symp. Wirel. Pervasive Comput., 2006.

[9] V. Sulc, "IQMESH , Technology for Wireless Mesh Networks : Implementation Case Studies," no. c, pp. 97–104, 2012.

[10] B. Leong, S. Mitra, and B. Liskov, "Path vector face routing-Geographic routing with local face information"MIT Computer Science and Artificial Intelligence Laboratory 32 Vassar Street Cambridge, MA 02139, USA

[11] A. Pal, "Localization Algorithms in Wireless Sensor Networks : Current Approaches and Future Challenges," vol. 2, no. 1, pp. 45–73, 2010.

[12] D. Lindgren and H. Habberstad, "Acoustic Source Localization in a Network of Doppler Shift Sensors," pp. 1281–1288, 2013.

[13] Tan, J.; Koo, S.G.M. "A Survey of Technologies in Internet of Things", Distributed Computing in Sensor Systems (DCOSS), 2014 IEEE International Conference on, On page(s): 269 – 274

[14] R. Morello, C. De Capua, and M. Lugarà, "the Design of a Sensor Network Based on Iot Technology for Landslide Hazard Assessment," 4th Imeko TC19 Symp. Environ. Instrum. Meas. Prot. Environ. Clim. Chang. Pollut. Control, pp. 99–103, 2013.

[15] K. Machado, D. Rosário, E. Cerqueira, A. a F. Loureiro, A. Neto, and J. N. De Souza, "A routing protocol based on energy and link quality for Internet of Things applications.," Sensors (Basel)., vol. 13, no. 2, pp. 1942–64, 2013.

[16] T. Oliveira, M. Raju, and D. P. Agrawal, "Accurate Distance Estimation Using Fuzzy based combined RSSI/LQI Values in an Indoor Scenario: Experimental Verification," Netw. Protoc. Algorithms, vol. 4, no. 4, pp. 174–199, 2012.

[17] M. V Ramesh, "Real-Time Wireless Sensor Network for Landslide Detection," Sens. Technol. Appl. 2009. SENSORCOMM '09. Third Int. Conf., no. 003914, pp. 405–409, 2009.