Advances in Computer Science Research

Volume 11

Oficyna Wydawnicza Politechniki Białostockiej

Białystok 2014

ISSN 2300-715X

This journal is the continuation of *Zeszyty Naukowe Politechniki Białostockiej. Informatyka*
issued in the years 2002-2012, ISSN 1644-0331

THE ADRESS FOR THE CORRESPONDENCE / ADRES DO KORESPONDENCJI:

„*Advances in Computer Science Research*"
Faculty of Computer Science / Wydział Informatyki
Bialystok University of Technology / Politechnika Białostocka
ul. Wiejska 45a, 15-351 Białystok
tel. 85 746 90 50, fax 85 746 97 22
e-mail: znpb@irys.wi.pb.edu.pl
http://irys.wi.pb.edu.pl/acsr

Printing / Druk: Oficyna Wydawnicza Politechniki Białostockiej
Issue / Nakład: 100 copies / egzemplarzy

CONTENTS

WARPED S-TRANSFORM FOR ANALYSING THE BRAIN WAVES

Adam Borowicz

Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

Abstract: In this paper the warped S-transform is introduced as a tool for non-uniform time-frequency representation (TFR) of the brain electrical activity. The brain oscillations are classified as the five basic rhythms. The center frequencies and frequency ranges of these rhythms are non-uniformly distributed over frequency scale. Unlike the conventional S-transform the proposed technique is based on the warped discrete Fourier transform (WDFT), that allows for frequency scale warping. This can improves a spectral resolution of the TFR in particular oscillation band. In opposition to the time-domain filtering techniques, the brain rhythms can be analysed more precisely in the time-frequency plane as a full-band signal.

Keywords: WDFT, Stockwell transform, EEG

1. Introduction

The use of electro-encephalography (EEG) for registering brain activity has gained growing interest in recent years. Five simple periodic rhythms recorded in the EEG are *alpha*, *beta*, *gamma*, *delta* and *theta*. Since these rhythms are related to different brain activities they are usually analysed independently. Most commonly, the signals within the delta band ($<$4 Hz) correspond to a deep sleep, theta frequencies (4-8 Hz) are typical for dreamlike state, alpha band (8-13 Hz) signals correspond to relaxed state, beta band (13-35 Hz) is related to waking activity and gamma frequencies ($>$ 35 Hz) are characteristics for mental activities [3]. It should be noted that those are the basic rhythms only and the full classification of the brain waves includes slow ($<$1.5 Hz), fast and ultra fast waves ($>$80 Hz). An important brain wave is also *mu* rhythm that is commonly used as a control feature for brain-computer interfaces (BCIs) [8]. The *mu* oscillations (7.5–12.5 Hz) occupies the same frequency range as the *alpha* rhythm but specifically, they occur in the sensorimotor (SM) cortex.

Advances in Computer Science Research, vol. 11, pp. 5-16, 2014.

Unfortunately, the exact mechanisms of most brain oscillations are not known. Some oscillations are independently generated, but many of them are active simultaneously. Therefore it is usually assumed that the frequency content of the EEG recordings is non-stationary and multicomponent.

The discrete Fourier transform (DFT) is widely used in spectral analysis. It provides global information about the amplitude and phase spectra of the signal at each frequency. However the EEG recordings (due to its non-stationary nature) are better described by time-frequency representations (TFRs). Short-time Fourier transform (STFT) was one of the first TFR technique [16]. Its major disadvantage is inability to obtain a good frequency and time resolution of low and high frequency events at once. In order to overcome this limitation the S-transform has been proposed [19]. In opposition to the STFT, it employs a variable window length providing better time-frequency resolution. The S-transform is similar to a continuous wavelet transform (CWT) [4] but in opposition to the CWT the amplitude and phase spectra of the S-transform are directly related to the spectra of the Fourier transform. It has found applications in many fields including the EEG data analysis [15], [14].

Those methods are powerful tools in uniform spectral analysis. However the center frequencies and the frequency ranges of the popular oscillation bands are non-uniformly distributed over linear frequency scale. Some studies [12] indicate that, they rather form a geometric progression on the linear scale (and a linear progression on a natural logarithmic scale). Therefore the uniform spectral analysis of the full-band EEG signal may not be the best choice. Usually, in order to analyse particular brain wave the EEG signal is filtered using time-domain methods [11]. However in this case, some important signal features can be lost or not visible enough (in time domain). On the other hand some studies suggest [8], [10] that amplitude/phase coupling exists between two or more oscillation bands. Thus, the non-uniform TFR technique that is able to increase spectral resolution in arbitrarily selected oscillation band while the preserving the full-band information can be interesting alternative.

A typical example of the non-uniform frequency decomposition tool is warped discrete Fourier transform (WDFT) [9]. Number of the WDFT applications can be found in the literature of signal processing, including filter banks [9] and frequency estimation [5]. The WDFT was also employed in perceptual speech enhancement [13], [2].

In this article a warped S-transform is introduced as a tool for non-uniform time-frequency analysis of the brain wave EEG recordings. The proposed technique is based on the WDFT and allows for an improving frequency resolution in particular oscillation band at cost of the lower resolution in other bands. We propose to use second order allpass function to obtain a proper frequency scale warping. In oppo-

sition to the time-domain filtering methods, the warped S-transform is strictly TFR technique thus the all bands can be analysed at once on the time-frequency plane.

2. The S-Transform

The continuous S-transform [19] of the function $x(t)$ is defined as follows

$$S(\tau, f) = \int_{-\infty}^{\infty} x(t) g(\tau - t, f) e^{-i2\pi f t} dt, \tag{1}$$

where τ and f denote the time variable and Fourier frequency, respectively and $g(t, f)$ is a Gaussian window

$$g(t, f) = \frac{|f|}{\sqrt{2\pi}} e^{-(tf)^2/2}. \tag{2}$$

The frequency domain definition of the discrete S-transform, for $n = 0, 1, ..., N-1$ and $k = 1, 2, ..., N-1$ is given by

$$S\left[n\Delta, \frac{k}{N\Delta}\right] = \sum_{m=0}^{N-1} X\left[\frac{k+m}{N\Delta}\right] e^{-2\pi^2 m^2/k^2} e^{i2\pi mn/N}, \tag{3}$$

where Δ denotes sampling interval and $X\left[\frac{k}{N\Delta}\right]$ is the DFT of the N-point time series $x[m\Delta]$ (with $m = 0, 1, ..., N-1$). For $k = 0$ and any n the S-transform is simply equal to arithmetic mean of $x[m\Delta]$, i.e.

$$S[n\Delta, 0] = \frac{1}{N} \sum_{m=0}^{N-1} x[m\Delta]. \tag{4}$$

The TFR is strictly redundant, the inverse of the S-transform can be computed using inverse DFT of the time-averaged spectra, i.e.

$$x[m\Delta] = \sum_{k=0}^{N-1} \frac{1}{N} \sum_{n=0}^{N-1} S\left[n\Delta, \frac{k}{N\Delta}\right] e^{i2\pi km/N}. \tag{5}$$

There are computational advantages that come from using the frequency domain definition (3). Namely it can be implemented using the fast Fourier transform (FFT) algorithm.

3. The WDFT definition

The WDFT is not in itself the TFR technique, it transform the one-dimensional sequence of the time domain samples to one-dimensional frequency domain representation. It can be considered as a special case of non-uniform discrete Fourier transform [1]. In the case of the WDFT, the frequency samples are allocated non-uniformly but regularly over the unit circle. For the sequence $x[n]$ of the length N, it is defined by

$$\hat{X}[z_k] = X[\hat{z}_k] = \sum_{n=0}^{N-1} x[n]\hat{z}_k^{-n},\ k = 0,1,...,N-1, \tag{6}$$

with $\hat{z}_k$ being the images of allpass transformed equidistant points of the unit circle

$$z_k^{-1} = e^{-i2\pi k/N} \rightarrow \hat{z}_k = A(z_k^{-1}), \tag{7}$$

where $A(z)$ can be an arbitrary, stable allpass function. As a generalization of the DFT, the WDFT maintains some of its properties related to the linearity, symmetry and shifting [9], [5]. Unfortunately, certain important properties are lost.

The matrix representation of the WDFT is given by

$$\begin{bmatrix} \hat{X}_0 \\ \hat{X}_1 \\ \vdots \\ \hat{X}_{N-1} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & \hat{z}_0^{-1} & \dots & z_0^{-N+1} \\ 1 & \hat{z}_1^{-1} & \dots & z_1^{-N+1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \hat{z}_{N-1}^{-1} & \dots & z_{N-1}^{-N+1} \end{bmatrix}}_{\mathbf{D}} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_N - 1 \end{bmatrix}, \tag{8}$$

with $\hat{X}_k$ denoting $\hat{X}[z_k]$ and x_k denoting $x[k]$. This representation is also a basis for computation inverse transform. The matrix $\mathbf{D}$ is in fact the Vanermonde matrix and for distinct points $\{z_k\}_{k=0}^{N-1}$ its inverse is guaranteed from the theoretical point of view. Note that the elements of the WDFT matrix are no longer the roots of unity, thus the construction of fast computation algorithm like the FFT, seems impossible. The currently fastest algorithm was proposed in [9]. It exploits the factorization of the WDFT matrix into the product of the three matrices: real, the DFT (implemented via the FFT) and complex diagonal one. Its complexity is significantly reduced, but still of $O(n^2)$.

4. Warped discrete S-transform

The TFR-based version of the WDFT can be implemented using a sliding-window technique in a similar manner as the STFT. However due to fixed window length a

such technique allows only for a rough time-frequency representation and is rather dedicated to real-time processing applications. Therefore we propose a novel TFR technique that is rather based on the S-transform and exploits some WDFT features.

The discrete S-transform can be defined for points on a complex plane $\{z_k\}_{k=0}^{N-1}$ as follows

$$S[n\Delta, z_k] = \sum_{m=0}^{N-1} x[m\Delta] g_k(n\Delta - m) z_k^{-m}, \tag{9}$$

where

$$g_k(n\Delta - m) = g\left(n\Delta - m, \frac{\arg z_k}{2\pi}\right). \tag{10}$$

The warped discrete S-transform can be considered as a special case of the non-uniformly sampled continuous S-transform. Similarly to (6), it can be obtained from (9) by replacing z_k with $\hat{z}_k$, i.e.:

$$\hat{S}[n\Delta, z_k] = S[n\Delta, \hat{z}_k]. \tag{11}$$

Unfortunately we can not use the frequency domain definition similar to (3) and the FFT algorithm, thus the computational advantages are lost. Instead we propose to use a vector/matrix notation which can be more suitable for hardware implementation on some platforms. Let $\mathbf{x} = [x[0]\, x[\Delta] \ldots x[(N-1)\Delta]]^T$ be an input vector and $\hat{\mathbf{s}}_k = [\hat{S}[0, z_k]\, \hat{S}[\Delta, z_k] \ldots \hat{S}[(N-1)\Delta, z_k]]^T$ be a corresponding S-transformed vector for kth spectral bin. It can be verified that

$$\hat{\mathbf{s}}_k^T = \left(\mathbf{D}_{\{k,:\}} \otimes \mathbf{x}^T\right) \mathbf{G}_k, \tag{12}$$

where $\mathbf{D}_{\{k,:\}}$ denotes kth row of the WDFT matrix, $\otimes$ is Kronecker product and

$$\mathbf{G}_k = \begin{bmatrix} g_k(0) & g_k(1) & \ldots & g_k(N-1) \\ g_k(1) & g_k(0) & \ldots & g_k(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ g_k(N-1) & g_k(N-2) & \ldots & g_k(0) \end{bmatrix}, \tag{13}$$

is a symmetric Toeplitz matrix composed from Gaussian window coefficients. From (12) we can derive

$$\mathbf{x}^T = \left(\hat{\mathbf{s}}_k^T \mathbf{G}_k^{-1}\right) \otimes \mathbf{D}^*_{\{k,:\}},\ k > 0, \tag{14}$$

which can be viewed as the inverse transformation. This expression is rather not surprising since the time-frequency representation is redundant. Also note that for some $k << N-1$ the signal is highly averaged, thus the matrix $\mathbf{G}_k$ can be ill-conditioned.

5. Adjusting the allpass function to the brain waves

The allpass function of the warped S-transform is determined by particular application. In the case of the brain waves spectral analysis, we propose to increase frequency resolution within a specific oscillation band. It can be done by appropriate adjustment of the allpass filter parameters that results in squeezing/compressing the z-transform points on a particular section of the unit circle. Although there is some control over the behaviour of the first-order allpass filter [5], more control is afforded using the second-order allpass filter, whose the transfer function is given by [17]

$$A(z) = \frac{a_2 + a_1 z^{-1} + z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}, \tag{15}$$

where a_1, a_2 are real valued parameters. For stability reasons we assume $|a_m| < 1$ for all m. In order to ensure a proper mapping range, for $k = 0, 1, ..., N/2$, we propose to use a modified allpass function

$$\hat{z}_k = \hat{A}(z_k^{-1}) = A(z_k^{-1} e^{i\phi}) e^{i\theta}, \tag{16}$$

where $0 < \theta < \pi$ determines the location on unit circle where the frequency samples are concentrated and ϕ is a phase offset. In our discussion we assumed that N is even, for simplicity. Thus the z-transform points, for $k = N/2 + 1, ..., N - 1$, can be computed according to

$$\hat{z}_k = \hat{A}^*(z_{N-k}^{-1}). \tag{17}$$

It can be verified that (16) is in itself a complex allpass function

$$\hat{A}(z) = \frac{\alpha_2^* + \alpha_1^* z^{-1} + \alpha_0^* z^{-2}}{\alpha_0 + \alpha_1 z^{-1} + \alpha_2 z^{-2}}, \tag{18}$$

with

$$\begin{aligned} \alpha_0 &= e^{-i(\theta/2 - \phi)}, \\ \alpha_1 &= a_1 e^{-i\theta/2}, \\ \alpha_2 &= a_2 e^{-i(\theta/2 + \phi)}. \end{aligned} \tag{19}$$

Similarly to the first-order allpass function [5], the magnitude of the parameter α_2 (i.e. a_2) determines the local resolution and can be viewed as the independent variable. For example, for $a_2 = 0$, all frequency samples are placed uniformly on the unit circle, whereas for $0 < a_2 < 1$ the frequency samples are compressed around the point $e^{i\theta}$. We found it empirically that the parameter a_1 together with ϕ controls phase mapping range. In order to cover entire Nyquist frequency range (from 0 to π) we assume that

$$\begin{cases} \hat{A}(e^{i0}) = e^{i0} = 1 \\ \hat{A}(e^{i\pi}) = e^{i\pi} = -1 \end{cases} \tag{20}$$

Since the parameters a_2 and θ are given, we can solve above equation set for a_1 and ϕ. As a result we obtain

$$\phi = \arg z_c, \;\; a_1 = \frac{a_2^2 - 1}{|z_c|}, \tag{21}$$

with $z_c = a_2 e^{-i\theta} - e^{i\theta}$. Thus, the only thing we have to do is to adjust the parameters a_2 and θ, for a particular oscillation band. As mentioned before, the parameter a_2 controls the strength of the frequency scale warping, whereas the parameter θ corresponds to the warping location on the unit circle. Therefore we propose to set

$$\theta = 2\pi f_c / f_s, \tag{22}$$

and

$$a_2 = \frac{1 - \tan(\pi f_b / f_s)}{1 + \tan(\pi f_b / f_s)}, \tag{23}$$

where f_s is a sampling rate and f_c, f_b is respectively the center frequency and bandwidth of the selected oscillation band. It can be verified that (23) is inversely proportional to the bandwidth f_b, thus for narrower bands we get stronger warping. In fact the equation (23) is commonly used to adjust 3dB attenuation bandwidth of notch filters [17].

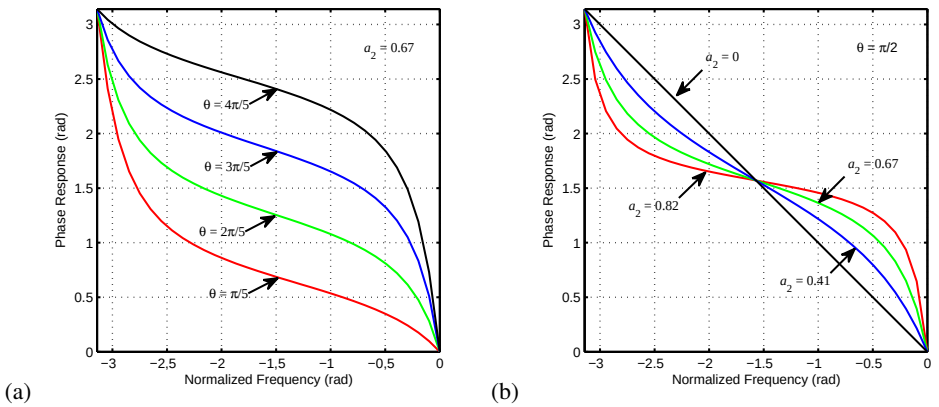


Fig. 1. Adjusting the phase responses of the modified allpass filter: (a) illustrates how the frequency warping location can be adjusted by varying a parameter θ, while (b) illustrates how the warping strength can be tuned by varying a_2.

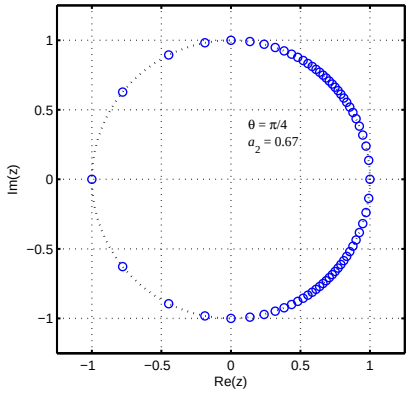


Fig. 2. Locations of the allpass-transformed points $\{\hat{z}_k\}_{k=0}^{N-1}$ on the unit circle.

In Figure 1 we depicted phase responses of the modified allpass filter (16), for different values of the parameters θ and a_2. Figure 2 presents the locations of the allpass-transformed points on the unit circle, for $\theta = \pi/4$, $a_2 = 0.67$ and $N = 64$.

The proposed TFR technique has also been verified using real EEG data. The recordings have been selected from PhysioNet database [6], [18]. They contain the brain activity related to different motor/imagery tasks (opening and closing either both fists or both feet). The EEG data were recorded at 160 Hz sampling rate from 64 electrodes placed according to the international 10-10 system [7] (excluding electrodes Nz, F9, F10, FT9, FT10, A1, A2, TP9, TP10, P9, and P10). For our purposes we selected the signal from electrode C3 (placed above SM cortex of the left hemisphere). The *mu* waves are present when SM cortex is in the idle state and they are suppressed when subject performs or imagines a motor action.

Figure 3a presents the spectrogram obtained using conventional S-transform with uniform frequency scale. The spectrograms presented in Fig. 3b and Fig. 3c have increased spectral resolution within the *beta* and *mu* band respectively. They have been both obtained using warped S-transform for parameters $f_c = 10$ Hz, $f_b = 5$ Hz (*mu* band) and $f_c = 24$ Hz, $f_b = 22$ Hz (*beta* band). As can be seen the time-frequency plane is non-uniformly stretched (warped) along the Y-axis. Note that the strongest stretch is around the center frequency of a given oscillation band. In other words the spectral resolution decreases with increasing distance from the center frequency. Thus we can analyse the selected brain wave in details while preserving some signal features of the full-band activity. For instance, in Fig. 3c although the warped

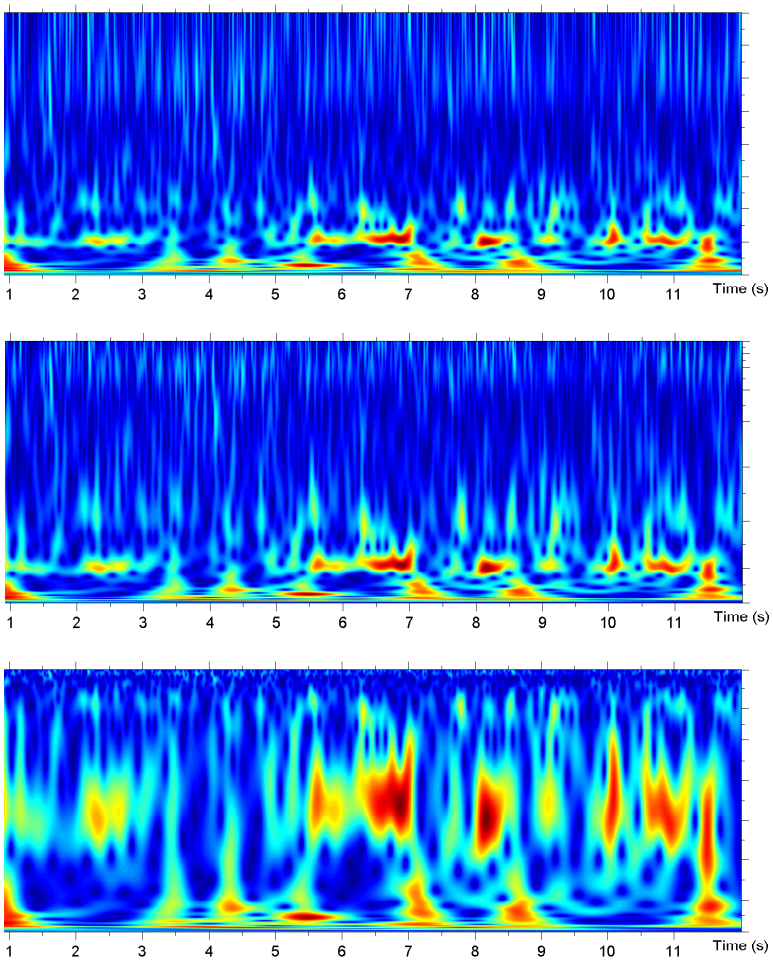


Fig. 3. Spectrograms of the example EEG signal recorded at electrode C3, obtained using: conventional S-transform (a), warped S-transform adjusted to reveal activity in *beta* (b) and warped S-transform tuned to *mu* band (c).

S-transform has been tuned to *mu* band, some activities in *beta* and lower bands are still visible.

In the case of relatively large bandwidths the deformation of the frequency scale is rather small, thus the Fig. 3b does not introduce much information comparing to Fig. 3a. Otherwise, for relatively narrow oscillation bands (i.e. *mu* rhythm) the time-frequency components are visualised in details. It should be noted that the increase in spectral resolution comes from data interpolation. In fact true frequency resolution depends only on window size. However by relocating transform points on a unit circle we can minimize spectral leak in a particular oscillation band and some spectral components can be more visible.

6. Summary

We have introduced the warped S-transform as non-uniform TFR technique for analysis of the brain waves. In fact the novel approach can be considered as a special case of the continuous S-transform with non-uniformly sampled frequency bins. In order to achieve a proper frequency scale warping we exploited the allpass function of the second order in a similar manner as in the case of the WDFT-based methods. The expressions for allpass function parameters have been provided that allows for adjusting the frequency scale warping to a particular oscillation band. It results in stretching the time-frequency plane along the Y-axis and around the center frequency of the selected brain wave. In this way, the rhythms can be better visualized on a time-frequency plane. At the same time by representing the full-band signal, we retain information about relationships between different bands.

Acknowledgment

This work was supported by the Polish National Science Centre under Decision No. DEC-2012/07/D/ST6/02454.

References

[1] S. Bagchi and S.K. Mitra. The nonuniform discrete fourier transform and its applications in filter design. i. 1-d. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Trans. on*, 43(6):422–433, Jun 1996.

[2] A. Borowicz, M. Parfieniuk, and A. Petrovsky. An application of the warped discrete fourier transform in the perceptual speech enhancement. *Speech Comm.*, 48(8):1024–1036, 2006.

[3] G. Buzsáki. *Rhythms of the Brain*. Oxford University Press, Inc., New York 2006.

[4] I. Daubechies. The wavelet transform, time–frequency localization and signal analysis. *IEEE Trans. Inf. Theory*, 36:961–1005, 1990.

[5] S. Franz, S.K. Mitra, and G. Doblinger. Frequency estimation using warped discrete fourier transform. *Signal Process.*, 83(8):1661–1671, Aug 2003.

[6] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000 (June 13).

[7] V. Jurcak, D. Tsuzuki, and I. Dan. 10/20, 10/10, and 10/5 systems revisited: Their validity as relative head-surface-based positioning systems. *NeuroImage*, 34(4):1600 – 1611, 2007.

[8] D.J. Krusienski, G. Schalk, D.J. McFarland, and J.R. Wolpaw. A μ-rhythm matched filter for continuous control of a brain-computer interface. *IEEE Trans. Biomed. Eng.*, 54(2):273–280, 2007.

[9] A. Makur and S.K. Mitra. Warped discrete-fourier transform: Theory and applications. *IEEE Trans. Circuits Systems I*, 48(9):1086–1093, 2001.

[10] L.F. Márton, L. Bakó, S.T. Brassai, and L. Losonczi. Multichannel {EEG} signal recording analysis based on cross frequency coupling method. *Procedia Technology*, 12(0):133 – 140, 2014.

[11] J.B. Nitschke, G.A. Miller, and E.W. Cook. Digital filtering in eeg/erp analysis: Some technical and empirical comparisons. *Behavior Research Methods, Instruments, & Computers*, 30(1):4–67, 1998.

[12] M. Penttonen and G. Buzsaki. Natural logarithmic relationship between brain oscillators. *Thalamus and Related Systems*, 2(2):145–152, 2003.

[13] A. Petrovsky, M. Parfieniuk, and A. Borowicz. Warped DFT based perceptual noise reduction system. In *Proc. AES 116th*, Berlin, Germany, May 2004, 14 p.

[14] C.R. Pinnegar, H. Khosravani, and P. Federico. Frequency phase analysis of ictal eeg recordings with the s-transform. *Biomedical Engineering, IEEE Trans. on*, 56(11):2583–2593, Nov 2009.

[15] Kok-Kiong Poh and P. Marziliano. Analysis of neonatal eeg signals using stockwell transform. In *Engineering in Medicine and Biology Society, 2007. 29th Annual International Conference of the IEEE*, pages 594–597, Aug 2007.

[16] M. Portnoff. Time-frequency representation of digital signals and systems based on short-time fourier analysis. *IEEE Trans. Acoust. Speech Signal Process.*, 28(1):55–69, 1980.

[17] P.A Regalia, S.K. Mitra, and P.P. Vaidyanathan. The digital all-pass filter: a versatile signal processing building block. *Proceedings of the IEEE*, 76(1):19–37, Jan 1988.

[18] G. Schalk, D.J. McFarland, T. Hinterberger, N. Birbaumer, and J.R. Wolpaw. Bci2000: A general-purpose brain-computer interface (bci) system. *IEEE Trans. on Biomedical Engineering*, 51(6):1034–1043, 2004.

[19] R. G. Stockwell, L. Mansinha, and R. P. Lowe. Localization of the complex spectrum: the s transform. *Sig. Process., IEEE Trans. on*, 44(4):998–1001, Apr 1996.

SPACZONA TRANSFORMATA S DO ANALIZY FAL MÓZGOWYCH

Streszczenie: W artykule wprowadzamy spaczoną transformatę S, jako narzędzie nierównomiernej reprezentacji czasowo-częstotliwościowej aktywności elektrycznej mózgu. Oscylacje mózgowe klasyfikowane są, jako pięć podstawowych rytmów. Częstotliwości środkowe oraz zakresy odpowiadające tym rytmom rozmieszczone są nierównomiernie na skali częstotliwości. Proponowana technika, w przeciwieństwie do konwencjonalnej transformaty S, opiera się na spaczonej dyskretnej transformacie Fouriera, która pozwala na deformowanie skali częstotliwości. Umożliwia to zwiększenie rozdzielczości widmowej reprezentacji czasowo-częstotliwościowej w określonym paśmie oscylacji. W odróżnieniu od klasycznych metod filtracji dziedziny czasu, rytmy mózgowe mogą być dokładniej analizowane w płaszczyźnie czasowo-częstotliwościowej, jako sygnał pełno-pasmowy.

Słowa kluczowe: WDFT, transformata S, EEG

Artykuł zrealizowano w ramach grantu badawczego Narodowego Centrum Nauki nr DEC-2012/07/D/ST6/02454.

A ROBUST GENERALIZED SIDELOBE CANCELLER EMPLOYING SPEECH LEAKAGE MASKING

Adam Borowicz

Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

Abstract: A novel speech enhancement method based on generalized sidelobe canceller (GSC) structure is presented. We show that it is possible to reduce audible speech distortions and preserve residual noise level under acoustic model uncertainties. It can be done by constraining a speech leakage power according to masking phenomena and conditional minimizing the residual noise power. We implemented the proposed approach using a simple delay-and-sum beamformer model. Finally a comparative evaluation of the selected methods is performed using objective speech quality measures. The results show that the novel method outperforms conventional one providing lower speech distortions.

Keywords: GSC, psychoacoustics, speech enhancement

1. Introduction

A major objective of the speech enhancement is to reduce environmental noise while preserving speech intelligibility. In a context of the multichannel methods the dereverberation and interference suppression is also expected. The most commonly used dereverbaration methods are beamforming techniques [2]. The key idea of the beamforming is to process the microphone array signals to listen the sounds coming from only one direction. Particularly the noise reduction can be implicitly achieved by avoiding noise directions. The linearly constrained minimum variance (LCMV) algorithm has been originally proposed by Frost [4] and it is probably the most studied beamforming technique since then. It minimizes beamformer output variance subject to the set of linear equations that ensure a constant gain in a specified listening direction. The minimum variance distortion-less (MVDR) method [11] can be considered as a special case of the LCVM approach. Another popular technique is generalized sidelobe canceller [5] [12]. The noisy signal domain is split into two orthogonal subspaces where the dereverberation and noise suppression can be performed separately.

Advances in Computer Science Research, vol. 11, pp. 17-29, 2014.

In order to work reasonably well in the reverberant environments, classical beamforming techniques often require a system model identification i.e. knowledge of the acoustic room impulse responses or its relative ratios. These parameters can be fixed or estimated adaptively, however in general it is a difficult task. In addition the beamforming methods are usually very sensitive to the model uncertainties. Recently, much efforts have been made to reformulate the multichannel speech enhancement problem so that the noise reduction can be achieved without performing speech dereverberation [7]. However these methods are out of scope of this article.

The proposed system is based on the GSC beamformer. We directly assume the presence of the system model uncertainties, which results in the estimation errors (speech leakage effect) and thus increased speech distortions. Instead to minimize these errors we propose to use perceptual properties of the auditory system (simultaneous masking phenomena) to make the speech distortions inaudible. In particular, it is observed that for a given spectral power level, there is a masking threshold so that any interferer below this threshold becomes unnoticed. A similar strategy has been proved to be useful in several single channel methods [3] but according our best knowledge it was not used in a field of the multichannel speech enhancement.

2. Notation

Consider an array of N microphones with arbitrary geometry and single speech source $s(t)$ located inside reverberant enclosure. The observation signal at nth microphone is given by:

$$x_n(t) = a_n(t) * s(t) + v_n(t) = y_n(t) + v_n(t), \tag{1}$$

where $*$ denote a convolution operator, a_n is a room acoustic impulse response from the source speech signal to the nth microphone and $y_n(t)$, $v_n(t)$ are the clean speech and noise components received at nth microphone.

The multichannel systems are often implemented in the frequency-domain using the discrete Fourier transform (DFT). The samples are processed on frame-by-frame basis using analysis window of the length M. Let $X_n(\omega)$, $A_n(\omega)$, $S(\omega)$, $Y_n(\omega)$ and $V_n(\omega)$ denote the DFTs of $x_n(t)$, $a_n(t)$, $s(t)$, $y_n(t)$ and $v_n(t)$ respectively. For sufficiently large M (compared to the length of the room impulse response), we can approximate the model (1) as follows [5]:

$$\mathbf{x}(\omega) = \mathbf{a}(\omega)S(\omega) + \mathbf{v}(\omega) = \mathbf{y}(\omega) + \mathbf{v}(\omega), \tag{2}$$

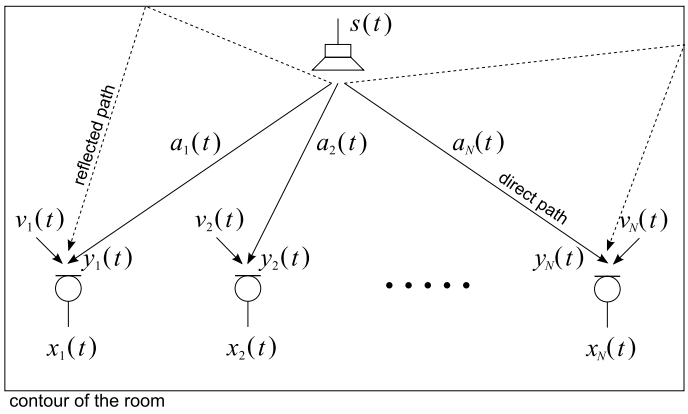


Fig. 1. Graphic illustration of the multimicrophone signal model (1).

where

$$\begin{aligned}
\mathbf{x}(\omega) &= [X_1(\omega), X_2(\omega), \ldots, X_N(\omega)]^T, \\
\mathbf{a}(\omega) &= [A_1(\omega), A_2(\omega), \ldots, A_N(\omega)]^T, \\
\mathbf{y}(\omega) &= [Y_1(\omega), Y_2(\omega), \ldots, Y_N(\omega)]^T, \\
\mathbf{v}(\omega) &= [V_1(\omega), V_2(\omega), \ldots, V_N(\omega)]^T.
\end{aligned} \tag{3}$$

For example, a correlation matrix for an arbitrary vector $\mathbf{z}(\omega)$ is defined as: $\mathbf{R_{zz}}(\omega) = E\{\mathbf{z}(\omega)\mathbf{z}^H(\omega)\}$, where $E\{.\}$ is an expectation operator and superscript H denotes conjugate transpose. We assumed that the speech and noise processes are wide-sense stationary and uncorrelated, i.e.: $\mathbf{R_{xx}}(\omega) = \mathbf{R_{yy}}(\omega) + \mathbf{R_{vv}}(\omega)$.

3. Speech enhancement: GSC and speech leakage masking

The signal model that we use here (1) can also be presented in the graphical form (Fig. 1). In the case of the frequency domain implementation, our aim is to estimate complex spectrum of the source speech signal, i.e. $S(\omega)$ (then the signal $s(t)$ is obtained from $S(\omega)$ via inverse DFT). The most straightforward way is to apply a linear filter $\mathbf{h}(\omega)$ to observation vector $\mathbf{x}(\omega)$ for each frequency bin:

$$\hat{Y}(\omega) = \mathbf{h}^H(\omega)\mathbf{x}(\omega). \tag{4}$$

Above formula can be viewed as the frequency domain implementation of the finite-impulse-response (FIR) filter. The derivation of the optimal filter $\mathbf{h}(\omega)$ depends on some criteria which we will investigate in the next subsections.

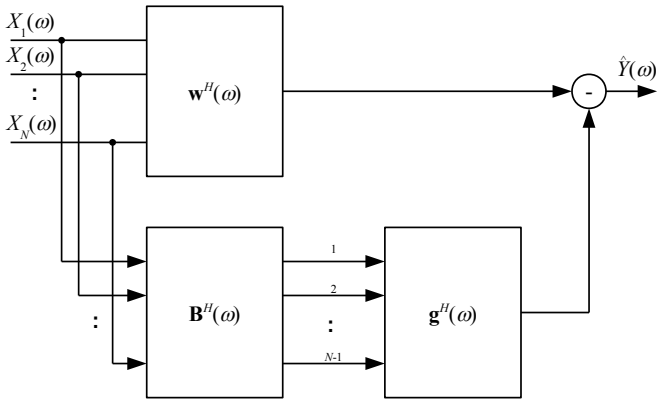


Fig. 2. Block diagram of the GSC beamformer.

3.1 Generalized sidelobe caceller

The GSC approach assumes that the filtering for each channel can be performed in two orthogonal subspaces. It can be expressed mathematically using decomposition of the weighting vector:

$$\mathbf{h}(\omega) = \mathbf{w}(\omega) - \mathbf{B}(\omega)\mathbf{g}(\omega) \tag{5}$$

where $\mathbf{w}(\omega)$ is a steering vector of size N, and $\mathbf{B}(\omega)$ is a blocking matrix of size $N \times (N-1)$ that spans the null space of $\mathbf{A}(\omega)$. The corresponding block diagram of the GSC beamformer is depicted in Fig. 2.

The objective of the GSC approach is to find optimal noise cancellation vector $\mathbf{g}(\omega)$ of size $N-1$. It can be done by solving the following (unconstrained) optimization problem:

$$\min_{\mathbf{g}(\omega)} E\{|\mathbf{w}^H(\omega)\mathbf{v}(\omega) - \mathbf{g}^H(\omega)\mathbf{B}^H(\omega)\mathbf{v}(\omega)|^2\}. \tag{6}$$

Note that this is equivalent to minimizing average residual noise power at the GSC output. An explicit solution for (6) is multichannel Wiener filter [5]:

$$\mathbf{g}_{\mathrm{W}}(\omega) = [\mathbf{B}^H(\omega)\mathbf{R}_{\mathbf{vv}}(\omega)\mathbf{B}(\omega)]^{-1}\mathbf{B}^H(\omega)\mathbf{R}_{\mathbf{v}}(\omega)\mathbf{w}(\omega). \tag{7}$$

Although the GSC and the LCMV beamformers are equivalent, the GSC approach have some interesting interpretation. Note, that the objective of the first vector $\mathbf{w}(\omega)$ is to perform dereverberation on the signal $\mathbf{x}(\omega)$, while the objective of the second

component $\mathbf{B}(\omega)\mathbf{g}(\omega)$ is to suppress the interferences and additive noise. It is worthwhile to note that computationally efficient, adaptive implementations are preferred [5]. However in our experiments we use non-recursive implementation for simplicity.

3.2 Speech leakage constrained method

A major drawback of the GSC beamformer is a high sensitivity to model uncertainties. In fact the performance of the GSC method is affected by the accuracy of the steering vector and blocking matrix estimates. Unfortunately these parameters depend on true channel transfer functions which are usually unknown. Although they can be roughly estimated using second-order statistics [5], [11], in general it is a difficult task. Similarly assuming a simpler acoustic model, can also result in the estimation errors. For example the delay-and-sum beamformer is reliable only in less-reverberant environments.

In our approach, we assume a presence of the estimation errors in the model, explicitly. The output of the GSC beamformer can be decomposed as follows:

$$\hat{Y}(\omega) = \hat{S}(\omega) - \hat{S}_{\mathrm{N}}(\omega) + \hat{V}(\omega) - \hat{V}_{\mathrm{N}}(\omega) \tag{8}$$

where

$$\begin{aligned} \hat{S}(\omega) &= \mathbf{w}^H(\omega)\mathbf{a}(\omega)S(\omega), \\ \hat{V}(\omega) &= \mathbf{w}^H(\omega)\mathbf{v}(\omega), \\ \hat{S}_{\mathrm{N}}(\omega) &= \mathbf{g}^H(\omega)\mathbf{B}^H(\omega)\mathbf{a}(\omega)S(\omega), \\ \hat{V}_{\mathrm{N}}(\omega) &= \mathbf{g}^H(\omega)\mathbf{B}^H(\omega)\mathbf{v}(\omega). \end{aligned} \tag{9}$$

are the beamformer speech component, beamformer noise component, speech leakage and noise reference respectively. If steering vector $\mathbf{w}(\omega)$ is estimated inaccurately, the speech component contains reverberations. Similarly, if $\mathbf{B}^H(\omega)\mathbf{a}(\omega) \neq 0$, the speech signal leakages to the noise cancellation loop i.e. $\hat{S}_{\mathrm{N}}(\omega) \neq 0$, which results in the cancellation of the speech components at the output of the GSC beamformer. It is difficult to improve dereverberation efficiency, however we can minimize the speech leakage effect at expense of some residual noise increase.

Let's define average power of residual noise and speech leakage respectively at the output of the GSC beamformer:

$$\begin{aligned} \varepsilon_{\mathbf{v}}^2(\omega) &= E\{|\hat{V}(\omega) - \hat{V}_{\mathrm{N}}(\omega)|^2\}, \\ \varepsilon_{\mathbf{s}}^2(\omega) &= E\{|\hat{S}_{\mathrm{N}}(\omega)|^2\}. \end{aligned} \tag{10}$$

Optimization problem for the GSC method can be reformulated as follows:

$$\min_{\mathbf{g}(\omega)} \varepsilon_{\mathbf{v}}^2(\omega), \text{ subject to: } \varepsilon_{\mathbf{s}}^2(\omega) = \alpha(\omega), \tag{11}$$

where $\alpha(\omega)$ is a some predefined level of the speech leakage power. The complex Lagrange functional is given by:

$$L(\mathbf{g}(\omega), \lambda(\omega)) = \varepsilon_{\mathbf{v}}^2(\omega) + \lambda(\omega)(\varepsilon_{\mathbf{s}}^2(\omega) - \alpha(\omega)). \tag{12}$$

Differentiating (12) with respect to $\mathbf{g}(\omega)$ and equating to zero we find the solution:

$$\mathbf{g}_{\mathrm{SLC}}(\omega) = \mathbf{M}(\omega)^{-1}\mathbf{B}^H(\omega)\mathbf{R}_{\mathbf{vv}}(\omega)\mathbf{w}(\omega), \tag{13}$$

where

$$\mathbf{M}(\omega) = \mathbf{B}^H(\omega)[\mathbf{R}_{\mathbf{vv}}(\omega) + \lambda(\omega)\mathbf{R}_{\mathbf{yy}}(\omega)]\mathbf{B}(\omega). \tag{14}$$

The Lagrange multiplier $\lambda(\omega)$ provides a trade-off between speech leakage and noise reduction. It can be easily verified that for $\lambda(\omega) \to \infty$ speech leakage power is decreased at the expense of increased residual noise. If $\lambda(\omega) = 0$, the conventional GSC method is obtained.

The simplest approach is to set this parameter to empirically chosen fixed value. However an optimal (from the perceptual point of view) solution is to find λ_{opt} such that the speech distortion is inaudible and the residual noise is as low as possible. It can be done by substituting the masking threshold of the clean speech - $\phi_{\mathrm{m}}(\omega)$ for $\alpha(\omega)$ and solving the optimization constraint (11), i.e.:

$$\mathbf{g}_{\mathrm{SLC}}^H(\omega)\mathbf{B}^H(\omega)\mathbf{R}_{\mathbf{yy}}(\omega)\mathbf{B}(\omega)\mathbf{g}_{\mathrm{SLC}}(\omega) = \phi_{\mathrm{m}}(\omega), \tag{15}$$

In this way the speech distortions can be effectively reduced. This situation is also depicted in the Fig. 3. Unfortunately derivation of an explicit expression for $\lambda(\omega)$ seems to be a difficult task. It can be done numerically but we found that for certain cases the solution may not exists or be unstable (i.e. when the masking threshold level is very small). Therefore instead trying to solve (15) explicitly, we propose a suboptimal solution:

$$\lambda(\omega) = \lambda_{\max}\min(\mathrm{MNR}(\omega), 1), \tag{16}$$

where

$$\mathrm{MNR}(\omega) = \frac{\phi_{\mathrm{m}}(\omega)}{E\{|\hat{V}(\omega)|^2\}} = \frac{\phi_{\mathrm{m}}(\omega)}{\mathbf{w}^H(\omega)\mathbf{R}_{\mathbf{vv}}(\omega)\mathbf{w}(\omega)} \tag{17}$$

is the mask to noise ratio and $\lambda_{\max}$ is a maximum value for $\lambda(\omega)$. In our experiments it was empirically set to 0.25. The speech correlation matrix $\mathbf{R}_{\mathbf{yy}}(\omega)$ may be

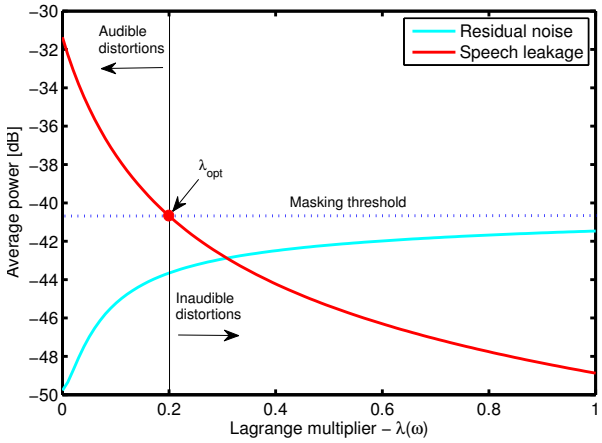


Fig. 3. Example of speech leakage masking

semi-positive definite thus the limiting the Lagrange multiplier improves a numerical stability of the matrix inversion in (13). Note that if the noise power level at beamformer output is below the masking threshold ($\mathrm{MNR}(\omega) \geq 1$) the noise is not audible, thus there is no need for noise cancellation and the speech leakage may be minimized as much as possible. Otherwise, if $0 < \mathrm{MNR}(\omega) < 1$, the noise is audible, thus $\lambda(\omega)$ is scaled proportionally to the MNR value, giving a better noise attenuation.

Theoretically instead of using the MNR one can use local signal-to-noise ratio (SNR), however it is known that the SNR estimate is rather erroneous and says nothing about masking effects [14]. In fact, most psychoacoustic models compute $\phi_{\mathrm{m}}(\omega)$ by performing some smoothing operations on speech power spectrum. Therefore we estimate the clean speech power spectral density (PSD), first:

$$\phi_{\mathrm{s}}(\omega) \approx E\{|\hat{S}(\omega)|^2\} = \mathbf{w}^H(\omega)\mathbf{R}_{\mathbf{yy}}(\omega)\mathbf{w}(\omega). \tag{18}$$

Then we use (18) as an input for Johnston's psychoacoustic model [9]. The correlation matrix of the microphone speech signal is computed as $\mathbf{R}_{\mathbf{yy}}(\omega) = \mathbf{R}_{\mathbf{xx}}(\omega) - \mathbf{R}_{\mathbf{vv}}(\omega)$.

4. Experiments

In this section we compare the performance of the conventional GSC beamformer with the proposed speech leakage constrained approach (denoted as GSC-SLC). The

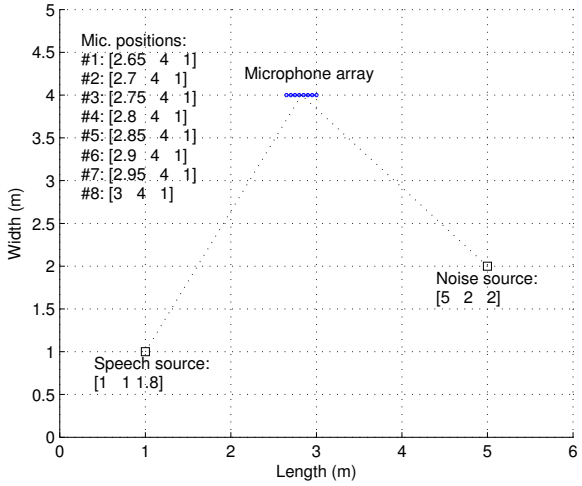


Fig. 4. Floor plan of the simulated enclosure (all coordinates in meters).

methods were implemented in MATLAB using overlap-save procedure. The microphone signals are cut into 50% overlapping frames of size $M = 1024$ samples that corresponds to time window of 128ms long (assuming 8kHz sampling rate). Once the signals are filtered in the DFT domain they are transformed back to time domain and only last $M/2$ samples are saved. In order to determine the system performance under model uncertainties we assumed simple direct-path acoustic model and use delay-and-sum beamformer, thus the steering vector and blocking matrix were computed using propagation delays. To efficiently compute the frequency filters the correlation matrix of the noise signal have to be estimated. However for comparative purposes we put aside this problem and compute $\mathbf{R}_{\mathbf{vv}}(\omega)$ directly from data. In practice any voice activity detector (VAD) can be used to update noise statistics in speech pauses only. Similarly we estimate microphone delays for delay-and-sum beamformer using an exact value of the direction of arrival (DOA) angle.

Two acoustic environments were simulated using the image method [1]: the first one with absorptive surfaces ($T_{60} = 33$ms) and the second one with reflective surfaces ($T_{60} = 135$ms). The parameter T_{60} denote reverberation time defined as the time taken for the sound to decay to 60dB below its value at cessation [10]. In both cases we assumed the rectangular enclosure with dimensions $6 \times 5 \times 2.8$ (all dimensions and coordinates are in meters). We considered an uniform linear array of 8 microphones placed on the x-axis with the first microphone at the position $(2.65, 4, 1)$ and spacing 0.05. The speech source signal was positioned at $(1, 1, 1.8)$. It was about 30s-long

comprised of eight shorter phonetically balanced sentences, uttered by eight of the speakers (four males and four females). These sentences have been selected from TIMIT database [6]. Originally they were recorded at 16kHz sampling rate but for our purposes they were low-pass filtered and downsampled to 8kHz. The noise source was located at $(5, 2, 2)$. The locations of the microphones and the sound sources are also depicted in Fig. 4. During the experiments two noise types have been considered: white Gaussian noise and babble noise both selected from NOISEX-92 database [13]. The microphone signals were obtained by convolving the speech source signal with the room impulse responses and adding to the corresponding noise signals at different SNRs, according to (1).

Our experiments were based on objective performance measurement. The amount of noise reduction was measured using noise attenuation factor defined as the mean ratio between the input noise power and output noise power. The speech distortion factor was defined as segmental signal to noise ratio where the noise is interpreted as a difference between the original and enhanced speech, thus the higher the factor the better. These measures mainly reflect the statistical differences between the signals. Therefore the cepstral distance [15] and modified Bark spectral distortion (MBSD) measure [16] were also used for evaluation of the audible differences. For computation of the cepstral distance we use first 16 cepstral coefficients, that are excepted to carry tonal information. The lower the cepstral distance and/or MBSD measure, the less audible speech distortion. Additionally PESQ measure [8] was exploited for overall evaluation of the speech quality.

The objective measurement results are depicted in Fig. 5. The vertical error-bars denote 95% confidence intervals estimated using 1000 bootstrap data samples. As can be seen the relative improvements are similar for both noise types. Note that the noise statistics were estimated directly from data, often noise signal is not directly available and noise statistics must be estimated from the noisy speech signal, i.e. during speech pauses, thus in practice some performance drop is expected.

For non-reverberant environment ($T_{60} = 33$ms) the improvement is rather not significant. It is not surprising since in this case the direct path model is accurate enough (speech signal goes straight from the sound source to the listener), thus speech leakage is very low and the parameter $\lambda(\omega)$ has no impact on the system performance. In this case the proposed method is equivalent to the conventional GSC beamformer.

In the case of reverberant environment ($T_{60} = 135$ms) the direct path model is not sufficient (i.e. presence of the system model uncertainties) which results in increased speech leakage. However as can be seen in Fig. 5 (solid lines) the proposed method outperforms conventional one providing significantly better performance at lower SNRs in the terms of speech distortion and MBSD measure. One exception

is cepstral distance where confidence intervals are slightly overlapped and thus the improvement is not significant. In order to avoid overestimation of the noise attenuation factor, it should be measured in speech pauses only, however it is rather difficult to precisely mark these regions. Thus, this factor was estimated also in transients where mean squared error is substantially lower for the speech leakage constrained method. Theoretically this measure should be comparable for both methods. On the

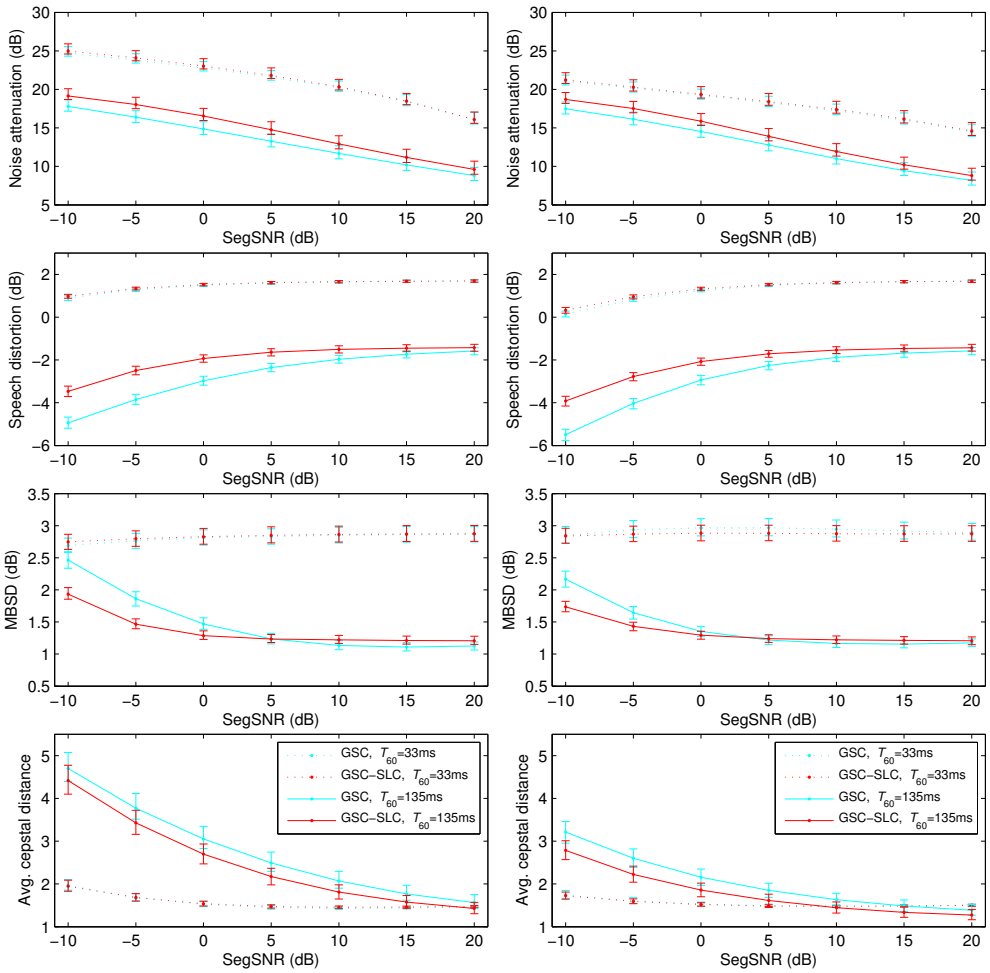


Fig. 5. Comparison of the objective performance measures for the conventional GSC beamformer and the proposed GSC-SLC method in two noisy environments: white noise (left) and babble noise (right); vertical lines denote 95% confidence intervals.

Table 1. Perceptual evaluation using PESQ.

	White Noise				Babble Noise			
	$T_{60} = 33$ms		$T_{60} = 135$ms		$T_{60} = 33$ms		$T_{60} = 135$ms	
SegSNR	GSC	GSC-SLC	GSC	GSC-SLC	GSC	GSC-SLC	GSC	GSC-SLC
-10	2.350	2.412	1.797	1.800	2.306	2.429	1.916	1.988
-5	2.575	2.648	1.914	1.992	2.548	2.643	2.027	2.147
0	2.775	2.847	2.041	2.139	2.757	2.838	2.133	2.262
5	2.947	3.006	2.150	2.262	2.945	3.021	2.243	2.340
10	3.113	3.165	2.249	2.343	3.128	3.191	2.335	2.389
15	3.293	3.356	2.334	2.387	3.301	3.369	2.390	2.437
20	3.479	3.566	2.404	2.423	3.468	3.542	2.433	2.454

other hand, it is clear that a residual noise increase is not proportional to the speech distortion decrease. In our experiments this increase is 'negative'.

Similar observations can be made for the PESQ scores (see Tab. 1). Although we observe lower performance results for the conventional GSC beamformer for both reverberation/noise conditions, in the case of reverberant environment relative improvement is higher.

5. Conclusion

The performance of the conventional GSC beamformer can be improved in the presence system model uncertainties by using auditory properties. We derived a noise cancellation filter which is able to reduce the speech leakage (and speech distortions) at expense of residual noise increase. However as we show this increase is rather small. In addition it is tolerated by auditory system as long as the noise level is placed below masking threshold. The experimental results show that the proposed method outperforms conventional GSC beamformer providing lower speech distortions and comparable residual noise level.

There are some possible improvements of the proposed method, i.e.: a derivation of an explicit formula for optimal Lagrange multiplier, a recursive implementation of the frequency filters or an estimation of the steering vector and blocking matrix using second-order statistics only. These issues will be considered in a future work.

Acknowledgment

This work was supported by Bialystok University of Technology under the grant S/WI/1/2013.

References

[1] J.B. Allen and D.A. Berkley. Image method for efficiently simulating small-room acoustics. *Journal Acoustic Society of America*, 65(4):943, 1979.

[2] J. Benesty, J. Chen, Y. Huang, and J. Dmochowski. On microphonearray beamforming from a mimo acoustic signal processing perspective. *IEEE Trans. Audio, Speech, Lang. Process.*, 15(3):1053–1065, 2007.

[3] A. Borowicz and A. Petrovsky. Signal subspace approach for psychoacoustically motivated speech enhancement. *Speech Comm.*, 53(2):210–219, 2011.

[4] O.L. Frost. An algorithm for linearly constrained adaptive array processing. In *Proc. IEEE*, volume 60, pages 926–935, Aug 1972.

[5] S. Gannot, D. Burshtein, and E. Winstein. Signal enhancement using beamforming and nonstationarity with applications to speech. *IEEE Trans. Signal Process.*, 49(8):1614–1626, 2001.

[6] J.S. Garofolo, L.F. Lamel, W.M. Fisher, J.G. Fiscus, D.S. Pallett, and N.L. Dahlgren. DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus. National Institute of Standards and Technology (NIST), CD-ROM, 1993.

[7] Y. Huang, J. Benesty, and J. Chen. Analysis and comparison of multichannel noise reduction methods in a common framework. *IEEE Trans. Audio, Speech, Lang. Process.*, 16(5):957–968, 2008.

[8] ITU-T. Perceptual evaluation of speech quality (PESQ). Rec. P.862, ITU, Geneva, 2001.

[9] J.D. Johnston. Transform coding of audio signals using perceptual noise criteria. *IEEE J. on Selected Areas in Comm.*, 6:314–323, February 1988.

[10] R. Ratnam, D.L. Jones, and W.D. O'Brien. Fast algorithms for blind estimation of reverberation time. *IEEE Signal Process. Lett.*, 11(6):537–540, 2004.

[11] M. Souden, J. Benesty, and S. Affes. A study of the LCMV and MVDR noise reduction filters. *IEEE Trans. Sig. Process.*, 58(9):4925–4935, Sept 2010.

[12] R. Talmon, I. Cohen, and S. Gannot. Convolutive transfer function generalized sidelobe canceler. *IEEE Trans. Speech, Lang. Process.*, 17(7):1420–1434, Sept 2009.

[13] A. Varga and H.J.M. Steeneken. Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems. *Speech Communication*, 12(3):247–251, 1993.

[14] D. Virette, P. Scalart, and C. Lamblin. Analysis of background noise reduction techniques for robust speech coding. In *Proc. EUSIPCO*, volume 3, pages 297–300, 2002.

[15] S. Wang, A. Sekey, and A. Gersho. An objective measure for predicting subjective quality of speech coders. *IEEE J. Sel. Areas Commun.*, 10:819–829, 1992.

[16] W. Yang, M. Benbouchta, and R. Yantorno. Performance of a modified bark spectral distortion measure as an objective speech quality measure. In *Proc. ICASSP*, pages 541–544, Seattle, USA, 1998.

SKUTECZNY TŁUMIK LISTKÓW BOCZNYCH Z WYKORZYSTANIEM MASKOWANIA PRZECIEKU MOWY

Streszczenie: Prezentowana jest nowa metoda uzdatniania mowy w oparciu o strukturę uogólnionego tłumika listków bocznych. Wykazujemy, że możliwe jest zmniejszenie słyszalnych zniekształceń mowy przy zachowaniu stałego poziomu szumu rezydualnego, dla modeli przybliżonych środowiska akustycznego. Może to być dokonane poprzez uwarunkowanie poziomu mocy przecieku mowy zgodnie ze zjawiskiem maskowania oraz minimalizację warunkową mocy szumu rezydualnego. Proponowane podejście zaimplementowano w oparciu o prosty model beamformera opóźniająco-sumującego. Ostatecznie przeprowadzono ocenę porównawczą wybranych metod z wykorzystaniem obiektywnych miar jakości mowy. Wyniki pokazują, że nowa metoda przewyższa konwencjonalną zapewniając mniejsze zniekształcenia mowy.

Słowa kluczowe: GSC, psychoakustyka, uzdatnianie mowy

Artykuł zrealizowano w ramach pracy badawczej S/WI/1/2013.

DATA PREPROCESSING IN THE CLASSIFICATION OF THE IMBALANCED DATA

Katarzyna Borowska[1], Magdalena Topczewska[2]

[1] Student of Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

[2] Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

Abstract: The article concerns the problem of imbalanced data classification. Two algorithms improving the standard SMOTE method have been created and tested. To measure the distance between objects the Euclidean or the HVDM metric was applied, depending on the number of nominal attributes in a dataset.

Keywords: class imbalance, oversampling, classification

1. Introduction

Dynamic development of expert systems brings uncountable benefits in many domains. Especially the medical diagnosis requires very accurate and infallible computer decision support systems. Introducing these kind of systems to real–world problems revealed that effectiveness of data mining depends on data distribution. Experts have found that standard classifiers become not sufficient when processing complex examples. One of the reasons of high complexity is an imbalanced class distribution. This problem occurs when one class is underrepresented in a dataset. It may lead to the reduction of performance when standard classifiers are used. Due to the fact that many real–life domains suffer from the class imbalance problem, it has emerged as one of the challenges in data mining community [9].

There is one fundamental difficulty when objects are classified: the assumption that concerns the class distribution. Typically, the distribution of examples in a dataset is predicted to be uniform and costs of misclassification are expected to be equal for all classes. This is excessively simplified assumption, because many datasets contain rare objects which represent the class of interest. Simple classifiers tend to generalize

Advances in Computer Science Research, vol. 11, pp. 31-46, 2014.

and create rules with broader data coverage. Hence, they usually ignore rare examples [12]. However, the accuracy may reach very high values, even when all instances from the minority class are misclassified. Therefore more appropriate methods for evaluating classification performance in case of occurrence of rare examples need to be applied.

It should be emphasized that the erroneous recognition of examples from the minority class may lead to disastrous consequences. Medical diagnosis, detection of fraudulent financial transactions, anomaly detection, learning word pronunciation, predicting pre-term births or detection of oil spills are only a few examples of areas affected by the imbalanced data problem [1,9]. It is obvious that the necessity of minimizing the number of wrong decisions in these domains is recognized as the significant issue. Due to the critical role of imbalanced data in the modern world, many proposals have been developed to decrease the negative effects of this problem. These techniques can be divided into three main categories: data level, algorithm level and cost-sensitive approaches.

Data level approaches are the most versatile. These algorithms are used independently of the classifier and that is considered as their main advantage. The principal aim of designing these solutions was to reduce disparity between the number of data from the minority and majority class. The ability to deal with additional difficulties of the learning process is very important in data processing.

One of the proposals concerning pre–processing imbalanced data is SMOTE. Although this technique has been used with success in many domains, it is not deprived of some drawbacks. Limitations of this algorithm may lead to considerable depletion of classifier performance. This problem usually occurs when dataset is not only imbalanced, but also have a complex distribution [4]. Complex distribution may be associated with factors such as overlapping, small disjunctions and noise. Recent studies have shown that these difficulties are the main source of problems in the classification [11,9,5].

The novel approach for mining imbalanced data is presented in this paper. Two algorithms are proposed as the improved version of the standard SMOTE technique.

2. Algorithm

Three main approaches can deal with the class imbalance problem. Although all of these solutions have been successfully applied in many domains, in this paper we focus only on the data level techniques as they are independent of the classifier and therefore flexible. Studies have shown that the application of preprocessing phase to balance the skewed class distribution usually improves the classifier performance [9].

Data preprocessing methods consist of various approaches. It is possible to list the following groups involving data level classification techniques:

- undersampling - a subset of the original dataset is created, some examples from the majority class have to be removed,
- oversampling - new examples are generated, especially from the minority class,
- hybrid - combination of the two previous methods.

Neither undersampling nor oversampling is deprived of disadvantages. The major drawback of undersampling is the risk of missing potentially important data. On the other hand, the oversampling in its simplest version, assuming random replication of minority class examples, may lead to overfitting [4]. The numerous proposals addressing these problems were developed.

SMOTE (Synthetic Minority Oversampling Technique) [6] represents the group of oversampling techniques. Unlike the simple random oversampling, SMOTE does not create new instances by generating identical copies of existing minority samples. The main idea of this algorithm is to create new minority class examples along the line segments between each positive class object and any of the k nearest neighbors. New instances are generated by randomly selecting appropriate number of the k nearest neighbors of a positive class example and creating a combination of features describing each of them and sample under consideration. The number of neighbors involved in oversampling depends on the number of needed minority examples. Algorithm 1 presents the pseudocode for SMOTE.

The objective of data preprocessing in SMOTE is to create synthetic examples regarding similarity between minority class instances. The similarity is defined in feature space by using the kNN algorithm, where the number of nearest neighbors k is a parameter. It is crucial to choose an appropriate value for the k parameter. Necessity of finding an adequate number of nearest neighbors is one of the SMOTE drawbacks. Over generalization and variance are considered as the other limitations of this method [4]. Although SMOTE avoids the overfitting problem and makes the decision boundaries for the minority class larger, the algorithm does not take into consideration the neighborhood of the minority class examples. It may lead to the overlapping between classes, which has a considerable negative impact on the classification process.

3. Methods

The method for creating synthetic items based on the combination of adjacent objects features seems to be so effective and groundbreaking tool that would be unreasonable

Algorithm 1 SMOTE (T, N, k)

Require: Number of minority class samples T;
Amount of examples to create $N\%$;
Number of nearest neighbors k

if $N < 100$ **then**
  Randomize the T minority class samples
  $T = (N/100) * T$
  $N = 100$
end if
$N = (int)(N/100)$
$numattrs$ = number of attributes
$Sample[][]$: array for the original minority class samples
$newindex$: keeps a count of number of synthetic samples generated, initialized to 0
$Synthetic[][]$: array for synthetic samples
for $i \leftarrow 0$ **to** T **do**
  Compute k nearest neighbors for i, save the indices in the $nnarray$
  $Populate(N, i, nnarray)$
end for
/* Function to generate the synthetic samples */
$Populate(N, i, nnarray)$
while $N \neq 0$ **do**
  Choose a random number between 1 and k, call it n. This step chooses one of the k nearest neighbors of i.
  for $attr \leftarrow 1$ **to** $numattrs$ **do**
    $dif = Sample[nnarray[nn]][attr] - Sample[i][attr]$
    $gap = randomnumberbetween0and1$
    $Synthetic[newindex][attr] = Sample[i][attr] + gap * dif$
  end for
  $newindex++$
  $N = N - 1$
end while
return

not to take advantage of the potential it brings. The original version of SMOTE technique turns out to be insufficient when faced with complex problems. On the other hand, it has been proven that many real data sets has a complicated structure and differences between the representatives of the different classes are not as obvious as it is expected [5,11]. When data is imbalanced, the high complexity has a negative impact mainly on identifying minority class instances. Especially small disjunctions, noise and overlapping handicap the process of classification (figure 1). Improved SMOTE (IS) algorithms were created to reduce negative impact of these impediments. The novel algorithms comprise a compound of the existing approaches and provide a brand new way of dealing with imbalanced data issue. Works such as [11,7,8] pose

an effective solutions for the imbalanced data problem. However, they were not sufficient for all kinds of specific domains. Therefore IS techniques were designed to be more flexible.

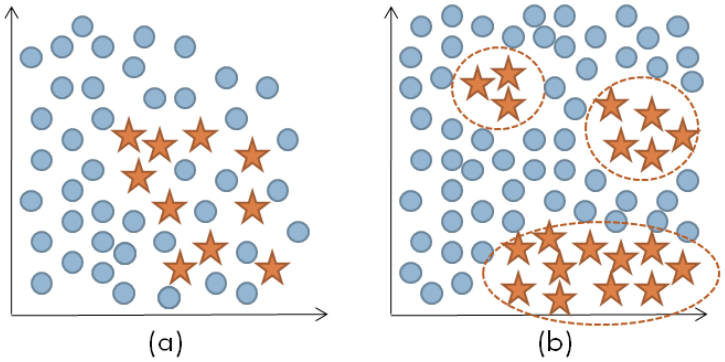


Fig. 1. (a) Class overlapping (b) Small disjunctions

Both of the developed algorithms are based on the same main concept (algorithm 27). At the beginning, right after loading the data, the metrics is choosen automatically. The analysis of attributes characteristics indicates whether the Euclidean distance is used or the HVDM. The HVDM metrics is applied, when more than half of the attributes is nominal. Otherwise, the Euclidean distance is used. The idea presented in [10] was the inspiration for developing this method to determine which distance function is the most proper for a specific issue. According to this work, the effectiveness of HVDM metrics should be closely related to the number of nominal attributes.

In the next step the k–NN algorithm is used to obtain the distance between each minority example and all other instances from both classes. The k is a parameter – user can specify the value of nearest neighbors. According to these calculations minority objects are divided into three groups (Algorithm 2):

- NOISE, when all of the k nearest neighbors represent the majority class,
- DANGER, if half or more than half of the k nearest neighbors come from the majority class,
- SAFE, when more than half of the k nearest neighbors represent the same class as the example under consideration.

Algorithm 2 IBA (S, M, k)

Require: Number of all instances S;
Number of minority class samples M;
Number of nearest neighbors k

metrics: keeps the name of used evaluation metric
numattrs: keeps the number of attributes
SampleMinority[][]: array for the original minority class samples
Sample[][]: array for all samples
Synthetic[][]: array for the new examples
nominal: keeps the number of nominal attributes
continuous: keeps the number of linear attributes
label[]: array for examples labels
for $i \leftarrow 0$ **to** *numattrs* **do**
    Verify the number of two kinds of attributes: nominal and linear. Save the number of nominal attributes in *nominal* variable and the number of linear attributes in *continuous* variable.
end for
if *continuous* $\leqslant$ *numeric* **then**
    $metrics := HVDM$
else
    $metrics := Euklides$
end if
for $i \leftarrow 0$ **to** M **do**
    Calculate the distance between minority class examples and all other examples using k–NN method with measure written in *metrics* variable.
    Indexes of k nearest neighbors write in *nnarray* array.
    $label[i] := LabelMinorityData(nnarray, i, k)$
end for
Calculate the needed number of minority class examples to create. The result save in N variable.
for $i \leftarrow 0$ **to** M **do**
    if $label \neq NOISE$ **then**
        Run the k–NN algorithm for the object i using distance measure saved in *metrics* variable, indexes save in *nnarray* array
        $Populate(N, i, nnarray, label[i])$
        /* The *Populate* method is different for the two algorithms */
    end if
end for

The mechanism of the above division considers the location of each minority example in the feature space. There are plenty of proposals in the literature regarding this approach. They vary in the way of distinguishing different objects types and processing them in the next phases. The method presented in this paper assumes that the NOISE examples are surrounded only by the majority class instances. It is obvious that this kind of rare data may lead to serious difficulties in the learning process [2]. Examples which occure in the area surrounding class boundaries are

labeled DANGER. The relatively homogeneous areas consists of the SAFE objects. The main process, responsible for generating new data, strictly relates to the assigned labels.

Algorithm 3 LabelMinorityData (nnarray, i, k)

Require: Number of nearest neighbors k;
Indexes of k nearest neighbors *nnarray*;
Index of the example under consideration i;
/* This method labels the minority class data */
minorityClass: keeps the number of minority class neighbors of the i example
majorityClass: keeps the number of majority class neighbors of the i example
if $majorityClass == k$ **then**
  $returnNOISE$
end if
if $majorityClass < k/2$ **then**
  $returnSAFE$
end if
if $majorityClass \geqslant k/2$ **then**
  $returnDANGER$
end if

The oversampling techniques necessitate the number of minority class examples which should be created. In proposed IS solution this number is selected automatically. The minority class instances are generated to even the amount of objects from both classes.

Next, the distances between samples representing only the minority class are calculated. The k–NN method is used for this purpose again. After this operation, it is possible to start the next phase – generating new synthetic samples in the number depending on the assigned labels and the algorithm version.

The main purpose of this paper is to verify the impact of DANGER examples on the learning process. These borderline instances determine the boundaries between different classes. The two approaches of preprocessing minority samples are proposed.

3.1 ASIS

ASIS (Amplify SAFE Improved SMOTE) is the first of presented algorithms. This is a modified version of the standard SMOTE technique. The main assumption in the SIS method is that excessive number of borderline minority examples may increase

the data complexity. After the steps described previously, new data is generated. Dependency between assigned labels and the number of created minority class examples is the following:

- SAFE – the numerous new objects are created, similarly to the standard SMOTE algorithm,
- DANGER – only one new example is created by combining features of the nearest neighbor and the instance under consideration; it is located closer to the DANGER object,
- NOISE – no new example is created.

The fact that the NOISE instances are omitted in this phase is dictated by their location in the feature space. Figure 2 illustrates the undesirable consequences of applying the SMOTE technique to this kind of instances. Figure 2 (a) presents the situation when the new object is created in the line segment between minority class example $A1$, considered as noisy, and its nearest neighbor $A2$, located in the homogeneous area. As it is showed, generated object $A3$ overlaps with the majority class example. The learner's ability to generalize may cause the misclassification of the $A3$. Situation is even worse in the figure 2 (b). When four neighbors are involved in creating new samples, the level of distribution disturbances is very high. The boundaries between classes are ambiguous. Hence, discriminative rules are hard to prepare. The avoidance of these problems is possible by omitting NOISE examples in preprocessing step.

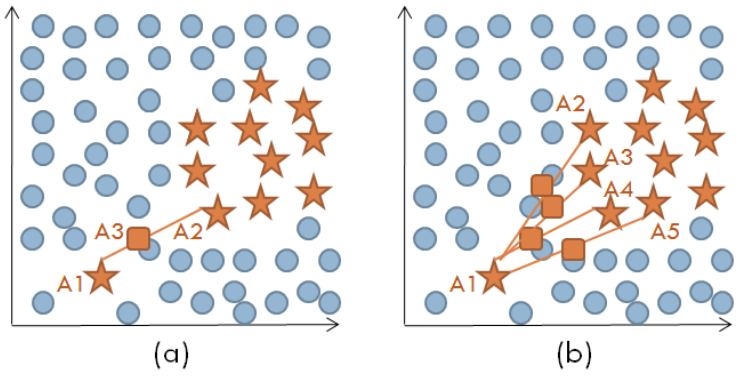


Fig. 2. Example of difficulties in processing NOISE instances

The ASIS algorithm doubles the number of DANGER data. It is assumed that this amount would be sufficient to make borderline minority examples more recognizable and not increase the overlapping level. Moreover, in this case synthetic objects are created closer to instance which is currently processed and only the nearest neighbor takes part in processing. On the other hand, the plenty of new data is created for SAFE objects. This kind of objects should be considered as the representatives of the minority class. Due to the fact that SAFE examples are located in relatively homogeneous areas this data has characteristic properties of the minority class.

3.2 ADIS

The second algorithm is named ADIS (Amplify DANGER Improved SMOTE). It represents different concept of treating respective groups of minority data. The novel strategy assumes that the ambiguity of class boundaries should be reduced by amplifying the minority class instances especially in these areas. Experiments made in [3] demonstrates that the local density of examples plays a key role in the analysis of overlapping regions containing examples from different classes. Hence, performance of the classifier should be improved when creating many new instances in DANGER objects neighborhood. In this strategy, the following processing is performed for the respective three groups:

- SAFE – one new object is created by interpolation of the example under consideration and its nearest neighbor,
- DANGER – the numerous minority instances are created, the synthetic example is placed closer to the object under consideration,,
- NOISE – no new example is created.

The amplification of the minority class representatives in the borderline regions should make learner to create more proper rules. However, it may lead to the degradation of the classifier performance regarding majority class examples.

4. Experiments

Two experiments have been performed to test the new methods as the improvement of the SMOTE algorithm.

4.1 Experiment 1

In the first experiment the artificial data set, containing only 25 objects, has been used to present the performance of new methods. This data are characterized by the

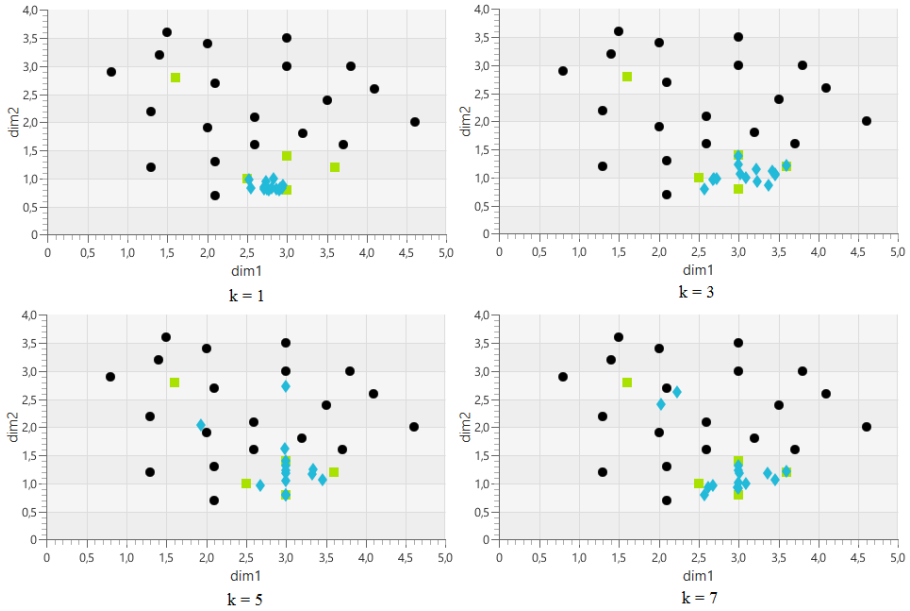


Fig. 3. Distributions after ASIS preprocessing

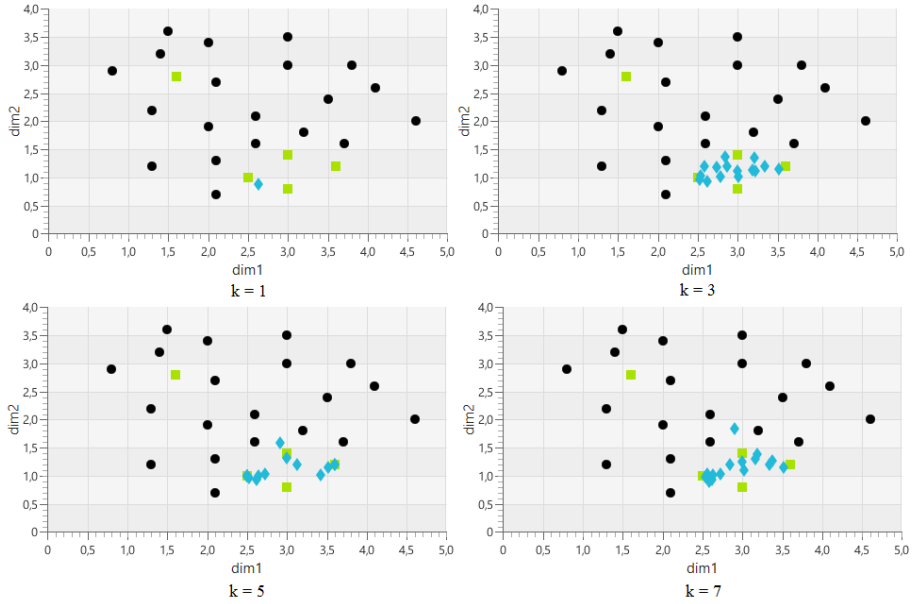


Fig. 4. Distributions after ADIS preprocessing

moderate IR value. The size of the majority class four times exceeds the number of objects in the minority class. The obtained results of classification are shown in the table 1. In the picture 3 and 4 the majority class objects are marked as circles, while the minority class objects are squares. The new generated objects are marked as diamonds.

Table 1. Results of the `artifficialData` classification (IR=4): Q – accuracy, TP_{rate} – rate of true positives, TN_{rate} – rate of true negatives, AUC – area under the curve, nIR – new value of IR

method	k	Q	TP_{rate}	TN_{rate}	*F-measure*	AUC	nIR
SMOTE	1	80.00	80.00	80.00	0.80	80.00	1.00
ASIS	1	80.00	90.00	70.00	0.82	80.00	1.00
ADIS	1	84.62	50.00	95.00	0.60	72.50	3.33
SMOTE	3	87.50	90.00	85.00	0.88	87.50	1.00
ASIS	3	95.00	90.00	100.00	0.95	95.00	1.00
ADIS	3	95.00	90.00	100.00	0.95	95.00	1.00
SMOTE	5	80.00	80.00	80.00	0.80	80.00	1.00
ASIS	5	90.00	85.00	95.00	0.89	90.00	1.00
ADIS	5	95.00	90.00	100.00	0.95	95.00	1.00
SMOTE	7	80.00	80.00	80.00	0.80	80.00	1.00
ASIS	7	75.00	65.00	85.00	0.72	75.00	1.00
ADIS	7	87.50	85.00	90.00	0.87	87.50	1.00

The analyzed data set may be specified as implicitly imbalanced, because the size of the minority class is extremely small. The best results have been obtained for the number of neighbours equaled 3. The ASIS and the ADIS methods occurred to give the highest values of parameters: the accuracy at the level 95%, the rate of true positives (90%), the rate of true negatives (100%), the F-measure (0.95) and the area under the curve (the level 95%). The same highest results have been gained for the number of neighbours 5. Thus we confirmed the assumption that the correctness of the classification depends largely on the complexity of the data distribution. Placement of objects from the minority class in a homogeneous area is one of the main success factors in creating the correct model.

4.2 Experiment 2

In the second experiment the data sets from the UCI (*University of California at Irvine Repository*) [13] are performed. The characteristics of the chosen data is presented in the table 2.

Table 2. Characteristics of datasets

dataset	number of objects	number of attributes (numeric; symbolic)	missing data	*IR*
abalone9-18	731	8 (7;1)	no	16.60
blood transfusion	748	4 (4;0)	no	3.20
breast cancer	286	9 (0;9)	yes	2.36
german credit	1000	20 (7;13)	no	2.33
hepatitis	155	19 (6;13)	yes	3.84
vowel0	988	13 (13;0)	no	9.98

Six chosen datasets have been preprocessed using the ASIS and ADIS methods. The detailed results of classification are demonstrated in the tables 3 and 4 regarding several values for the number of neighbours. The highest values are highlighted in bold.

In the case of proposed methods, both approaches proved to be effective in the real applications. For all chosen datasets the accuracy, the area under the curve, the F-measure and what is the most important – the rate of true positives are better than in the standard SMOTE case.The 3 neighbours occurred the most advantageous.

5. Conclusions

In the era of collecting increasingly large and large data volumes, the problem of the class imbalance in data becomes one of the biggest challenges for the scientists. Achieving high classification accuracy of data representing the minority class is not an easy task and the variety of methods created only for this purpose may be the confirmation of that growing need.

Among many, the algorithms belonging to the group called pre-processing of data can be noticed. Their aim is to increase in the number of objects of the positive (minority) class. The most well-known technique of this type is the SMOTE algorithm that became the inspiration to create and test the new, improved versions of the method.

In the first algorithm (`ASIS`) the number of safe objects increases primarily. They may be perceived as the best representation data of the minority class. Additionally, the number of border objects is doubled, while the objects recognized as the noise do not transform. In the second algorithm (`ADIS`) most of the new objects are created in the border area. The number of the safe objects is doubled, while similarly as in the previous case the noise objects do not share the transformation process. Comparing

Table 3. Results of the chosen UCI datasets classification: Q – accuracy, TP_{rate} – rate of true positives, TN_{rate} – rate of true negatives, AUC – area under the curve, nIR – new value of IR

method	k	Q	TP_{rate}	TN_{rate}	*F-measure*	AUC
			abalone9-18			
SMOTE	-	94.12	35.71	97.68	0.41	0.6670
ASIS	3	97.31	**96.81**	97.82	**0.97**	97.31
	5	96.73	96.37	97.10	**0.97**	96.73
	7	**97.39**	96.52	**98.26**	**0.97**	**97.39**
	11	93.80	62.32	96.95	0.65	79.64
	15	94.48	62.50	97.82	0.68	80.16
ADIS	3	97.02	**97.10**	96.95	0.97	97.02
	5	96.73	96.37	97.10	0.97	96.73
	7	96.08	96.37	95.79	0.96	96.08
	11	95.07	95.50	94.63	0.95	95.07
	15	94.27	95.21	93.32	0.94	94.27
			blood transfusion			
SMOTE	-	76.07	33.71	**89.30**	0.40	61.50
ASIS	3	**80.88**	79.65	82.11	**0.81**	**80.88**
	5	79.65	77.37	81.93	0.79	79.65
	7	79.91	77.72	82.11	0.79	79.91
	11	76.05	77.72	74.39	0.76	76.05
	15	78.85	78.07	79.82	0.79	78.95
ADIS	3	80.61	**84.04**	77.19	**0.81**	80.61
	5	77.98	80.53	75.44	0.79	77.98
	7	74.47	80.53	68.42	0.76	74.47
	11	76.49	73.86	79.12	0.76	76.49
	15	78.07	74.56	81.58	0.77	78.07
			breast cancer			
SMOTE	-	69.50	41.18	**81.59**	0.45	61.38
ASIS	3	**69.90**	72.14	67.66	0.71	69.90
	5	70.15	70.15	70.15	0.70	70.15
	7	73.38	74.13	72.64	0.74	73.38
	11	68.66	72.14	65.17	0.70	68.66
	15	69.15	71.14	67.16	0.70	69.15
ADIS	3	74.63	**76.62**	72.64	**0.75**	**74.63**
	5	73.63	75.62	71.64	0.74	73.63
	7	69.40	70.15	68.66	0.70	69.40
	11	72.78	68.82	76.12	0.70	72.47
	15	69.65	70.65	68.66	0.70	69.65

Table 4. Results of the chosen UCI datasets classification: Q – accuracy, TP_{rate} – rate of true positives, TN_{rate} – rate of true negatives, AUC – area under the curve, nIR – new value of IR

method	k	Q	TP_{rate}	TN_{rate}	*F-measure*	AUC
			german credit			
SMOTE	-	69.60	47.33	79.14	0.48	63.24
ASIS	3	**80.64**	**81.29**	**80.00**	**0.81**	**80.64**
	5	79.50	80.14	78.86	0.80	79.50
	7	78.57	80.00	77.14	0.79	78.57
	11	79.00	80.29	77.71	0.79	79.00
	15	79.14	80.29	78.00	0.79	79.14
ADIS	3	80.14	80.86	79.43	0.80	80.14
	5	78.21	78.14	78.29	0.78	78.21
	7	78.64	79.00	78.29	0.79	78.64
	11	79.00	79.86	78.14	0.79	79.00
	15	79.00	79.86	78.14	0.79	79.00
			hepatitis			
SMOTE	-	85.81	53.13	**94.31**	0.61	73.72
ASIS	3	89.84	93.50	86.18	0.90	89.84
	5	89.02	92.68	85.37	0.89	89.02
	7	87.80	89.43	86.18	0.88	87.80
	11	91.87	**96.75**	86.99	**0.92**	91.87
	15	88.62	91.06	86.18	0.89	88.62
ADIS	3	91.46	92.68	90.24	**0.92**	91.46
	5	**92.28**	93.50	**91.06**	**0.92**	**92.28**
	7	91.87	94.31	89.43	**0.92**	91.87
	11	88.21	91.87	84.55	0.89	88.21
	15	86.99	89.43	84.55	0.87	86.99
			vowel0			
SMOTE	-	99.49	96.67	99.78	0.97	98.22
ASIS	3	99.78	**100.00**	99.55	**1.00**	99.78
	5	**99.94**	**100.00**	**99.89**	**1.00**	**99.94**
	7	99.83	99.89	99.78	**1.00**	99.83
	11	99.83	99.89	99.78	**1.00**	99.83
	15	99.78	99.89	99.67	**1.00**	99.78
ADIS	3	99.54	98.33	99.78	0.99	99.06
	5	99.61	99.67	99.55	**1.00**	99.61
	7	99.67	99.67	99.67	**1.00**	99.67
	11	99.67	99.89	99.44	**1.00**	99.67
	15	99.72	99.78	99.67	**1.00**	99.72

to the standard SMOTE algorithm, new methods improved the classification results like the accuracy, the rate of true positives or the area under the curve.

References

[1] G. M. Weiss, Mining with Rarity: A Unifying Framework, SIGKDD Explor. Newsl., Springer Berlin Heidelberg, 6(1), 7–19, 2004.

[2] S. Barua, Md. M. Islam, K. Murase, A Novel Synthetic Minority Oversampling Technique for Imbalanced Data Set Learning, Neural Information Processing, Springer Berlin Heidelberg, 7063, 735–744, 2011.

[3] V. Garcia, R. A. Mollineda, J. S. Sanchez, On the k–NN performance in a challenging scenario of imbalance and overlapping, Pattern Analysis and Applications, Springer-Verlag, 11, 269–280, 2008.

[4] H. He, E. A. Garcia, Learning from Imbalanced Data, IEEE Trans. on Knowl. and Data Eng. on 21(9), 1263–1284, 2009.

[5] J. Taeho, N. Japkowicz, Class Imbalances Versus Small Disjuncts, SIGKDD Explor. Newsl. on 6(1), 40–49, 2004.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, SMOTE: Synthetic Minority Over-sampling Technique, J. Artif. Int. Res. on 16(1), 321–357, 2002.

[7] S. Hu, Y. Liang, L. Ma, Y. He, MSMOTE: Improving Classification Performance When Training Data is Imbalanced, Computer Science and Engineering, 2, 13–17, 2009.

[8] N. V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: Synthetic Minority Over-sampling Technique, J. Artif. Int. Res. on 16(1), 321–357, 2002.

[9] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera, A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 42(4), 463–484, 2012.

[10] G. E. A. P. A. Batista and D. F. Silva, How k-Nearest Neighbor Parameters Affect its Performance, Argentine Symposium on Artificial Intelligence, 1–12, 2009.

[11] K. Napierała, J. Stefanowski, S. Wilk, Learning from Imbalanced Data in Presence of Noisy and Borderline Examples, Proceedings of the 7th International Conference on Rough Sets and Current Trends in Computing, Springer-Verlag, Warsaw, 2010.

[12] Y. Sun, M. S. Kamela, A. K. C. Wongb, Y. Wangc, Cost-sensitive boosting for classification of imbalanced data, Pattern Recognition, 40(12), 3358—3378, 2007.

[13] UC Irvine Machine Learning Repository, http://archive.ics.uci.edu/ml/, (20.05.2014).

PRZETWARZANIE WSTĘPNE W PROBLEMIE KLASYFIKACJI DANYCH NIEZRÓWNOWAŻONYCH

Streszczenie: Artykuł dotyczy problemu klasyfikacji w przypadku, gdy mamy do czynienia z klasami niezrównoważonymi. W tym celu stworzone zostały dwa algorytmy poprawiające wyniki uzyskiwane za pomocą standardowego algorytmu SMOTE. Do pomiaru odległości między obiektami zastosowano metrykę euklidesową lub metrykę HVDM, w zależności od liczby cech nominalnych w zbiorze.

Słowa kluczowe: klasy niezrównoważone, tworzenie nowych obiektów, klasyfikacja

Artykuł w części zrealizowano w ramach pracy badawczej S/WI/2/2013.

AUTOMATIC SEGMENTATION OF CORNEAL ENDOTHELIAL CELLS USING ACTIVE CONTOURS

Kamil Charłampowicz[1], Daniel Reska[2], Cezary Boldak[2]

[1] Student of Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

[2] Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

Abstract: We present a new method for segmenting the corneal endothelial cells from microscopic images. It uses multiple active contours initialized by adaptive thresholding and limited with their growing to not overlap. Thanks to the inherent characteristics of the active contour both outcomes can be achieved: cell quantity and delimitation. The tool implementing this approach is built within the MESA framework - an environment for developing and evaluating segmentation techniques. The accuracy is estimated on the base of real microscopic cell images segmented manually.

Keywords: image processing, cell segmentation, active contour

1. Introduction and background

Cell segmentation, very important in many biomedical applications, is not a trivial task. Many approaches to this problem have been proposed, but the most often general techniques fail when applied to specific cell shapes and targeted methods perform the best. One example are the corneal endothelial cells, which play important role in the human visual mechanism and are of great interest for physicians [1]. The healthy cells have a characteristic regular hexagonal form (Fig. 1). With time or in presence of pathologies their number drops and their shape deforms, so their analysis can supply important diagnostic and monitoring information. Unfortunately, their standard microscopic imaging does not always provide a good quality of images. Specifically, near the image borders the focus can be lost and the illumination is not homogeneous. The noise is also present and the cell shape can be irregular. Several dedicated approaches have been proposed to segment such the cells, apart from the manual delineating. Vincent and Masters [2] firstly detect the cell centers by

Advances in Computer Science Research, vol. 11, pp. 47-60, 2014.

a dome extractor based on morphological grayscale reconstruction and then let the marker-driven watershed segmentation to extract the binary (so unstructured) result. Mahzoun at al. [3] focus on the cell edges, detecting them (including "tricorn" points) with specially designed convolution filters. Then they complete the shapes with manually initialized active contours, so the approach is not automatic but can provide the cell structure. Sanchez-Marin [4] proposes a fully automatic method using the Gauss low-pass filtering to remove the intensity gradient and also to close gaps in the cell borders, then thresholding and skeletonization (to obtain the one-pixel wide net) followed by some improvement (e.g. pruning) of the final cell structure. The result cells can be smoothed with the active contour. The method seems sensitive to the scale on the Gauss filtering stage. Khan et al. [5] apply the Frangi algorithm (originally detecting vessels) to images filtered by consecutive low-pass Gauss filters in order to track the cell borders. These borders are then improved by mathematical morphology operations and finally thinned to get the one-pixel wide net. The result is not structured - it consists of the border pixel collection. Bullet et al. [6] use composition of four steps to automatically detect the cell contours: FFT band-pass filtering to remove the intensity information and noise, binarization with the mean value threshold, watershed segmentation on the distance map and finally Voronoi diagram segmentation providing the final contours. All the above operations are performed within the standard image processing tool – ImageJ. The most recent work by Piorkowski and Gronkowska-Serafin [1] proposes two approaches. Both of them analyze local configurations of pixel levels in order to mark the border (intensity valley) between cells. The detected contours are then improved with the mathematical morphology.

This work is organized as follows. The next section describes the proposed segmentation methods in details and reveals some implementation aspects. Section 3. presents the method evaluation on real images of the corneal endothelium cells. The last section summarizes the article and proposes some ideas to improve the segmentation.

2. Method

The proposed algorithm is composed of the four steps:

- preprocessing,
- binarization,
- detection of the cell center points,
- segmentation of the cell contours.

All of them are automatic and do not require an operator interaction. Each steps is described in details in the following subsections.

2.1 Preprocessing

Our input images are noisy (Fig. 1), what is often the case while analyzing the medical data. The intensity levels of the cell bodies and borders are not constant, even locally. To alleviate these problems we apply the Gauss low-pass convolution filter to the raw images (as like it is common in many segmentation frameworks). This filtering extends also the high gradient zones located on the cell borders and makes them more "visible" to the active contours segmenting the final cell shapes.

2.2 Binarization

In order to distinguish the cell bodies (more intensive) from the cell borders (darker) the most obvious solution is to use the thresholding technique. But as it has been already mentioned above, the illumination of the input images is not equal in all the image regions – one can observe the intensity gradient superimposed on the local intensity distribution (Fig. 1). This artifact makes impossible to binarize the input with the simple thresholding with one global threshold.

One of the possible to employ techniques that has been used in our work is the *adaptive thresholding*. As the intensity variation coming from the varying illumination has very low frequency and can be neglected locally, the threshold is calculated and used also locally. The entire image is divided on square regions and the mean intensity values are calculated for all of them. For each such region its center pixel is given the local threshold equal to the mean intensity of this region, incremented by few levels to avoid detecting some ghost structures in more or less homogeneous areas. The rest of pixels obtain their local threshold by simple linear interpolation. Then for all the pixels in the loop: each one is marked "black" (cell body) if its intensity is above its own local threshold and it is marked "white" (cell border) if the intensity is below it. Example of such the binarization is presented Fig. 1.

2.3 Detection of the cell center points

The healthy corneal cells have they shape close to hexagonal, and even if it is not ideal, it can be approximated by a circle. After the binarization the next loop (going also through all the pixels) tries to localize the biggest (locally) circular regions containing only "black" pixels. A special condition promotes bigger regions if two or more of them overlap – it prevents from locating multiple circle regions inside one cell what can happen because of the order of visiting the pixels (the loop simply increments the pixel coordinates). This condition is effective for the most of cases but it sometimes fails, especially for elongated cells. In order to eliminate such the

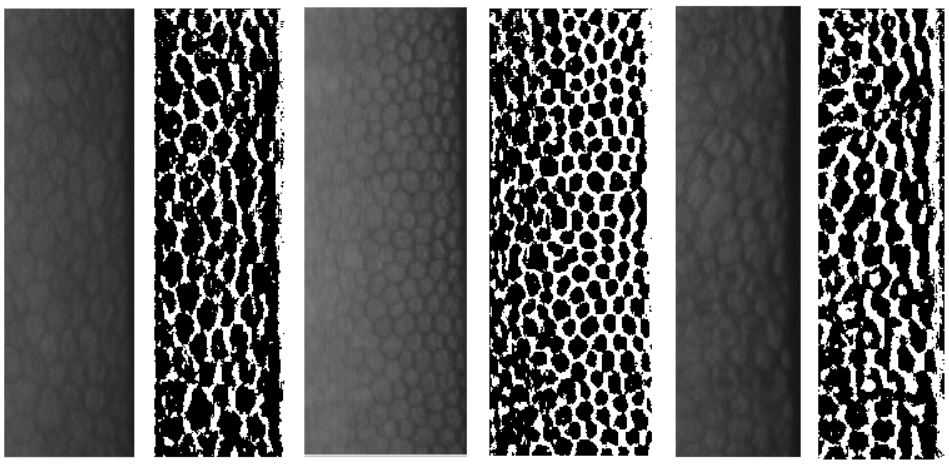


Fig. 1. Example images of the corneal endothelial cells and effects of binarization by adaptive thresholding (after the Gauss filtering)

situations all pairs of detected neighbor circles are examined to verify if the area separating them belongs also to the cell body (Fig. 2). If it is so the regions are merged to form a single cell body representation.

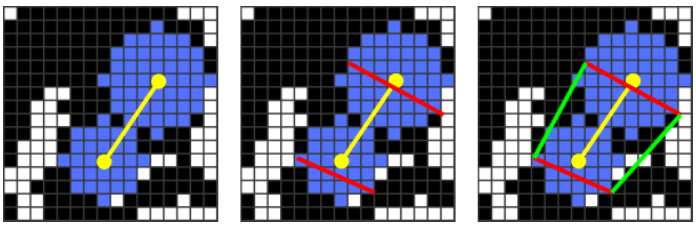


Fig. 2. Merging neighbor circle regions inside a single cell

The effect of this procedure is shown on Fig. 3. One can observe several merged circles inside elongated cells.

2.4 Final segmentation of the cell contours – active contours

The initial approximation of the cell bodies by the circular regions (or their unions) are further deformed to fit the actual cell borders using the well known *active con-*

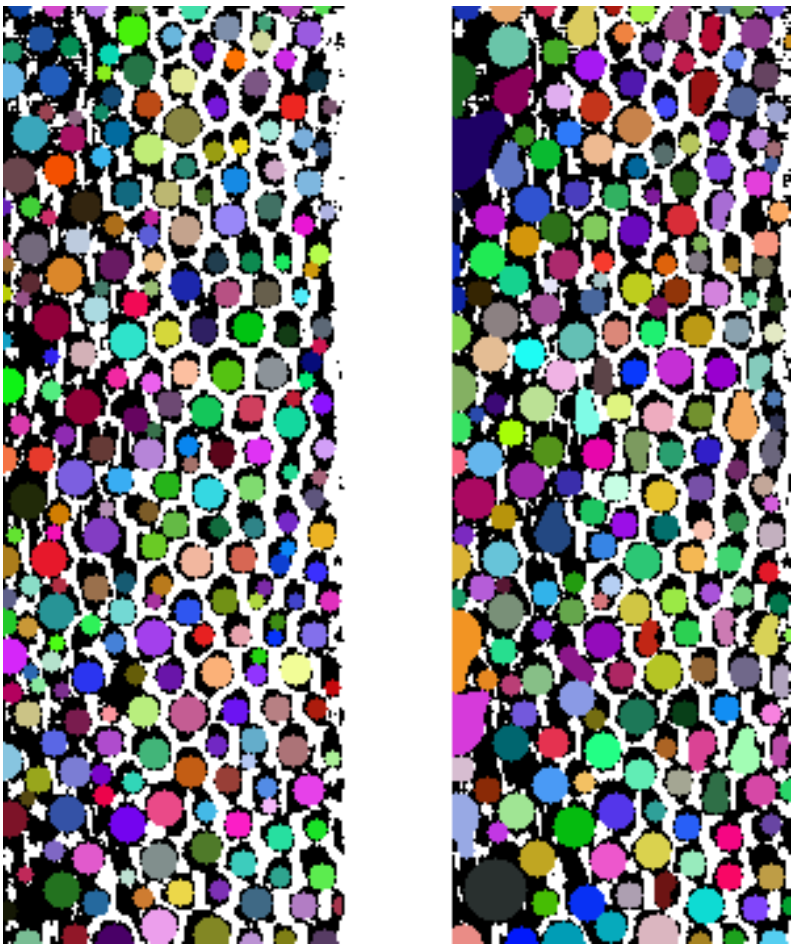


Fig. 3. Comparison of the cell body detection: on the left before merging the neighbor circles, on the right – after this procedure (colors are not corresponding on the two images)

tours – snake [7]. We use here its discrete version where the evolving curve minimizing its energy is represented by the collection of points given the local energy. Each previously marked region defines (by its external points) the initial form of the snake. All the snakes evolve on the original image with the full intensity range – its binary form is used only for the initialization. We use three energies and one balloon force (described below) that are applied to each potential (being examined) snake point (snaxel) position. The following notation is used: p – examined position of the snaxel, p_{prev} and p_{next} – neighbor snaxels (fixed while p is examined), $v_{prev} = \overrightarrow{p\ p_{prev}}$ and $v_{next} = \overrightarrow{p\ p_{next}}$) – vectors from p to its neighbors.

1. *Image energy* given by:
$$E_{image}(p) = I(p), \tag{1}$$
where $I(p)$ is the intensity level in the p position. Since the cell borders are darker (lower intensity), the snake tends to them.
2. *Internal energies* controlling the snake appearance:
   – *smoothness energy* promoting the linear location of snaxels and expressed as a length of sum of two normalized vectors v_{prev} and v_{next} (for the collinear points it has its minimum equal to zero):
$$E_{smoothness}(p) = \left| \frac{v_{prev}}{|v_{prev}|} + \frac{v_{next}}{|v_{next}|} \right|; \tag{2}$$
   – *regularity energy* favorizing the equal distance between all the snaxels and expressed as the two square differences between a mean inter-snaxel distance avg and the length of vectors v_{prev} and v_{next}, divided by the square avg:
$$E_{regularity}(p) = \frac{(avg - |v_{prev}|)^2 + (avg - |v_{next}|)^2}{avg^2}. \tag{3}$$
   .
3. *Balloon force* [8] responsible for pushing out the contours (initially located inside the cells). This force does not influence the p position optimization (with the above energies) – instead it shifts all the examined neighborhood of p by a vector $v_{balloon}$ perpendicular to the difference $v_{next} - v_{prev}$ and directed outward. Its length is the method parameter.

Evolution of snakes The process of minimization of the snake energy is decomposed on the separate snaxels – no global energy is formulated and for every snaxel p the local energy $E(p) = E_{image}(p) + E_{smootheness}(p) + E_{regularity}(p)$ is analyzed independently. In every iteration each snaxel neighborhood (shifted by $v_{balloon}$ as described

above) is examined and if a position with lower local energy is found – the snaxel is moved there. Because there are multiple evolving contours on the image (one for each cell) we impose a condition forbidding each snake to grow inside any other one. To avoid the curve discretization problems two procedures are introduced to keep the inter-snaxel distance reasonable (few pixels):

- each two neighbor snaxels too close one to another are replaced by a single one;
- if two neighbor snaxels are too distant one from another a new one is added in the middle of them.

The evolution stops when there is no more moving snaxels (all snakes reached their local optima). Two examples of automatic segmentations are given on Fig. 4.

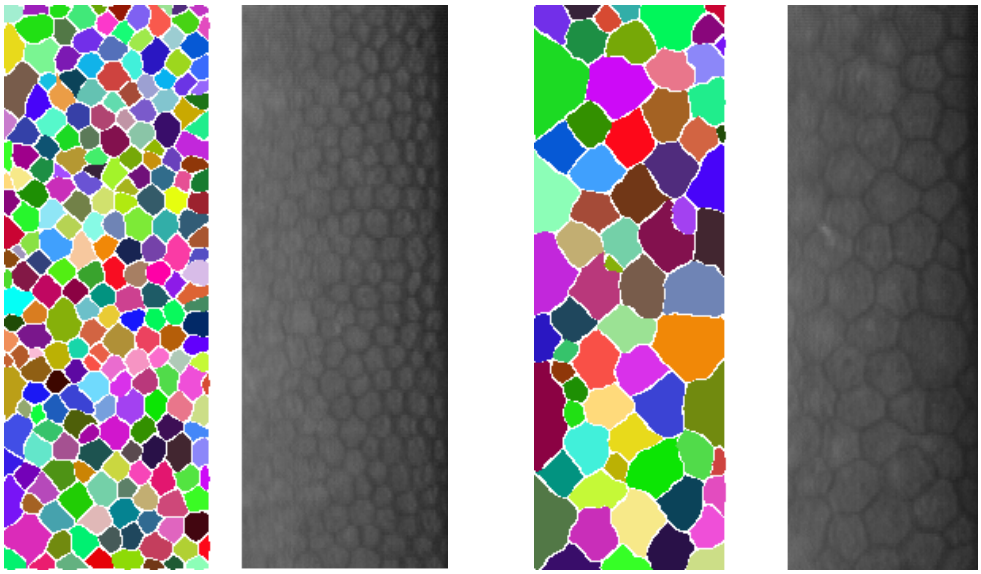


Fig. 4. Examples of the automatic cell segmentation

2.5 Implementation issues

The entire method has been implemented within the MESA framework [9]. It helps in designing and evaluating of new segmentations methods based on the deformable models. Its on-line version is accessible (mesa.wi.pb.edu.pl) – however its desktop version was used here (incorporation of the presented approach is planned). All the

standard operations (e.g. image loading, visualization, segmentation procedure) are offered there. The user is only responsible for defining the basic elements of his snake (writing new Java classes). In this work all three custom components were used:

- *model* replacing the standard one since multiple contour evolves in the image instead of the single one;
- *energies* defining the three ones as described in Section 2.4;
- *extensions* responsible for correcting the snake topology (adding and removing snaxels).

The segmentation of a single image was of order of seconds (after some optimization on controlling of the overlapping snakes) on a standard PC machine.

3. Experimental validation

Verification of quality of the medical image segmentation is not a trivial task. The most important problem is usually lack of the ground truth location and shape of the segmented object. A manual delineating by an expert can provide such a reference, but inter- and also intra-operator variabilities are very often significant. It is also time-consuming and tedious. Nevertheless, this procedure is sometimes the only possibility to quantitatively asses the precision of the method. In our work this approach was also selected and one of the authors (not being a medical expert) manually drew the cell contours of two chosen images (Fig. 5 and 6). He took into account only well visible cells. Then the presented method segmented the cells on the same images. Both segmentations took 10 snake iterations.

Having such the reference we choose two segmentation quality indices (after [10]):

- Overlap Error (*OE*)

$$OE(A,B) = 100(1 - (|A \cap B|/|A \cup B|)), \tag{4}$$

  where A and B are the pixel sets (in our work representing the cell bodies: segmented by our method and manually marked by the operator), value 0 characterizes two sets completely overlapping, value 100 – two separate sets;
- Relative Volume Difference (*RVD*)

$$RVD(A,B) = 100((|A| - |B|)/|B|), \tag{5}$$

  A and B meaning as above, value 0 means that the two compared sets have the same size (and says nothing about the set overlapping), negative value indicates undersegmentation, positive one – oversegmentation.

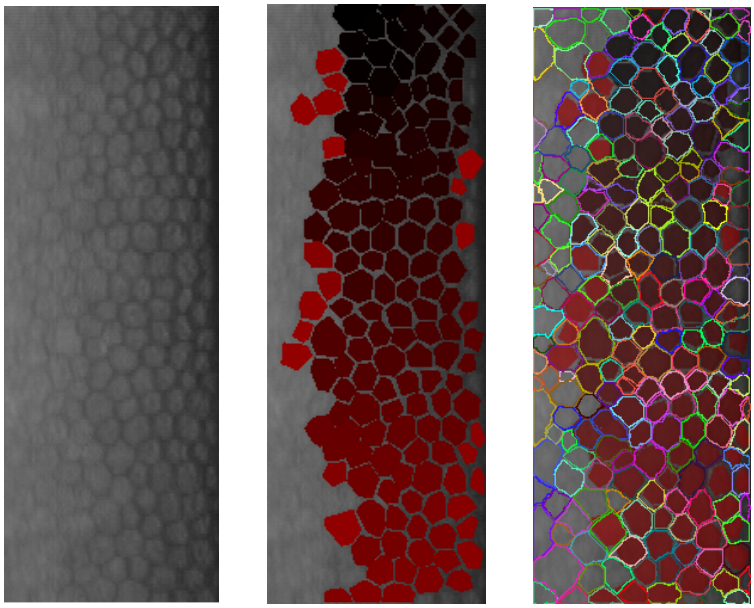


Fig. 5. Input image 1 (left), its manual segmentation – about 150 cells (center) compared with the automatic segmentation (right)

These indices are defined to be used with a single segmented object. In our cases there are multiple cells, so two problems are to be resolved. The first one is to find the correspondence between cells in two such images. This is done by finding for each reference cell an automatically segmented one with the highest overlapping ratio. The second problem is to gives one value (or at least few ones) characterizing the whole image and not separate values for each cell. So we give here: mean, median, minimum and maximum values for every index. The results are presented in Tables 1 and 2. The value distributions are also presented Fig. 7 and 8.

Table 1. Overlap Error statistics for the two selected images

Image	Median	Mean	Minimum	Maximum
Image 1 (150 cells)	22.34	24.92	7.12	86.18
Image 2 (56 cells)	22.41	26.85	7.79	71.47

As it can be observed, for the most of cases the cells are detected correctly. However one can see also some problems:

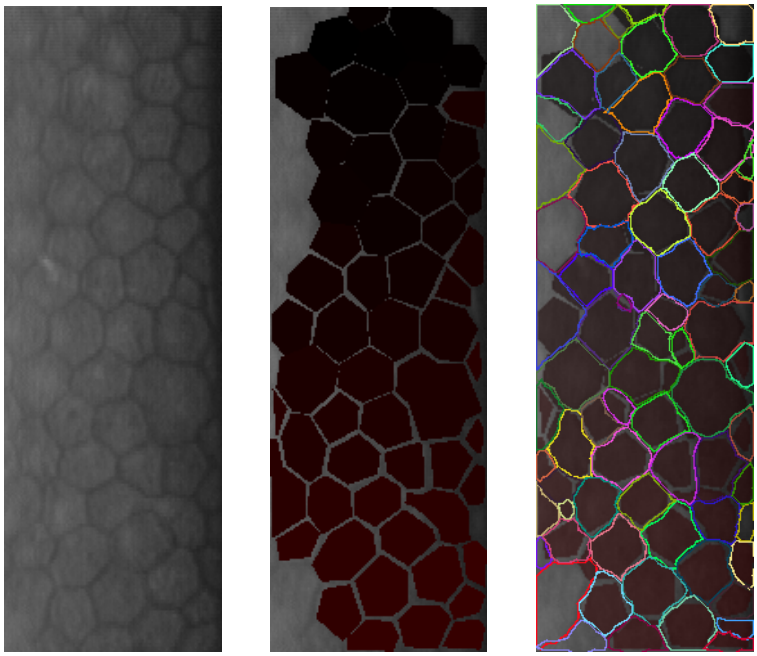


Fig. 6. Input image 2 (left), its manual segmentation – about 50 cells (center) compared with the automatic segmentation (right)

Table 2. Relative Volume Difference statistics for the two selected images

Image	Median	Mean	Minimum	Maximum
Image 1 (150 cells)	29.76	39.07	-48.83	363.83
Image 2 (56 cells)	5.35	35.25	-62.14	452.34

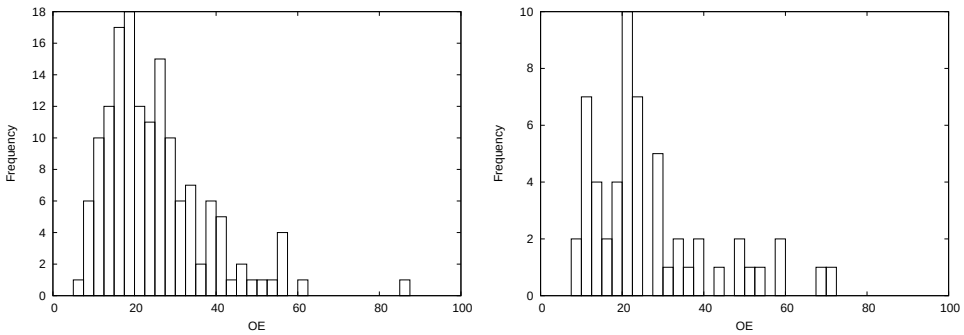


Fig. 7. Overlap Error histograms for image 1 (left) and image 2 (right)

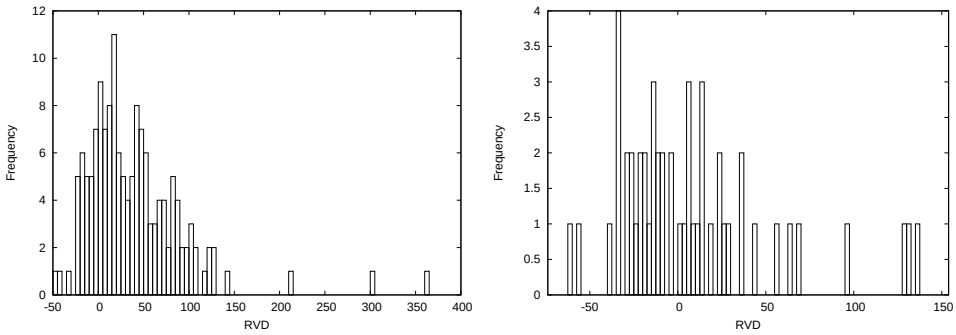


Fig. 8. Relative Volume Difference histograms for image 1 (left) and image 2 (right)

- sometimes one cell is segmented as two separate contours;
- some contours do not reach the cell borders, stopping their evolution inside the cells;
- some contours do not stop on the cell borders and they overpass them.

The first problem means that the detection of the cell centers does not work correctly there, marking two separate circles inside one cell and not merging them. Since the snake topology is fixed it results in two segmented cells inside the actual one. The next two problems show how hard is to balance the snake energies. When the balloon force is too weak (having too low weight in comparison to other forces, i.e. image and internal ones) to push the snake enough to reach the border, the contour evolution stops prematurely. But in the same time this weight is too high in other situations and it pushes the contour too much what results in overpassing the border. This behavior will be investigated more to improve the method.

The problems are reflected by the Overlapping Error values (especially the median and mean ones) what leaves a place to further working on the method. The positive Relative Volume Error shows a slight oversegmentation, but it is not very important.

In order to give even introductory comparison with other approaches (to be continued in future works) we took a single image from [1] and applied our method. The results can be visually compared Fig. 9.

4. Conclusions and future works

In this article we have presented the automatic method segmenting the corneal endothelial cells form microscopic images. The first results are promising, it detects

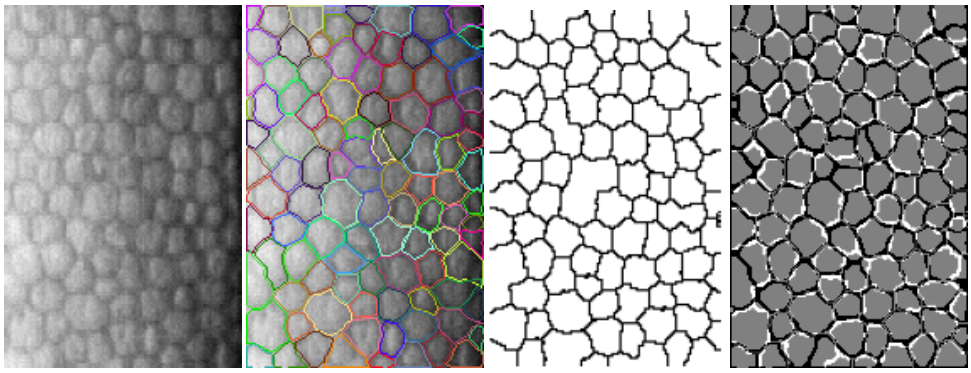


Fig. 9. Comparison of the presented method and [1]: original image (first), our segmentation superimposed on the original image (second), binary result from [1] (third), two segmentations on one image (fourth) - in gray common area, in black our segmentation, in white - [1].

even little visible cells and works quite fast (order of seconds on a standard machine). The implementation, for the moment limited to the standalone version, is realized in the technology (MESA environment) that allows an easy porting to the on-line web application and can be make available for the public use.

However there is still some work that can be done here. The results have been not yet seriously compared with the other existing methods and it is difficult to say where our approach is between them in term of quality of segmentation. Only two images (of rather good quality) was used to quantitatively assess the segmentation accuracy and the tedious manual cell delineation was done by the non expert. The parameter choice has not been extensively studied as well. Some of the parameters seem not critical (the constant in the adaptive thresholding) but others (the weights balancing the snake energies, especially the balloon one) have to be adjusted carefully and even then fail in some configurations. An iterative incrementation of the balloon force could solve the problems of not reaching some cell borders and overpassing others in the same image.

5. Acknowledgments

This work was supported with the Bialystok University of Technology Grants S/WI/2/2013 and W/WI/5/2014. We would like also to thank Jolanta Gronkowska-Serafin from Pomeranian Medical University for the cell images and Adam Piórkowski from AGH University of Science and Technology for making us interested in this problem and for other help.

References

[1] Piórkowski, A., Gronkowska-Serafin, J.: Towards Automated Cell Segmentation in Corneal Endothelium Images. Image Processing & Communications Challenges 6 Advances in Intelligent Systems and Computing 313, 2015, pp 179–186.

[2] Vincent, L.M., Masters, B.R.: Morphological image processing and network analysis of cornea endothelial cell images. Proc. SPIE vol. 1769, Image Algebra and Morphological Image Processing III, 1992, pp. 212–226.

[3] Mahzoun, M., Okazaki, K., Mitsumoto, H., Kawai, H., Sato, Y., Tamura, S., Kani, K.: Detection and complement of hexagonal borders in corneal endothelial cell image. Medical Imaging Technology 14(1), 1986, pp. 56–69.

[4] Sanchez-Marin F.J.: Automatic segmentation of contours of corneal cells. Computers in Biological and Medicine, 29(4), 1999, pp. 243–58.

[5] Khan, M.A.U., Niazi, M.K.K., Khan, M.A., Ibrahim, M.T.: Endothelial cell image enhancement using non-subsampled image pyramid. Information Technology Journal 6(7), 2007, pp. 1057–1062.

[6] Bullet, J., Gaujoux, T., Borderie, V., Bloch, I., Laroche, L.: A reproducible automated segmentation algorithm for corneal epithelium cell images from in vivo laser scanning confocal microscopy. Acta Ophthalmologica 92(4), 2014, pp. e312–6.

[7] Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International Journal of Computer Vision 1 (4), 1988, pp. 321–331.

[8] Cohen, L.D.: On active contour models and balloons. CVGIP: Image Understanding, 53(2), 1991, pp 211–218.

[9] Reska, D., Jurczuk, K., Boldak, C., Kretowski, M.: MESA: Complete approach for design and evaluation of segmentation methods using real and simulated tomographic images. Biocybernetics and Biomedical Engineering 34(3), 2014, pp. 146–158.

[10] Badakhshannoory, H., Saeedi, P.: A model-based validation scheme for organ segmentation in CT scan volumes. IEEE Transactions on Biomedical Engineering, 58(9), 2011, pp. 2681–2693.

AUTOMATYCZNA SEGMENTACJA KOMÓREK ŚRDBŁONKA ROGÓWKI OKA PRZY POMOCY AKTYWNYCH KONTURÓW

Streszczenie: W artykule zaprezentowano autorską automatyczną metodę segmentacji komórek śródbłonka rogówki oka z obrazów mikroskopowych. Metoda używa wielu aktywnych konturów zainicjalizowanych wewnątrz komórek za pomocą adaptacyjnego progowania i ograniczonych w swoim rozroście tak, aby nie pokrywać się. Metoda został zaimplementowana w środowisku MESA przeznaczonym do rozwoju i ewaluacji technik segmentacji. Jakość segmentacji została oszacowana na rzeczywistych obrazach mikroskopowych w odniesieniu do ręcznie zaznaczonych konturów komórek.

Słowa kluczowe: przetwarzanie obrazów, segmentacja komórek, aktywny kontur

Artykuł zrealizowano w ramach pracy statutowej Politechniki Białostockiej nr S/WI/2/2013 oraz pracy nr W/WI/5/2014.

TEXTURE ANALYSIS AS A TOOL FOR MEDICAL DECISION SUPPORT. PART 1: RECENT APPLICATIONS FOR CANCER EARLY DETECTION

Dorota Duda

Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

Abstract: A great number of works have been devoted to developing different medical decision support systems, based on an image data. Such systems combine a wide range of methods for digital image analysis and interpretation. It has been proven that one of the most useful sources of information encoded in the image is its texture. Texture Analysis (TA) provides many important discriminating characteristics, not normally perceptible with visual inspection. With properly chosen TA methods, an image-based diagnosis could be considerably improved. However, the choice of the methods is not an easy task and often depends on the nuances of each diagnostic problem.
The present work provides an overview of the most frequently used methods for texture analysis (statistical, model-based, and filter-based) and shows their advantages and limitations. It also includes an overview of texture-based medical decision support systems, recently proposed for cancer detection and classification.

Keywords: medical imaging, image analysis, texture characterization, feature selection, computer aided diagnosis, CAD, medical decision support

A list of abbreviations is given at the end of this article.

1. Introduction

Different imaging modalities are presently available to assist clinicians in the detection and the diagnosis of human pathologies. Among them are: Computed Tomography (CT), Positron Emission Tomography (PET), Single Photon Emission Computed Tomography (SPECT), Magnetic Resonance Imaging (MRI), Ultrasonography (US), or Optical Imaging. With a constant improvement of image acquisition devices, the

Advances in Computer Science Research, vol. 11, pp. 61-84, 2014.

amount of diagnostic information obtained within a single study has considerably increased. In such a situation, the interpretation of image content based only on its visual inspection goes far beyond the human abilities. Indeed, an unarmed human eye can distinguish barely 100 gray levels, whereas the gray-scale images obtained nowadays (still commonly used) can encode many thousands of gray levels.

Since an experienced physician is not able to read all the useful image data without any additional equipment, a great deal of work has been devoted to develop different methods for (semi)automatic medical image analysis, interpretation, and recognition. As a result, many Computer-Aided Diagnosis (CAD) systems have been proposed over the past two decades. These systems combine a broad range of image analysis methods (including image segmentation and tissue characterization techniques), feature selection, and classification algorithms. The literature describes many examples of image-based CAD systems that have already found their application in various problems, concerning different organs and/or different imaging modalities. Among them, we can enumerate the systems for hepatic diseases recognition, based on the CT images (they will be detailed in the second part of the work [1]), on the MR images [2], or on the contrast-enhanced ultrasonography [3]. Another example concerns the breast lesion classification based on the ultrasound images [4–6] or on the Dynamic Contrast Enhanced (DCE) MR images [7–13]. The recognition of the prostate cancers on the basis of the DCE-MR images were also studied [14–22]. An exhaustive overview of the CAD systems for lung cancer recognition, based on the CT and/or the PET images can be found in [23]. Finally, the usefulness of CAD systems for brain tumor detection and classification from MRI were investigated in [24–30].

The advantage of CAD systems is that they improve considerably the image-based diagnosis, which reduces the necessity of using other methods, such as a fine needle aspiration biopsy or a surgical biopsy. Moreover, medical imaging is becoming continuously cheaper, faster, and less wasteful. Finally, it is certainly much less invasive (or even non-invasive) in comparison with many gold standard procedures.

The choice of the most appropriate methods for (semi)automatic image analysis and interpretation is not a trivial task. The works presented so far have shown that each diagnostic problem requires practically a re-validation of methods previously tested in similar domains. However, a good selection of such methods is crucial to guarantee the satisfactory tissue recognition. In 1979, Haralick stated that one of the most important sources of analyzed image region could be its texture [31]. It characterizes the spatial relationships between gray levels describing pixels within a considered image region (so called "Region of Interest", commonly abbreviated as ROI). Since then, numerous review studies have shown that texture analysis could

be highly useful in various problems related to the recognition of medical (and also nonmedical) images [32–38], as it provides a crucial information in terms of tissue discrimination. Moreover, many works have revealed that digital image analysis enables to detect the high order texture properties not accessible to visual inspection (see, for example, [31, 39–41]).

The aim of present work is to review the most frequently used methods for extraction of textural features. Three main TA approaches are considered: statistical, model-based, and filter-based. The remaining sections are organized as follows. In the next section, different methods of texture analysis are presented. Section 3 presents some recent results of their application in several diagnostic domains. Finally, in the last section, the main conclusions are drawn.

2. Overview of texture analysis methods

2.1 Gray Level Histogram

Features derived from a Gray Level Histogram (GLH) are based solely on the distribution of pixel gray levels and do not consider the relationships between neighboring pixels. They provide knowledge on the most and the less often occurring gray levels, on the concentration of the gray levels around their average, or on the degree of asymmetry in their distribution. On the contrary, they do not contain any information neither about the possible direction of the texture, nor about its structure. Nevertheless they are often used because of their invariability to translation or rotation, simplicity, and low memory and time requirements for their calculation. The most popular first-order features are:

- *range of gray levels*,
- *mean of gray levels* (measure of image brightness),
- *median gray level* (the second quartile),
- *gray level energy* (indicates how the gray levels are distributed),
- *variance of gray levels* (characterizes the distribution of gray levels around the mean),
- *gray level skewness* (measures the asymmetry of the gray-level histogram),
- *gray level kurtosis* (indicates the relative flattening of the gray-level histogram),
- *coefficient of variation* (the ratio of the standard deviation to the mean).

2.2 Co-Occurrence Matrices

Co-Occurrence Matrices (COM) were introduced by Haralick et al. [42]. This method consists in analyzing all the possible pairs of pixels, spaced apart by a fixed distance,

d, and arranged in a given direction, θ. Four pixel alignment directions (0°, 45°, 90°, and 135°), and different distances between the pixels in pairs can be considered. Typically, d takes small values. A combination (d,θ) determines thus unequivocally the relative position of pixels composing the pairs to analyze. A co-occurrence matrix $C(d,\theta)$ is of size $G \times G$, where G is the number of gray levels possible to be encoded in an image. Each element of co-occurrence matrix, c_{ij} $(i, j = 0, ..., G-1)$, represents a probability of occurrence of a pair of pixels with gray levels of i and j, for the first and for the second pixel, respectively. Several texture characteristics can be obtained on the basis of a co-occurrence matrix [42]:

- *energy* or *angular second moment* (measure of homogeneity of gray levels characterizing the pixels within an analyzed ROI),
- *contrast* or *inertia* (measure of contrast or local variations in pixel gray levels),
- *inverse difference moment* (measure of local homogeneity),
- *entropy* (quantifies a degree of randomness of the pixel gray levels),
- *correlation* (measures linear dependency of gray levels on neighboring pixels).

Other features can be calculated from the sums of probabilities that relate to specified intensity sums or differences [34]. In practice, this requires the construction of vectors whose components are the co-occurrence probabilities of pairs of pixels with a determined sum or difference of the gray levels. All possible sum / difference values are taken into account. The probabilities form a vector and are sorted in increasing order of corresponding sum or difference values. Some features derived from such vectors are:

- *sum average*,
- *sum variance*,
- *sum entropy*,
- *difference average*,
- *difference variance*,
- *difference entropy*.

Conners et al. [43] proposed two additional COM-based features, that measure the skewness (the lack of symmetry) of the matrix $C(d,\theta)$:

- *cluster shade*,
- *cluster prominence*.

For non-directed textures, several values of the same feature, corresponding to different arrangement directions, θ, but obtained for the same distance between pixels in pairs, can be averaged. Often feature values corresponding to different distances, d, are also averaged.

2.3 Run Length Matrix

Run Length Matrix (RLM) features are based on probabilities of pixel runs of each possible length, arranged in a certain direction [44]. Like in previous method, four standard directions of pixel runs are considered, $\theta = 0^\circ$, 45°, 90°, or 135°. A run-length matrix, $R(\theta)$, has G columns and M rows, where G is the number of image gray levels, and M is the maximum length of pixel run which can exist in an analyzed image region. The element r_{mg} ($m = 1,...,M$, and $g = 0,...,G$) of a run length matrix $R(\theta)$ is the number of existing pixel runs of a gray level g, having a length of m, and oriented in a direction θ.

Galloway initially proposed the following features, derived form a run length matrix.

- *short run emphasis*,
- *long run emphasis*,
- *gray level non-uniformity (distribution)*,
- *run length non-uniformity (distribution)*,
- *fraction of image in runs*.

The proposition of two additional features can be found in [45]:

- *low gray level runs emphasis*,
- *high gray level runs emphasis*.

Yet another work [46] proposes to use a *run length entropy* as a texture feature.

Also in this method, the values of the same feature corresponding to different directions of pixel runs can be averaged.

2.4 Gray Level Difference Matrices

Gray Level Difference Matrices (GLDM) are constructed with consideration of only the absolute values of differences between the gray levels of pixels, still considered in pairs [47]. Similarly to a COM-based method, also here four pixel alignment directions θ, and different distances, d, between the pixels in pairs can be considered. Further, all possible absolute differences in gray levels that can be encoded in the image are taken into account. For each absolute difference, the probability of finding a pair of pixels with just such a difference in the gray levels is calculated. The probabilities sorted in increasing order of corresponding absolute gray level differences form a vector $l(d,\theta) = [l_0, l_1, ..., l_{G-1}]^T$, where G is the number of gray levels possible to be encoded in an image.

Five textural features can be derived from the $l(d,\theta)$ vector:

- *mean* (measures a level of texture diversity),
- *energy* or *angular second moment* (measure of homogeneity of gray levels),
- *contrast* or *inertia* (measure of intensity contrast between a pixel and its neighbors),
- *inverse difference moment* (measure of the local homogeneity),
- *entropy* (quantifies a degree of randomness of the pixel gray levels).

All above features could be averaged when they are calculated for different θ and/or d parameters.

2.5 Gradient Matrices

Gradient-based features were introduced by Lerski et al. [48]. A gray-level gradient at a particular image point is a function of the differences between the gray levels of its neighboring pixels, aligned on vertical and horizontal lines passing through the point. Often a neighborhood of 3×3 pixels or 5×5 pixels is considered. The Gradient Matrix (GM) contains the values of the absolute gradient at each point of an analyzed image region, excluding its boundaries.

Features derived from a gradient matrix are the following:

- *mean*,
- *variance*,
- *skewness*,
- *kurtosis*,
- percentage of pixels with nonzero gradient.

They can provide the information on the uniformity, homogeneity, or the roughness of the texture. They may also indicate the presence or absence of edges.

2.6 Texture Feature Coding Method

Texture Feature Coding Method (TFCM) was proposed by Horng et al. [49]. The method consists of three steps. First, an image is transformed. The transformation consists of assigning to each pixel (except for the pixels located at the edges of a considered ROI) a value that measures a degree of heterogeneity (of variation, of diversity) of the local gray levels of its neighbors. The authors call this measure a "Texture Feature Number" (TFN). Only a neighborhood of 3×3 pixels is considered. Afterwards, a histogram of Texture Feature Numbers, and co-occurrence matrices are constructed, based on a transformed image. Finally, several texture descriptors are obtained, either from a TFN histogram:

– *coarseness*,
– *homogeneity*,
– *mean convergence* (indicates how close the texture approximates the mean),
– *variance* (measures deviation of TFNs from the mean),

either from a TFN-based co-occurrence matrix:

– *entropy*,
– *code similarity* (assesses the density of the same TFNs in a 3×3 neighborhood),
– *resolution similarity* (measures the local homogeneity of TFNs).

2.7 Autocorrelation Coefficients

Autocorrelation Coefficients (AC) [50] expresses the correlation of the gray levels describing pixels within a defined neighborhood. It is the function of the vertical (Δx) and the horizontal (Δy) distance between the considered pixels in pairs. Usually such distances are relatively small. In order to normalize the autocorrelation coefficients a gray level of each pixel is decreased by the mean gray level. Normalized autocorrelation coefficients are independent of the image brightness, and can be regarded as textural features. They can provide knowledge on the spatial relationship between the texture patterns, and the average size of texture patterns.

2.8 Fractal Model (FM)

Fractal Model (FM) was described in several works [51–53]. Unfortunately, each of them gives different definitions of a fractal object. Mandelbrot characterized fractals as self-similar objects, whose parts are similar to the whole, and whose topological dimension is not an integer [51]. The fractal dimension of an object reflects the extent to which this object fills the space or the rate of its diversity, the degree of irregularity of the object.

A gray-level image can be considered as a topographic surface in three-dimensional space, where two dimensions are those of the image plane and the third one (height) is the gray level of image pixels. The fractal dimension of such a surface can be used as a texture descriptor. It can measure the irregularity and the roughness of the texture. Irregular surfaces (corresponding to a diversified textures) have relatively high fractal dimension, while the smooth ones are characterized by the low fractal dimension.

So far, several methods for calculating a fractal dimension of a texture have been reported in the literature. Among them we can mention: the approaches based on the fractional Brownian motion model [54, 55], the box-counting methods [56, 57], the mass-radius methods [58], the wavelet-based methods [59], and others [60–63].

2.9 Discrete Wavelet Transform

Discrete Wavelet Transform (DWT) of an image consists in its convolution with two filters, the low-pass and the high-pass one, separately throughout the image rows and separately throughout the image columns. The Mallat algorithm for DWT [64] divides the image into the four sub-images, each of which is linearly two times smaller than the decomposed image. Each sub-image can then be divided in the same manner. Thus multiple resolution levels are obtained. Four sub-band images created at each decomposition step are denoted: d^{LL}, d^{LH}, d^{HL}, and d^{HH}. They are created by applying, respectively: the low-pass filter for rows and columns (LL), the low-pass filter for rows and the high-pass filter for columns (LH), the high-pass filter for rows and the low-pass filter for columns (HL), the low-pass filter for rows and columns (LL). The component d^{LL} is created by calculating the average of disjoint groups of 2×2 pixels, using Haar transform. It is therefore an approximation (simplified representation) of the transformed image. Further sub-bands represent the vertical (LH), the horizontal (HL) and the diagonal (HH) image information. On the basis of them an edge energy at three directions can be calculated. It is also possible to analyze the energy distribution in each sub-band image. The image d^{LL} is used only for DWT calculation at the next scale.

2.10 Laws' Texture Energy measures

Laws' Texture Energy (LTE) measures [65] are useful for estimating the frequency of the image elements, such as ripples, edges, or spots. Laws proposed to transform the images using linear filters. During the transformation, each image pixel is assigned a value that is a combination of initial gray levels of pixels belonging to a neighborhood of a transformed pixel. Usually two types of neighborhood are considered: 3×3 pixels and 5×5 pixels. The weights of the neighboring pixels are defined by a zero-sum convolution matrix (so-called Laws' mask). For each pair of asymmetric masks, the resulting images could be added. In this case, images obtained with an application of symmetric masks are multiplied by two. On the basis of a transformed image, the entropy can be calculated. Also, the filtered images can be once again subjected to further transformation, that results in creation of texture energy images. Finally, the features such as: *mean*, *variance*, *skewness*, and *kurtosis* can be calculated from the resulting images.

3. Recent applications for cancer early detection

The following describes the most recently published works, considering textural features as useful tissue descriptors. Each of the systems aims at cancer detection and

characterization from MRI images. Three main diagnostic problems are considered: prostate cancer diagnosis, brain tumor diagnosis, and breast lesion classification.

3.1 Prostate cancer diagnosis

The (semi)automated CAD systems using a texture analysis as a prostatic tissue descriptor are still not broadly developed. Most of the existing works are based mainly on the pharmacokinetic models, employing the signal-to-time curves, in order to find perfusion parameters (e.g. [18]). Such models give an information about the propagation of contrast product, extracted from the T1-weighted DCE MRI sequences. Other systems exploit also diffusion weighted image features (e.g. [66]), not based on textural properties. However some works exploiting the potential of TA in prostatic tissue differentiation have appeared in recent few years [17, 19–22, 66, 67].

Lopes et al. [17] employed fractal and multifractal textural features to characterize prostatic tissues on T2-weighted MR images. Their system was able to recognize two types of tissue: a tumorous and a non-tumorous one. The fractal dimension was computed using the variance method. The multifractal spectrum was estimated by a modified multifractional Brownian motion model. The classification was performed with Support Vector Machines (SVM) [68] and an adaptive boosting voting scheme (AdaBoost) [69]. The best result was obtained by AdaBoost classifier: 85% and 93%, for sensitivity and specificity, respectively. Moreover, the results obtained by the proposed system were better than those corresponding to the application of classical textural features, derived from co-occurrence matrices, wavelets, or Gabor filters [70]. The fractal method turned also to be most robust against signal intensity variations.

Niaf et al. [19] analyzed simultaneously different MRI sequences (T1-, T2-, and diffusion-weighted) in order to differentiate between: (i) malignant vs benign prostatic tissues, and (ii) malignant vs nonmalignant, but suspicious ones. The CAD system proposed in their work combined functional parameters, extracted from DCE images, together with textural features derived from the three considered MRI sequences. First-order and second-order (COM-based) textural features were utilized. Four classifiers were applied: nonlinear SVM, Linear Discriminant Analysis (LDA) [71], k-Nearest Neighbors (k-NN) [72] and Bayesian one [72]. The system performances were assessed by the areas under the Receiver Operating Characteristic (ROC) Curves (AUC) [73]. The best result was achieved with the SVM classifier and was 0.89 and 0.82, for the first (i) and for the second (ii) discrimination problem, respectively.

Peng et al. [66] assessed the utility of T2-weighted MRI texture features and diffusion weighted image features in distinguishing prostate cancer from normal tissue.

Here, the LDA and the areas under the ROC curves were used to evaluate the performance of each feature. Among many tested texture characteristics, the *sum average* turned to be the best feature. Nevertheless it did not outperform some of the *Apparent Diffusion Coefficient* (ADC) features, that measure the magnitude of diffusion (of water molecules) within a tissue. The combination of three features (*sum average* and two ADCs) yielded AUC values of 0.94 and 0.89 on the images acquired with the Phillips and the GE scanner, respectively.

Duda et al. [20] proposed to analyze simultaneously triplets of prostate MR images, corresponding to the same prostate slice, but derived from different image series: the contrast-enhanced T1-, the T2-, and the diffusion-weighted one. Two classes of prostatic tissue were differentiated: tumorous and healthy. Six different texture analysis methods were used: GLH-, COM-, RLM-, GM-, AC-, and FM-based. Their ability of characterizing prostatic tissue was assessed with three classifiers: Logistic Regression (LR) [74], Neural Network (NN) [75, 76] and SVM. The 10-fold cross-validation [71] was used to assess the classification accuracies. The best overall classification result exceeded 99% and corresponded to the application of the SVM classifier.

Ginsburg et al. [67] tried to predict the probability of developing biochemical recurrence risk (associated with raised risk of metastases and prostate cancer-related mortality) following the radiation therapy. In their work they evaluated the efficiency of different textural features, extracted from the T2-weighted images. They considered: first-order statistical features, gradients (involving image convolutions with Sobel and Kirsch operators [77]), co-occurrence matrices-based, and Gabor wavelet features [78]. For each feature, its prognostic potential and its contribution to classification results was assessed. As a classifier, a Logistic Regression was used. Despite of poor resolution of images, available for the experiments, the area under the ROC curve for the best three features (the Gabor wavelet ones) reached 0.83.

Litjens et al. [21] developed a fully automated computer-aided detection system, which was able to differentiate between patients with and without prostate cancer. Their study based on: T2-weighted, proton density-weighted, dynamic contrast enhanced, and diffusion-weighted images. Thus several types of features could be used. They were: based on signal intensity, representing pharmacokinetic behavior, anatomical features, blobness, and finally – texture descriptors, based on Gaussian texture models. These latter characteristics contributed to the ability of the system to achieve a performance comparable to the one achieved by radiologists.

Finally, Molina et al. [22] proposed a system that combined different features (anatomic, textural, and functional) in order to recognize three classes of prostatic tissue: cancerous, unhealthy non-cancerous, and healthy. Three different series of

MR images were considered in the work: T2-weighted, Dynamic-Contrast Enhanced Plasma Flow (DCE-PF) and DCE Mean Transit Time (DCE-MTT). Nevertheless, the texture information was extracted from only the structural T2-weighted images. In the work, three groups of features were used: the first-order statistical descriptors, and the second-order ones, derived from: a Neighboring Gray Level Dependence Matrix (NGLDM) [79] and a Neighborhood Gray-Tone Difference Matrix (NGTDM) [80]. Experiments showed that the use of texture descriptors could provide more relevant discriminative information than the considered functional parameters. The average sensitivity and specificity obtained with the system was of 84.46% and 78.06%, respectively.

3.2 Brain tumor diagnosis

A few algorithms have been recently developed for brain tumor detection and classification, based on MR images of different modalities.

In 2012, John [25] proposed a system for brain tumor classification from T2-weighted MR images. The system was able to recognize the three tissue types: normal, non-cancerous (benign) brain tumor and cancerous (malignant) brain tumor. The tissue characterization process consisted of two stages. First, the images were decomposed with the wavelet transform. Next – the co-occurrence matrix-based textural features were extracted from the LH and HL sub-bands of the first five levels of wavelet decomposition. Five textural features were considered: *energy*, *contrast*, *correlation*, *homogeneity* and *entropy*. Finally, they were fed into a Probabilistic Neural Network (PNN) [81] for further classification and tumor detection. The system achieved the classification accuracy of near 100%.

Patil et al. [26] tried to differentiate the four grades of Astrocytoma (from Grade I to Grade IV). Their approach consisted of several stages: image preprocessing, segmentation, feature extraction and classification. Feature extraction involved using the co-occurrence techniques, providing a set of 11 features. Finally, a Probabilistic Neural Network has been developed to differentiate between different grades of considered brain tumor. The overall accuracy of the system (obtained on the test set) was of 94.87%.

The system presented by Islam et al. [27] was designed for the detection and the segmentation of brain tumors from non-enhanced T1-weighted, T2-weighted, and FLAIR images. Two different tumor groups were considered in their study: astrocytoma and medulloblastoma. The tissue was characterized here using fractal and multifractal (based on fractional Brownian motion model) methods. The features corresponding to different modalities were fused. As a classifier, an extension of AdaBoost

algorithm was used. The system was tested on 14 patients with over 300 images and showed its high efficacy.

Jayachandran et al. [28] evaluated a system for detection and classification of brain tumors on T1-weighted post contrast (gadolinium-based) images. Their system was able to find tissue characteristics, to reduce the feature space, and to classify tissues into two categories: tumorous and non-tumorous one. Texture analysis was performed with the co-occurrence matrices. The Principal Component Analysis (PCA) [82] was used in order to reduce the feature space. Finally – the Fuzzy based Support Vector Machine was applied for a classification. Experiments were conducted on 80 brain MRI images. The proposed methodology resulted in quite high rates of correctly classified cases (more than 95%).

Sachdeva et al. [29] developed a system aimed at the differentiation of six tissue classes. They corresponded to the primary brain tumors: astrocytoma, multiform glioblastoma, childhood tumor – medulloblastoma, meningioma, secondary tumor – metastatic, and normal regions. Their analyses involved using post-contrast T1-weighted MR images. First, tumors were segmented with the content-based active contour model [83]. Then over two hundred intensity and texture features were used as tissue characteristics. Texture analysis considered: Laplacian of Gaussian (LoG) filters [77], co-occurrence matrices, rotation invariant Local Binary Patterns (LBP) [84], directional Gabor texture features [85], gray-level histogram, and rotation invariant circular Gabor features [86]. Due to the large number of candidate features, a feature space was reduced with the PCA. Then, an artificial Neural Network was applied in order to perform the classification. The robustness of the proposed system was tested using quite a large database (856 ROIs), with a partitioning of data into a training and a test set. The overall classification accuracy was of 85.23%.

Most recently, Tiwari et al. [87] assessed different groups of textural features, in terms of their ability to differentiate radiation necrosis (a radiation induced treatment effect) from recurrent brain tumors. In fact this task is very difficult to the human observer, because both pathological processes results in almost the same morphological appearance on standard MRI. So far, the diagnosis was possible only through a surgical intervention. The aim of the study was thus to find a set of features that could accentuate subtle differences between both pathologies, and – further – to determine which MRI protocol could provide the most discriminating information. Three MR image series were considered: T1-weighted, T2-weighted and FLAIR. The examined textural features were derived from: co-occurrence matrices, neighboring gray-level dependence matrices, Laplacian pyramids [88], Laws' texture energy measures, and Histogram of Gradient orientations (HoG) [89]. In total, 119 features were assessed. Each feature was assessed by Principal Component Analysis-based Variable

Importance Projection (PCA-VIP), developed by authors. Then a random-forest classifier [90] was used to differentiate between considered pathologies. The experiments showed that HoG, COM, and Laws' features were the most suitable ones for the problem solution. The best MRI image series turned out to be the contrast-enhanced T1-weighted one.

3.3 Breast lesion classification

The breast lesion characterization and differentiation from MRI images involves often a simultaneous analysis of images derived from different series, like T1-weighted (non-enhanced or contrast-enhanced), T2-weighted, diffusion-weighted, or others. In this case it is important not only to find potentially reliable pathology-related features, but also to combine properly the tissue descriptors corresponding to different image series. Here, textural features are often combined with other lesion descriptors: morphological, intensity kinetic features (based on signal-to-time curve), or shape descriptors.

Bhooshan et al. [8] combined textural features from both DCE T1- and T2-weighted MR images in order to recognize benign and malignant breast lesions. For the T1-weighted sequences, only the first post-contrast image was used for texture analysis. As texture descriptors, again the COM-based features were used. The contrast product propagation was characterized by typical kinetic parameters obtained from signal-to-time curves. The system was able to perform an automatic lesion segmentation, then – the features were automatically extracted. In the experimental stage, a stepwise feature selection was performed by Linear Discriminant Analysis. The selected features were merged by with Bayesian artificial Neural Network classifier. The leave-one-out cross-validation [71], and the areas under the ROC curve were used to assess the performance of tested sets of features. The experiments showed, that the combination of texture characteristics, obtained from both T1-, and T2-weighted images may outperform the conventional analysis of T1-weighted contrast-enhanced sequences. When all the features were considered, the best result was achieved. It gave the area under the ROC curve of 0.85.

Agner et al. [12] also tried to distinguish malignant from benign lesions. In their work, they compared several approaches to lesion characterization, giving different types of tissue descriptors: morphological features, signal intensity kinetic features, and textural features. The latter ones were based on gray-level histogram, gradients, and co-occurrence matrices. The study introduced a notion of "textural kinetics", that characterized texture evolution under contrast product propagation in DCE MRI. At first, textural features were calculated at each moment of contrast product propaga-

tion, and the "textural kinetics curve" was created basing on the set of feature values. Afterwards, a third order polynomial was fitted to such curve in order to characterize its shape. Four polynomial coefficients constituted the feature vector. Feature vectors were classified with the SVM and the AdaBoost classifiers. Experiments on 41 cases, showed that textural kinetics features outperformed the other ones (morphological, descriptors of signal intensity kinetics, and these based on "static" textures). The best classification accuracy was about 90%.

Nagarajan et al. [13] used a multi-image texture analysis for breast lesion classification. In order to differentiate two types of small lesions (benign and malignant) they analyzed simultaneously five post-contrast T1-weighted images. A multi-image texture was characterized by five values of the same textural feature. Each value corresponded to a different moment of contrast agent propagation. Only the COM-based textural features were considered. The tissue recognition was performed with Support Vector Regression and a fuzzy k-Nearest Neighbor classifier. The classifier performances were determined through the ROC analysis. The highest AUC value observed was of 0.82. Experiments also showed that textural features extracted from the third and fourth post-contrast image contributed the most to the correct tissue differentiation.

Recently, Cai et al. [10] combined dynamic contrast-enhanced and diffusion-weighted images (DWI), also to recognize benign lesions and malignant ones. The lesion regions were obtained with a semi-automated segmentation method. Then, four types of tissue descriptors were considered: kinetic, morphological, textural (co-occurrence matrix-based), and DWI features. In order to select the most robust features, a hybrid filter-wrapper algorithm [91] was applied. Finally, various classifiers (SVM, Bayesian, k-NN, LR) were used to evaluate the diagnostic performance of the selected features. The study comprised of 234 female patients. The classification accuracy was assessed with a 10-fold cross-validation and ROC characteristics. Finally, seven selected features (among them – three textural features) were found to be statistically different between the malignant and the benign groups, and their combination gave the highest classification accuracy – 93%.

Still in 2014, Pang et al. [11] presented a fully automated CAD system for the classification of malignant and benign masses. The system included a breast segmentation method, the mass segmentation method (described in [92]), feature extraction stage, feature selection (with the ReliefF [93] algorithm), and the SVM classifier. As tissue descriptors, morphological and textural features were used. Like in previous study, the texture analysis was performed using a co-occurrence matrix-based approach. A database comprised 120 cases. For the leave-one-out classifier assessment method, the accuracy was of 90.0%.

4. Conclusion

The present study enumerated the most commonly used methods for texture analysis. In addition to a brief description of the methods, it also included a short interpretation of the meaning of the parameters derived from each method. An overview of recently proposed works, considering textural features as reliable tissue descriptors in different classification problems, showed that the list of TA methods, presented in Section 2, is not exhaustive. The variety of methods proposed in the literature is far much larger. Experiments show, that each diagnostic problem, each image modality, may require the use of newer and newer procedures guaranteeing satisfactory classification results. Nevertheless, it could be noticed, that some groups of textural features show their huge potential more often than others. Among them the most powerful ones are the statistical features, obtained from the co-occurrence matrices. Such features were considered in almost all the quoted systems. The first-order statistics are less popular, however they are also tested because of their simplicity. Quite often a fractal model is used to find reliable texture characteristics. Also good are methods involving an image filtering.

To sum up the first part of the work, it can be concluded, that texture analysis has repeatedly demonstrated its valuable potential in the cancer early detection and differentiation. The implementation of many referred systems might certainly improve the image-based diagnosis, reducing the need for invasive procedures. The further development of imaging techniques, and the continuous work to improve the digital image-analysis methods may result in physicians more frequently refraining from the use of any invasive procedure.

Abbreviations

AdaBoost: Adaptive Boosting algorithm
ADC: Apparent Diffusion Coefficient
AUC: Area Under the ROC Curve
CAD: Computer-Aided Diagnosis
COM: Co-Occurrence Matrix
CT: Computed Tomography
DCE: Dynamic Contrast Enhanced (in MRI)
DCE-MTT: DCE Mean Transit Time (in MRI)
DCE-PF: DCE Plasma Flow (in MRI)
DWI: Diffusion-Weighted Imaging (in MRI)

DWT: Discrete Wavelet Transform
FLAIR: Fluid-Attenuated Inversion Recovery (in MRI)
FM: Fractal Model
GLDM: Gray Level Difference Matrix
GLH: Gray Level Histogram
GM: Gradient Matrix
HoG: Histogram of Gradient orientations
k-NN: *k*-Nearest Neighbors (classifier)
LBP: Local Binary Pattern
LoG: Laplacian of Gaussian
LR : Logistic Regression (classifier)
LTE: Laws' Texture Energy
MR: Magnetic Resonance
MRI: Magnetic Resonance Imaging
AC: Autocorrelation Coefficient
NGLDM: Neighboring Gray Level Dependence Matrix
NGTDM: Neighborhood Gray-Tone Difference Matrix
NN: Neural Network (classifier)
PCA: Principal Component Analysis
PCA-VIP: PCA-based Variable Importance Projection
PET: Positron Emission Tomography
PNN: Probabilistic Neural Network (classifier)
RLM: Run Length Matrix
ROC: Receiver Operating Characteristic
ROI: Region of Interest
SPECT: Single Photon Emission Computed Tomography
SVM: Support Vector Machines (classifier)
T1: longitudinal (or spin-lattice) relaxation time (in MRI)
T2: transverse (or spin-spin) relaxation time (in MRI)
TA: Texture Analysis
TFCM: Texture Feature Coding Method
TFN: Texture Feature Number
US: Ultrasonography

References

[1] Duda, D.: Texture analysis as a tool for medical decision support. Part 2: Classification of liver disorders based on Computed Tomography images, Advances in Computer Science Research 11, 2014, pp. 85-108.

[2] Zhang, X., Kanematsu, M., Fujita, H., Zhou, X., Hara, T., Yokoyama, R., Hoshi, H.: Application of an artificial neural network to the computer-aided differentiation of focal liver disease in MR imaging, Radiol. Phys. Technol. 2(2), 2009, pp. 175-182.

[3] Shiraishi, J., Sugimoto, K., Moriyasu, F., Kamiyama, N., Doi, K.: Computer-aided diagnosis for the classification of focal liver lesions by use of contrast-enhanced ultrasonography, Medical Phys. 35(5), 2008, pp. 1734-1746.

[4] Chabi, M.L., Borget, I., Ardiles, R, Aboud, G., Boussouar, S., Vilar, V., Dromain, C., Balleyguier, C.: Evaluation of the accuracy of a computer-aided diagnosis (CAD) system in breast ultrasound according to the radiologist's experience, Acad. Radiol. 19(3), 2012, pp. 311-319.

[5] Tan, T., Platel, B., Huisman, H., Sanchez, C.I., Mus, R., Karssemeijer, N.: Computer-aided lesion diagnosis in automated 3-D breast ultrasound using coronal spiculation, IEEE Trans. Med. Imag. 31(5), 2012, pp. 1034-1042.

[6] Wang, Y., Wang, H., Guo, Y., Ning, C., Liu B., Cheng, H.D., Tian, J.: Novel computer-aided diagnosis algorithms on ultrasound image: effects on solid breast masses discrimination, J. Digit. Imaging 23(5), 2012, pp. 581-591.

[7] Agliozzo, S., De Luca, M., Bracco, C., Vignati, A., Giannini, V., Martincich, L., Carbonaro, L.A., Bert, A., Sardanelli, F., Regge, D.: Computer-aided diagnosis for dynamic contrast-enhanced breast MRI of mass-like lesions using a multiparametric model combining a selection of morphological, kinetic, and spatiotemporal features, Med. Phys. 39(4), 2012, pp. 1704-1715.

[8] Bhooshan, N., Giger, M., Lan, L., Li, H., Marquez, A., Shimauchi, A., Newstead, G.M.: Combined use of T2-weighted MRI and T1-weighted dynamic contrast-enhanced MRI in the automated analysis of breast lesions, Magn. Reson. Med. 66(2), 2011, pp. 555-564.

[9] Rakoczy, M., McGaughey, D., Korenberg, M.J., Levman, J., Martel, A.L.: Feature selection in computer-aided breast cancer diagnosis via dynamic contrast-enhanced magnetic resonance images, J. Digit. Imaging 26(2), 2013, pp. 198-208.

[10] Cai, H., Peng, Y., Ou, C., Chen, M., Li, L.: Diagnosis of breast masses from dynamic contrast-enhanced and diffusion-weighted MR: A machine learning approach, PLOS ONE 9(1), 2014, e87387, pp. 1-7.

[11] Pang, Z., Zhu, D., Chen, D., Li, L., Shao, Y.: A computer-aided diagnosis system for dynamic contrast-enhanced MR images based on level set segmentation and ReliefF feature selection, Computational and Mathematical Methods in Medicine, Article ID 450531, 2014, pp. 1-10.

[12] Agner, S.C., Soman, S., Libfeld, E., McDonald, M., Thomas, K., Englander, S., Rosen, M.A., Chin, D., Nosher, J., Madabhushi, A.: Textural kinetics: a novel dynamic contrast-enhanced (DCE)-MRI feature for breast lesion classification, J. Digit. Imaging. 24(3), 2011, pp. 446-463.

[13] Nagarajan, M.B., Huber, M.B., Schlossbauer, T., Leinsinger, G., Krol, A., Wismuller, A.: Classification ofsmall lesions in breast MRI: Evaluating the role of dynamically extracted texture features through feature selection, J. Med. Biol. Eng. 33(1), 2013, pp. 59-68.

[14] Puech, P., Betrouni, N., Makni, N., Dewalle, A.S., Villers, A., Lemaitre, L.: Computer-assisted diagnosis of prostate cancer using DCE-MRI data: design, implementation and preliminary results, Int. J. Comput. Assist. Radiol. Surg. 4(1), pp. 1-10, 2009.

[15] Firjany, A., Elnakib, A., El-Baz, A., Gimel'farb, G., El-Ghar, M.A., Elmagharby, A.: Novel Stochastic Framework for Accurate Segmentation of Prostate in Dynamic Contrast Enhanced MRI, In Madabhushi, A., Dowling, J., Yan, P., Fenster, A., Abolmaesumi, P., Hata, N. (Ed): Lect. Notes Comput. Sc. 6367, Springer-Verlag Berlin Heidelberg, 2010, pp. 121-130.

[16] Vos, P.C., Hambrock, T., Barenstz, J.O., Huisman, H.J.: Computer-assisted analysis of peripheral zone prostate lesions using T2-weighted and dynamic contrast enhanced T1-weighted MRI, Phys. Med. Biol. 55(6), 2010, pp. 1719-1734.

[17] Lopes, R., Ayache, A., Makni, N., Puech, P., Villers, A., Mordon, S., Betrouni, N.: Prostate cancer characterization on MR images using fractal features, Med. Phys. 38(1), 2011, pp. 83-95.

[18] Sung, Y.S., Kwon, H.J., Park, B.W., Cho, G., Lee, C.K., Cho, K.S., Kim, J.K.: Prostate cancer detection on dynamic contrast-enhanced MRI: computer-aided diagnosis versus single perfusion parameter maps, AJR Am. J. Roentgenol. 197(5), 2011, pp. 1122-1129.

[19] Niaf, E., Rouviere, O., Mege-Lechevallier, F., Bratan, F., Lartizien, C.: Computer-aided diagnosis of prostate cancer in the peripheral zone using multiparametric MRI, Phys. Med. Biol. 57(12), 2012, pp. 3833-3851.

[20] Duda, D., Kretowski, M., Mathieu, R., de Crevoisier, R., Bezy-Wendling, J.: Multi-image texture analysis in classification of prostatic tissues from MRI.

Preliminary results. In Pietka, E., Kawa, J., Wieclawek, W. (Ed): Adv. Intell. Soft. Comp. 283, Springer, Switzerland, 2014, pp. 139-150.

[21] Litjens, G., Debats, O., Barentsz, J., Karssemeijer, N., Henkjan H.: Computer-aided detection of prostate cancer in MRI, IEEE Trans. Med. Imag. 33(5), 2014, pp. 1083-1092.

[22] Molina, J.F.G., Zheng, L., Sertdemir, M.. Dinter, D.J., Schonberg, S., Radle, M.: Incremental Learning with SVM for Multimodal Classification of Prostatic Adenocarcinoma, PLoS ONE 9(4), e93600, 2014, pp. 1-14.

[23] El-Baz, A., Beache, G.M., Gimel'farb, G., Suzuki, K., Okada, K., Elnakib, A., Soliman, A., Abdollahi, B.: Computer-aided diagnosis systems for lung cancer: challenges and methodologies, Int. J. Biomed. Imaging 2013, 942353, 2013, pp. 1-46.

[24] Ambrosini, R.D., Wang, P., O'Dell, W.G.: Computer-aided detection of metastatic brain tumors using automated 3-D template matching, J. Magn. Reson. Imaging 31(1), 2010, pp. 85-93.

[25] John, P.: Brain tumor classification using wavelet and texture based neural network, Int. J. Sci. Eng. Research 3(10), 2012, pp. 1-7.

[26] Patil, S., Udupi, V.R.: A Computer Aided Diagnostic System for Classification of Brain Tumors Using Texture Features and Probabilistic Neural Network, Int. J. Comput. Sc. Eng. Inf. Technol. Research 3(1), 2013, pp. 61-66.

[27] Islam, A., Reza, S.M.S., Iftekharuddin, K.M.: Multifractal Texture Estimation for Detection and Segmentation of Brain Tumors, IEEE Trans. Biomed. Eng. 60(11), 2013, pp. 3204-3215.

[28] Jayachandran, A., Dhanasekaran, R.: Brain tumor detection and classification of MR images using texture features and fuzzy SVM classifier, Res. J. Appl. Sci. Eng. Technol. 6(12), 2013, pp.2264-2269.

[29] Sachdeva, J., Kumar, V., Khandelwal, N., Ahuja, C.K.: Segmentation, feature extraction, and multiclass brain tumor classification J. Digit. Imaging. 26(6), 2013, pp. 1141-1150.

[30] Selvanayaki, K., Karnan, M.: CAD System for Automatic Detection of Brain Tumor through Magnetic Resonance Image-A Review, Int. J. Eng. Sc. Technol. 2(10), 2010, pp. 5890-5901.

[31] Haralick, R.M.: Statistical and structural approaches to texture, Proc. of the IEEE 67(5), 1979, pp. 786-804.

[32] Castellano, G., Bonilha, l., Li, L.M., Cendes, F.: Texture analysis of medical images, Clin. Radiol. 59(12), 2004, pp. 1061-1069.

[33] Hajek, M., Dezortova, M., Materka, A., Lerski R.A. (Ed): Texture Analysis for Magnetic Resonance Imaging, Med4Publishing, Prague, Czech Republic, 2006.

[34] Bankman, I.N.: Handbook of Medical Image Processing and Analysis, Second Edition, Academic Press, 2008.

[35] Nielsen, B., Albregtsen, F., Danielsen, H.E.: Statistical nuclear texture analysis in cancer research: a review of methods and applications, Crit. Rev. Oncog. 14(2-3), 2008, pp. 89-164.

[36] Szczypinski, P.M., Strzelecki, M., Materka, A., Klepaczko, A.: MaZda – A software package for image texture analysis, Comput. Meth. Prog. Bio. 94(1), 2009, pp. 66-76.

[37] Nailon, W.H.: Texture analysis methods for medical image characterisation, In Mao, Y. (Ed): Biomedical Imaging, InTech, Open, 2010, pp. 75-100.

[38] Kassner, A., Thornhill, R.E.: Texture Analysis: A Review of Neurologic MR Imaging Applications, Am. J. Neuroradiol. 31, 2010, pp. 809-816.

[39] Tuceryan, M., Jain, A. K.: Texture analysis, In Chen, C.H., Pau, L.F., Wang, P.S.P. (Ed): The Handbook of Pattern Recognition and Computer Vision, World Scientific Publishing Co., Singapore, 2nd edition, 1998, pp. 207-248.

[40] Zhang, J., Tan, T.: Brief review of invariant texture analysis methods, Pattern Recognition 35(3), 2002, pp. 735-747.

[41] Xie, X.: A Review of Recent Advances in Surface Defect Detection using Texture analysis Techniques, Electron. Lett. Comp. Vision Image Anal. 7(3), 2008, pp. 1-22.

[42] Haralick, R. M., Shanmugam K., Dinstein I.: Textural features for image classification, IEEE Trans. Syst., Man, Cybern., Syst 3, 1973, pp. 610-621.

[43] Conners, R.W., Harlow, C.A.: Toward a structural textural analyzer based on statistical methods, Comput. Vision Graph. 12(3) 1980, pp. 224-256.

[44] Galloway, M.M.: Texture analysis using gray level run lengths, Comput. Vision Graph. 4(2), 1975, pp. 172-179.

[45] Chu, A., Sehgal, C.M., Greenleaf, J.F.: Use of gray value distribution of run lengths for texture analysis, Pattern Recognition Letters, 11(6), 1990, pp. 415-420.

[46] Albregtsen, F., Nielsen, B., Danielsen, H.E.: Adaptive gray level run length features from class distance matrices, Proc. 15th Int. Conf. on Pattern Recognition, 3, 2000, pp. 738-741.

[47] Weszka, J.S., Dyer, C.R., Rosenfeld, A.: A Comparative Study of Texture Measures for Terrain Classification, IEEE Trans. Systems, Man, Cybernetics, 6, 1976, pp. 269-285.

[48] Lerski, R.A., Straughan, K., Shad, L., Boyce, D., Bluml, S., Zuna, I.: MR Image Texture Analysis – An Approach to Tissue Characterization, Magn. Reson. Imaging 11(8), 1993, pp. 873-887.

[49] Horng, M.H., Sun, Y.N., Lin, X.Z.: Texture feature coding method for classification of liver sonography, Comput. Med. Imag. Grap 26(1), 2002, pp. 33-42.

[50] Gonzalez, R.C., Woods, R.E.: Digital Image Processing, Second edition, Reading, MA: Addison-Wesley, 2002.

[51] Mandelbrot, B.: The Fractal Geometry of Nature, block W. H. Freeman and Co., NY, 1982.

[52] Falconer, K.: Fractal Geometry, Mathematical Foundations and Applications, Second edition, Willey, 2003.

[53] Edgar, G.: Measure, Topology and Fractal Geometry, Second edition, Springer-Verlag, 2008.

[54] Chen, C., Daponte, J.S., Fox, M.D.: Fractal feature analysis and classification in medical imaging, IEEE Trans. Med. Imag. 8(2), 1989, pp. 133-142.

[55] Chen, E.L., Chung, P.C., Chen, C.L., Tsai, H.M., Chang, C.I.: An automatic diagnostic system for CT liver image classification, IEEE Trans. Biomed. Eng. 45(6), 1998, pp. 783-794.

[56] Li, J., Du, Q.: An improved box-counting method for image fractal dimension estimation, Pattern Recognition 42(11), 2009, pp. 2460-2469.

[57] Sankar, D., Thomas. T.: Fractal Features based on Differential Box Counting Method for the Categorization of Digital Mammograms, International Journal of Computer Information Systems and Industrial Management Applications 2, 2010, pp. 11-19.

[58] Landini, G., Rippin, J.W.: Notes on the implementation of the mass-radius method of fractal dimension estimation, Comput. Appl. Biosci. 9(5), 1993, pp. 547-550.

[59] Jones, C.L., Jelinek, H.F.: Wavelet packet fractal analysis of neuronal morphology, Methods 24(4), 2001, pp. 347-458.

[60] Pentland, A.P.: Fractal-Based Description of Natural Scenes, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-6(6), 1984, pp. 661-674.

[61] Maragos, P., Sun, F.K.: Measuring the Fractal Dimension of Signals: Morphological Covers and Iterative Optimization, IEEE Trans. Signal Process. 41(1), 1993, pp. 108-121.

[62] Backes, A.R., Bruno, O.M.: A new approach to estimate fractal dimension of texture images, In Elmoataz, A., Lezoray, O., Nouboud, F., Mammass, D. (Ed): Lect. Notes Comput. Sc. 5099, Springer, 2008, pp. 136-143.

[63] Kilic, K.I., Abiyev, R.H.: Exploiting the synergy between fractal dimension and lacunarity for improved texture recognition, Signal Processing 91(10), 2011, pp. 2332-2344.

[64] Mallat, S.G.: A theory for multiresolution signal decomposition: The wavelet representation, IEEE Trans. Pattern Anal. Mach. Intell. 11(7), 1989, pp. 674-693.
[65] Laws, K.I.: Textured image segmentation, PhD thesis, University of Southern California, 1980.
[66] Peng, Y., Jiang, Y., Antic, T., Giger, M.L., Eggener, S., Oto, A.: A study of T2-weighted MR image texture features and diffusion-weighted MR image features for computer-aided diagnosis of prostate cancer, Proc. SPIE 8670, 86701H, 2013.
[67] Ginsburg, S.B., Rusu, M., Kurhanewicz, J., Madabhushi, A.: Computer extracted texture features on T2w MRI to predict biochemical recurrence following radiation therapy for prostate cancer, Proc. SPIE 9035, 903509, 2014.
[68] Vapnik, V.N.: The Nature of Statistical Learning Theory, Second Edition, Springer, NY, 2000.
[69] Freund, Y., Shapire, R.: A decision-theoretic generalization of online learning and an application to boosting, J. Comput. Syst. Sci. 55(1), 1997, pp. 119-139.
[70] Clausi, D.A., Jernigan, M.E.: Designing Gabor filters for optimal texture separability, Pattern Recognition 33(11), 2000, pp. 1835-1849.
[71] Duda, R., Hart P., Stork D.: Pattern Recognition, Second Edition, John Willey and Sons, 2001.
[72] Fukunaga, K.: Introduction to Statistical Pattern recognition, Second edition, Academin Press, San Diego, CA, USA, 1990.
[73] Hanley, J.A.: Receiver operating characteristic (ROC) methodology: the state of the art, Crit. Rev. Diagn. Imaging 29(3), 1989, pp. 307-335.
[74] Hosmer, D.W., Lemeshow, S., Sturdivant R.X.: Applied Logistic Regression, Third Edition, John Wiley & Sons, Inc., Hoboken, NJ, US, 2013.
[75] Bishop, C.M.: Neural Networks for Pattern Recognition, Oxford University Press, NY, USA, 1995.
[76] Ripley, B.D.: Pattern Recognition and Neural Networks, Cambridge University Press, Cambridge, UK, 1996.
[77] Costa, L. da F., Cesar, R.M.: Shape Analysis and Classification: Theory and Practice, CRC Press, Inc., Baca Raton, FL, USA, 2000.
[78] Turner, M.R.: Texture discrimination by Gabor functions, Biol. Cybern. 55(2-3), 1986, pp. 71-82.
[79] Sun, C., Wee, W.G.: Neighboring gray level dependence matrix for texture classification, Comput. Vision, Graph. Imag. Process. 23(3), 1983, pp. 341-352.
[80] Amadasun, M., King, R.: Textural features corresponding to textural properties, IEEE Trans. Syst. Man Cybern. 19(5), 1989, pp. 1264-1274.

[81] Specht, D.F.: Probabilistic Neural Networks, Neural Networks 3(1), 1990, pp. 109-118.

[82] Pearson, K.: On lines and planes of closest fit to systems of points in space, Phil. Mag. 2(6), 1901, pp. 559-572.

[83] Sachdeva, J., Kumar, V., Gupta, I., Khandelwal, N., Ahuja, C.K.: A novel content-based active contour model for brain tumor segmentation, Magn. Reson. Imaging 30(5), 2012, pp. 694-715.

[84] Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24(7), 2002, pp. 971-987.

[85] Idrissa, M., Acheroy , M.: Texture classification using Gabor filters, Pattern Recognition Letters 23(9), 2002, pp. 1095-1102.

[86] Zhang, J., Tan, T., Ma, L.: Invariant texture segmentation via circular Gabor filters, Proc: 16th Int. Conf. Pattern Recognition vol. 2, 2002, pp. 901-904.

[87] Tiwari, P., Prasanna, P., Rogers, L., Wolansky, L., Badve C., Sloan, A., Cohen, M., Madabhushi, A.: Texture descriptors to distinguish radiation necrosis from recurrent brain tumors on multi-parametric MRI, Proc, SPIE, 9035, 90352B, 2014.

[88] Burt, P.J., Adelson, E.H.: The Laplacian pyramid as a compact image Code, IEEE Trans. Commun. 31(4), 1983, pp. 532-540.

[89] Dalal N., Triggs, B.: Histograms of oriented gradients for human detection, IEEE Computer Society Conf. Comput. Vision and Pattern Recognition, CVPR2005 vol. 1, 2005, pp. 886-893.

[90] Random Forests, Machine Learning 45(1), 2001, pp. 5-32.

[91] Yang, C., Chuang, L., Yang, C.: IG-GA: a hybrid filter/wrapper method for feature selection of microarray data, J. Med. Biol. Eng. 30(1), 2010, pp. 23-28.

[92] Chan, T.F., Vese, L.A.: Active contours without edges, IEEE Trans. Image Process. 10(2), 2001, pp. 266-277.

[93] Kononenko, I.: Estimating attributes: analysis and extensions of relief, In Bergadano, F., de Raedt, L. (Ed): Lect. Notes Comput. Sc. 784, Springer-Verlag Berlin Heidelberg, 1994, pp. 171-182.

ANALIZA TEKSTUR JAKO NARZĘDZIE WSPOMAGANIA DECYZJI MEDYCZNYCH. CZĘŚĆ 1: NAJNOWSZE ZASTOSOWANIA DO WCZESNEGO WYKRYWANIA NOWOTWORÓW

Streszczenie: W ciągu ostatnich dwudziestu lat zaproponowano wiele komputerowych systemów wspomagania decyzji medycznych, opierających się na danych obrazowych. Systemy te są w stanie zlokalizować patologicznie zmienione obszary, opisać właściwości rozpatrywanych tkanek, jak również dokonać ich klasyfikacji. Istotnym źródłem informacji zawartej w obrazie jest jego tekstura. Cyfrowa analiza tekstur pozwala wykryć znacznie więcej szczegółów obrazu, niż zwykła analiza wizualna. Odpowiedni dobór metod analizy tekstur może przyczynić się do znacznego podwyższenia liczby trafnie rozpoznanych schorzeń. Wybór ten często zależy od niuansów danego problemu diagnostycznego.
Niniejsza praca stanowi przegląd najczęściej stosowanych metod analizy tekstur (statystycznych, opierających się na modelach, wykorzystujących filtry) oraz pokazuje ich zalety i ograniczenia. Zawiera również przegląd najnowszych systemów do wczesnego wykrywania i rozpoznawania nowotworów, opierających się na analizie tekstury.

Słowa kluczowe: obrazowanie medyczne, analiza obrazów, tekstura, selekcja cech, wspomaganie decyzji medycznych, diagnoza wspomagana komputerowo

Artykuł zrealizowano w ramach pracy statutowej S/WI/2/2013.

TEXTURE ANALYSIS AS A TOOL FOR MEDICAL DECISION SUPPORT. PART 2: CLASSIFICATION OF LIVER DISORDERS BASED ON COMPUTED TOMOGRAPHY IMAGES

Dorota Duda

Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

Abstract: Texture analysis has already demonstrated its great potential in many digital image-based diagnostic systems. It allows to extract from an image many important diagnostic information, impossible to capture with only the visual appreciation. The first attempts to use a texture analysis (TA) as a tool for characterization of an image content took place in the 70's of the last century. Since then a variety of methods have been proposed and found their application in many domains, also – in the medical field. However, it is still difficult to indicate a method that would ensure satisfactory results for any diagnostic problem.
The present work gives an overview of the texture analysis methods, that have been applied for hepatic tissue characterization from Computed Tomography (CT) images. It includes details of about forty studies, presented over the past two decades, devoted to (semi)automatic detection or/and classification of different liver pathologies. Quoted systems are divided into three categories: (i) based on a single-image texture of non-enhanced CT images of the liver, (ii) based on a single-image texture of contrast-enhanced images, and (iii) based on a multi-image texture. The latter ones concern a simultaneous analysis of sets of textures, each of which corresponds to the same liver slice, but is related to a different contrast agent concentration in hepatic vessels.

Keywords: medical imaging, image analysis, texture characterization, feature selection, Computer Aided Diagnosis, CAD, medical decision support, liver, computed tomography, CT

A list of abbreviations is given at the end of this article.

1. Introduction

In clinical practice, when dynamic CT of the liver is performed, three image series are usually acquired: the first one – before the contrast agent injection, the next two

Advances in Computer Science Research, vol. 11, pp. 85-108, 2014.

ones – after its injection, at arterial and at portal phase of its propagation [1]. The two post-injection acquisition moments correspond, respectively, to the maximal concentration of contrast agent that reaches the liver first via the hepatic artery, next – via the portal vein. The arterial phase starts after about $25 \div 35$ seconds after the intravenous injection of contrast agent, the portal one – after about $60 \div 70$ seconds. In some cases a fourth – delayed hepatic phase is considered [2]. It takes place after about $5 \div 10$ minutes succeeding the injection. Each of the three (or even four) images enhances a different tissue property, that could reveal a development of a pathology. In the case of the liver CT – it can be excessive or insufficient growth of the arterial or of the portal vascular tree. After injection of the contrast agent, the high vascularization regions are more enhanced than those with normal vasculature, and less vascularized regions appear darker. The presence of contrast agent in hepatic vessels results also in changes of texture properties, imperceptible to the naked eye.

In the first part of the study [3], several approaches to characterization of image textures were presented. They based on:

- Gray Level Histogram (GLH), giving the First Order Statistics (FOS),
- Co-Occurrence Matrices (COM) [4–6],
- Run Length Matrices (RLM) [7–9],
- Gray Level Difference Matrices (GLDM) [10],
- Gradient Matrices (GM) [11],
- Texture Feature Coding Method (TFCM) [12],
- Autocorrelation Coefficients (AC) [13],
- Fractal Model (FM) [14–24],
- Discrete Wavelet Transform (DWT) [25],
- Laws' Texture Energy (LTE) [26].

The aim of this part is to examine which of these methods have found their application in Computer-Aided Diagnosis (CAD) systems, based on CT images of the liver.

The first attempts to liver texture characterization from CT images (about 20 years ago) considered only the non-enhanced images. Over the time, with a development of imaging technique and with more and more wider access to studies it has become possible to perform frequent imaging after administration of contrast agent. Despite the availability of several series of images depicting the same part of liver, texture analysis was performed yet for a long time on only one image (contrast-enhanced, or still non-enhanced) . It is about 10 years ago, that the systems adapted for multi-image texture analysis were introduced. Such systems tried to find tissue characteristics based on analysis of several CT liver images, acquired at the same slice location, but under different conditions (different moments of contrast agent

propagation). Given the above, the review will present at first the systems based on the analysis of a single image, acquired without contrast. Next, the systems dealing with single-image textures, acquired after administration of contrast agent, will be described. Finally, the multi-image texture-based systems will be quoted. The work will be finished by general conclusions, drawn on the basis of the three parts of the review.

2. Application of texture analysis in classification of liver disorders based on CT images

2.1 Texture analysis of single liver CT image acquired without contrast agent

One of the earliest studies on the possibility of applying a texture analysis for the characterization and recognition of liver tissue, from CT images, was presented in 1995 [27]. This work had two main objectives. The first was to investigate whether the texture could be used to discriminate between various tissue types, providing the information not accessible to human perception. The second was to find the most useful features, in terms of tissue classification. In the study textural features obtained by the COM (12 features), RLM (15 features) and the GLDM (20 features) methods were used. Three types of hepatic tissue were characterized: normal liver, abnormal liver with the clearly visible malignancy, and abnormal one with the invisible malignancy. The performance of features was compared on the basis of statistical significance. It was found that the three features: *entropy*, *local homogeneity* (COM method) and *gray level distribution* (RLM method) were the most appropriate to detect an invisible (early) liver malignancy with a confidence level of above 99%

From this moment, quite a lot of semi-automatic systems for liver tissue recognition from CT images have been proposed. In many of them, especially in the earlier ones, the tissue was characterized on the basis of only one image, acquired without injection of contrast agent [16, 28–38]. Quoted systems utilized several methods for extraction of textural features. They included: gray-level histogram, co-occurrence matrices, run length matrices, gray level difference matrices, fractal model, Laws' texture energy measures, autocorrelation coefficients, or different frequency methods. A list of systems and tested methods is given in Table 1. Due to the fact that each of the systems used different classifiers and that the methods for their quality assessment were also different, the table do not contain the best classification results.

For example, the system evaluated by Chen et al. [16] was able to automatically find the liver, to extract its boundaries and to recognize two types of liver tumors: hepatoma and hemangioma. In this system, the image texture was characterized by

Table 1. Comparison of systems based on single liver CT images acquired without contrast agent

Work	Year	TA Methods	Tissue Classes and number of cases
Mir et al. [27]	1995	– COM – RLM – GLDM	– normal (200) – abnormal, clearly visible (200) – abnormal invisible (200)
Chen et al. [16]	1998	– COM – FM	– hepatoma (20) – hemangioma (10)
Husain et al. [28]	2000	– FOS – COM	– normal – abnormal
Sariyanni et al. [29]	2001	– FM	– healthy (99) – HCC (50)
Gletsos et al. [30]	2003	– COM – FOS	– healthy (76) – liver cysts (19) – hemangioma (28) – HCC (24)
Valavanis et al. [31]	2004	– FOS – COM – GLDM – LTE – FM	– healthy (76) – liver cysts (19) – hemangioma (28) – HCC (24)
Mala et al. [32]	2005	– OWT & FOS – OWT & COM	– steatosis (70) – cirrhosis (70)
Huang et al. [33]	2006	– AC	– malignant (80) – benign (84)
Stoitsis et al. [34]	2006	– FOS – COM – GLDM – LTE – FM	– healthy (76) – liver cysts (19) – hemangioma (28) – HCC (24)
Mougiakakou et al. [35]	2007	– FOS – COM – GLDM – LTE – FM	– healthy (76) – liver cysts (19) – hemangioma (28) – HCC (24)
Ganeshan et al. [36]	2009	– filters & FOS – filters & COM	– absence of malignancy (15) – malignancy not related to the liver (9) – liver metastases (8)
Kumar et al. [38]	2013	– FOS – COM – CCT & FOS – CCT & COM – WCT & FOS – WCT & COM	– HCC (150) – hemangioma (150)

features obtained from the co-occurrence matrices (here, the *correlation* and *sum entropy* turned out to be the best ones) and its fractal dimension, evaluated from a fractional Brownian motion model (the method developed by authors and described in their work). A probabilistic Neural Network (NN) [39] was used as a classifier. The proposed system was tested on 30 liver cases and shown to be quite efficient.

Another system, described in [28], was also able to recognize a liver region (normal and abnormal) on CT images. The system used the gray-level histogram features (*mean gray level*, *standard deviation*, and *skewness*) in combination with the COM-based features (*entropy*, *homogeneity*), and a back-propagation Neural Network [40] as a classifier. The system was able to recognize correctly more than 95% of analyzed cases.

Sariyanni et al. [29] tried to recognize a healthy liver tissue and a tissue affected by hepatocellular carcinoma (HCC). As texture descriptors, they used a fractal dimensions calculated from four different methods: the power spectrum method (belonging to the fractional Brownian motion methods) [21], the box-counting method, the morphological fractal estimator (belonging to the area measurement methods) [22], and the kth-Nearest Neighbor estimator (k-NN), proposed by authors. The Fuzzy C-Means algorithm [41] was then applied for clustering the input data into two clusters. It revealed that the k-NN estimator, introduced by authors, outperforms the other methods.

The work of Gletsos et al. [30] described a CAD system adapted to the recognition of four types of liver tissue: healthy, liver cysts, hemangioma, and hepatocellular carcinoma. It used 48 texture descriptors derived from the co-occurrence matrices, and the average gray level of the Regions of Interest (ROIs). The classification module consisted of three sequentially placed feed-forward Neural Networks, each adapted to perform a pairwise classification. The first one distinguished normal from pathological liver regions, the second one recognized pathological regions and distinguished cysts from "other pathologies", and the third one distinguished between "other pathologies" – hemangioma and HCC. Three feature selection techniques were used separately for each binary classifier: the Sequential Forward Selection (SFS) [42], the Sequential Floating Forward Selection (SFFS) [43], and Genetic Algorithm for feature selection (GAs) [44] with the implementation based on the work [45]. The feature selection used a criterion based on the squared Mahalanobis distance between the populations of the two classes for each binary NN classifier. Finally, several subsets of features were considered for classification experiments. The CAD performance was tested with validation and testing sets, each containing a portion $(1/5)$ of the initial data. Results obtained for different sub-sets of features differed from one another. The best overall classification accuracy was of 97%.

A more developed system was presented by Valavanis et al. [31]. It was evaluated in the process of recognition of four types of focal liver lesions, the same that were considered in [30]. The number of ROIs for each tissue class was also the same. Here, the relevance of the five texture characterization methods was assessed. Among the tested methods were: the method based on the gray-level histogram, the co-occurrence matrices, the run length matrices, the Laws' texture energy measures, and the fractal model. The most useful features were found using a feature selection, based on Genetic Algorithms. Classification was carried out by Neural Networks (three-layer feed-forward NN and Radial Basis Function (RBF) NN) and statistical methods (k-NN [46] with different k). Here, the best classification accuracy was equal to 90.63%. Similar works have been described, some years later, in [34], and further – in [35]. The continuation of this research has finally resulted in the creation of a telematics-enabled system for image archiving, management, and diagnosis support [37]. This integrated CAD system performed an image preprocessing, a semi-automatic image segmentation, an extraction of texture features, and a classification.

Another CAD system was proposed by Mala et al. [32], in order to classify two diffused liver diseases, steatosis and cirrhosis. First, it performed an automatic extraction of liver, using adaptive threshold and morphological processing. Second, images were transformed into frequency domain using the Orthogonal Wavelet Transform (OWT). Then, the statistical features were calculated based on the horizontal, the vertical, and the diagonal details extracted from the images. They included: *mean*, *standard deviation*, *contrast*, *entropy*, *homogeneity*, and *angular second moment*. Finally, the two-layer probabilistic Neural Network was used as a classifier. The system was trained on 40 cases and tested on 100 ones. Both classes were equally numerous in the train end the test set. The classification accuracy of 95% was achieved.

The next CAD example was described by Huang et al. [33]. Their system was adapted for differentiation between two groups of liver tumors: malignant (primary tumor – HCC or secondary tumors – metastases) and benign. As texture parameters, only the normalized autocorrelation coefficients were used. The classification was performed with the Support Vector Machines (SVM) [47]. The k-fold cross-validation [48] was used to evaluate the performance of the proposed diagnostic system. The classification accuracy was of nearly 82%.

The objective of yet another research, presented by Ganeshan et al. [36], was to determine whether the textures corresponding to the apparently healthy liver regions were altered by the presence of malignancy in patients with colorectal cancer. Three types of liver tissue were considered. The first one corresponded to an absence of malignancy, the second one – to the presence of a malignancy but not related to the liver, and the third one – to the presence of liver metastases. Here the frequency methods

in combination with statistical approaches were used to characterize hepatic tissue. The following statistical descriptors of texture were derived from both unfiltered and filtered images (highlighting fine, medium, and coarse texture): *mean gray level*, *entropy*, and *uniformity*. The experiments showed that textural features obtained from the filtered images were statistically different for each of the three considered tissue classes.

The most recently, Kumar et al. [38] developed a texture-based CAD system, specialized in discrimination between malignant (hepatocellular) and benign (hemangioma) liver tumors. Their work tested several sets of features: gray-level texture features (first order statistics and second order, COM-based texture descriptors), Wavelet Coefficient Texture (WCT) features (first- and second- order statistics), and Contourlet Coefficient Texture (CCT) features [49, 50] (also of the first- and of the second order). As numbers of considered features were quite large (in total about 300 features were tested) the Principal Component Analysis (PCA) [51] was applied for a dimensionality reduction. The ability of each feature set in differentiating malignant from benign tissues was assessed with a probabilistic Neural Network classifier. The areas under the Receiver Operating Characteristic (ROC) curves (AUC) [52] were used for measuring the system performance. The highest classification accuracy (96.7%), as well as the highest sensitivity and specificity (97.3% and 96%, respectively) were obtained with the contourlet coefficient co-occurrence features.

2.2 Texture analysis of single liver CT image acquired after administration of contrast agent

Preliminary studies on processing of contrast-enhanced CT images for semi-automatic recognition of liver disorders were reported by Krętowski [53]. The work aimed at comparing the classification accuracy obtained for the three acquisition moments, typical for the CT of abdominal organs (without injection, arterial phase, portal phase). Five types of liver tissue were differentiated: the healthy liver and four types of its metastases: insulinoma, adenocarcinoma (kidney), adenocarcinoma (intestine) and leiomyosarcoma. The image database was divided into three parts, each of which being composed of images corresponding to one (of the three considered) acquisition moment. The tissue was characterized by features calculated with the FOS, GM, COM and RLM methods. The texture classification was performed by oblique (multivariate) Dipolar Decision Trees [54], separately for each of the three parts of the database. The classification accuracy for acquisitions with contrast material outperformed the results obtained for those without contrast. The highest classification accuracy was observed for the arterial phase.

The systems for liver tissue characterization and recognition from enhanced CT images began to appear after this moment [55–65]. The texture analysis in these systems was performed with the following methods: FOS, COM, RLM, GLDM, LTE, or frequency methods (see the comparison in Table 2). However, all of those systems were still limited to the analysis of only one image at a time and they did not consider the changes in texture properties during the propagation of contrast material.

For example, Bilello et al. [55] presented a system working on portal-phase images. It combined the methods for detection, characterization and classification of liver hypodense hepatic tissue (cysts, hemangiomas, and metastases). Its texture was characterized with the frequency methods. Then the Support Vector Machines were used to perform a pairwise lesion classification. In order to evaluate the system performance, the Free-Response Receiver Operator Characteristic Curves (FROC) [66] were utilized. The system assured perfect discrimination (100% of correctly recognized cases) between hemangiomas and cysts, good discrimination between cysts and metastases (at 95% sensitivity for detection of metastases, only about 5% of cysts were incorrectly classified as metastases), and was least accurate in discriminating between hemangiomas and metastases (at 90% sensitivity for detection of hemangiomas, about 28% of metastases were incorrectly classified as hemangiomas).

The system described by Lambrou et al. [56], differentiated between healthy and tumorous tissue. To extract texture features, it used a wavelet transform method, in combination with three statistical methods (based on the gray level histogram, the co-occurrence matrices, and the run length matrices). Three statistical classifiers were employed in the study: minimum distance classifier, quadratic minimum distance classifier, and Bayes classifier [46]. The performance of the classifiers was assessed with the leave-one-out method [48]. The first- and the second order statistics turned out to be better (yielding the classification accuracies exceeding 90%) than those derived from the wavelet-based techniques.

Another system, developed by Smutek et al. [57], focused on the analysis of focal liver lesions (HCC and cysts). It used the first- and the second order texture features (COM-based). The analyzed images corresponded to the late portal phase. In the system, an ensemble of Bayesian classifiers was applied. The classification accuracy was assessed by leave-one-out method. The system was able to classify correctly even 100% of recognized cases.

Mala et al. [58] presented a system adapted to the recognition of four types of liver diseases: HCC, cholangiocarcinoma, hemangeoma and hepatoadenoma. This system was able to automatically detect the areas affected by a disease, to characterize a tissue (using Biorthogonal Wavelet Transform (BWT) and co-occurrence matrices derived from the transformed images), to select the best texture features, and, finally,

Table 2. Comparison of systems based on single liver CT images acquired after administration of contrast agent

Work	**Year**	**TA Methods**	**Tissue Classes and number of cases**
Krętowski [53]	2002	– FOS – GM – COM – RLM	– healthy (192) – insulinoma (126) – adenocarcinoma / kidney (104) – adenocarcinoma / intestine (107) – leiomyosarcoma (68)
Bilello et al. [55]	2004	– filters	– hemangiomas (11) – cysts (25) – metastases (52)
Lambrou et al. [56]	2006	– WT & FOS – WT & COM – WT & RLM	– healthy (425) – tumor (425)
Smutek et al. [57]	2006	– FOS – COM	– HCC (425) – cysts (110)
Mala et al. [58]	2007	– DTW & COM	– HCC (60) – cholangiocarcinoma (60) – hemangeoma (60) – hepato adenoma (30)
Lee et al. [59]	2009	– FOS – GTF	– cyst (70) – hepatoma (70) – cavernous hemangioma (33) – normal liver (60)
Wang et al. [61]	2009	– FOS – COM – GLDM – RLM	– HCC (30) – hemangioma (30) – normal (30)
Mala et al. [62]	2010	– BWT & FOS – BWT & COM)	– fatty (100) – cirrhotic (100)
Kayaalti et al. [63]	2014	– COM – RLM – GTDM – LTE – DWT – DFT – GF – FOS	– fibrosis, stage 0 (21) – fibrosis, stage 1 (16) – fibrosis, stage 2 (12) – fibrosis, stage 3 (16) – fibrosis, stage 4 (13) – fibrosis, stage 5 (13) – fibrosis, stage 6 (25)
Rao et al. [64]	2014	– filters & FOS – filters & COM	– without metastases (15) – synchronous metastases (10) – metachronous metastases (4)
Simpson et al. [65]	2014	– COM	– postoperative liver failure (12) – no liver failure (24)

to classify tissues, using a probabilistic Neural Network. The BWT enabled to obtain horizontal, vertical, and diagonal details of images. Then, 10 features were extracted for each of three resulting images. Feature selection was performed here with a Sequential Backward Elimination (SBE) [42]. Regarding the classification experiments – all the available data were randomly divided into two equally numerous sets (for training and testing). Each experiment was repeated 5 times. The best classification result was of 90.2%.

The aim of another work [59] was to automatically discriminate liver diseases using a sigmoid Radial Basis Function Neural Network with growing and pruning algorithm (described by the authors). This time cyst, hepatoma, cavernous hemangioma, and normal liver tissue were recognized. The ROIs were characterized using gray level and Gabor Texture Features (GTF) [67, 68]. The ROC curves were used to evaluate the diagnosis performance, and the area under ROC curve measured the classification accuracies. The best classification result exceeded 99%.

The study presented in [60] aimed at the assessment of the utility of texture analysis of liver CT images, and at the comparison of the abilities of texture analysis and hepatic perfusion CT to help predict survival for patients with colorectal cancer. The texture analysis comprised two stages. The first one was the image filtration (here, a Laplacian of Gaussian band-pass filter was chosen). The second one was the quantification of texture (here, the *mean gray-level intensity* and *uniformity* were used). The study provided preliminary evidence that analysis of liver texture on portal phase CT images was potentially a superior predictor of survival for patients with colorectal cancer than the CT perfusion imaging.

Wang et al. [61] tested yet another diagnostic system, which worked with the three types of liver tissue: HCC, hemangioma, and normal one. This system used four texture analysis methods (based on the gray level histogram, the co-occurrence matrices, the gray level difference matrices, and the run length matrices). As a classifier the Support Vector Machines were used, and two strategies were considered in order to ensure a multi-class classification: One-Against-All (OAA) [69] and One-Against-One (OAO) [70]. The performance of the CAD system was estimated by the 5-fold cross-validation. The experiments on 90 ROIs, described by set of 22 textural features, gave the overall classification accuracy of about 94% and 98%, for the OAA and the OAO strategy, respectively.

In yet another work [62], a CAD system used the wavelet-based statistical textural features as tissue descriptors. The system was able to extract the liver, and to recognize between fatty and cirrhotic liver tissue. In this work, the original images were first decomposed using a biorthogonal wavelet transform. Then, as in the previous work of the same team [32], the second order statistical features were extracted in

horizontal, vertical and diagonal directions. After performing a feature selection, the most robust texture descriptors were fed to the three types on Neural Networks. The 10-fold cross-validation procedure was used to evaluate the system abilities. The experiments on 200 patients resulted in quite high percentages of correctly recognized characters (reaching 96%).

The most recent studies, published this year, are also based on texture analysis of contrast-enhanced CT images, acquired at portal venous phase [63–65]. For example, Kayaalti et al. [63] recognized seven possible stages of liver fibrosis. For this purpose, eight methods for texture feature extraction were tested. They were based on: co-occurrence matrix, run length matrix, Gray Tone Difference Matrix (GTDM) [71], Laws' filters, Discrete Wavelet Transform [72], Discrete Fourier Transform (DFT), Gabor Filters (GF) [73], and first order statistics. For each combination of classes, a sequential floating forward selection and exhaustive search methods were used in order to find the best texture descriptors. The pairwise classification experiments with Support Vector Machines and k-NN classifier showed that DWT, Gabor, COM, and Laws' features were more successful than the others. The performances of the classifiers were assessed by 2- or 3-fold cross-validation. When only 5 features were used, the mean classification accuracy in pairwise group comparisons was approximately 95% for both the k-NN and the SVM method.

Rao et al. [64] evaluated the potential of analysis of the whole liver with apparently disease-free parenchyma, for discriminating between three types of colorectal cancer patients: without liver metastases, with synchronous liver metastases, and with metachronous metastases. In their work, a texture characterization comprised two stages. First, images were filtered with a Laplacian of Gaussian band-pass filter with different bandwidths. Afterwards, three features were calculated from the filtered and the unfiltered images: *entropy*, *uniformity*, and *mean gray-level intensity*. The ROC analyses were conducted to determine the diagnostic performance of the considered features. As a result, mean *entropy and uniformity* in patients with synchronous metastases were significantly different compared with the non-metastatic patients, while texture parameters for the metachronous metastases group were not significantly different neither from the non-metastatic group nor from synchronous metastases group.

Finally Simpson et al. [65] used some COM-based features of preoperative CT images of the liver, in order to predict a postoperative liver failure after hepatic resection. The study was undertaken on 36 patients. It was discovered that the following features: *contrast*, *correlation*, *cluster prominence*, and normalized *inverse difference moment* were significantly different between patients with and without postoperative liver failure.

2.3 Multi-image texture analysis, involving non-enhanced and contrast-enhanced liver CT images

In [74] a simultaneous analysis of triplets of liver textures, corresponding to the three aforementioned typical acquisition moments, was proposed. At first, the three corresponding simple textures were characterized separately by features obtained from gray-level histogram, Laws' filtering, COM and RLM methods. As a result, three feature sets, each characterizing one of the three textures, were obtained. Then, all the features from those three sets were placed together in one feature vector, characterizing a multi-image ("triphase") texture. As a classifier, an oblique Decision Dipolar Tree was used. The 5-times repeated 10-fold cross-validation procedure was applied to estimate the classification accuracy. Three types of liver tissue were recognized: healthy liver and its two main primary malignant tumors (HCC and cholangiocarcinoma). The classification accuracies obtained for triphase textures were significantly higher than those corresponding to each acquisition moment separately. For example, the best classification accuracies obtained with the set of the 8 RLM-based features were: 95.5%, 93.9%, and 95.5% for the no-contrast, the arterial, and the portal phase, respectively, while considering simultaneously the three phases resulted in the 99.7% of correctly diagnosed cases. Further work of the same team [75] has confirmed that a simultaneous analysis of images, corresponding to the three acquisition moments could lead to better results that the simple texture analysis – performed when only one acquisition moment is considered.

An approach similar to the two preceding ones was used by Quatrehomme et al. [76]. In their work, the analysis of multi-image textures was performed on four-phase CT scans of the liver: the first one – taken in pre-injection phase, the next three ones – after injection of contrast material, in arterial, portal and late phase. Five types of hepatic lesions were differentiated: cysts, adenomas, haemangiomas, HCC and metastases. Four techniques for feature extraction were used. They based on: gray-level histogram, Gaussian Markov Random Fields (GMRF) measures [77], LTE measures, and Unser Histograms Statistics (UHS) [78]. Features, calculated separately for four considered acquisition moments were placed side by side in a multiphase vector, describing four-phase textures. As a classifier, the SVM were used. Its performance was evaluated by the leave-one-out technique. The results obtained with multi-image approach were significantly better than for the case of a single-image texture analysis.

A multi-phase liver images, derived from the four image series (non-enhanced, arterial, portal, and delayed) were also considered by Chi et al. [79]. Their system was designed in order to help radiologists in characterization of various focal liver lesions. Six types of lesions were considered: HCC, hemangioma, cysts, liver abscess,

Focal Nodular Hyperplasia (FNH), and metastases. The latter class included: pancreatic carcinoma, sigmoid carcinoma, rectal carcinoma, colorectal carcinoma, and gallbladder carcinoma. The system first localized a lesion on multi-phase CT using a hybrid generative-discriminative method [80]. Then, a lesion was selected in one phase, and nonrigid B-spline registration [81] was employed in order to align the images of all the four phases. The tissue was characterized by a simultaneous analysis of textures corresponding to the four considered phases. Feature vectors were composed of multi-phase density characteristics and combinations of the co-occurrence matrix-based parameters, calculated for each of four phases. The system compared a tested lesion with the model lesions from a reference database (characterized by vectors of features), and measured their similarities using the L1-norm-based similarity scores. The reference cases which were the most similar to the examined one were finally provided to the users for their later studies. The system was tested on a database of 69 cases and evaluated using the precision-recall curves and the "Bull's Eye Percentage" (BEP) score [82]. A multi-image texture analysis resulted in a BEP value of 78%, while the best results for a single-phase cases were about 63% – 65%.

The aim of two other studies [83, 84] was to determine preliminarily how some of the hepatic texture features (*entropy*, *uniformity*) change during the propagation of contrast agent and to assess whether the differences in these changes between tumorous and non-tumorous liver tissue were statistically significant. The potential utility of Dynamic Contrast-Enhanced (DCE) texture analysis of the liver was compared to the potential of the measurements of hepatic attenuation and perfusion, obtained from the kinetic modeling. The study concerned patients following a resection of colorectal cancer and having apparently normal hepatic morphology. It showed that the temporal changes of the two considered textural features were different from those for hepatic attenuation and they were statistically significant between tumorous and non-tumorous patients. It also demonstrated that the textural features were less sensitive to changes in CT acquisition conditions (current and voltage variations).

In yet another work [85], four images of the same slice location, corresponding to subsequent moments of contrast agent propagation (pre-contrast, arterial, portal venous and delayed phase) were analyzed simultaneously. Here, four hepatic tissue classes were differentiated: normal, cyst, haemangioma and HCC. In contrast to the above cited works, this work considered only the combinations of the mean pixel values of ROI in different phases as temporal features. These were: *relative signal intensity*, *intensity change tendency*, and *signal enhancement ratio*. In addition, a few sets of textural features (gray-level histogram-based, COM-based, and selected features) were used in the four single-phase classification tasks. As a classifier, three hierarchically organized binary SVMs were used. The classification accuracy was

assessed by k-fold cross validation. Here, the application of temporal characteristics did not result in better tissue recognition, in comparison with the best results obtained with textural features for each separate moment of contrast agent propagation. It is with a set of combined features (FOS, COM, and temporal) that the best classification accuracy was achieved: 95.5%, 97.2% and 96.4% for normal vs abnormal, cyst vs other disease and carcinoma vs haemangioma sub-problems, respectively.

Finally, in [86], 61 textural features were evaluated in the task of distinguishing between four classes of liver tissue: HCC, cholangiocarcinoma, cirrhosis, and healthy. The study involved: 4 first order statistics, 4 gradient-based features, 11 COM features, 8 RLM features, 5 GLDM features, 19 features obtained with Laws' filtering, 2 fractal dimension estimates, 7 TFCM-based statistics, and 1 normalized autocorrelation coefficient. Such features were calculated separately for each for the three considered moments of contrast agent propagation: no-contrast, arterial phase, and portal phase. In total $3 \times 61 = 183$ tissue descriptors were considered. The choice of the most useful features proceeded in two stages. At the beginning, unstable features (sensitive to small changes in ROI size and/or in ROI position) were rejected. Then, a simplified Monte Carlo feature selection (initially proposed by Draminski et al. [87]) was performed in order to find the most robust features. Classification experiments were performed using an Adaptive Boosting (AdaBoost) algorithm [88] with a C4.5 tree [89]. They revealed that a small set of 12 features was able to ensure classification accuracy exceeding 90%, while all of the 183 features provided an accuracy rate of 88.94%.

Table 3 summarizes the most important information about selected CAD systems, adapted for characterization of multi-image liver CT textures.

3. Conclusion

A vast variety of CAD systems adapted for recognition of liver disorders from CT images were developed over the past 20 years. The most frequently diagnosed pathologies were: primary malignant liver tumors (like HCC or cholangiocarcinoma), secondary tumors (different types of metastases), benign liver tumors (hemangiomas) or other benign liver changes, like steatosis (fatty change), cirrhosis, or fibrosis. Despite numerous proposals for texture analysis methods, that can be found in the literature, the presented systems use only a few approaches for texture characterization. The most popular are those that use: co-occurrence matrices, run length matrices, first order statistics, fractal models, Laws' texture energy measures, and different frequency methods. Each of described systems was tested on different data (different were: image resolutions – spatial and in gray levels, preprocessing techniques, numbers of

Table 3. Comparison of selected CAD systems based on multi-image texture analysis, involving non-enhanced and contrast-enhanced liver CT images

Work	**Year**	**TA Methods**	**Tissue Classes**	**Phases**
Duda et al. [74]	2006	– FOS – COM – RLM – LTE	– healthy (150) – HCC (150) – cholangiocrcinoma (150)	– no contr. – arterial – portal
Ye et al. [85]	2009	– FOS – COM	– normal (64) – cysts (14) – haemangioma (27) – HCC (26)	– no contr. – arterial – portal – delayed
Quatrehomme et al. [76]	2013	– FOS – MRF – LTE – UHS	– cysts (25) – adenomas (10) – HCC (13) – metastases (38)	– no contr. – arterial – portal – delayed
Chi et al. [79]	2013	– FOS – COM	– HCC (16) – hemangioma (16) – cysts (15) – liver abscess (7) – FNH (5) – metastases (10)	– no contr. – arterial – portal – delayed
Duda et al. [86]	2013	– FOS – COM – RLM – GLDM – GM – TFCM – AC – LTE	– normal (573) – cirrhosis (433) – HCC (319) – cholangiocarcinoma (222)	– no contr. – arterial – portal

ROIs, ROI sizes, classification algorithms, ...). Different methods were used for the evaluation of the system classification performance (leave-one-out, cross-validation, using a training set). Therefore, it is difficult to conclude which TA method could be the best possible one. Nevertheless, it can be noticed, that some methods have proven to be reliable for each classification task. For example, the COM-based method was successfully used for both the classification of non-enhanced images (acquired without contrast agent) and of enhanced images (acquired after administration of contrast agent). Other methods were frequently considered only for one type of images. In the case of the cited works, the fractal model-based texture features were of frequent consideration for non-enhanced images, whereas the first order statistics and run length matrices were most often utilized for the enhanced ones. Some experiments have also shown that image pre-filtering (like with WT, DWT, BWT, DFT), performed before extraction of the first- and the second order texture features, could lead to better tissue characterization (in terms of classification process) than the use of statistical methods alone. The comparison of results for non-enhanced an enhanced single-image textures shows that considering the texture changes introduced with the presence of the contrast agent could be a better solution. Finally, it is with the multi-image texture analysis, that the best results could be achieved.

Abbreviations

AdaBoost: Adaptive Boosting algorithm
AUC: Area Under the ROC Curve
BEP: "Bull's Eye Percentage"
BWT: Biorthogonal Wavelet Transform
CAD: Computer-Aided Diagnosis
CCT: Contourlet Coefficient Texture features
COM: Co-Occurrence Matrix
CT: Computed Tomography
DCE: Dynamic Contrast-Enhanced
DFT: Discrete Fourier Transform
DWT: Discrete Wavelet Transform
FM: Fractal Model
FNH: Focal Nodular Hyperplasia
FOS: First Order Statistics
FROC: Free-Response ROC Curves
GAs: Genetic Algorithm for feature selection

GF: Gabor Filters
GLDM: Gray Level Difference Matrix
GLH: Gray Level Histogram
GM: Gradient Matrix
GMRF: Gaussian Markov Random Fields
GTDM: Gray Tone Difference Matrix
GTF: Gabor Texture Features
HCC: Hepatocellular Carcinoma
k-NN: *k*-Nearest Neighbors (classifier)
LTE: Laws' Texture Energy
NA: Autocorrelation Coefficients
NN: Neural Network (classifier)
OAA: One-Against-All
OAO: One-Against-One
OWT: Orthogonal Wavelet Transform
PCA: Principal Component Analysis
RBF: Radial Basis Function
RLM: Run Length Matrix
ROC: Receiver Operating Characteristic
ROI: Region of Interest
SBE: Sequential Backward Elimination
SFFS: Sequential Floating Forward Selection
SFS: Sequential Forward Selection
SVM: Support Vector Machines (classifier)
TA: Texture Analysis
TFCM: Texture Feature Coding Method
UHS: Unser Histograms Statistics
WCT: Wavelet Coefficient Texture features

References

[1] Calvien P.A.: Malignant Liver Tumors: Current and Emerging Therapies, Second Edition, Jones and Bartlett Publishers, Inc., London, UK, 2004.

[2] Lencioni, R., Cioni, D., Batolozzi, C.: Focal Liver Lesions: Detection, Characterization, Ablation (Medical Radiology / Diagnostic Imaging), Springer-Verlag Berlin-Heidelberg, 2005.

[3] Duda, D.: Texture analysis as a tool for medical decision support. Part 1: Recent applications for cancer early detection, Advances in Computer Science Research 11, 2014, pp. 61-84.
[4] Haralick, R. M., Shanmugam K., Dinstein I.: Textural features for image classification, IEEE Trans. Syst., Man, Cybern., Syst 3, 1973, pp. 610-621.
[5] Bankman, I.N.: Handbook of Medical Image Processing and Analysis, Second Edition, Academic Press, 2008.
[6] Conners, R.W., Harlow, C.A.: Toward a structural textural analyzer based on statistical methods, Comput. Vision Graph. 12(3) 1980, pp. 224-256.
[7] Galloway, M.M.: Texture analysis using gray level run lengths, Comput. Vision Graph. 4(2), 1975, pp. 172-179.
[8] Chu, A., Sehgal, C.M., Greenleaf, J.F.: Use of gray value distribution of run lengths for texture analysis, Pattern Recognition Letters, 11(6), 1990, pp. 415-420.
[9] Albregtsen, F., Nielsen, B., Danielsen, H.E.: Adaptive gray level run length features from class distance matrices, Proc. 15th Int. Conf. on Pattern Recognition, 3, 2000, pp. 738-741.
[10] Weszka, J.S., Dyer, C.R., Rosenfeld, A.: A Comparative Study of Texture Measures for Terrain Classification, IEEE Trans. Systems, Man, Cybernetics, 6, 1976, pp. 269-285.
[11] Lerski, R.A., Straughan, K., Shad, L., Boyce, D., Bluml, S., Zuna, I.: MR Image Texture Analysis – An Approach to Tissue Characterization, Magn. Reson. Imaging 11(8), 1993, pp. 873-887.
[12] Horng, M.H., Sun, Y.N., Lin, X.Z.: Texture feature coding method for classification of liver sonography, Comput. Med. Imag. Grap 26(1), 2002, pp. 33-42.
[13] Gonzalez, R.C., Woods, R.E.: Digital Image Processing, Second edition, Reading, MA: Addison-Wesley, 2002.
[14] Mandelbrot, B.: The Fractal Geometry of Nature, block W. H. Freeman and Co., NY, 1982.
[15] Chen, C., Daponte, J.S., Fox, M.D.: Fractal feature analysis and classification in medical imaging, IEEE Trans. Med. Imag. 8(2), 1989, pp. 133-142.
[16] Chen, E.L., Chung, P.C., Chen, C.L., Tsai, H.M., Chang, C.I.: An automatic diagnostic system for CT liver image classification, IEEE Trans. Biomed. Eng. 45(6), 1998, pp. 783-794.
[17] Li, J., Du, Q.: An improved box-counting method for image fractal dimension estimation, Pattern Recognition 42(11), 2009, pp. 2460-2469.
[18] Sankar, D., Thomas. T.: Fractal Features based on Differential Box Counting Method for the Categorization of Digital Mammograms, International Journal

of Computer Information Systems and Industrial Management Applications 2, 2010, pp. 11-19.

[19] Landini, G., Rippin, J.W.: Notes on the implementation of the mass-radius method of fractal dimension estimation, Comput. Appl. Biosci. 9(5), 1993, pp. 547-550.

[20] Jones, C.L., Jelinek, H.F.: Wavelet packet fractal analysis of neuronal morphology, Methods 24(4), 2001, pp. 347-458.

[21] Pentland, A.P.: Fractal-Based Description of Natural Scenes, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-6(6), 1984, pp. 661-674.

[22] Maragos, P., Sun, F.K.: Measuring the Fractal Dimension of Signals: Morphological Covers and Iterative Optimization, IEEE Trans. Signal Process. 41(1), 1993, pp. 108-121.

[23] Backes, A.R., Bruno, O.M.: A new approach to estimate fractal dimension of texture images, In Elmoataz, A., Lezoray, O., Nouboud, F., Mammass, D. (Ed): Lect. Notes Comput. Sc. 5099, Springer, 2008, pp. 136-143.

[24] Kilic, K.I., Abiyev, R.H.: Exploiting the synergy between fractal dimension and lacunarity for improved texture recognition, Signal Processing 91(10), 2011, pp. 2332-2344.

[25] Mallat, S.G.: A theory for multiresolution signal decomposition: The wavelet representation, IEEE Trans. Pattern Anal. Mach. Intell. 11(7), 1989, pp. 674-693.

[26] Laws, K.I.: Textured image segmentation, PhD thesis, University of Southern California, 1980.

[27] Mir, A.H., Hanmandlu, M., Tandon, S.N.: Texture analysis of CT images, IEEE Eng. Med, Biol. Mag. 14(6) 1995, pp. 781-786.

[28] Husain, S.A., Shigeru E.: Use of Neural Networks for feature based recognition of liver region on CT images, Proc. of the IEEE NNSP Work. 2, 2000, pp. 831-840.

[29] Sariyanni, C.P.A., Asvestas, P., Matsopoulos, G.K., Nikita, K.S., Nikita, A.S., Kelekis, D.: A fractal analysis of CT liver images for the discrimination of hepatic lesions: A comparative study, Proc. of the 23rd Annual EMBS Int. Conf., 2001, pp. 1557-1560.

[30] Gletsos, M., Mougiakakou, S.G., Matsopoulos, G.K., Nikita, K.S., Nikita, A.S., Kelekis, D.: A computer-aided diagnostic system to characterize CT focal liver lesions: design and optimization of a Neural Network classifier, IEEE Trans. Inf. Technol. Biomed. 7(3), 2003, pp. 153-162.

[31] Valavanis, I., Mougiakakou, S.G., Nikita, K.S., Nikita, A.: Computer aided diagnosis of CT focal liver lesions by an ensemble of Neural Network and statistical classifiers, Proc. of the IEEE IJCNN Conf. 3, 2004, pp. 1929-1934.

[32] Mala, K., Sadasivam, V.: Automatic segmentation and classification of diffused liver diseases using wavelet based texture analysis and Neural Network, Proc. of the Annual IEEE INDICON Conf., 2005, pp. 216-219.

[33] Huang, Y.L., Chen, J.H., Shen, W.C.: Diagnosis of hepatic tumors with texture analysis in nonenhanced computed tomography images, Acad. Radiol. 13(6), 2006, pp. 713-720.

[34] Stoitsis, J., Valavanis, I., Mougiakakou, S.G., Golemati, S., Nikita, A., Nikita, K.S.: Computer aided diagnosis based on medical image processing and artificial intelligence methods, Nucl. Instrum. Methods Phys. Res., Sect. A 569(2), 2006, pp. 591-595.

[35] Mougiakakou, S.G., Valavanis, I.K., Nikita, A., Nikita, K.S.: Differential diagnosis of CT focal liver lesions using texture features, feature selection and ensemble driven classifiers, Artif. Intell. Med. 41(1), 2007, pp. 25-37.

[36] Ganeshan, B., Miles, K.A., Young, R.C., Chatwin, C.R.: Texture analysis in non-contrast enhanced CT: impact of malignancy on texture in apparently disease-free areas of the liver, Eur. J. Radiol. 70(1), 2009, pp. 101-110.

[37] Mougiakakou, S.G., Valavanis, I.K., Mouravliansky, N.A., Nikita, A., Nikita, K.S.: DIAGNOSIS: A telematics-enabled system for medical image archiving, management, and diagnosis assistance, IEEE Trans. Instrum. Meas. 58(7), 2009, pp. 2113-2120.

[38] Kumar, S.S, Moni, R.S. Rajeesh, J.: An automatic computer-aided diagnosis system for liver tumours on computed tomography images, Comput. Electr. Eng. 39(5), 2013, pp. 1516-1526.

[39] Specht, D.F.: Probabilistic Neural Networks, Neural Networks 3(1), 1990, pp. 109-118.

[40] Fahlman, S.E.: Fast learning variations on back-propagation: An empirical study, Proc. of the 1988 Connectionist Models Summer School, Morgan-Kaufmann, Los Altos CA, 1988.

[41] Beale, M., Demuth, H.: Fuzzy system toolbox for use with Matlab, PWS Publishing Company, 1994.

[42] Jain, A.K., Duin, R.P.W., Mao, J.: Statistical pattern recognition: A review, IEEE Trans. Pattern Anal. Machine Intell. 22(1), 2000, pp. 4-37.

[43] Pudil, P., Novovicova, J., Kittler, J.: Floating search methods in feature selection, Pattern Recogn. Lett. 15(11), 1994, pp. 1119-1125.

[44] Siedlecki W., Sklansky, J.: A note on genetic algorithms for large-scale feature selection, Pattern Recogn. Lett. 10(5), 1989, pp. 335-347.

[45] Goldberg, D.: Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1989.

[46] Fukunaga, K.: Introduction to Statistical Pattern recognition, Second edition, Academin Press, San Diego, CA, USA, 1990.

[47] Vapnik, V.N.: The Nature of Statistical Learning Theory, Second Edition, Springer, NY, 2000.

[48] Duda, R., Hart P., Stork D.: Pattern Recognition, Second Edition, John Willey and Sons, 2001.

[49] Do, M.N., Vetterli, M.: The contourlet transform: an efficient directional multiresolution image representation, IEEE Trans. Image Process. 14(12), 2005, pp. 2091-2106.

[50] Nguyen, T.T., Liu, Y., Chauris, H., Oraintara, S.: Implementational aspects of the contourlet filter bank and application in image coding, EURASIP J. Adv. Signal Process., 2008.

[51] Pearson, K.: On lines and planes of closest fit to systems of points in space, Phil. Mag. 2(6), 1901, pp. 559-572.

[52] Hanley, J.A.: Receiver operating characteristic (ROC) methodology: the state of the art, Crit. Rev. Diagn. Imaging 29(3), 1989, pp. 307-335.

[53] Kretowski, M.: Tissue modeling and classification in biomedical imaging, Ph.D. Dissertation, University of Rennes 1, Rennes, France and Bialystok Technical University, Bialystok, Poland, 2002.

[54] Bobrowski, L., Kretowski, M.: Induction of multivariate decision trees by using dipolar criteria, In Zighed, D.A., Komorowski, J., Zytkow J. (Ed): Lect. Notes Artif. Int. 1910, Springer-Verlag Berlin Heidelberg, 2000, pp. 331-336.

[55] Bilello, M., Gokturk, S.B., Desser, T., Napel, S., Jeffrey, R.B., Beaulieu, C.F.: Automatic detection and classification of hypodense hepatic lesions on contrast-enhanced venous-phase CT, Med. Phys. 31(9), 2004, 2584-2593.

[56] Lambrou, Y., Linney, A.D., Todd-Pokropek, A.: Wavelet transform analysis and classification of the liver from computed tomography datasets, Proc. of the 6th Int. IEEE EMBS Special Topic Conf., 2006.

[57] Smutek, D., Tesar, L., Kobatake, H., Nawano, S., Svacina, S.: Automatic internal medicine diagnostics using statistical imaging methods, Proc. of the 19th Int. IEEE CBMS Symp., 2006, pp. 405-412.

[58] Mala, K., Sadasivam, V., Alagappan, S.: Neural Network based texture analysis of liver tumor from Computed Tomography images," Int. J. Biol. Life Sci. 2(1), 2007, pp. 33-40.

[59] Lee, C.C., Shih, C.Y.: Classification of liver disease from CT image using radial basis function Neural Network, Proc. of the IEEE CSIE World Congress 7, 2009, pp. 656-660.

[60] Miles, K.A., Ganeshan, B., Griffiths, M.R., Young, R.C., Chatwin, C.R.: Colorectal cancer: texture analysis of portal phase hepatic CT images as a potential marker of survival, Radiology 250(2), 2009, pp. 444-452.
[61] Wang, L., Zhang, Z., Liu, J., Jiang, B., Duan, X., Xie, Q., Hu, D., Li, Z.: Classification of hepatic tissues from CT images based on texture features and multiclass Support Vector Machines, In Yu, W., He, H., Zhang, N. (Ed): Lect. Notes Comput. Sc. 5552 part 2, Springer-Verlag Berlin Heidelberg, 2009, pp. 374-381.
[62] Mala, K., Sadasivam, V.: Classification of fatty and cirrhosis liver using wavelet-based statistical texture features and Neural Network classifier, Int. J. Softw. Inform. 4(2), 2010, pp. 151-163.
[63] Kayaalti, O, Aksebzeci, B.H., Karahan, I.O., Deniz, K., Ozturk, M., Yilmaz, B., Kara, S., Asyali, M.H.: Liver fibrosis staging using CT image texture analysis and soft computing, Appl. Soft Comput. 25, 2014, pp. 399-413.
[64] Rao, S.X., Lambregts, D.M.J., Schnerr, R.S., van Ommen, W., van Nijnatten, T.J.A., Martens, M.H., Heijnen, L.A., Backes, W.H., Verhoef, C., Zeng, M.S., Beets, G.L., Beets-Tan, R.G.H.: Whole-liver CT texture analysis in colorectal cancer: Does the presence of liver metastases affect the texture of the remaining liver? United European Gastroenterol. J. 2(6), 2014, pp. 530-538.
[65] Simpson, A.L., Do, R.K., Parada, E.P., Miga, M.I., Jarnagin, W.R.: Texture feature analysis for prediction of postoperative liver failure prior to surgery, Proc. SPIE, 9034, 903414, 2014.
[66] Chakraborty, D.P.: Maximum likelihood analysis of free-response receiver operating characteristic (FROC) data, Med. Phys. 16(4), 1989, pp. 561-568.
[67] Manjunath, B.S., Ma, W.Y.: Texture features for browsing and retrieval of image data, IEEE Trans. Pattern Anal. Machine Intell. 18(8), 1996, pp. 837-842.
[68] Clausi, D.A., Jernigan, M.E.: Designing Gabor filters for optimal texture separability, Pattern Recognition 33(11), 2000, pp. 1835-1849.
[69] Liu, Y., Zheng, Y.F.: One-against-all multi-class SVM classification using reliability measures, Proc. 2005 IEEE Int. Joint Conf. on Neural Networks 2, 2005, pp. 849-854.
[70] Hsu, C.W., Lin, C.J.: A comparison of methods for multiclass support vector machines, IEEE Trans. Neural Netw. 13(2), 2002, pp. 415-425.
[71] Amadasun, M., King, R.: Textural features corresponding to textural properties, IEEE Trans. Syst. Man Cybern. 19(5), 1989, pp. 1264-1274.
[72] Van de Wouwer, G., Scheunders, P., Van Dyck, D.V.: Statistical texture characterization from discrete wavelet representations, IEEE Trans. Image Process. 8(4), 1999, pp. 592-598.
[73] Daugman, J.G.: Two-dimensional spectral analysis of cortical receptive field profiles, Vision Res. 20(10), 1980, pp. 847-856.

[74] Duda, D., Kretowski, M., Bezy-Wendling, J.: Texture-based classification of hepatic primary tumors in multiphase CT, In Barillot, C., Haynor., D.R., Hellier, P. (Ed): Lect. Notes Comput. Sc. 3217 part 2, Springer-Verlag Berlin Heidelberg, 2004, pp. 1050-1051.

[75] Duda, D., Kretowski, M., Bezy-Wendling, J.: Texture characterization for Hepatic Tumor Recognition in Multiphase CT, Biocybern. Biomed. Eng. 26(4), 2006, pp. 15-24.

[76] Quatrehomme, A., Millet, I., Hoa, D., Subsol, G., Puech, W.: Assessing the classification of liver focal lesions by using multi-phase Computer Tomography scans, In Greenspan, H., Muller, H., Syeda-Mahmood, T. (Ed): Lect. Notes Comput. Sc 7723, Springer-Verlag Berlin Heidelberg, 2013, pp. 80-91.

[77] Cross, G.R., Jain, A.K.: Markov random fields texture models, IEEE Trans. Pattern Anal. Mach. Intell. 5(1), 1985, pp. 25-39.

[78] Unser, M.: Sum and difference histograms for texture classification. IEEE Trans. Pattern Anal. Mach. Intell. PAMI-8(1), 1986, pp. 118-125.

[79] Chi, Y., Zhou, J., Venkatesh, S.K., Tian, Q., Liu, J.: Content-based image retrieval of multiphase CT images for focal liver lesion characterization, Med Phys. 40(10), 103502, 2013.

[80] Chi, Y., Zhou, J., Venkatesh, S.K., Huang, S., Tian, Q., Hennedige, T., Liu, J.: Computer-aided focal liver lesion detection, Int. J. Comput. Assist. Radiol. Surg. 8(4), 2013, pp. 511-525.

[81] Yushkevich, P.A., Piven, J., Hazlett, H.C., Smith, R.G., Ho, S., Gee, J.C., Gerig, G.: User-guided 3D active contour segmentation of anatomical structures: significantly improved efficiency and reliability, NeuroImage 31(3), 2006, pp. 1116-1128.

[82] Manjunath, B.S., Salembier, P., Sikora, T. (Ed): Introduction to MPEG-7: Multimedia Content Description Interface, John Wiley & Sons, Inc., New York, NY, USA, 2002.

[83] Ganeshan, B., Miles, K.A., Young, R.C., Chatwin, C.R.: Hepatic entropy and uniformity: additional parameters that can potentially increase the effectiveness of contrast enhancement during abdominal CT, Clin. Radiol. 62(8), 2007, 761-768.

[84] Ganeshan, B., Burnand, K., Young, R., Chatwin, C., Miles, K.: Dynamic contrast-enhanced texture analysis of the liver: initial assessment in colorectal cancer, Invest. Radiol. 46(3), 2011, pp. 160-168.

[85] Ye, J., Sun, Y., Wang, S., Gu, L., Qian, L., Xu, J.: Multi-phase CT image based hepatic lesion diagnosis by SVM, Proc. of 2nd Int. BMEI Conf., 2009, pp. 1-5.

[86] Duda, D., Kretowski, M., Bezy-Wendling, J.: Computer-Aided Diagnosis of Liver Tumors Based on Multi-Image Texture Analysis of Contrast-Enhanced

CT,. Selection of the Most Appropriate Texture Features, Studies in Logic, Grammar and Rhetoric 35(48), 2013, pp. 49-70.

[87] Draminski, M., Rada-Iglesias, A., Enroth, S., Wadelius, C., Koronacki, J., Komorowski, J.: Monte Carlo feature selection for supervised classification, Bioinformatics 24(1), 2008, pp. 110-117.

[88] Quinlan, J.: C4.5: Programs for Machine Learning, Morgan Kaufmann, San Francisco, 1993.

[89] Freund, Y., Shapire, R.: A decision-theoretic generalization of online learning and an application to boosting, J. Comput. Syst. Sci. 55, 1997, pp. 119-139.

ANALIZA TEKSTUR JAKO NARZĘDZIE WSPOMAGANIA DECYZJI MEDYCZNYCH. CZĘŚĆ 2: KLASYFIKACJA PATOLOGII WĄTROBY NA OBRAZACH TOMOGRAFII KOMPUTEROWEJ

Streszczenie: Analiza tekstur jest szeroko stosowana w wielu cyfrowych systemach wspomagania decyzji medycznych, na podstawie danych obrazowych. Pozwala ona wydobyć z obrazu istotne szczegóły, których nie można dostrzec podczas analizy wizualnej. Pierwsze próby analizy tekstur miały miejsce w latach siedemdziesiątych ubiegłego wieku. Od tamtej pory zaproponowano wiele metod analizy tekstur. Trudno jest jednak wskazać metodę uniwersalną, która zapewniłaby zadowalające wyniki dla każdego problemu diagnostycznego. Niniejsza praca stanowi przegląd metod analizy tekstur, stosowanych do opisu tkanki wątrobowej na obrazach tomografii komputerowej. Przedstawia informacje o około czterdziestu systemach diagnostycznych, zaproponowanych w ciągu ostatnich dwóch dekad, poświęconych (pół)automatycznemu wykrywaniu lub / i klasyfikacji schorzeń wątroby. Opisywane systemy zostały podzielone na trzy kategorie: (i) opierające się na teksturze pojedynczego obrazu, pozyskanego bez podawania pacjentowi środka kontrastującego, (ii) opierające się na teksturze pojedynczego obrazu, pozyskanego po podaniu pacjentowi środka kontrastującego, oraz (iii) opierające się na jednoczesnej analizie wielu tekstur. Te ostatnie odnoszą się do analizy zestawów tekstur przedstawiających ten sam wycinek wątroby, lecz odpowiadających różnym stężeniom środka kontrastowego w jej naczyniach krwionośnych.

Słowa kluczowe: obrazowanie medyczne, analiza obrazów, tekstura, selekcja cech, wspomaganie decyzji medycznych, diagnoza wspomagana komputerowo, wątroba, tomografia komputerowa

Artykuł zrealizowano w ramach pracy statutowej S/WI/2/2013.

ESTIMATION OF PARAMETERS OF GAUSSIAN MIXTURE MODELS BY A HYBRID METHOD COMBINING A SELF-ADAPTIVE DIFFERENTIAL EVOLUTION WITH THE EM ALGORITHM

Wojciech Kwedlo

Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

Abstract: In the paper the problem of learning of Gaussian mixture models (GMMs) is considered. A new approach based on hybridization of a self-adaptive version of differential evolution (DE) with the classical EM algorithm is described. In this approach, called DE-EM, the EM algorithm is run until convergence to fine-tune each solution obtained by the mutation and crossover operators of DE. To avoid the problem with parameter representation and infeasible solutions we use a method in which the covariance matrices are encoded using their Cholesky factorizations. In a simulation study GMMs were used to cluster synthetic datasets differing by a degree of separation between clusters. The results of experiments indicate that DE-EM outperforms the standard multiple restart expectation-maximization algorithm (MREM). For datasets with high number of features it also outperforms the state-of-the-art random swap EM (RSEM).

Keywords: Gaussian mixture models, differential evolution, expectation maximization, model-based clustering

1. Introduction

Gaussian mixture models (GMMs) [18] are one of the most versatile probability density models, which are used commonly in machine learning and pattern recognition. They are capable of approximating any multimodal distribution. Applications of GMMs include clustering [7] discriminant analysis [9], speaker recognition [22] and texture segmentation [19].

The standard method for maximum likelihood estimation (MLE) of parameters of GMMs is the expectation-maximization (EM) algorithm [21]. It starts from an initial set of mixture parameters and generates a sequence of mixture parameters with

Advances in Computer Science Research, vol. 11, pp. 109-123, 2014.

increasing log likelihood. However, the application of the EM algorithm to GMM parameter learning has several issues. The most important of these is ease of getting trapped in a local maxima of the log likelihood. Consequently the quality of the final solution is strongly dependent on the initial guess of the mixture parameters.

The most common approach proposed to overcome the above problem is to run the EM algorithm many times, starting each run from different random initial conditions, and return the solution with the highest log likelihood. We call this method multiple restart EM (MREM). However this approach lacks effective utilization of available CPU time, as multiple independent EM procedures are likely to exploit similar local maxima.

In [3], an extension of MREM called *emEM* was proposed. The idea of *emEM* involves performing several short EM runs using different random starting points and a lax convergence criterion. The mixture parameters obtained by the best (in the sense of the highest $\log p(X|\Theta)$) short run are used as a starting point for a long EM run. This strategy can be improved by repeating it many times until the available CPU time is exhausted. A variant of *emEM* called *rndEM* [16] reduces the short EM phase to the evaluation of $\log p(X|\Theta)$ of the random starting position.

Researchers investigating the problem of local maxima of the log likelihood have increasingly started to apply population based global optimization algorithms such as genetic algorithms [1,17,20], particle swarm optimization (PSO) [2] or differential evolution [12]. However, a random nature of of search operators employed by these algorithms makes it difficult to represent covariance matrices, because a random modification of individual elements of covariance matrix usually results in a matrix that is not valid (i.e. symmetric and positive definite). Consequently many applications of global optimization algorithms to problem of GMM learning use diagonal (or even spherical) covariance structure.

To avoid the above restriction on covariance structure, the encoding of covariance matrices in candidate solutions must allow for independent modification of individual parameters [2]. Two such encodings have been proposed so far. In [2] covariance matrix of d-dimensional Gaussian distribution was encoded using d eigenvalues and $d(d-1)/2$ Givens rotation angles. In our previous work [12], the covariance matrix was represented by its Cholesky factorization.

The main contribution of this paper, in comparison with our previous works [12], is the inclusion of the EM algorithm into the process of differential evolution (DE). We show, that DE augmented in such way is able to compete with state-of-the-art GMM parameter estimation methods such as random swap EM algorithm (RSEM) [24].

The rest of the paper is organized as follows. Section 2 presents the problem of GMMs parameter estimation. Section 3 describes the EM algorithm, which is the standard method for GMMs learning. Section 4 presents differential evolution algorithm. Section 5 describes a self-adaptation scheme for two key DE parameters. Section 6 presents the application of a hybrid self-adaptive DE to the problem of GMM parameter estimation. Section 7 presents the results of simulation study in which GMMs were used for data clustering. The last section concludes the paper.

2. Background on GMMs

A finite mixture model $p(\mathbf{x},\Theta)$ is defined by a weighted sum of K components:

$$p(\mathbf{x}|\Theta) = \sum_{m=1}^{K} \alpha_m p_m(\mathbf{x}|\theta_m), \tag{1}$$

where α_m is m-th mixing proportion and p_m is the probability density function of the m-th component. In (1) θ_m is the set of parameters defining the m-th component and $\Theta = \{\theta_1, \theta_2, \ldots, \theta_K, \alpha_1, \alpha_2, \ldots, \alpha_K\}$ is the complete set of the parameters needed to define the mixture. The mixing proportions $\alpha_m \in (0,1)$ are constrained to sum up to 1. In this work we assume, that the number of components K is known a priori.

In GMMs m-th component follows a multivariate Gaussian distributions with mean vector μ_m and covariance matrix Σ_m. Its probability density function is given by:

$$p_m(\mathbf{x}|\theta_m) = \frac{1}{(2\pi)^{d/2}|\Sigma_m|^{1/2}} \exp(-\frac{1}{2}(\mathbf{x}-\mu_m)^T \Sigma_m^{-1}(\mathbf{x}-\mu_m)), \tag{2}$$

where $|\cdot|$ denotes a determinant of a matrix, ^{T} denotes transposition of a matrix, and d is the dimension of the feature space. Thus, for a GMM Θ is defined by: $\Theta = \{\mu_1, \Sigma_1, \ldots, \mu_K, \Sigma_K, \alpha_1, \ldots, \alpha_K\}$.

A standard method for learning the parameters of GMMs is the maximum likelihood estimation (MLE). Given a training set of independent and identically distributed feature vectors $X = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N\}$, where $\mathbf{x}^i = [x_1^i, x_2^i, \ldots, x_d^i] \in R^d$, the log likelihood corresponding to the K-component GMM is given by:

$$\log p(X|\Theta) = \log \prod_{i=1}^{N} p(\mathbf{x}^i|\Theta) = \sum_{i=1}^{N} \log \sum_{m=1}^{K} \alpha_m p_m(\mathbf{x}^i|\theta_m). \tag{3}$$

The maximum likelihood estimate of the parameters is given by:

$$\Theta_{ML} = \underset{\Theta}{\operatorname{argmax}}\{\log p(X|\Theta)\}. \tag{4}$$

It is a well known fact, that a solution of this maximization problem cannot be obtained in a closed form [4]. For that reason a numerical optimization algorithm must be employed to find it.

Model-based clustering [7] is an important application of GMMs. The aim of clustering is to group similar feature vectors together. In this application of GMMs each feature vector is assumed to originate from one K mixture components. We also assume that mixture components are well-separated. The goal of the model-based clustering is to identify, for each feature vector, the mixture component from which it was generated. If we are able to estimate mixture parameters Θ, we can achieve this by allocating e feature vector $\mathbf{x}^i$ to a cluster (mixture component) the highest posterior probability. Using the Bayes theorem this probability for mixture component m can be expressed as:

$$h_m(\mathbf{x}^i) = \frac{\alpha_m p_m(\mathbf{x}^i|\theta_m)}{p(\mathbf{x}^i|\Theta)}. \tag{5}$$

Maximization of (5) is equivalent to finding the mixture index m with the highest value $\alpha_m p_m(\mathbf{x}|\theta_m)$.

3. EM algorithm for GMM learning

The standard method for maximizing (3) is the EM algorithm. It is an iterative algorithm, which, starting from initial guess of a parameters $\Theta^{(0)}$, generates a sequence of estimations $\Theta^{(1)}, \Theta^{(2)}, \ldots, \Theta^{(j)}, \ldots$, with increasing log likelihood (i.e., $\log p(X|\Theta^{(j)}) > \log p(X|\Theta^{(j-1)})$. Each iteration j of the algorithm consists of two steps called expectation step (E-step) and maximization step (M-step) followed by a convergence check. For the GMMs these steps are defined as follows [21]:

1. E-step: Given the set of mixture parameters $\Theta^{(j-1)}$ from the previous iteration, for each $m = 1, \ldots, K$ and $i = 1, \ldots, N$, the posterior probability that a feature vector $\mathbf{x}^i$ was generated from mth component is computed as:

$$h_m^{(j)}(\mathbf{x}^i) = \frac{\alpha_m^{(j)} p_m(\mathbf{x}^i|\theta_m^{(j-1)})}{\sum_{k=1}^{K} \alpha_k^{(j)} p_k(\mathbf{x}^i|\theta_k^{(j-1)})}, \tag{6}$$

   where $\theta_m^{(j-1)}$ and $\theta_k^{(j-1)}$ denote parameters of components m and k, in the iteration $j-1$, respectively.
2. M-step: Given the posterior probabilities $h_m^{(j)}(\mathbf{x}^i)$ obtained in the E-step the set of parameters $\Theta^{(j)}$ is calculated as:

$$\alpha_m^{(j)} = \frac{1}{N} \sum_{i=1}^{N} h_m^{(j)}(\mathbf{x}^i) \tag{7}$$

$$\mu_m^{(j)} = \frac{\sum_{i=1}^{N} h_m^{(j)}(\mathbf{x}^i) * \mathbf{x}^i}{\sum_{i=1}^{N} h_m^{(j)}(\mathbf{x}^i)} \tag{8}$$

$$\Sigma_m^{(j)} = \frac{\sum_{i=1}^{N} h_m^{(j)}(\mathbf{x}^i)(\mathbf{x}^i - \mu_m^{(j)})(\mathbf{x}^i - \mu_m^{(j)})^T}{\sum_{i=1}^{N} h_m^{(j)}(\mathbf{x}^i)} \tag{9}$$

3. Convergence check: The log likelihood $\log p(X|\Theta^{(j)})$ is computed according (3). The algorithm is terminated if the following convergence criterion is met.

$$\frac{\log p(X|\Theta^{(j)}) - \log p(X|\Theta^{(j-1)})}{\log p(X|\Theta^{(j)})} < \varepsilon, \tag{10}$$

where $\varepsilon \ll 1$ is a user defined termination threshold. If the convergence criterion is not met algorithm proceeds to Step 1.

The above algorithm is easy to implement. However it has one important drawback. It is highly sensitive to initialization and easily gets trapped in a local maximum of the log likelihood function. For that reason the quality of the final solution is strongly dependent on the initial guess of the mixture parameters $\Theta^{(0)}$. The problem can be to some degree alleviated by performing multiple runs of the algorithm, each of them starting from different random initial conditions, and returning the result with the highest $\log p(X|\Theta)$. We call this approach multiple restart EM (MREM).

4. Differential evolution

Differential evolution, proposed in [23], is an evolutionary algorithm, which in each generation maintains a population of S solutions to optimization problem. In this section the most common variant with rand/1/ mutation and binomial crossover is described.

Let $u_{i,G}$ denote the i-th member ($i = 1, \ldots, S$) of the population in the G-th iteration. It is assumed that $u_{i,G}$ Is a D-dimensional real-valued vectors (i.e., $u_{i,G} \in \Re^D$).

At the start of the algorithm all population members are initialized randomly. Each generation G consists of three steps. Two of them are mutation and crossover, which for each population element $u_{i,G}$ create a trial solution $y_{i,G}$. The mutation and crossover are followed by a selection, in which fitness of population member $u_{i,G}$ is compared to fitness of the trial solution $y_{i,G}$. The solution with the better (i.e., higher in our application) fitness survives into the next generation:

$$y_{i,G+1} = \begin{cases} y_{i,G} & \text{if} f(y_{i,G}) > f(y_{i,G}) \\ u_{i,G} & \text{otherwise} \end{cases}, \tag{11}$$

where $f : \Re^D \rightarrow \Re$ is the fitness function.

The mutation operator of DE generates a creates a mutant vector $v'_{i,G}$ according to the equation:

$$v'_{i,G} = u_{a,G} + F * (u_{b,G} - u_{c,G}), \tag{12}$$

where $F \in [0,2]$ is a user-supplied parameter called amplification factor and $a,b,c \in 1,\ldots,S$ are randomly selected in such way that $a \neq b \neq c \neq i$.

The final trial vector $y_{i,G}$ is obtained by the crossover operator, which mixes the mutant vector $v_{i,G}$ with the original vector $u_{i,G}$. Let us assume that $u_{i,G} = (u_{1i,G}, u_{2i,G}, \ldots, u_{Di,G})$. Each element $y_{ji,G}$ (where $j = 1,\ldots,D$) of the trial vector $y_{i,G}$ is generated as:

$$y_{ji,G} = \begin{cases} v_{ji,G} & \text{if } rnd(j) < CR \text{ or } j = e \\ u_{ji,G} & \text{otherwise} \end{cases}. \tag{13}$$

where $CR \in [0,1]$ is another user-supplied parameter called crossover factor, $rnd(j)$ denotes a random number from the uniform distribution on $[0,1]$ which is generated independently for each j. $e \in 1,\ldots,S$ is a randomly chosen index which ensures that at least one element of the trial vector $y_{i,G}$ comes from the mutant vector $y'_{i,G}$.

5. Self adaptation of DE control parameters

Experimental studies have shown, that the choice of control parameters F and CR has a significant impact on the performance of DE. In the first experiments with DE [23] the parameters were fixed during the run of the algorithm. Later, some methods for parameter control [6], which change the parameters during the run, were developed. Among these, the approach called self-adaptive parameter control attracted many researchers. In this approach the parameters are encoded into individuals and undergo evolution. The better values of the parameters result in better individuals which are more likely to reproduce and produce offspring and, thus, disseminate these better parameters.

In our DE-EM method, a self-adaptation scheme proposed by Brest et al. [5] was used. It works as follows. Each population element and each trial vector is augmented with its own amplification factor and crossover factor. Let us denote by $F^u_{i,G}$ and $F^y_{i,G}$ the amplification factors associated with the vectors $u_{i,G}$ and $y_{i,G}$, respectively. Similarly, let us denote by $CR^u_{i,G}$ and $CR^y_{i,G}$ the crossover factors associated with the vectors $u_{i,G}$ and $y_{i,G}$, respectively.

Before the mutation $F^y_{i,G}$ is generated as:

$$F^y_{i,G} = \begin{cases} L + rnd_2 * U & \text{if } rnd_1 < \tau_1 \\ F^u_{i,G} & \text{otherwise} \end{cases}. \tag{14}$$

rnd_1 and rnd_2 are uniform random values from $[0,1]$, $\tau_1 \in [0,1]$ is the probability of choosing new random value of $F^y_{i,G}$, L and U are the parameters determining the range for $F^y_{i,G}$.

Similarly to $F^y_{i,G}$, $CR^y_{i,G}$ is generated before the mutation as:

$$CR^y_{i,G} = \begin{cases} rnd_3 & \text{if } rnd_4 < \tau_2 \\ CR^u_{i,G} & \text{otherwise} \end{cases}, \tag{15}$$

where $\tau_2 \in [0,1]$ is the probability of choosing new random value of $CR^y_{i,G}$.

It may seem that self-adaptation of F and CR introduces another four parameters (L, U, τ_1, τ_2) which require a costly fine-tuning using the trial-and-error approach. However, Brest et al. [5] used fixed values of these parameters obtaining very good results for a very diverse range of benchmark numerical optimization problems. Following their advice in our experiments we set $\tau_1 = \tau_2 = 0.1$. L and U were set to 0.05 and 0.35 respectively, which ensured that $F^y_{i,G} \in [0.05, 0.4]$.

6. Application of hybrid self-adaptive DE to the problem of GMM learning

6.1 Representation of GMM parameters

Since DE represents the problem solutions as real-valued vectors the encoding of mixing proportions and mean vectors is very straightforward: they simply are stored in solution vectors using the floating point representation. Unfortunately, this method cannot be use in case of covariance matrices. A covariance matrix of Σ of d-dimensional Gaussian distribution is symmetric, and thus has $d(d+1)/2$ free parameters. However, if a distribution is non-degenerate, this matrix must be positive definite i.e., for each non-zero $\mathbf{x} \in \Re^d$ $\mathbf{x}^T \Sigma \mathbf{x} > 0$ [11]. For that reason it is impossible to store these parameters directly, because matrices obtained by a random operators of crossover and mutation would violate the positive-definiteness constraint [2].

To overcome this obstacle DE-EM uses the representation of covariance matrices, first proposed in [12], based on their Cholesky factorization. Each positive-definite matrix Σ can be decomposed as a product of a lower triangular matrix L with

positive diagonal elements and its transpose [8]:

$$\Sigma = LL^T. \tag{16}$$

The Cholesky factorization of a positive-definite matrix is unique [8]. The matrix L is called the Cholesky factor of Σ or the square root of Σ.

In the DE-EM method the covariance matrices of a GMM are represented in solution vectors of DE by their Cholesky factors. The constraints on Cholesky factors (the diagonal elements must be positive) are easily handled by DE, because each constraint on solution handled independently from the other constraints.

6.2 Fitness function

The fitness function used by DE-EM is $\log p(X|\Theta)$. The selection method is configured to maximize the fitness.

6.3 Hybridization with the EM algorithm

Before the selection step of DE, each candidate solution is fine-tuned by the EM algorithm. First, the solution is used to initialize the EM. Next, the EM algorithm is run until the convergence criterion (10) is met. Then, the solution obtained by the EM algorithm is used in the selection step.

Similar fine-tuning by the EM algorithm is performed on random initial solutions in generation 0.

7. Experimental results

In this section the results of the computational experiments on synthetic datasets, in which the GMMs were used for model-based clustering, are reported. We compared our DE-EM method to two other approaches: the standard multiple restart EM (MREM) and recently proposed [24] random swap EM (RSEM), which is state-of-the-art method for GMM parameters estimation, capable of escaping from local maxima of log likelihood.

The algorithms were implemented in C++ language and compiled by the Intel C++ compiler version 14.0.1 using optimizing options (-O3 -ipo -march=core2 -fno-alias). The compiled programs were run on a Dell Poweredge 1950 server with two quad-core Intel Xeon 5355 (2.66 GHz) processors and 16 GB of RAM, running Ubuntu Linux 12.04. The implementation of EM was parallelized [14] using

OpenMP standard for shared memory computers, taking advantage of all eight cores of the system.

In the experiments we used a generator proposed by [15], which generates randomly Gaussian clusters according to the user-defined overlap characteristic. The overlap ω_{ij} between two clusters i and j is defined as the sum of two misclassification probabilities $\omega_{j|i}$ and $\omega_{i|j}$, where: $\omega_{j|i} = Pr[\alpha_i p(\mathbf{x}|\mu_i,\Sigma_i) < \alpha_j p(\mathbf{x}|\mu_j,\Sigma_j)|\mathbf{x} \sim \mathcal{N}(\mu_i,\Sigma_i)]$, and similarly $\omega_{i|j} = Pr[\alpha_j p(\mathbf{x}|\mu_j,\Sigma_j) < \alpha_i p(\mathbf{x}|\mu_i,\Sigma_i)|\mathbf{x} \sim \mathcal{N}(\mu_j,\Sigma_j)]$.

The overlap characteristic of the generator [15] was controlled by one parameter ω expressing the average pairwise overlap between clusters. In our experiments the number of components K was fixed at 20. Figure 1 shows example two-dimensional training sets simulated from mixtures obtained from the generator for different values of ω. It can be seen that by using different values of ω we can control the separation of clusters.

In our experiments we generated mixtures with dimension $d \in \{5, 10, 25\}$. For each dimension we used $\omega \in \{0.0001, 0.0002, 0.0005, 0.001, 0.0025, 0.005, 0.01, 0.0250, 0.05, 0.1\}$. We used the adjusted Rand index (ARI) [10] to measure the degree of agreement between partitions of data discovered by the clustering algorithms and the original partitions (we knew them because we used synthetic datasets drawn by a random generator, which allowed us to track the source of each feature vector). The ARI is bounded between -1 and 1. The expected value of ARI in case of randomly generated partitions is 0. A higher value of ARI indicates a higher similarity between partitions; a maximum value of 1 means, that two partitions are identical. A similar experimental setting was used for comparison of different EM initialization methods in [13].

The feature vectors were clustered according to the MAP rule, as described in Section 2. Since in this experiment the original (ground truth) mixture parameters were available, we also performed clustering using them.

The experimental protocol was as follows. For every combination of d and ω 50 different random mixtures were generated. For each mixture a single dataset was realized. For $d = 5$ and $d = 10$ the number of feature vectors in dataset was set to 6000. For $d = 25$ we had to increase this number to 30000 to avoid issues with the singularity of covariance matrices. To assure a fair comparison, each of three algorithms was allocated equal CPU time.

Figure 2a shows the obtained values of ARI, (averaged over 50 different mixtures) when clustering was performed on the basis of the ground truth parameters. As expected, whereas for clusters with very small overlap ARI close to 1 (indicating very good agreement between original partitions and clustering results) could be obtained, an increase of overlap between clusters led to lower values of ARI.

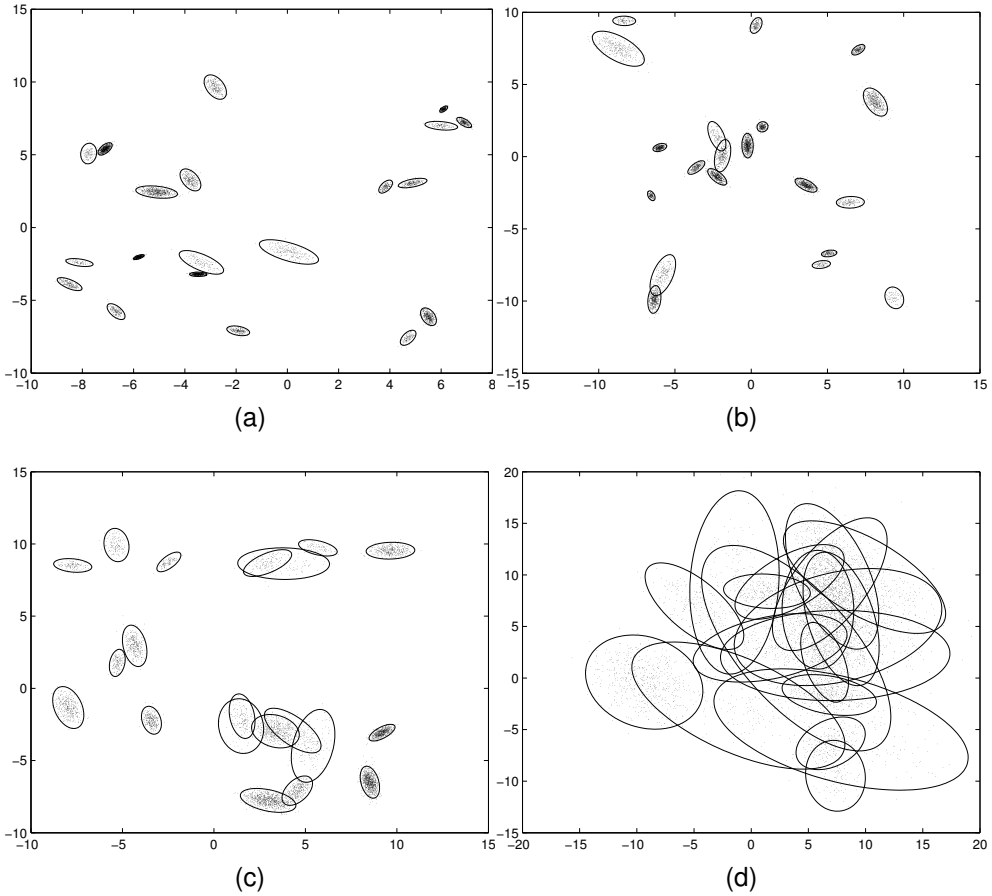


Fig. 1. Two-dimensional training sets simulated from 20–component mixtures with (a) $\omega = 0.0001$, (b) $\omega = 0.001$, (c) $\omega = 0.01$, (d) $\omega = 0.1$. The ellipses are centered around component means and represent 95% confidence regions.

The average ARI values obtained for ground truth mixture parameters were used as the baseline for comparison of three GMM parameter estimation methods. The results concerning these methods are shown on Figures 2b, 2c, 2d. The result of each method is shown as a % error relative ground truth mixture parameters. The % error of the method A is computed as $(\mathrm{ARI}_T - \mathrm{ARI}_A)/\mathrm{ARI}_\mathrm{T} * 100$, where ARI_T is the average (over 50 different mixtures) ARI obtained using the ground truth mixture parameters and ARI_A is average ARI obtained using mixture parameters estimated by the method A. A lower value of % error indicates a better performance, values

close to 0 indicate that clustering using a given GMM parameter estimation method achieves similar results as clustering using the ground truth parameters. The results

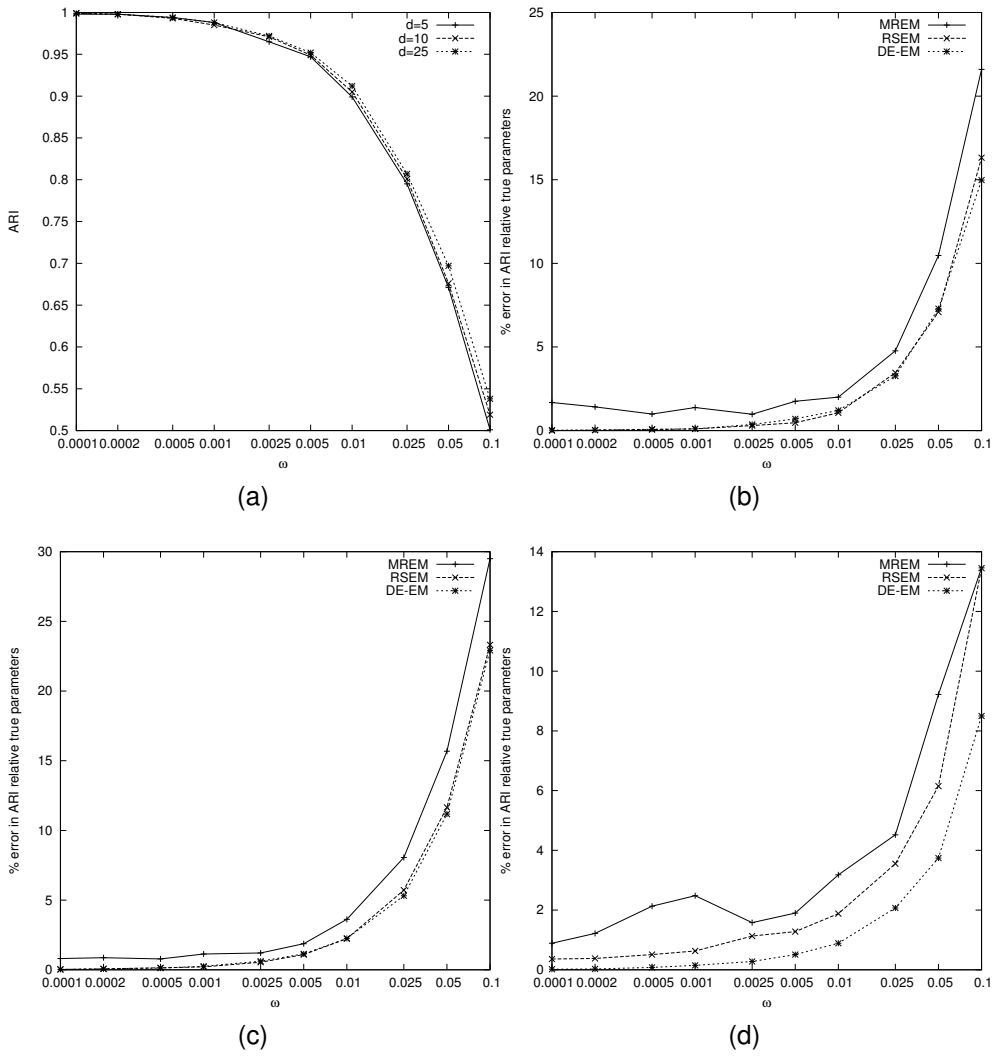


Fig. 2. (a) The average ARI values obtained for clustering using true mixture parameters. % Error in ARI relative true mixture parameters for (b) $d = 5$, (c) $d = 10$, (d) $d = 25$.

from Figures 2b – 2d are summarized by Table 1, which shows the results averaged

over 10 different values of ω separately for each dimension d. The last row of the table shows the total average result for each of compared methods.

Table 1. The average error in ARI relative known mixture parameters

d	MREM	RSEM	DE-EM
5	4.70	2.89	2.81
10	6.35	4.50	4.39
25	4.06	2.93	1.63
Total	5.04	3.44	2.94

The results achieved by the three algorithms indicate that:

- The difference between the results obtained by clustering using each of three estimation methods and clustering using the ground truth parameters widens as the average overlap between clusters is increased.
- MREM is the worst estimation method irrespectively from the dimension d of feature space.
- For $d = 5$ and $d = 10$ the results of the DE-EM method are on par with the RSEM approach. However, our method clearly outperforms RSEM for $d = 25$.

8. Conclusions

In this paper a new method for GMMs learning which combines the self-adaptive differential evolution with the EM algorithm was proposed. To avoid the problem with infeasibility of solutions we used an representation, in which covariance matrices were encoded using their Cholesky factorization.

The results of our study allow us to recommend DE-EM method over the MREM and RSEM algorithms in application of GMMs to clustering problems. Although there was little difference between DEEM and RSEM in experiments where $d = 5$ and $d = 10$ our method was clear winner for more difficult problems where $d = 25$.

In future works we are going to compare the performance of the DE-EM method to other well established hybrid evolutionary algorithms, for instance the GA-EM algorithm [20]. We also plan to use other encodings of covariance matrices, especially based on Givens angles [2]. Finally, we are going to to test the performance of DE-EM method in other applications of GMMs, for instance in discriminant analysis [9].

Acknowledgments

This work was supported by the grant S/WI/2/2013 from Bialystok University of Technology.

References

[1] J. L. Andrews and P. D. McNicholas. Using evolutionary algorithms for model-based clustering. *Pattern Recognit. Lett.*, 34(9):987–992, 2013.

[2] C. Ari, S. Aksoy, and O. Arikan. Maximum likelihood estimation of Gaussian mixture models using stochastic search. *Pattern Recognit.*, 45(7):2804–2816, 2012.

[3] Christophe Biernacki, Gilles Celeux, and Gérard Govaert. Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. *Comput. Stat. Data Anal.*, 41(3):561–575, 2003.

[4] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006.

[5] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*, 10(6):646–657, 2006.

[6] A. E. Eiben, R. Hinterding, and Z. Michalewicz. Parameter control in evolutionary algorithms. *IEEE Trans. Evol. Comput.*, 3(2):124–141, 1999.

[7] C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *J. Am. Stat. Assoc.*, 97(458):611–631, 2002.

[8] G. H. Golub and C. F. van Loan. *Matrix Computations*. Johns Hopkins, Baltimore, MD, 1996.

[9] T. Hastie and R. Tibshirani. Discriminant analysis by Gaussian mixtures. *J. Royal Stat. Soc. Ser. B*, 58(1):155–176, 1996.

[10] L. Hubert and P. Arabie. Comparing partitions. *J. Classif.*, 2(1):193–218, 1985.

[11] R.A. Johnson and D.W. Wichern. *Applied Multivariate Statistical Analysis*. Prentice Hall, 6th edition, 2007.

[12] W. Kwedlo. Learning finite Gaussian mixtures using differential evolution. *Zeszyty Naukowe Politechniki Białostockiej. Informatyka*, 5:19–33, 2010.

[13] W. Kwedlo. A new method for random initialization of the EM algorithm for multivariate Gaussian mixture learning. In *Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013*, pages 81–90. Springer, 2013.

[14] W. Kwedlo. A parallel EM algorithm for Gaussian mixture models implemented on a NUMA system using OpenMP. In *Proceedings of the 22nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing PDP 2014*, pages 292–298. IEEE CPS, 2014.

[15] R. Maitra and V. Melnykov. Simulating data to study performance of finite mixture modeling and clustering algorithms. *J. Comput. Graph. Stat.*, 19(2):354–376, 2010.

[16] Ranjan Maitra. Initializing partition-optimization algorithms. *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, 6(1):144–157, 2009.

[17] A. M. Martinez and J. Vitria. Learning mixture models using a genetic version of the EM algorithm. *Pattern Recognition Letters*, 21(8):759–769, 2000.

[18] G. McLachlan and D. Peel. *Finite Mixture Models*. Wiley, New York, 2000.

[19] H. Permuter, J. Francos, and I. Jermyn. A study of Gaussian mixture models of color and texture features for image classification and segmentation. *Pattern Recognit.*, 39(4):695–706, 2006.

[20] F. Pernkopf and D. Bouchaffra. Genetic-based EM algorithm for learning Gaussian mixture models. *IEEE Trans. Pattern Analysis Mach. Intell.*, 27(8):1344–1348, 2005.

[21] R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Rev.*, 26(2):195–239, 1984.

[22] D.A. Reynolds, T.F. Quatieri, and R.B. Dunn. Speaker verification using adapted Gaussian mixture models. *Digit. Signal Process.*, 10(1):19–41, 2000.

[23] R. Storn and K. Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.*, 11(4):341–359, 1997.

[24] Q. Zhao, V. Hautamäki, I. Kärkkäinen, and P. Fränti. Random swap EM algorithm for Gaussian mixture models. *Pattern Recognit. Lett.*, 33(16):2120–2126, 2012.

ESTYMACJA PARAMETRÓW MODELI MIESZANIN ROZKŁADÓW NORMALNYCH PRZY POMOCY METODY HYBRYDOWEJ ŁĄCZĄCEJ SAMOADPTACYJNĄ EWOLUCJĘ RÓŻNICOWĄ Z ALGORYTMEM EM

Streszczenie: W pracy poruszono problem uczenia modeli mieszanin rozkładów normalnych. Zaproponowano nowe podejście, nazwane DE-EM, oparte na hybrydyzacji samodaptacyjnego algorytmu ewolucji różnicowej i klasycznego algorytmu EM. W nowej metodzie rozwiązanie otrzymane jako wynik operatorów mutacji i krzyżowania jest poddawane optymalizacji lokalnej, prowadzonej aż do momentu uzyskania zbieżności, przez algorytm EM. Aby uniknąć problemu z reprezentacją macierzy kowariancji i niedopuszczalnością rozwiązań użyto metody, w której macierze kowariancji są kodowane przy pomocy dekompozycji Cholesky'ego. W badaniach symulacyjnych modele mieszanin rozkładów normalnych zastosowano do grupowania danych syntetycznych. Wyniki eksperymentów wskazują, że metoda DE-EM osiąga lepsze wyniki niż standardowa technika wielokrotnego startu algorytmu EM. Dla zbiorów danych z dużą liczbą cech, metoda osiąga lepsze wyniki niż technika losowej wymiany rozwiązań połączona z algorytmem EM.

Słowa kluczowe: Mieszniny rozkładów normalnych, ewolucja różnicowa, algorytm EM, grupowanie danych

Artykuł zrealizowano w ramach pracy badawczej S/WI/2/2013.

REAL-TIME ENGINE SOUND GENERATOR BASED ON ANALYSIS OF VIDEO AND RECORDED SAMPLES

Marcin Skoczylas

Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

Abstract: Generating engine sound samples is a broad topic known for decades, mostly because of high usage of such algorithms in driving car simulations, especially in games. These algorithms differ from very simple looped short sound players that change the frequency of prerecorded samples to sophisticated algorithms that model the engine sound based on some defined characteristics. The latter are computationally extensive and can't be used in mobile environment (such as smartphones). In this paper author presents own approach to use visual analysis techniques to prepare a database of multiple recorded sound samples and a mixer that can replay these sounds in proper order to mimic an engine sound in real-time.

Keywords: engine sound generator, recorded samples, mixer, visual anaylysis, Hough Transform

1. Introduction

Generating automotive engine sound samples is a broad topic known for decades, mostly because of high usage of such algorithms in games, especially driving car simulations, also professional ones such as presented in [14]. Engine sounds generator is very important for the feeling of speed during the simulation and is inseparable unit of the whole experience, giving constant impression of speed. To accurately model the sound of a motor vehicle, in direct response to the interactivity of the simulation, there are many challenges in order to represent it as realistic as possible. These algorithms differ from very simple looped short sound players that change the frequency of prerecorded samples to sophisticated algorithms that model the engine sound based on some defined characteristics. First approaches to simulate car engine sounds were done back in '80s using sine waves with eventually distortion added. The first racing vehicle sounds were simulated using Revolutions Per Minute (RPM) motor parameter and simple wave synthesis, so that the increase of RPM increased

Advances in Computer Science Research, vol. 11, pp. 125-136, 2014.

tone and motor sounds frequency. Even at this basic level, driving speed was communicated to the user and though these algorithms sounded very simple but were enough to attract. One of very popular algorithms to represent engine sound is to use simple, looped sounds with playback frequency correlated to the RPM value. The original engine sound is recorded, played in a loop and then its frequency is changed depending on target RPM of the engine. Other algorithms use procedural engine models to properly create dynamics in response to the player interaction, for example using FFT analysis and granular synthesis, distortion and effects such as delay or flange. These sophisticated algorithms model the sound based on some defined characteristics of the engine. The latter are computationally extensive and can't be used in mobile environment (such as smart phones).

In this paper author presents own approach to use visual analysis techniques to prepare a database of multiple recorded sound samples and a mixer that can replay these sounds in proper order to mimic an engine sound in real-time.

2. Related work

Modeling of engine sound is not very popular research topic. Most existing solutions are done commercially for the purpose of generating sounds for games and the idea behind algorithms is not publicly shared. However, there are few publications that are worth noting here. In [4] authors modeled and synthesized engine sounds using a deterministic-stochastic signal decomposition approach, the deterministic component was extracted using a FFT method and it was subtracted out from the original signal and then the stochastic component was modeled and synthesized using a new multipulse excited time-series modeling technique. The technique gives very good results, however is quite computationally extensive, thus can't be implemented in mobile smartphones. Very often the sound generation is connected with implementation of a physics engine, as presented recently in [15] or [13]. Authors created a framework that allows virtual object contact sounds to be synthesised in real time, eventually adding a possibility to create car engine sounds. A very promising work is presented in [9], where authors presented deep analysis of engine sounds and some ideas are also base for this publication. Other approaches also exist to generate engine sounds in real time, recently very popular became hardware boxes from Sonory Engine Sound Synthesis [1]. These boxes can be hooked up to in-car stereo radio system to replay artificial sounds of V8 engine based on RPM readings from internal computer. Although these are implemented in hardware, still the sound is simulated and synthesized thus it is not easily possible to change the characteristics to generate different engine sounds.

Engine sound analysis is much more popular topic in research, and mainly concerns engine fault detection as for example in [10], a mobile application that analyzes the engine sound and detects engine faults using discrete wavelet transform.

In this paper author presents own approach to use visual analysis techniques to prepare a database of multiple recorded sound samples and a mixer that can replay these sounds in proper order to mimic an engine sound in real-time.

3. Engine sound generator method

In this paragraph idea of the method to generate arbitrary engine sounds is presented. To avoid computationally extensive engine sound synthesis and obtain realistic effect, taking into account that the algorithm should be able to generate natural sounding vehicle engine audio of multiple different engines and it should be relative easy to switch and obtain sounds of other cars, author decided to create an algorithm that creates and automatically organizes a bank of short sample sounds that will be used for mixing by the engine sound player in real-time. In overall, to correlate engine sounds with RPM readings one can use a plug to the car's computer to obtain current RPM readings. Unfortunately, not all available cars easily support sending a feedback of the current RPM to a PC, especially old cars that do not have on-board computers. Thus to avoid this problem and allow engineers to record and model sounds of very old cars, a video analysis algorithm was created and current RPM is obtained from a video frame.

In overall, this method contains a step that involves sound and video analysis of recordings of the original sound of the engine. The setup consists of a video camera that records readings of the RPM dial and in addition microphone that is attached to the car body to record sounds. Sounds are recorded together with video of the RPM dial on site, eventually operator drives a car with a load attached. These recordings are then marked and transferred to a PC that performs video analysis. Sounds and video frames are extracted and analyzed using the algorithm described below to create a bank of organized short audio samples marked with the RPM readings.

3.1 Video analysis and extracting RPM readings

To allow users to record sounds of cars with significant load, recordings must be done within a car that is in motion. Very often the mounted camera creates shaky videos, especially when car drives on uneven terrain. Thus before to obtain a RPM reading from one video frame, first that video needs to be stabilized.

Many different video stabilization techniques exist. Main purpose of this step is to reduce in-between frames motion. Author decided to create his own approach to video stabilization based on detection of features in images and so-called keypoints detectors and descriptors. First a marker is defined manually, this could be a vehicle logo or some other element which is visible on all frames and that does not overlap the RPM dial. This marker image is used as a reference point to stabilize the video. From this marker image keypoints are detected and descriptors are calculated.

Image feature descriptors are becoming a standard in current state of the art of image recognition algorithms. For this study, author selected most common and popular feature detectors: Scale-Invariant Feature Transform (SIFT [12]), Speeded Up Robust Features (SURF [5]) and also recently presented Binary Robust Invariant Scalable Keypoints (BRISK [11]). Results of the accuracy of marker image detection are presented in section 4.

In a new video frame a marker image is detected. First, keypoints are found using the same feature detector, these keypoints form a set $K_c = \{p_1, p_2, ...\}$ and are considered as candidates for keypoints that correspond to the marker image. For all keypoints in the set K_c feature descriptors D_c are calculated, so that each element from set K_c corresponds to one descriptor from set D_c. A nearest neighbour kNN search is performed on detected keypoints in a new frame and marker keypoints with[1] $K = 2$. Found pairs are filtered to find good matches using technique described in [7]: first, the minimum distance (*min*) is found from all matches, and then all distances that are bigger than $2 \cdot min$ are discarded. If the amount of keypoints in a set containing found matches is less than 4 (thus, at least four corners), then the marker is not detected and that frame is skipped: stabilization is not performed, the marker image is not visible. In other case the marker is detected and a homography is found using a RANSAC [8] algorithm using pairs of keypoint matches and then perspective matrix transformation of vectors is performed. If the transformed polygon is not convex, then the marker is considered not detected.

When a position of the marker on a new video frame is known, then a relative position to the original position of the marker can be calculated. The whole video frame is then shifted in the opposite direction to overlay a marker from new frame on a marker in the previous (original) frame and video stabilization is continued for next frames.

[1] The k=2 in kNN is suggested by J. Beis and D. Lowe in their Best-Bin-First (BBF) algorithm [6]

3.2 Detection of angle of the RPM pointer

When video is stabilized a next step of the algorithm is detection of the RPM pointer and then its angle. Knowing the angle one can calculate also the RPM reading. All video frame images are processed to obtain number of RPMs connected with a video frame using following algorithm steps:

1. Image is converted to a gray scale color space.
2. Binarisation filter is applied with threshold calculated using Mean Iterative Selection [2]. During each iteration the average brightness T_B is determined for all pixels below the estimated threshold, and similarly also the average brightness T_W of all pixels above value of the estimated threshold. The new value of the estimated threshold value is calculated as the average of the two values T_B and T_W. The general formula for calculating the estimated value of the threshold for the histogram h:

$$T_k = \frac{\sum_{i=0}^{T_{k-1}} i * h[i]}{2\sum_{i=0}^{T_{k-1}} h[i]} + \frac{\sum_{i=T_{k-1}+1}^{N} i * h[i]}{2\sum_{i=T_{k-1}+1}^{N} h[i]} \tag{1}$$

   The moment of stopping the algorithm is the condition:

$$(T_B = T_W) \vee (T_{k-1} = T_k) \tag{2}$$

3. Boundaries of the dial are detected using a sweep algorithm and pixels that reside inside a circle of the RPM dial are extracted and considered for further analysis.
4. A Hough Transform is calculated. From the resulting Hough Transform matrix the best representative of the angle (θ) and offset of the most visible line is chosen. When these representatives are multiple then values are averaged.
5. RPM pointer angle is detected using data from the previous step. It is possible that angle will be not correct. The Hough Transform result can detect angle from opposite part of the circle, and in such cases that angle needs to be corrected. First a RPM dial is divided into 4 equal parts. Then, amount of pixels that reside in each quarter is calculated and from that it is known in which quarter the pointer is located. Then simple correction calculations are performed, for example, if pointer is detected in left-down quarter and θ is higher than 220°then $\theta = |180^\circ - \theta|$, etc.
6. Finally, a RPM angle value is linearly scaled to reflect RPM values range, thus for example 180° becomes 3000 RPMs and so on.

RPM pointer readings calculated from the angle are stored for all video frames. For each frame as a result from this step a RPM reading is stored together with short sound sample which has a duration of one frame: $1/(frames\ per\ second)$. These readings will be organized into bulk groups of samples. The algorithm is explained in following paragraph.

Example of the RPM pointer detection is presented in Figure 1, and Hough Transform matrix used to RPM pointer angle detection is visible in Figure 2.

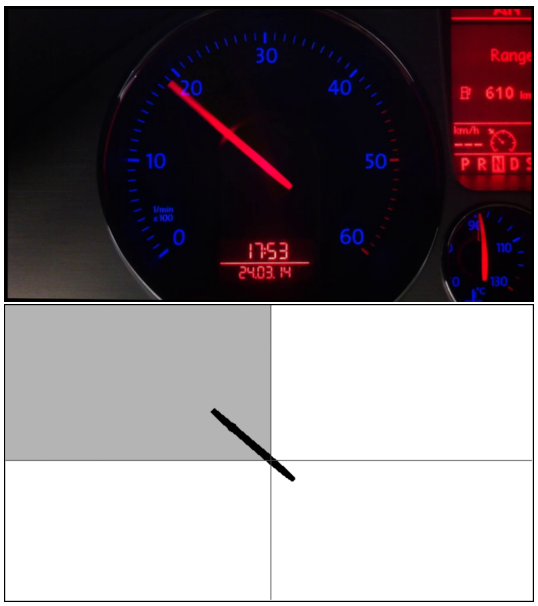


Fig. 1. Example of RPM pointer detection steps. Original video frame image, segmented pointer and divided dial with selected quarter highlighted.

3.3 Samples database creation

In previous paragraph all video frames were analyzed and RPM readings were stored. Next step organizes sound samples into three groups: constant RPM, accelerate and decelerate groups of samples. For this purpose vector of all RPM values is analyzed. First the RPM signal is smoothed using gaussian smooth filter and then for each frame discrete derivative is calculated. Knowing derivatives then groups of samples that consist of acceleration, deceleration or constant RPM values are found, however

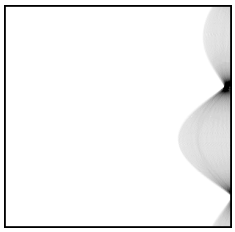


Fig. 2. Hough Transform matrix calculated on segmented pointer example from Figure 1.

Table 1. Example temporary data structure for storage of detected RPMs together with video frames. From this structure three groups of samples are created.

Frame #	RPM	State	Start Frame #	End Frame #	Min RPM	Max RPM
1	750	Constant	1	3	750	770
2	770	Constant	1	3	750	770
3	760	Constant	1	3	750	770
4	800	Accelerate	4	6	760	900
5	850	Accelerate	4	6	760	900
6	900	Accelerate	4	6	760	900
7	850	Decelerate	7	9	900	750
8	800	Decelerate	7	9	900	750
9	750	Decelerate	7	9	900	750

to avoid erroneous state changes with too rapid differences, frames are analyzed in clusters of 3 frames. Thus, to confirm detection of a state change from constant to acceleration, then three consecutive frames have to show the RPM acceleration. In addition an information in which frame this group started and in which frame that state ended is stored, as well as RPM for that group of samples. All groups of samples are kept in three separate buckets: constant RPMs, accelerations and decelerations. See as an example Table 1 where the concept is shown. Please note that RPM for a frame starts as a previous value and finishes as current frame's RPM (for example see frame #4). This example stores links to three groups of samples detected, one constant, one accelerate and one decelerate group. Start, end frames and start, end RPMs are stored with groups and dataset is created.

3.4 Method to select group of samples during playback

The real-time player of the engine sounds is playing recorded samples based on information obtained after analysis from the previous sections. The input parameters to the player are target RPM (R_{target}) and the engine load. The engine load parameter

just selects proper bank of previously analyzed samples. R_{target} is a RPM value to which the engine should aim and select consecutive sound samples.

The algorithm selects samples to be played based on dependencies of current RPM ($R_{current}$) and target RPM (R_{target}). Knowing RPM of the sound sample that was previously played the algorithm selects new samples to be played as follows:

1. If the R_{target} is equal $R_{current}$ then select a random group of samples from 'Constant' samples bucket that has RPM equal to the R_{target}.
2. If the R_{target} is higher than $R_{current}$ then select a random group of samples from 'Accelerates' samples bucket that has minimum RPM equal or below the $R_{current}$ and the maximum RPM equal or higher than the R_{target}. If the $R_{current}$ is not equal starting RPM of this group of samples then scroll the sample to frame that matches $R_{current}$.
3. If the R_{target} is less than $R_{current}$ then select a random group of samples from 'Decelerates' sample bucket that has minimum RPM equal or higher than the $R_{current}$ and maximum RPM equal or below the R_{target}. If the $R_{current}$ is not equal starting RPM of this group of samples then scroll the sample to frame that matches $R_{current}$.

The above algorithm is run every sample step. Note, that if a current group of samples that is played resides in 'Accelerates' bucket and maximum RPM from the group is higher than R_{target}, then that group is replaced by a group of samples from the 'Constant' bucket when $R_{current}$ reaches R_{target}. Analogously, if a current group of samples that is played resides in 'Decelerates' bucket and minimum RPM in a group is lower than R_{target}, then that group is replaced by a group of samples from the 'Constant' bucket when $R_{current}$ reaches R_{target}. That group replacement is achieved by the first step of the algorithm.

To quickly select proper group of samples a Red-Black tree is used as a storage structure and search algorithm. In addition number of uses of a particular group of samples is recorded. When a group of samples is selected to be played then the number of usages is increased. Algorithm selects new group of samples based on the amount of previous usages, thus it favors selection of a group that was not played before than replaying the same again. This approach avoids repeatable loops of the same group of samples.

Furthermore, a classical volume (amplitude) ramping method [3] is used, so each sample group to be played is overlapped during mixing with previous to avoid clicks and noise in a place of cut (see Figure 3).

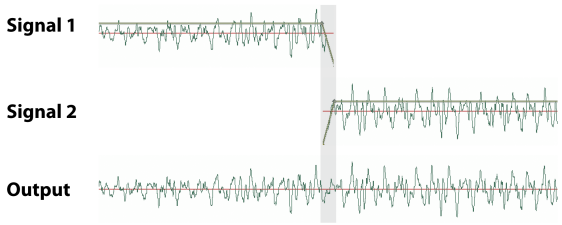


Fig. 3. Ramp-mixing of two groups of samples (signals). Green horizontal lines over the signals show output volume. The Output signal does not have clicks when two signals are joined.

4. Results

4.1 Comparison of marker detection algorithm

First results of marker detection algorithm from section 3.1 for the purpose of video stabilization will be presented. Image marker was extracted from first video frame and other video frames were processed. Resolution of the marker was 240×160 pixels. Normally video recording parameters can change, so to mimic this video frame images were changed: scale, noise, rotation, blur and lightness filters were applied and detection results gathered. A *ratio* parameter was obtained in such way: first from the original marker image keypoint descriptors were calculated, then video frames were changed and on these amended images marker was detected using algorithm from section 3.1. The *ratio* is a number of properly matched keypoints in a new image (that are positioned inside marker area) divided by the original number of keypoints from the marker. A ratio of 1.0 means that all the keypoints from the marker image were found properly in the amended image. Results of this experiment are presented in Table 2. It is clearly seen that SURF algorithm is performing best video stabilization, surprisingly weak result of BRISK algorithm can be explained by low resolution of the marker image.

4.2 Engine sound generator

It is not easy to create a comparatory review of the algorithm presented in this paper. It is possible however to create a spectrogram of sounds generated by real vehicle engine and compare it with results generated by the algorithm presented. These spectrograms should not be equal, as real engine sound is not generating exactly the same sound every time and also it depends on the RPM throttle, so frequencies will vary over time. But visualization of such spectrograms gives information about common

Image filter	Ratio			Time		
	SURF	SIFT	BRISK	SURF	SIFT	BRISK
LIGHTNESS	96.9	94.0	60.7	27	28	2
NOISE	99.1	98.7	95.7	75	50	5
SCALE	90.5	87.8	40.4	133	123	9
ROTATE	98.4	96.6	66.7	42	49	5
BLUR	97.0	96.8	18.5	39	38	2

Table 2. Video-stabilization marker detection accuracy and algorithm run time. The higher ratio value, the better. The lower time value, the better.

frequencies to some extent. Example spectrograms that show differences in real vehicle sound and generated ones are visible in Figure 4, so they can be compared. Author performed also a subjective study on the quality of the playback on a limited number of people, however this topic needs to be studied further.

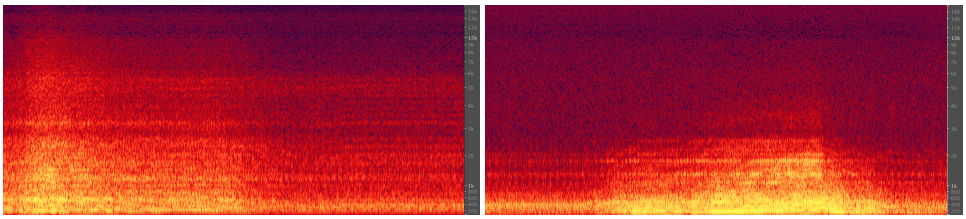


Fig. 4. Spectrogram of real car engine sound (left) and generated by the algorithm (right).

5. Conclusions

Modeling of engine sound is not very popular research topic. Most existing solutions were implemented commercially for the purpose of generating sounds for games and the idea behind algorithms is not publicly shared. Author created his own method to generate vehicle engine sounds that uses visual analysis techniques to prepare a database of multiple recorded sound samples and a mixer that can replay these sounds in proper order to mimic an engine sound in real-time. The solution can be used in a limited environment, such as for example on mobile smartphones. The algorithm generates audio that can successfully mimic sounds of vehicle engine and it can be used for the purpose of driving car simulation or computer games, especially in a limited environment.

References

[1] http://www.sonory.org/engine-sound-processor.html.

[2] http://www.olympusmicro.com.

[3] http://music.columbia.edu/pipermail/music-dsp/2006-April/065216.html.

[4] S.A. Amman and M. Das. An efficient technique for modeling and synthesis of automotive engine sounds. *Industrial Electronics, IEEE Transactions on*, 48(1):225–234, Feb 2001.

[5] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008.

[6] J. S. Beis and D. G. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. *In Conference on Computer Vision and Pattern Recognition, Puerto Rico*, pages 1000 – 1006, 1997.

[7] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[8] M. Fischler and C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[9] J. Jaglaa and J. Maillard. Sample-based engine noise synthesis using an enhanced pitch-synchronous overlap-and-add method. *J Acoust Soc Am.*, 132(5):3098–108, Nov 2012.

[10] Y. Karunakar, A. Kuwadekar, and K. Al Begain. A mobile based application for detecting fault by sound analysis in car engines using triangular window and wavelet transform. In *Computational Intelligence and Communication Networks (CICN), 2010 International Conference on*, pages 523–528, Nov 2010.

[11] S. Leutenegger, M. Chli, and R. Siegwart. BRISK: Binary Robust invariant scalable keypoints. *Computer Vision, IEEE International Conference on*, 0:2548–2555, 2011.

[12] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[13] E. Mullan. Driving sound synthesis from a physics engine. In *Games Innovations Conference, 2009. ICE-GIC 2009. International IEEE Consumer Electronics Society's*, pages 1–9, Aug 2009.

[14] Tao Ni, Dingxuan Zhao, and Hongyan Zhang. Realistic vehicle driving simulator with dynamic terrain deformation. In *Mechatronics and Automation, 2009. ICMA 2009. International Conference on*, pages 4795–4800, Aug 2009.

[15] Ting Wei and Hao Zheng. Sound effect of physical engine in game design. In *Information Technology, Computer Engineering and Management Sciences (ICM), 2011 International Conference on*, volume 3, pages 148–151, Sept 2011.

Marcin Skoczylas

GENERATOR DŹWIĘKU SILNIKA W CZASIE RZECZYWISTYM NA PODSTAWIE ANALIZY WIDEO I ZAREJESTROWANYCH PRÓBEK

Streszczenie: Generowanie dźwięku silnika jest szerokim tematem znanym od dziesięcioleci, głównie z powodu zastosowania takich algorytmów we wszelakiego rodzaju symulacjach jazdy samochodem, a w szczególności grach komputerowych. Algorytmy te stosują różne podejścia, m.in. od bardzo prostych odtwarzaczy zapętlonych dźwięków, które zmieniają częstotliwość nagranych uprzednio próbek do zaawansowanych algorytmów modelowania dźwięku silnika na podstawie określonych cech charakterystyki silnika. Algorytmy te są obliczeniowo skomplikowane i nie mogą być stosowane w urządzeniach przenośnych (takich jak np. smartfony) w czasie rzeczywistym. W tym artykule autor przedstawia wĹ‚asne podejście do korzystania z technik analizy wizualnej aby automatycznie przygotować bazę wielu nagranych krótkich próbek dźwiękowych oraz miksera, który odtwarza uporządkowane dźwięki w odpowiedniej kolejności, tak aby naśladować sterowalny dźwięk silnika w czasie rzeczywistym.

Słowa kluczowe: generator dźwięku silnika, nagrane próbki, mikser, analiza wizualna, Hough Transform

Artykuł zrealizowano w ramach pracy badawczej MB/WI/3/2012.

DIAGNOSING GUEST OPERATING SYSTEMS OF VIRTUAL MACHINES LEVERAGING AGENT ARCHITECTURE

Kamil Szczygieł[1], Krzysztof Bielawski[2]

[1] Intratel Sp. z o.o

[2] Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

Abstract: Maintaining large amount of virtual machines requires a lot of dedication and time from administrator. Using tools provided by virtualization vendors help in daily maintenance. Additionally it is often required to predict future problems. To address this need there are solutions which include analytic mechanisms to lower the risk of possible issues. In open source world there are many competitive tools, but none of them is integrated with virtualization solution. Maintaining large infrastructure using many administrative consoles is difficult and creates a potential for human mistakes. Mentioned software miss one key functionality - it is not possible to deeply monitor guest operating system of the virtual machine while maintaining integration with virtualization software. Solution proposed in this paper was created to solve this issue with agent based diagnostic mechanisms to provide information about network connectivity, resource usage, applications state, system settings and health.

Keywords: virtualization, diagnosis, KVM, virtio

1. Introduction

Running a modern data center is very difficult task even when major part of the infrastructure is virtualized. Maintaining large amount of virtual machines requires a lot of dedication and time from administrator. Diversity of hardware, systems, vendors and technologies is not helping in daily tasks. With introduction of software defined approach in virtualization administration became more centralized, but still requires a lot of time.

Managing virtual infrastructure can be easier by leveraging software provided by virtualization vendors. There are two leading commercial hypervisors - VMware

Advances in Computer Science Research, vol. 11, pp. 137-150, 2014.

ESXi [1] and Microsoft Hyper-V [2]. Maintaining virtual machines running under control of first is done through VMware vCenter Server [3], while managing virtual machines under control of latter is done through System Center Virtual Machine Manager [4]. They allow to perform basic control actions such as creating, editing, destroying, powering on and off and suspending virtual machines. To ensure access to the virtual machines they have high availability mechanisms that in case of failure move them to another physical server. When access to the application running inside virtual infrastructure is critical both technologies provide fault tolerance mechanisms. There is an copy of a virtual machine that is synchronized in real time and will take place of the original one in event of failure. Additionally there are functionalities that allow moving virtual machines between physical servers or storage without interruption.

Managing open source hypervisors such as KVM [5] or Xen [6] can be done through many different solutions [7], starting from less popular such as PetiteCloud [8], through oVirt [9] and OpenNebula [10], to rapidly developing OpenStack [11]. However, they are not as advanced as their commercial counterpart. They provide basic functionalities to manage and control virtual infrastructure, but they lack advanced mechanisms such as high availability, fault tolerance or moving virtual machines between storage. These functionalities are often available through additional software not integrated into one solution. For instance high availability can be delivered through Pacemaker [12] - open source resource manager. Because of fact that these mechanisms are not integrated into one platform they require additional configuration and it is not possible to manage them from one management console.

Mentioned software provides management mechanisms only for virtual machines. In modern datacenters it is often required to manage all of the components such as storage, network, physical, virtual servers and guest operating systems. Administrators are expected to predict future problems and bottlenecks. Virtualization vendors are aware of these requirements and released software such as VMware vCenter Operations Manager [13] or Microsoft System Center Operations Manager [14] providing advanced analytics of virtual and physical components. They are able to predict future problems for instance not enough storage, computing power or performance bottlenecks. Additionally these tools have ability to help administrator to evaluate health of the virtual infrastructure. There is variety of available metrics such as workload (showing how high virtual machine load is), anomalies (rating behaviour of virtual machine compared to the past) or efficiency (showing how efficient your virtual infrastructure is and how you can improve it). Using these tools it is easier to manage and improve virtual platform. While using additional tools that extend functionalities of these tools such as VMware vRealize Hyperic [15] or Management

Packs for System Center Operations Manager it is possible to monitor applications running inside of the guest operating system such as database servers, web servers, SAP [16], application servers, email servers and more.

On the contrary there is much open source software able to monitor all infrastructure components for instance Zenoss [17], Nagios [18] or Zabbix [19]. These tools are able to monitor [20] both physical and virtual infrastructure which allows to see complete infrastructure health from single management platform. However, these solutions are not aware of virtualized components such as virtual switches, virtualized storage or vendor specific features. Therefore they are not able to leverage mechanisms available through virtualization software. It is possible to extend functionalities of these tools by using additional software. However, while managing large virtual infrastructure administrator does not have time to switch between consoles and compare results from many tools. It is very inconvenient and create a potential for human mistakes while evaluating management feedback.

All mentioned software miss one key functionality - it is not possible to deeply monitor guest operating system of the virtual machines while maintaining integration with virtualization software. This shortcoming inspired us to create a solution that will solve this issue. The proof of concept was done for KVM hypervisor which is rapidly developing, but is still missing important tools. By leveraging private communication channel between hypervisor and virtual machine there is no network requirement to exchange messages. By using agent architecture it is possible to diagnose high resource usage of running processes to avoid performance bottlenecks of the virtual machine. Additionally to ensure highest service availability solution is able to diagnose network failures and notify administrator about it. Often there are critical applications running inside guest operating system and ensuring their availability is a priority. Software presented in this paper is able to diagnose and notify when application is not responding. Biggest source of diagnostic data are log files, therefore published solution is also able to track system settings and logs to diagnose occurring errors.

2. Principle of operation

Presented solution leverage agent architecture introduced in paper [21].

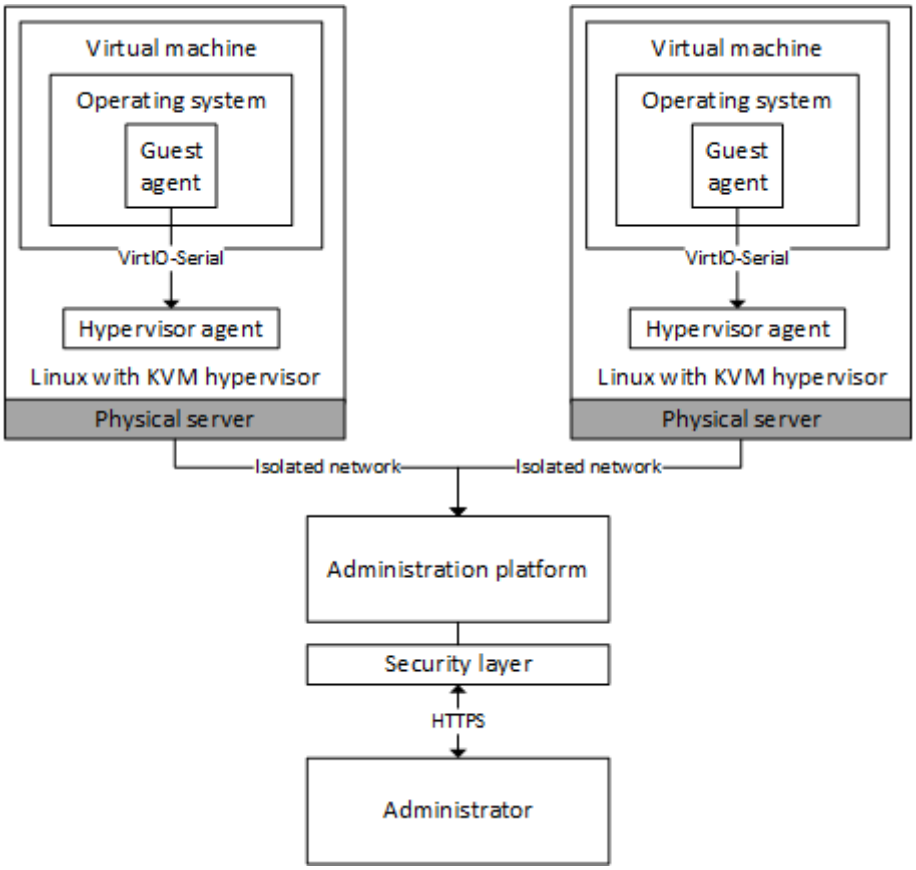


Fig. 1. Architecture of the proposed solution

Open source Linux operating system is installed on each physical server. Hypervisor of choice is KVM which is built into operating system kernel and because of that it does not require any additional modules to be installed. Administration is performed through administration platform - web application intended to maintain and run virtual infrastructure allowing for diagnosis and monitoring. It gathers all diagnostic information from virtual machines and presents them to the administrator. It

has built in resolution knowledge base using Windows event IDs [22] to help administrator to easier fix occurring problems. By having all the data in one place, administrator is able to analyse and troubleshoot problems in virtual infrastructure more accurate and faster. Communication is secured by using HTTPS protocol and software security layer. Hypervisor agent is an agent installed on every physical server with hypervisor role. It relays all the diagnostic data from virtual machines to the administration platform using dedicated, secure network. Guest operating system agent is a system service with elevated rights installed inside virtual machine operating system. It periodically performs the diagnostic tasks, gathers results and sends them to the hypervisor agent through hardware communication channel.

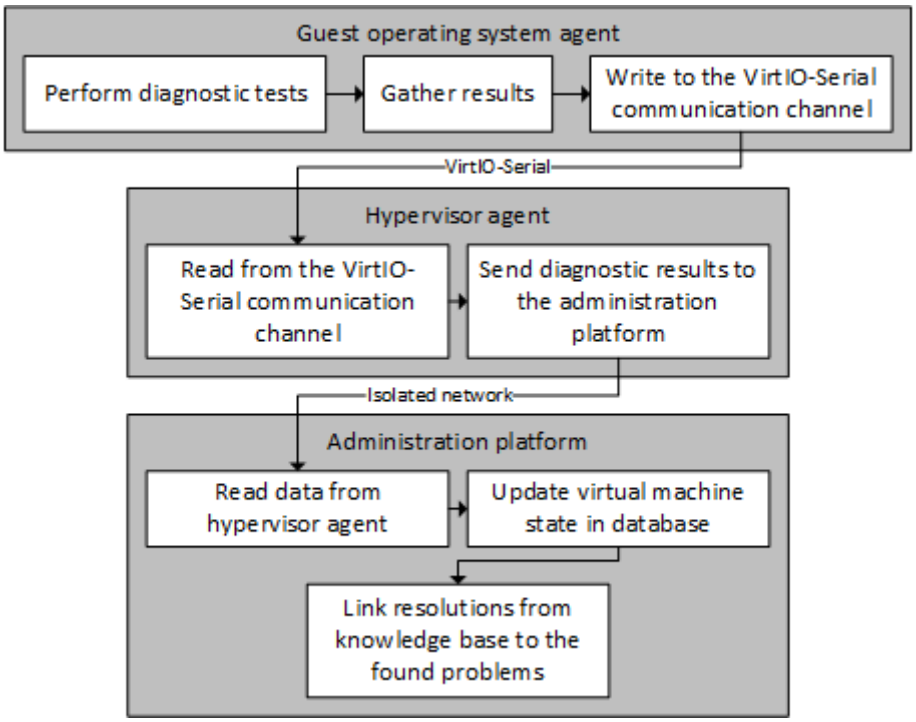


Fig. 2. Communication between components

Communication in proposed solution is one way only. The agent running inside guest operating system is performing diagnostic tasks such as network, applications, log files, system configuration and resource usage tests periodically. After all tasks are

finished, results are converted to the JavaScript Object Notation (JSON) [23] format and sent to the administration platform through VirtIO-Serial [24] communication channel. VirtIO-Serial is a transparent, hardware bridge between virtual machine operating system and hypervisor. Inside virtual machine it acts as a PCI device allowing write and read operations. Then in the physical server VirtIO-Serial can act as many types of output devices such as named pipes, text files, TCP servers, UNIX sockets and more. Because of hardware nature of communication channel, proposed solution is resistant to network problems. Hypervisor agent listens for incoming messages containing diagnostic results from virtual machines and relays them to the administration platform using dedicated, isolated network. Administration platform gathers all messages incoming from hypervisor agents and updates database containing virtual machine information. When diagnostic results contained problems, knowledge base is searched for resolutions allowing for administrator to view quick fix to the occurring problem.

3. Diagnostic cases

This section presents main diagnostic cases included in proposed solution in order of importance with brief description.

One of the major problems in virtual infrastructure maintenance software is that they require network connection to the guest operating system in order to diagnose processes running inside. Therefore we focused on solution that does not have this requirement and is able to perform network connectivity diagnosis and notify administrator in event of failure.

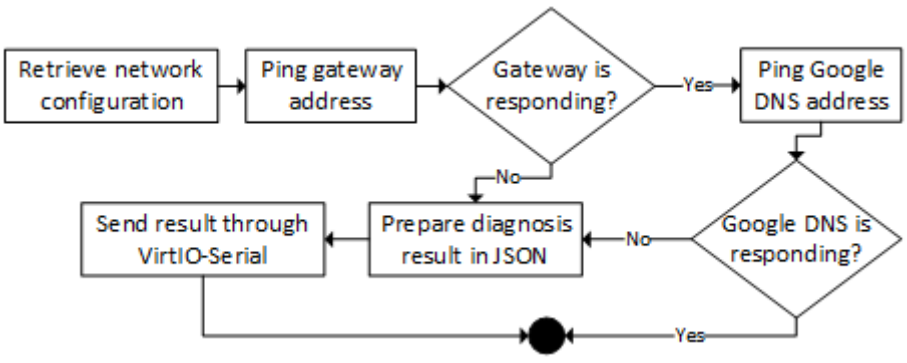


Fig. 3. Tasks performed during network diagnosis

At first agent is retrieving network configuration of network interface that has is active, is not loopback and has gateway address set. Then it perform simple ping tests to the gateway. If ping fails result of the test is sent through VirtIO-Serial communication channel to inform administrator about complete network failure. If not another ping test is performed to the Google DNS servers. If it fails result is sent to the hypervisor agent which will relay it to the administration platform indicating lack of Internet connection. Otherwise all tests passed successfully and virtual machine has properly functioning network connectivity.

Another significant aspect is that most of the virtual machines have important processes running inside guest operating system that must be maintained. Making sure that they are running properly is crucial. Therefore we developed a diagnostic test that notifies administrator when process is not responding.

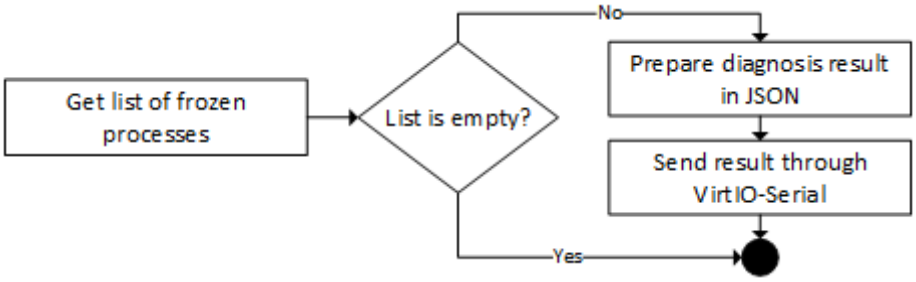


Fig. 4. Tasks performed during processes diagnosis

At first guest operating system agent retrieves list of all frozen processes. If the list is not empty result of the diagnostic test is sent through VirtIO-Serial communication channel. Otherwise, test passed successfully and all processes are running properly.

In most of the enterprises there are internal politics regarding security. We have noticed that there are repeating rules in most of the rules sets. One of them is firewall policy - it should be always enabled. Because of that we implemented diagnostic mechanism that is monitoring firewall and will notify administrator when settings have been changed.

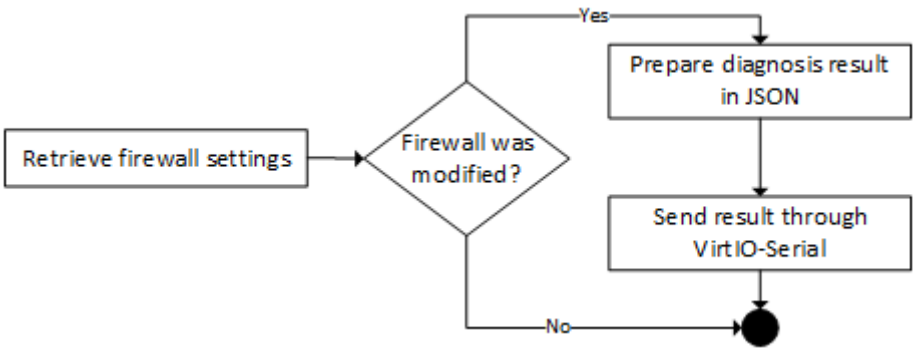


Fig. 5. Tasks performed during firewall diagnosis

Guest operating system agent periodically retrieves firewall settings and compares them to the previous result. If change of settings was detected result of this diagnosis is sent through VirtIO-Serial communication channel to inform administrator of potential security breach.

Log files are heart of every system. They contain a lot of diagnostic information therefore analysing them provides large amount of feedback. In our solution we focused on error entries that administrator should be notified of. They are sent through communication channel to the administration platform. Administrator can look up resolution in built in knowledge base and decide to fix it.

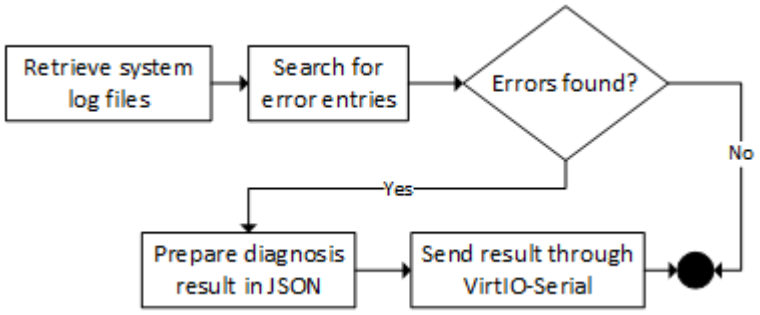


Fig. 6. Tasks performed during logs diagnosis

At first guest operating system agent retrieves all system log files and searches for error entries. If they are found result of this diagnosis is sent through VirtIO-

Serial communication channel to inform administrator about failure. If resolution to the error has been found in knowledge base it will be linked to the particular virtual machine allowing administrator to look up resolution to the occurring errors.

Performance is very important factor in virtualization. Being aware when virtual machine is using a lot of resources may prevent application performance issues. With this knowledge administrator can either diagnose why virtual machine is under such high load or assign more resources to the virtual machine. Therefore proposed solution has diagnostic test that notifies administrator when virtual machine is using a lot of resources.

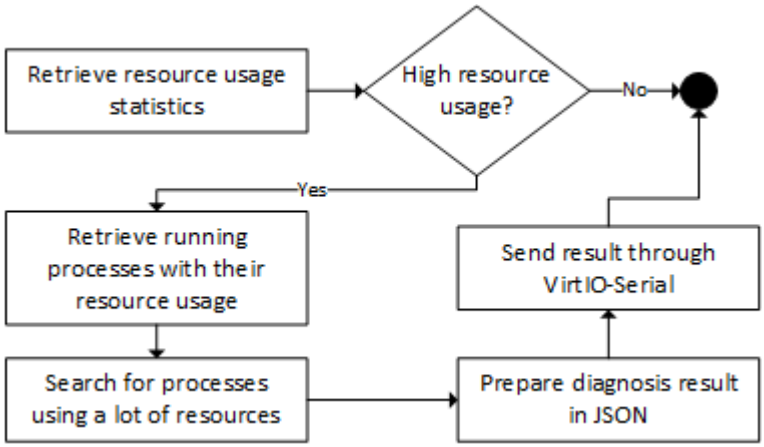


Fig. 7. Tasks performed during resources diagnosis

At first agent retrieves resource usage statistics to analyze usage. When very high usage is detected it will retrieve list of all running processes with their resource usage, filter it and prepare diagnosis result containing all resource intensive applications. Such list is sent through VirtIO-Serial to the administration platform.

Important concern for the administrator are applications running inside of operating system. Installation of software may leave system vulnerable and therefore compromised. Diagnostic test proposed in our solution compares installed software and notifies administrator through administration platform when application has been installed or removed. It allows for administrator to have more detailed view on the software that is being installed or removed inside of operating system.

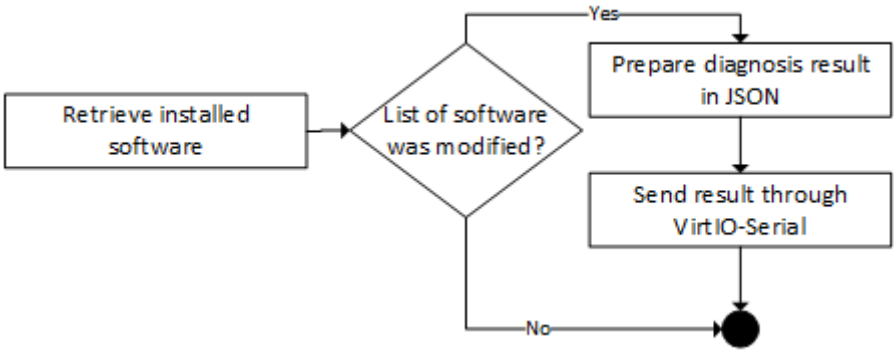


Fig. 8. Tasks performed during applications diagnosis

Guest operating system agent periodically retrieves list of installed software and compares it to the previous result. If any application was installed or removed result of this diagnosis is sent through VirtIO-Serial communication channel to inform administrator.

4. Case study

Presented solution was leveraged to create a self-service portal where customers are able to order a customized virtual machines. They are able to choose an operating system of their preference from predefined list containing Windows and Linux. Then it is possible to adjust virtual machine hardware such as amount of virtual CPUs, RAM and additional disks. To provide even more personal solution they are able to modify system hostname and administrator password. Additionally customer is able to order a virtual machine with preinstalled and preconfigured software such as databases, identity, web and mail servers. Parameters required by additional components are also customized during virtual machine creation process. User is able to control complete lifecycle of the instance with power on, off, suspend and delete actions. Additionally it is possible to preserve the current state of the virtual machine and return to it later with included snapshots mechanism. Self-service portal delivers variety of information about instance such as resource usage, health and network state.

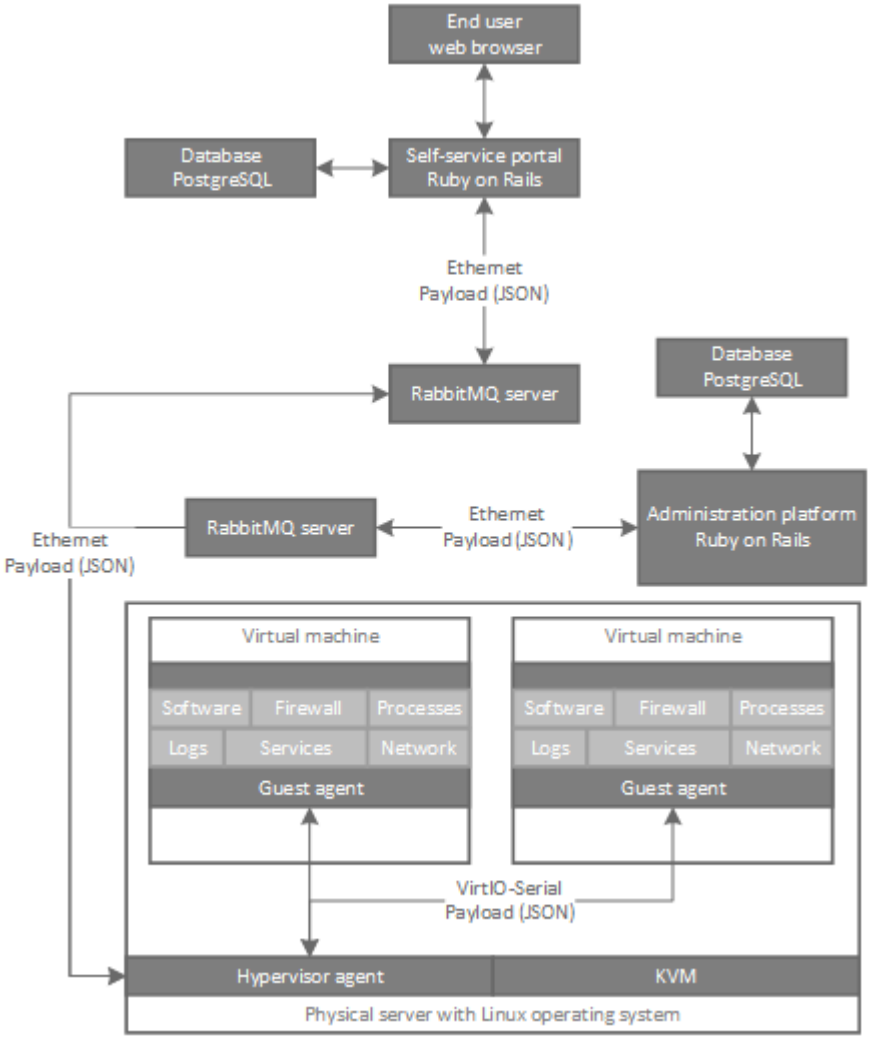


Fig. 9. Architecture of presented case study

Communication between components in this case study is depicted in figure 9. Self-service portal and administration platform are built using Ruby on Rails [25], open source web framework written in Ruby [26]. To meet requirements of modern, responsive web design technologies such as jQuery [27] and asynchronous calls are leveraged. Self-service portal is using PostgreSQL [28] open source database engine to store user and virtual machine data. To integrate with solution proposed in

this paper RabbitMQ [29] was introduced. RabbitMQ is open source message broker to exchange messages between components in asynchronous manner. It implements Advanced Message Queueing Protocol (AMQP) [30] to control payload flow. In presented case there are two RabbitMQ servers – one to handle communication between self-service portal and hypervisor agent, second to handle communication between administration platform and hypervisor agent. By separating message brokers components are completely independent and can work separately. Failure of self-service RabbitMQ server does not influence availability of administrative capabilities through administration platform. Payload in messages exchanged between components is in JSON format. Presented use case leverage solution depicted in this paper to install and preconfigure software delivered to the virtual machines. When a virtual machine with database server is requested, new virtual machine is cloned from predefined template image and configured with user hostname and password. Then self-service portal requests an installation of database server software inside of the virtual machine and preconfigure configuration files. In meantime, user can observe whole process in self-service portal, because guest agent is notifying portal about currently performed operation. After whole process is completed, user is notified that a virtual machine is ready to use. Minimized results of diagnostic cases presented in this paper are visible to the user in virtual machine information window after deployment process is finished.

5. Conclusion

In this paper a complete solution is depicted that includes functionalities providing diagnostic tests of the most frequently occurring problems inside virtual machines. Agent running inside of guest operating systems performs network connection diagnosis to ensure network connectivity. Additionally it diagnoses system settings and installed applications to make sure their configuration meets internal security policies. The agent periodically diagnoses state of running processes to ensure that critical applications are running properly. To avoid performance bottlenecks, the solution is diagnosing resource usage inside a virtual machine. Additionally the agent diagnoses system log files and events to provide most detailed diagnostic information. By gathering results provided by these tests and depicting them in administration platform, administrator is able to evaluate virtual infrastructure health and react to problems as they emerge, before they influence performance or availability of the services. Because of hardware private communication channel network failure does not influence ability to inform administrator about problems. Presented mechanisms are able to integrate into one management platform without additional configuration

and provide management mechanisms for virtual machines, storage, network, physical servers and guest operating systems including services and applications.

References

[1] VMware vSphere: [http://www.vmware.com/products/vsphere]

[2] Microsoft Hyper-V: [http://www.microsoft.com/en-us/server-cloud/solutions/virtualization.aspx]

[3] VMware vCenter Server: [http://www.vmware.com/products/vcenter-server]

[4] System Center Virtual Machine Manager: [http://technet.microsoft.com/en-us/library/gg610610.aspx]

[5] KVM: [http://www.linux-kvm.org/]

[6] Xen: [http://www.xenproject.org/]

[7] Sotomayor, B., Montero, Ruben S.m Llorente, I.M., Foster, I.: Virtual Infrastructure Management in Private and Hybrid Clouds, Internet Computing, IEEE, 2009.

[8] PetiteCloud: [http://petitecloud.org/]

[9] oVirt: [http://www.ovirt.org/]

[10] OpenNebula: [http://opennebula.org/]

[11] OpenStack: [http://www.openstack.org/]

[12] Pacemaker: [http://clusterlabs.org/]

[13] VMware vCenter Operations Manager: [http://www.vmware.com/products/vrealize-operations/]

[14] System Center Operations Manager: [http://technet.microsoft.com/en-us/library/hh205987.aspx]

[15] VMware vRealize Hyperic: [http://www.vmware.com/products/vrealize-hyperic/]

[16] SAP: [http://www.sap.com]

[17] Zenoss: [http://www.zenoss.com/]

[18] Nagios: [http://www.nagios.org/]

[19] Zabbix: [http://www.zabbix.com/]

[20] Marik, O., Zitta, S.: Comparative analysis of monitoring system for data networks, Multimedia Computing and Systems (ICMCS), Marrakech, 2014.

[21] Szczygieł, K., Bielawski, K.: Controlling KVM virtual machine guest with VirtIO, Design, development and implementation of real-time systems, Warsaw, 2013.

[22] Windows Events: [http://technet.microsoft.com/en-us/library/dd299434(v=ws.10).aspx]

[23] JavaScript Object Notation: [http://json.org/]
[24] VirtIO-Serial: [http://fedoraproject.org/wiki/Features/VirtioSerial]
[25] Ruby on Rails: [http://www.rubyonrails.org]
[26] Ruby: [http://www.ruby-lang.org]
[27] jQuery: [http://www.jquery.com]
[28] PostgreSQL: [http://www.postgresql.org]
[29] RabbitMQ: [http://www.rabbitmq.com]
[30] Advanced Message Queueing Protocol: [http://www.amqp.org]

DIAGNOZOWANIE SYSTEMÓW OPERACYJNYCH MASZYN WIRTUALNYCH PRZY WYKORZYSTANIU ARCHITEKTURY AGENTOWEJ

Streszczenie: Zarządzanie dużą ilością maszyn wirtualnych wymaga od administratora wiele czasu oraz poświęcenia. Wykorzystanie narzędzi dostarczonych wraz z oprogramowaniem wirtualizacyjnym ułatwia utrzymanie infrastruktury. Dodatkowo często wymagane jest przewidywanie problemów, które mogą wystąpić w środowisku wirtualnym. W tym celu powstało oprogramowanie zawierające mechanizmy analityczne zmniejszające ryzyko awarii. W świecie oprogramowania open source istnieje wiele narzędzi, lecz żadne z nich nie jest zintegrowane z platformą wirtualizacyjną, a w związku z tym zarządzanie infrastrukturą jest trudne. Przedstawionym rozwiązaniom brak jest jednej istotnej funkcjonalności - możliwości dokładnego monitorowania systemów operacyjnych. Zaproponowane w publikacji oprogramowanie w oparciu o architekturę agentową stara się rozwiązać ten problem poprzez wykorzystanie mechanizmów dostarczających informacji o stanie sieci, zużyciu zasobów, stanie aplikacji, ustawieniach systemu oraz jego zdrowiu.

Słowa kluczowe: wirtualizacja, diagnostyka, KVM, virtio

Artykuł zrealizowano w ramach pracy badawczej S/WI/2/13.

PROBABILISTIC GRAPHICAL MODEL SUPPORTING EARLY DIAGNOSIS OF AUTISM SPECTRUM DISORDER

Justyna Szczygieł[1], Agnieszka Oniśko[1], Jolanta Świderska[2], Elżbieta Krysiewicz[2], Jerzy Sienkiewicz[3]

[1] Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

[2] Fundacja Oswoić Świat, Białystok

[3] Towarzystwo Przyjaciół Dzieci, Oddział Białystok

Abstract: Bayesian networks are recognized as a suitable tool for modelling diagnostic problems. The power of this modelling is that it can combine knowledge coming from different sources. For example, in case of medical domain, the expert knowledge can be merged along with the medical data. This paper presents a Bayesian network model for early diagnosis of autism. The model was built based on the medical literature and then was revised by two domain experts. Our tool is dedicated to parents that can perform an early diagnosis of their child before visiting a specialist.

Keywords: Bayesian networks, medical diagnosis, autism spectrum disorder

1. Introduction

The support of medical diagnosis by computer-based tools has a long history with the first approaches proposed in the 1960s and 1970s (e.g., [6,9]). The medical diagnostic support systems built in the last few decades were based on various approaches that can be divided into two categories: (1) statistical modeling and (2) artificial intelligence modeling that includes fuzzy sets, neural networks, decision trees, or probabilistic graphical models. Probabilistic graphical models such as Bayesian networks have proven to be powerful tools for modeling complex diagnostic problems involving uncertain knowledge. They have been employed in solving a variety of medical diagnostic problems reaching the size of hundreds or thousands of variables (e.g., [1,2,3,4,10]).

Advances in Computer Science Research, vol. 11, pp. 151-164, 2014.

Autism spectrum disorder (ASD) is a neurodevelopmental disorder with genetic origins that leads to an impaired social interaction. In the last decades, a dramatic increase of the ASD prevalence has been observed [5]. For example, the U.S. 2010 statistics show that the prevalence of ASD in children aged 8 is 1 in 68 [8]. ASD is not easy to diagnose especially in children before the age of 24 months. We know that the disease is more prevalent in males than in females with the ratio 4.5 : 1 [8]. However, there is no definitive cause of ASD identified, i.e., usually there is a combination of different risk factors and symptoms that have to be present to establish a final diagnosis. Early diagnosis is important since different types of therapy can improve child's development. For example, the therapeutic interventions can help the child to talk, walk, and communicate with others and then can increase child's chances for living independently in a society when they are adult.

There is no standard procedure or screening examination for early ASD diagnosis in Poland. Parents are often not aware of this disease and may overlook its first symptoms. We have proposed a tool — a web-based application to support early diagnosis of ASD. This tool is dedicated to parents that observe an odd behaviour in their child. The core of our application is the AutismNET model, a Bayesian network that was built based on the medical literature and then revised by two domain experts. The model allows for estimating the probability of developing ASD based on the observed signs and symptoms entered into the model. This probability can be further interpreted by parents suspecting ASD in their child. The AutismNET model was developed and presented in [13].[4]

The reminder of this paper is structured as follows. Section 1. provides a brief introduction to the problem of the ASD diagnosis. Section 2. presents an overview of Bayesian networks. Section 3. describes the AutismNET model and its application. Section 4. concludes the paper.

2. Bayesian networks

Bayesian networks [14] are acyclic directed graphs modeling probabilistic influences among variables. The graphical part of a Bayesian network reflects the structure of a modeled problem, while conditional probability distributions quantify local interactions among neighboring variables.

Figure 1 captures a simple Bayesian network model. This example model includes one risk factor and two symptoms of autism. A left hand side of the figure

[4] Justyna Pawłowska is a maiden name of Justyna Szczygieł.

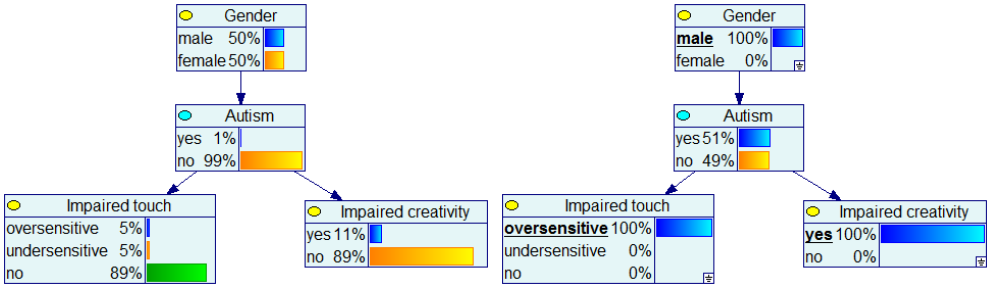


Fig. 1. A simple example of a Bayesian network

shows the model along with marginal probabilities for each node while a right hand side of the figure shows the same model but with observed three nodes and a posteriori probability distribution for the node *Autism*. Each arc of this graph represents a probabilistic relationship. For example, the arc between the variables *Gender* and *Autism* indicates that autism in males is more prevalent, i.e., males are around four times more probable to be diagnosed with autism than females. Furthermore, this simple example captures two possible symptoms of autism: (1) impaired touch and (2) impaired creativity. The numerical parameters of a Bayesian network model include a conditional probability distribution for the nodes that have parents (e.g., *Autism*, *Impaired touch*, and *Impaired creativity*) and a prior distribution for the nodes without parents (e.g., *Gender*). These probability distributions can be learned from the data or can be assessed by the domain experts.

After creating a Bayesian network model, we can perform a reasoning that involves calculating a posteriori probability distribution for the node *Autism* given the observations that were entered into the model. This calculation consists of repetitive application of a Bayes theorem that spreads over the network and leads to a derivation of conditional posterior probabilities in every node of the network. A right hand side of Figure 1 shows the result of such probabilistic reasoning and answers the question: *What is a probability of developing autism for a boy that has impaired creativity and that has oversensitive touch?* The probability of developing autism in this example model is equal to 51%.

3. The AutismNET model

The following section describes the process of building the AutismNET model for early diagnosis of ASD. The first part of the section shows a graphical structure of the

model while the second presents a quantitative part that includes conditional probability distributions. While referring to the nodes of a Bayesian network model we will use three terms: *target*, *observation*, and *auxiliary* indicating three different types of nodes. The category *target* stands for the nodes representing diagnoses, *observation* represents all these nodes that we would usually observe. For example, the nodes *Impaired touch*, *Impaired creativity*, and *Gender* in Figure 1 have a status *observation*. A type *auxiliary* indicates the nodes that we would never observe.

3.1 Graphical structure of the model

We have started building the model from browsing and studying the medical literature related to the ASD diagnosis.

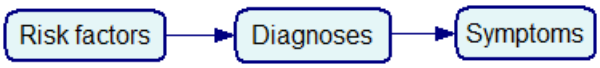


Fig. 2. The three layers of AutismNET

First version of the model The first version of the model was built based on knowledge encountered from the medical literature. A knowledge engineer identified 85 variables that were modeled in the framework of a Bayesian network model. The variables have belonged to three categories: (1) risk factors, (2) diagnoses, and (3) signs and symptoms. These three categories were mapped into three layers of the AutismNET model (see Figure 2). Additionally, we decided to group the variables within these three layers into submodels. This procedure helped us to organize the models' variables and to facilitate the process of navigation. The concept of a submodel is implemented in GeNIe [11] and it is simply a logical concept that does not introduce any additional relationships in the model.

Figure 3 presents the first version of the AutismNET model. The model consists of 85 nodes grouped in 14 submodels. A top layer of the model includes 9 submodels representing 43 risk factors while a bottom layer consists of 5 submodels representing 31 different symptoms. A middle layer represents two diseases: *Autism spectrum disorder* and *ADHD* (Attention Deficit Hyperactivity Disorder). We included *ADHD* as a part of differential diagnosis for *Autism spectrum disorder*.

One of the problems that we encountered during building the model was the number of parents per node. For example, the node *Autism spectrum disorder* had

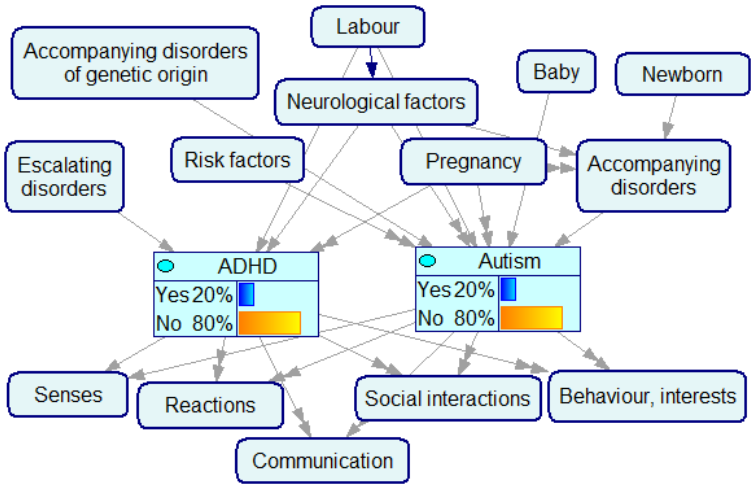


Fig. 3. The first version of the AutismNET model

initially 20 parents. Since it would be almost impossible to estimate the numerical parameters for the node with 20 parents,[5] we have applied a technique called in Bayesian networks as "parent divorcing" [12]. This technique helps to decrease a complexity of the network by reducing the number of parents per node. We have created 6 auxiliary nodes that divorced the parents of the node *Autism spectrum disorder*. The result of this procedure was a decrease of the number of parents for the node *Autism spectrum disorder* from 20 to 10. All the auxiliary nodes that we have created were modeled as the NoisyMAX gates. The advantage of applying the NoisyMAX gates is that we can estimate conditional probability distribution of a node with a smaller number of numerical parameters [7].

Figure 4 shows an example of "parent divorcing" that we have performed in the AutismNET model. The left hand side of the figure captures the four out of 20 parents of *Autism spectrum disorder*, while the right hand side of the figure shows the result of divorcing these parents. The auxiliary node *Labour complications* that divorced the parents was further modeled as a Noisy MAX gate.

[5] Assuming that all nodes are binary, the node with 20 parents needs 2^{20} independent probabilities to elicit.

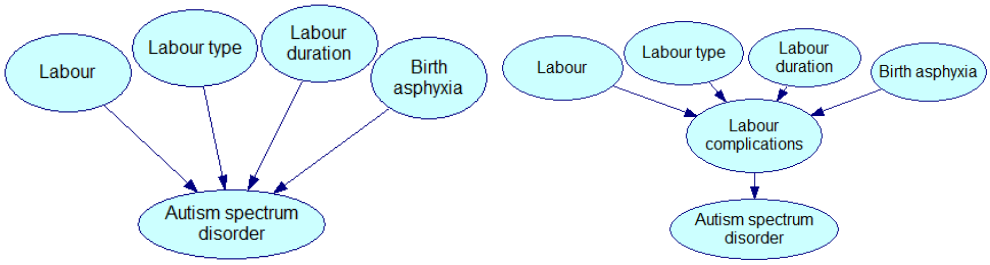


Fig. 4. An example of parent divorcing in AutismNET

Second version of the model After building the first version of the AutismNET model, we have scheduled five meetings with the experts[6] to verify it. Each meeting lasted around two hours. The first two meetings were devoted to verification of the model variables, while during the next two meetings we elicited the numerical parameters from the experts. The last meeting was devoted to model evaluation.

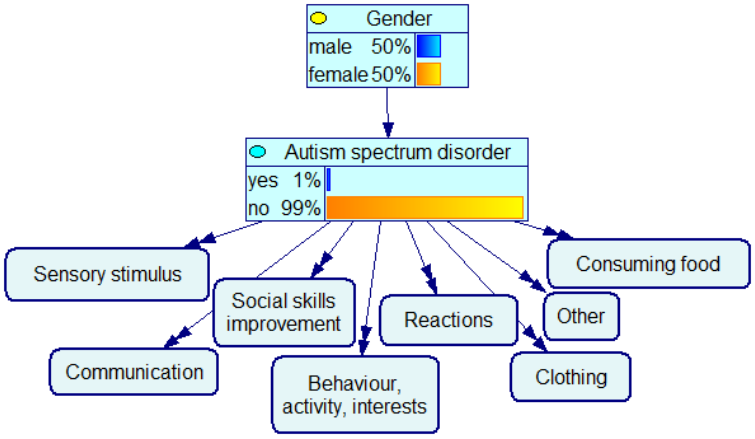


Fig. 5. The second version of the AutismNET model

For the first meeting we have prepared a list of the model's variables printed for each of the two experts — the variables were grouped by submodels. During this session with experts we went through this list and performed a clarity test for each

[6] The third and fourth author of this paper.

variable. The experts have excluded all the variables representing the risk factors of ASD, but the gender i.e., 43 nodes were removed from the model along with 9 auxiliary nodes. The experts claimed that risk factors that we had included in the model do not have much diagnostic value and that the ASD diagnosis should be performed mainly based on signs and symptoms. At the same time, the experts proposed to include in the model new variables, for example: *No response to name*, *No response to reading books by parents*, *Gestural communication*, *Impaired speech*. We included in the model 22 additional variables that along with previous variables were grouped into 8 submodels. The resulting model had three new submodels: *Clothing*, *Consuming food*, and *Other*. Three out of five submodels in *Version 2.0* changed their name, for example, the submodel *Senses* was changed to *Sensory stimulus*. The experts excluded also the variable *ADHD* claiming that this disease is not crucial in differential diagnosis of *Autism spectrum disorder*. During the second meeting with the experts, we again verified the model's variables – although this time it involved verification of variables states. For example, the variable *Impaired sleep* was initially modeled as a binary node with two states: *Short sleep* and *Normal sleep*. The experts claimed that it should be modeled as the variable with three states: (1) *Short sleep*, (2) *Interrupting sleep*, and (3) *Normal sleep*.

Figure 5 presents the second version of the model after two meetings with the experts. In fact, after removing from the model all the variables representing risk factors, the second version of AutismNET became a naive Bayesian network. The model consists of 50 nodes: one target node and 49 observation nodes.

Third version of the model Although our experts believed initially that only signs and symptoms play a significant role in a diagnosis of ASD, we agreed after a short discussion with them that it would be interesting to include the risk factors in the AutismNET model. Therefore, we have created the third version of the model that includes again three layers of the variables, i.e., risk factors, diagnoses, and signs and symptoms. Figure 6 presents the third version of the AutismNET model. The model consists of 100 nodes grouped in 16 submodels. Similarly to Figure 3, a top layer of the model includes the nodes representing risk factors while a bottom layer captures the signs and symptoms. The third version of the AutismNET model is essentially a hybrid of two previous versions: with a top layer of risk factors from *Version 1.0* and two bottom layers from *Version 2.0*.

Tables 1 and 2 present the properties for the three versions of the AutismNET model. Table 1 contains the information about the nodes and submodels of the three versions of AutismNET. For example, *Version 1.0* of the model has in total 76 nodes

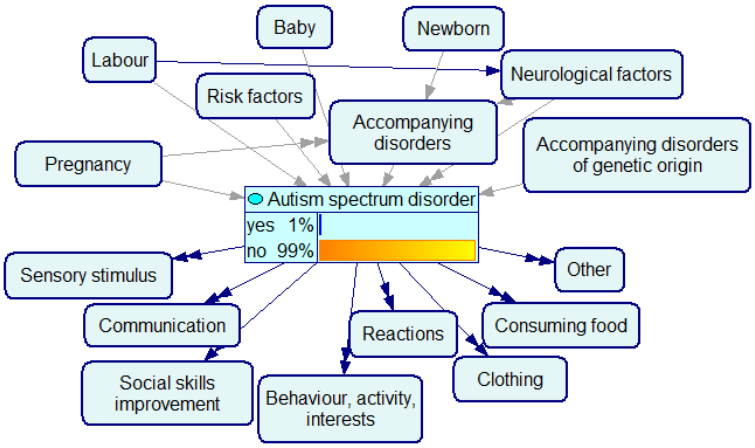


Fig. 6. The third version of the AutismNET model

modeled as CPTs[7] and 9 nodes modeled as NoisyMAX distributions. There were 2 target nodes, 74 observation nodes, and 9 auxiliary nodes; this version of the model was grouped into 14 different submodels. Table 2 captures additional structural statistics of AutismNET and shows a complexity of the models. For example, *Version 1.0* of the model has 92 arcs, on average 1.08 parents per node (*Avg indegree*), and a maximal number of parents equal to 10 (*Max indegree*). The model has on average 2.12 outcomes per node and a maximal number of outcomes per node is equal to 4.

Table 1. Characteristics of the AutismNET models; #nodes indicates the number of all nodes, #CPT stands for the number of nodes with the CPT distributions, #NoisyMAX stands for the number of nodes with the NoisyMAX distributions, etc.

Version	#nodes	#CPT	#NoisyMAX	#target	#observation	#auxiliary	#submodels
Version 1.0	85	76	9	2	74	9	14
Version 2.0	50	50	0	1	49	0	8
Version 3.0	100	90	10	1	89	10	16

[7] CPTs stands for Conditional Probability Tables

Table 2. Structural statistics of the AutismNET models; #arcs stands for the number of arcs

Version	#arcs	Avg indegree	Max indegree	Avg outcomes	Max outcomes
Version 1.0	92	1.08	10	2.12	3
Version 2.0	49	0.98	1	2.28	4
Version 3.0	102	1.02	10	2.19	4

3.2 Elicitation of numerical parameters

Quantitative part of Bayesian network model includes conditional probability distributions. These probability distributions can be learned from the data or assessed by a domain expert. There were no data available to us, therefore, we had to rely on the expert opinion while quantifying the model. In the first version of the model we simply assigned the distribution $(0.2, 0.8)$ for binary nodes, or a uniform distribution for non-binary nodes. The quantification of the model was conducted for the second version of the model. Two meetings with the experts were devoted to elicitation of the numerical parameters. During the first meeting we were posing the following type of questions: *What is the probability that a symptom is present if a child has ASD?* For example, we asked the following question:

What is the probability that oversensitive touch is present if a child has ASD?

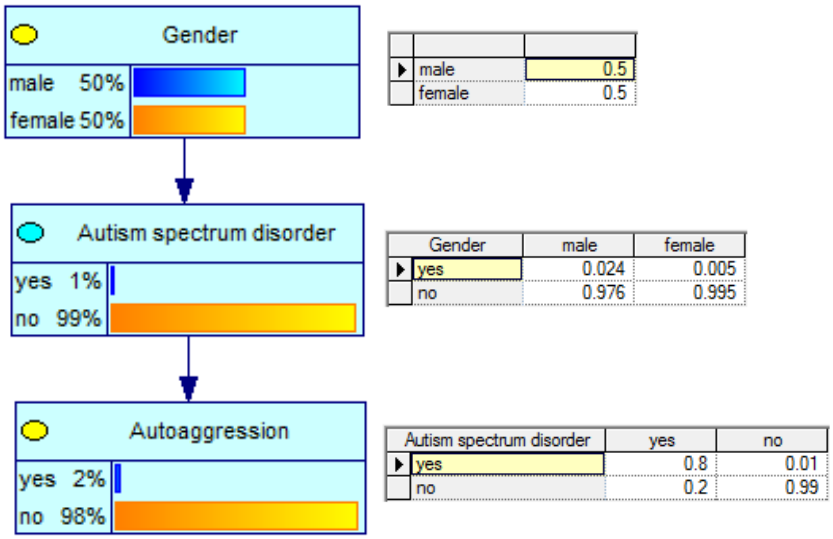
During the second meeting we were posing the following type of questions: *What is the probability that a symptom is present if a child does not have ASD?* For example, we asked the following question:

What is the probability that oversensitive touch is present if a child does not have ASD?

During these two meetings, that lasted four hours together, we elicited 122 independent numerical parameters. We have noticed that it was easier for the experts to assess the parameters for the first scenario, e.g., when we were asking for the probability of a symptom being present if a child had ASD.

During elicitation of the numerical parameters we have identified the variable that was not significant in diagnosis of ASD. While assessing the probabilities for the variable *Mood swinging* we have noticed that the experts specified the same probability distribution for autistic and non autistic population. This led us to removing this variable from the model.

Figure 7 presents a fragment of AutismNET along with its probability distributions. The node *Gender* with two states *male* and *female* is described by a prior



male	0.5
female	0.5

Gender	male	female
yes	0.024	0.005
no	0.976	0.995

Autism spectrum disorder	yes	no
yes	0.8	0.01
no	0.2	0.99

Fig. 7. A fragment of the AutismNET model along with conditional probability tables

probability distribution and it represents a general population distribution. The node *Autism spectrum disorder* is described by a conditional probability distribution and it reflects the relationship between gender and ASD. These numerical parameters were specified based on the published statistics of the ASD prevalence in a general population depending on a gender. The last node *Autoaggression*, again, is described by a conditional probability distribution that was elicited from our experts. In fact, to quantify this distribution, the experts had to specify only two independent numerical probabilities: 0.8 and 0.01.

Table 3 shows a summary of quantitative part of the AutismNET models along with the number of probabilities for target, observation, and auxiliary nodes. The table shows also the number of dependent probabilities that are part of the CPT or NoisyMAX distributions. For example, *Version 1.0* of the model has 2,551 dependent probabilities modeled by CPT and 96 dependent probabilities modeled by NoisyMAX.

Since the third version of the model is a hybrid of the two previous versions, only the variables from a bottom layer has the probabilities elicited by the experts.

Table 3. Numerical parameters of the AutismNET models

Version	CPT	NoisyMAX	Target	Observation	Auxiliary
Version 1.0	2,551	96	2,112	417	118
Version 2.0	228	0	2	226	0
Version 3.0	1,527	100	1,024	503	104

3.3 Model evaluation

We did not have any access to objective data to evaluate the AutismNET model, therefore, we have performed only a subjective expert evaluation. We were entering the data representing a typical autistic child and then the experts were observing how the a posteriori probability of developing ASD is changing. During this evaluation we also looked at the variables with the highest diagnostic value (calculated in GeNIe based on cross-entropy measure) and asked the experts whether indeed these variables are important in a diagnostic process. The experts had confirmed that the indicated variables have a high diagnostic value.

While playing with AutismNET, we have noticed that the model is too sensitive with respect to observed symptoms, i.e., after observing a few symptoms as present, the probability of ASD was approaching the value of 1.0. For example, after we had observed the following symptoms in the model: an oversensitive touch, unusual preoccupation with toys, and repeating unusual movements or actions, the calculated model probability of developing ASD was 99.9%. This value suggested that the model's probabilities need additional revision and refinement.

3.4 Application of AutismNET

We have built a web-based interface for the AutismNET model. This interface allows to access the model through Internet and to perform a diagnosis by answering the questions of a survey. These questions correspond to the variables modeled in AutismNET and they are grouped by submodels. This interface is dedicated to parents that would like to perform an initial diagnosis of their child. The model could possibly indicate a need for a more detailed diagnosis by a specialist. Figure 8 presents a screen shot of the application. A list of 8 elements on a left hand side of the window corresponds to 8 submodels from *Version 2.0* of AutismNET.[8] A right hand side

[8] Currently, only *Version 2.0* is fully quantified. Therefore, it is used in a web-based interface of AutismNET.

of the window shows a list of questions of the survey that correspond to the variables of the submodel *Behaviour, activity, interests*. The application is available at `http://www.autismnet.pl`.

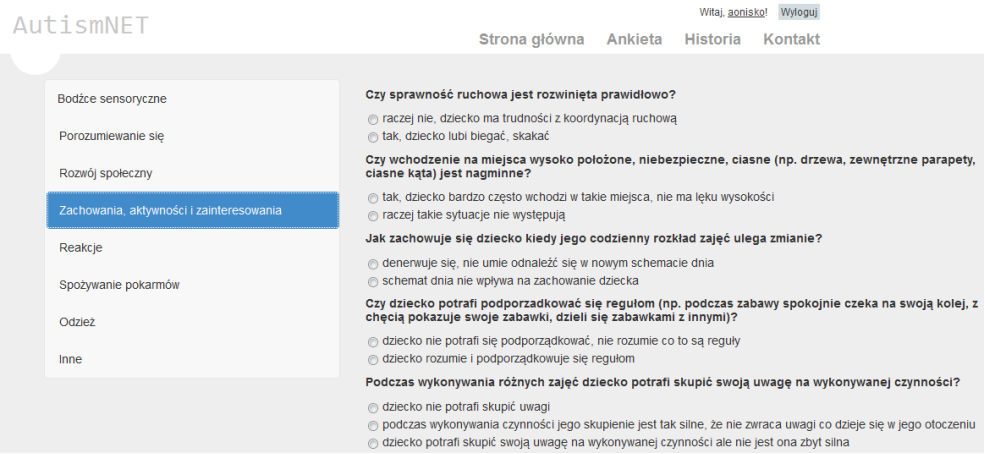


Fig. 8. Screen shot of AutismNET web-based application (in Polish)

4. Conclusions

We have built the AutismNET model for early diagnosis of ASD. The model was entirely built based on the medical literature and experts' knowledge. The model calculates the posteriori probability of developing ASD given entered observations. AutismNET has also a web-based interface that facilitates the interaction with the model. This tool is dedicated to parents that observe an odd behaviour in their child and suspect ASD.

Our project has several shortcomings that we plan to address in a future version of the model. The model requires a revision of the numerical parameters since they are too sensitive towards observed symptoms. A sensitivity analysis has to be performed to identify these numerical parameters of the model that should be further tuned. We also need to refine the variables representing risk factors modeled in AutismNET and then elicit the numerical parameters for the corresponding nodes.

It is an interesting research question whether a simple diagnostic model would perform better than a complete extended model. We plan to answer this question by

comparing the two models: the AutismNET model in *Version 2.0* consisting of 50 nodes and the AutismNET model in *Version 3.0* including 100 nodes.

We also plan to implement AutismNET user interface for a mobile device – this will even increase the availability of the model to its potential users.

Acknowledgments

The model was created and tested using SMILE, an inference engine, and GeNIe, a development environment for reasoning in graphical probabilistic models, both developed at the Decision Systems Laboratory, University of Pittsburgh and available at `https://dslpitt.org/genie/`.

References

[1] S. Andreassen, M. Woldbye, B. Falck, and S. K. Andersen. MUNIN – A causal probabilistic network for interpretation of electromyographic findings. In J. McDermott, editor, *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 366–372, Los Altos, CA, 1987. Morgan Kaufmann Publishers.

[2] I.A. Beinlich, H.J. Suermondt, R.M. Chavez, and G.F. Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proceedings of the Second European Conference on Artificial Intelligence in Medical Care*, pages 247–256, London, 1989.

[3] Gregory F. Cooper. *NESTOR: A Computer-based Medical Diagnostic Aid that Integrates Causal and Probabilistic Knowledge*. PhD thesis, Stanford University, Computer Science Department, 1984.

[4] F. J. Díez, J. Mira, E. Iturralde, and S. Zubillaga. DIAVAL, a Bayesian expert system for echocardiography. *Artificial Intelligence in Medicine*, 10:59–73, 1997.

[5] E. Fombonne. Epidemiological trends in rates of autism. *Molecular Psychiatry*, 7:S4–S6, 2002.

[6] G. Anthony Gorry. Computer-assisted clinical decision-making. *Methods of Information in Medicine*, 12:45–51, 1973.

[7] Max Henrion. Some practical issues in constructing belief networks. In L.N. Kanal, T.S. Levitt, and J.F. Lemmer, editors, *Uncertainty in Artificial Intelligence 3*, pages 161–173. Elsevier Science Publishers B.V., North Holland, 1989.

[8] Developmental Disabilities Monitoring Network Surveillance Year 2010 Principal Investigators, Centers for Disease Control, and Prevention (CDC). Prevalence of autism spectrum disorder among children aged 8 years - autism and developmental disabilities monitoring network, 11 sites, United States, 2010. *MMWR Surveill Summ*, 63(2):1–21, 2014.

[9] R. S. Ledley and L. B. Lusted. Reasoning foundations of medical diagnosis. *Science*, 130(3366):9–21, July 1959.

[10] Peter J. F. Lucas, Linda van der Gaag, and Ameen Abu-Hanna. Bayesian networks in biomedicine and health-care. *Artificial Intelligence in Medicine*, 30:201–214, 2004.

[11] Decision Systems Laboratory University of Pittsburgh. GeNIe and SMILE software available at `https://dslpitt.org/genie/`.

[12] K. G. Olesen, U. Kjaerulff, F. Jensen, F. V. Jensen, B. Falck, S. Andreassen, and S. K. Andersen. A MUNIN network for the median nerve — a case study on loops. *Applied Artificial Intelligence*, 3:384–403, 1989.

[13] Justyna Pawlowska. Probabilistic graphical model for diagnosis of autism (in Polish). Master's thesis, Faculty of Computer Science, Bialystok University of Technology, Bialystok, Poland, July 2014.

[14] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1988.

PROBABILISTYCZNY MODEL WSPIERAJĄCY WCZESNE DIAGNOZOWANIE AUTYZMU

Streszczenie: Sieci bayesowskie są często używanym narzędziem w rozwiązywaniu problemów diagnostycznych. Jedną z zalet tego narzędzia jest możliwość łączenia wiedzy pochodzącej z różnych źródeł. Na przykład, wiedza ekspertów może być połączona z danymi. W naszym artykule prezentujemy model sieci bayesowskiej wspomagający wczesne diagnozowanie autyzmu. Model został zbudowany w oparciu o literaturę medyczną, a następnie zweryfikowany przez ekspertów. Narzędzie, które stworzyliśmy jest dedykowane rodzicom, którzy mogą dokonać wstępnej diagnozy zanim skontaktują sie ze specjalistą.

Słowa kluczowe: sieci bayesowskie, diagnozowanie medyczne, autyzm

Artykuł zrealizowano w ramach pracy badawczej S/WI/2/2013.

THE LIST OF REVIEWERS
(2014)

1. Krzysztof Benedyczak
2. Johanne Bezy-Wendling
3. Wojciech Bieniecki
4. Paweł Czarnul
5. Małgorzata Ćwiklińska-Jurkowska
6. Jakub Gałka
7. Tomasz Grudniewski
8. Piotr Kisielewski
9. Marcin Kołodziej
10. Ewelina Majda-Zdancewicz
11. Michał Matuszak
12. Szczepan Paszkiel
13. Maria Piotrkiewicz
14. Małgorzata Plechawska-Wójcik
15. Andrzej Skalski
16. Ewa Straszecka
17. Cristian Zet