

Zeszyty Naukowe
Politechniki Białostockiej
INFORMATYKA
Numer 7

Oficyna Wydawnicza Politechniki Białostockiej
Białystok 2011

REDAKTOR NACZELNY / EDITOR-IN-CHIEF:

Marek Krętowski (m.kretowski@pb.edu.pl, 85 746 90 95)

SEKRETARZE NAUKOWI / SCIENTIFIC EDITORS:

Magdalena Topczewska (m.topczewska@pb.edu.pl, 85 746 90 86)

Marek Parfieniuk (m.parfieniuk@pb.edu.pl, 85 746 91 08)

SEKRETARZ TECHNICZNY / TECHNICAL EDITOR:

Tomasz Łukaszuk (t.lukaszuk@pb.edu.pl, 85 746 92 07)

RADA NAUKOWA/THE SCIENTIFIC BOARD:

Przewodniczący / Chairman:

Jarosław Stepaniuk (Białystok)

Witold Pedrycz (Edmonton)

Alexandr Petrovsky (Mińsk, Białystok)

Zbigniew Raś (Charlotte, Warszawa)

Członkowie/ Members:

Johanne Bezy-Wendling (Rennes)

Waldemar Rakowski (Białystok)

Leon Bobrowski (Białystok, Warszawa)

Leszek Rutkowski (Częstochowa)

Ryszard Choraś (Bydgoszcz)

Andrzej Salwicki (Warszawa)

Wiktor Dańko (Białystok)

Dominik Sankowski (Łódź)

Marek Drużdżel (Pittsburgh, Białystok)

Franciszek Seredyński (Warszawa)

Piotr Jędrzejowicz (Gdynia)

Władysław Skarbek (Warszawa, Białystok)

Józef Korbicz (Zielona Góra)

Andrzej Skowron (Warszawa)

Halina Kwaśnicka (Wrocław)

Ryszard Tadeusiewicz (Kraków)

Jan Madey (Warszawa)

Sławomir Wierzchoń (Gdańsk, Warszawa)

Andrzej Marciniak (Poznań)

Vyacheslav Yarmolik (Mińsk, Białystok)

Artykuły zamieszczone w *Zeszytach Naukowych Politechniki Białostockiej. Informatyka* otrzymały pozytywne opinie recenzentów wyznaczonych przez Redaktora Naczelnego i Radę Naukową

The articles published in *Zeszyty Naukowe Politechniki Białostockiej. Informatyka* have been given a favourable opinion by reviewers designated by Editor-In-Chief and Scientific Board

© Copyright by Politechnika Białostocka 2011

ISSN 1644-0331

Publikacja nie może być powielana i rozpowszechniana, w jakikolwiek sposób, bez pisemnej zgody posiadacza praw autorskich

ADRES DO KORESPONDENCJI/THE ADDRESS FOR THE CORRESPONDENCE:

„*Zeszyty Naukowe Politechniki Białostockiej. Informatyka*”

Wydział Informatyki /Faculty of Computer Science

Politechnika Białostocka /Białystok University of Technology

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/znpb>

Druk: Oficyna Wydawnicza Politechniki Białostockiej

Nakład: 100 egzemplarzy

CONTENTS

1.	Jarosław Baszun	5
	PASSIVE SOUND SOURCE LOCALIZATION SYSTEM	
	Pasywny system lokalizacji źródeł dźwięku	
2.	Irena Bulatowa , Mateusz Radziwoniuk	17
	DESIGN OF PSEUDO-EQUIVALENT MICROPROGRAM	
	AUTOMATA ON PROGRAMMABLE LOGIC DEVICES	
	Projektowanie pseudoekwiwalentnych automatów mikroprogramowalnych na układach PLD	
3.	Marta Chodyka , Włodzimierz Mosorow	31
	LOGOTYPE DETECTION AS A NEW METHOD	
	OF THE BLOCKING SYSTEM OF INAPPROPRIATE	
	FOR CHILDREN TRANSMISSIONS IN INTERNET TV	
	Detekcja logo jako nowa metoda blokowania nieodpowiednich dla dzieci transmisji w telewizji internetowej	
4.	Jerzy Krawczuk	47
	FORECASTING STOCK INDEX MOVEMENT DIRECTION	
	WITH CPL LINEAR CLASSIFIER	
	Prognozowanie kierunku zmiany indeksów giełdowych za pomocą klasyfikatora liniowego typu CPL	
5.	Anna Piwonska	59
	AN IMPROVED GENETIC ALGORITHM FOR SOLVING	
	THE SELECTIVE TRAVELLING SALESMAN PROBLEM	
	ON A ROAD NETWORK	
	Ulepszony algorytm genetyczny do rozwiązania selektywnego problemu komiwojażera w sieci drogowej	
6.	Daniel Reska , Marek Krętowski	71
	HIST - AN APPLICATION FOR SEGMENTATION	
	OF HEPATIC IMAGES	
	HIST - aplikacja do segmentacji obrazów wątroby	

PASSIVE SOUND SOURCE LOCALIZATION SYSTEM

Jarosław Baszun

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

Abstract: Acoustic source localization system for speech signals based on five microphone array was developed. Three dimensional position computation is based on time delay estimation between pairs of microphones. The psychoacoustically motivated voice activity detector was used to robustly determine activity of speaker in presence of background noise. The detector was based on modulation properties of human speech. Good performance was obtained by selecting frames with speech and nulling frequency bands without speech components. As the result more precisely computation of the time delay was possible. Real experiments shown good immunity of the proposed algorithm to noise and reverberation.

Keywords: phase transform, source localization, microphone arrays.

1. Introduction

Location of sources of waves using array of sensors is the important field of research in radar, seismology and sonar systems. Also similar techniques were developed over for four decades in acoustics. The knowledge about spatial position of sound source can be useful in many audio applications such as automatic camera tracking for video conferencing, suppressing noise and reverberation in voice control for robots hearing systems and audio surveillance. This work concerns the tracking of voice source.

Localization methods in acoustics can be divided into three categories: steered beamformer, high-resolution spectral estimation and time delay estimation based techniques. The most widely used localization techniques are based on time delay estimation in which localization systems computes the location of source in two step process. In the first step a set of time delay of arrivals (TDOA) among different microphone pairs is calculated. The relative time delay for each pair of microphones is determined. In the second step this set is used to estimate the acoustic source location based on knowledge of used microphone array geometry. To perform this different methods can be used to source position calculation: e.g. the triangulation, the maximum likelihood method, the spherical intersection method, the spherical interpolation

method [9]. In time delay estimation approach an important role plays a parametric model for an acoustical environment. Usually two models are used: free-field model and reverberation based model. The time delay estimation algorithm estimates TDOA based on the model.

In this paper passive voice localization system was proposed based on computation TDOA using generalized cross-correlation method with modifications which allow to distinguish between speech and non speech signals in time-frequency domain. The speech to noise estimate is computed in modulation frequency domain for each band separately and used as a feature for speech-pause detection. This psychoacoustically motivated voice activity detector was integrated with computation of weighting function for cross-correlation. Detecting of speech content for each frequency band separately allows for nulling non speech components to avoid influence of disturbing sources on position computation.

2. Time delay estimation

The problem of finding the distance between the sound source and the microphone array is usually defined for near-field case as shown in Fig. 1, but is also possible under some conditions for far-field case. The radius of near-field for array of microphones is defined

$$R_{nf} = \frac{2R_a^2}{\lambda}, \quad (1)$$

where R_a is the size of the array and λ is the wavelength of the operating frequency.

In such situation it is always possible to estimate angle of arrival for wave and the distance between the source and microphones. The time difference of arrival (TDOA) for the pair of microphones is

$$\tau_{1,2} = \frac{r_2 - r_1}{c}, \quad (2)$$

Where c is a speed of sound calculated based on air temperature t_{air} in deg. Celsius, from formula [9]:

$$c = 331 + 0.61t_{air}, \quad (3)$$

If the distance between microphones is know it is possible to calculate unknown parameters $r_1, r_2, \dots, \theta_1, \theta_2, \dots$. When information about TDOA is available it is

possible to calculate position of source in relation to the array using for example triangulation rule.

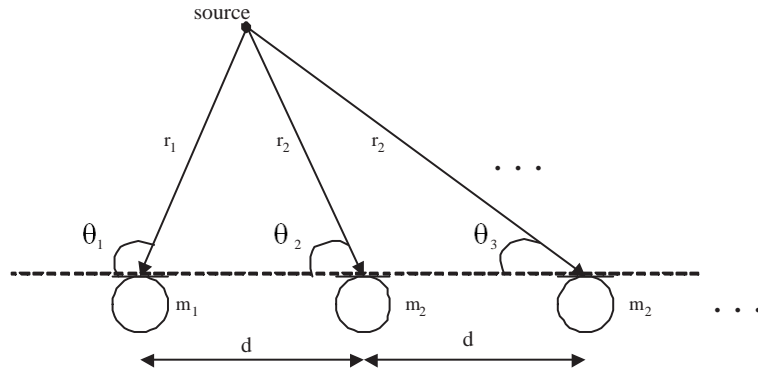


Fig. 1. Linear array

Many approaches can be applied to locate source of signals. In case of multiple narrow band sources, methods based on the eigenvalue analysis of the spatial covariance matrix of the signals from matrix of sensors are commonly used. Such methods were successfully applied especially for radiolocation technology. For wideband signals where we cannot assume the hypothesis about statistical properties of the source of signal and interferences other solutions must be used.

In acoustics we can distinguish two main methods. The first approach is based tuned beamformer. The algorithm scans a set of positions to find the place where the maximum acoustic power is received by the array of sensors. This method have some disadvantages: the computational complexity and the poor resolution in space and in time for moving objects. The advantage of this method is possibility of creation of acoustics maps for objects search [5].

Second approach is based on computation of Time Difference of Arrival (TDOA) between pairs of sensors - microphones. There is a lot of literature on this approach e.g. [2]. One of the basic method of computing of time delay between two signals is to compute maximum of cross-correlation function. But this simple approach gives bad results in case of narrowband signals and in presence of strong reverberation. To overcome this shortage methods of pre-filtering of signals can be applied in case were statistics of source and noise is known or in case when statistics is unknown an effective method is based on whitening the input signals, so only the

phase information in cross-power spectrum of the two signals is used. Such methods are known as Generalized Cross-Correlation (GCC) [10].

Different models can be employed to describe an acoustic environment in the TDOA problem [9]. The ideal model it is assumed that the signal acquired by each sensor is a delayed and attenuated version of the original source signal plus additive noise. This model takes into account the direct signal path only and do not consider multipath signal propagation encountered in many real environments due to reflections. Much more realistic is the multipath model in which received signal is described as a sum of direct signal and weighted sum of delayed reflections [12]. This multipath effect is widely used in the oceanic propagation environment. The drawback of the multipath propagation model is the difficulty to estimate time difference of arrival for pairs of sensors in case of many different paths. So more realistic model for room acoustic environment seems the reverberation model in which for the source signal $s(t)$, the signal received at the two sensors can be described as follows:

$$\begin{aligned} x_1(t) &= h_1 * s(t) + n_1(t), \\ x_2(t) &= h_2 * s(t) + n_2(t) \end{aligned} \quad (4)$$

Where $x_1(t)$, $x_2(t)$ - received signals, h_1 , h_2 represent reverberations and n_1 , n_2 are noise signals received at two sensors. It is assumed additive noise conditions. This model in case of weak reverberation can be simplified to the following model:

$$\begin{aligned} x_1(t) &= k_1 s(t) + n_1(t), \\ x_2(t) &= k_2 s(t + D) + n_2(t), \end{aligned} \quad (5)$$

where $s(t)$ is a source signal, $x_1(t)$, $x_2(t)$ - received signals, k_1 , k_2 are certain weights, D - the delay of the signal arrival between the two sensors and $n_1(t)$, $n_2(t)$ are additive noise. It can be shown that for slowly changed environment parameters the cross-correlation function of signals $x_1(t)$ and $x_2(t)$ can be used to determine the time delay D :

$$R_{x_1 x_2}(\tau) = E[x_1(t)x_2(t - \tau)], \quad (6)$$

where E denotes expectation. For the model from Eq. 5 assuming that noise is not correlated with the signal $s(t)$ the cross-correlation is:

$$R_{x_1 x_2}(\tau) = k_1 k_2 R_{ss}(\tau - D) + R_{n_1 n_2}(\tau), \quad (7)$$

The cross-power spectrum of (the Fourier transform of cross-correlation) is:

$$G_{x_1 x_2}(\omega) = k_1 k_2 G_{ss}(\omega) e^{-j\omega D} + G_{n_1 n_2}(\omega) \quad (8)$$

Background noise can be correlated due to the fact that it is produced by single source e.g. computer fan and can be estimated using energy detector and next subtracted. The cross power spectrum with subtracting noise becomes

$$\hat{G}_{x_1x_2}(\omega) = G_{x_1x_2}(\omega) - G_{n_1n_2}(\omega) = k_1k_2G_{ss}e^{-j\omega D} \quad (9)$$

This leads to normalized cross-correlation called Phase Transform (PHAT) [4], [10]:

$$\hat{R}_{x_1x_2}(\tau) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{\hat{G}_{x_1x_2}(\omega)}{|G_{x_1x_2}(\omega)|} e^{j\omega\tau} d\omega = (\tau - D). \quad (10)$$

Weighting factor:

$$\Psi(\omega) = \frac{1}{|G_{x_1x_2}(\omega)|}, \quad (11)$$

cause the whitening of the input signals. In effect only the phase information in the cross-power spectrum of the two signals is used. Because this operation weights $\hat{G}_{x_1x_2}(\omega)$ as the inverse of $G_{s_1s_2}(\omega)$ so errors arise for frequencies where signal power is small in compare to interferences. As a result in case of lack of signal source $s(t)$ knowledge appropriate weighting in spectrum domain of signals from sensors is required to avoid influence of this errors.

The TDOA between two microphones can be found by selecting the maximum location of the Eq. 10:

$$D = \operatorname{argmax}_{\tau} \hat{R}_{x_1x_2}(\tau), \quad (12)$$

In [14], [3] was shown that phase correlation approach can be used also in case of reverberation using some additional processing.

3. Voice activity detector

In this system psychoacoustically motivated voice activity detector (VAD) was used for two proposes: to select frames with speech and to select frequency bands where speech signal is dominant to minimalize noise influence on time delay calculation. This voice detector is an expansion of the idea developed in previous work in this area [1]. The detector exploits properties of modulation spectrum of human speech [6], [11]. It is known that modulations of sound are the carrier of information in speech. The background noise encountered in real environments is usually stationary

or changing differently in compare to the rate of change of speech. Modulation components of speech are mainly concentrated in range between 1 and 16 Hz with higher energies around 3 – 5 Hz what corresponding to the number of syllables pronounced per second [8]. Slowly-varying or fast-varying noises will have components outside the speech range. Further, steady tones will only have constant component in modulation domain. Additive noise reduces the modulation peak in speech modulation spectrum. System capable of tracking speech components in modulation domain allows to distinguish between frequency bands with dominant speech signal and band with dominant background noise. This operation is the key element of effective computation of GCC-PHAT algorithm because it is possible to set to zero signal in bands classified as noise.

The block diagram of the voice activity detector was shown in Fig. 2. Signal from microphone with sampling frequency 16 kHz is split into $M = 512$ frequency bands using Short Time Fourier Transform (STFT) with Hamming window and 25 % overlapping. Next amplitude envelope is calculated for first 256 bands:

$$y_k(nM) = \sqrt{\text{Re}^2[x_k(nM)] + \text{Im}^2[x_k(nM)]} \quad (13)$$

Then amplitude envelope is filtered by passband IIR filters with center frequency 3.5 Hz and frequency response shown in Fig. 3. The output of the filters is half-wave filtered to remove negative values from output of the filters. The following computation is carried out on the filtered and not filtered envelopes:

$$S(nM) = \frac{Y'}{Y - \text{mean}(Y) - Y' - \text{mean}(Y')} \quad (14)$$

Above parameter is an estimate of speech to noise ratio for each of analyzed channels. Mean value of filtered and nonfiltered envelope is computed based on exponential averaging with time constant approximately 1 s. Then all channels are summed and the square of this estimate is used as a classification parameter for voice activity detector. Speech decision is based on comparison between classification parameter and the threshold computed based on the following statistics [13]:

$$\text{Thr} = \text{mean}(d) + \alpha \cdot \text{std}(d) \quad (15)$$

where d is a classification parameter and α controls confidence limits and is usually in the range 1 to 2, here was set to be equal 2. Both mean value and standard deviation is estimated by exponential averaging in pauses. Frame is considered to be active if value of the classifier is greater than the threshold. Speech to noise computed

parameter for each channel in combination with the threshold is used to select channels with speech signal and nulling channels with noise in time delay computation algorithm.

To avoid isolated errors on output of VAD caused by short silence periods in speech or short interferences correction mechanism described in [7] was implementing. If current state generating by the VAD algorithm does not differ from n previous states then current decision is passed to detector output otherwise the state is treated as a accidental error and output stays unchanged.

4. Implementation of time delay estimation algorithm

In Fig. 4 block diagram of time delay estimation algorithm was shown for two channels x_1 and x_2 . Signals from both sensors are grouped into frames. One of the channels from microphone array is used by voice activity detector to calculate which frame contain speech and in what channels the speech signal is present. When speech signal is detected power spectra of signals for pair of channels are calculated using Fast Fourier Transform (FFT). Then cross-spectrum is calculated. The cross-spectrum of signal is averaged over several frames. The averaged cross-spectrum is then normalized according to equation:

$$G'_{x_1x_2}(\omega) = \frac{\hat{G}_{x_1x_2}(\omega)}{|G_{x_1x_2}(\omega)|}. \quad (16)$$

The normalized cross-spectrum $G'_{x_1x_2}(\omega)$ of the frequencies that were classified as non speech components are set to zero. Then inverse FFT is calculated on the averaged and normalized cross-spectrum. In classical GCC-PHAT algorithm the time delay is chosen as the lag that corresponds to the maximum of the normalized cross-correlation function. To increase resolution of time delay estimation three sample interpolator was implemented. The maximum value from the normalized cross-correlation function is selected and its both sides neighbours, as shown in Fig. 5. Therefore, time delay D becomes:

$$D = \frac{A_{i-1}t_{i-1} + A_it_i + A_{i+1}t_{i+1}}{A_{i-1} + A_i + A_{i+1}} \quad (2 \leq i \leq N-1), \quad (17)$$

where A_i is the largest value of the normalized cross-correlation function and t_i corresponding time delay. This method makes it possible to calculate time delay more accurately without a large number of FFT samples.

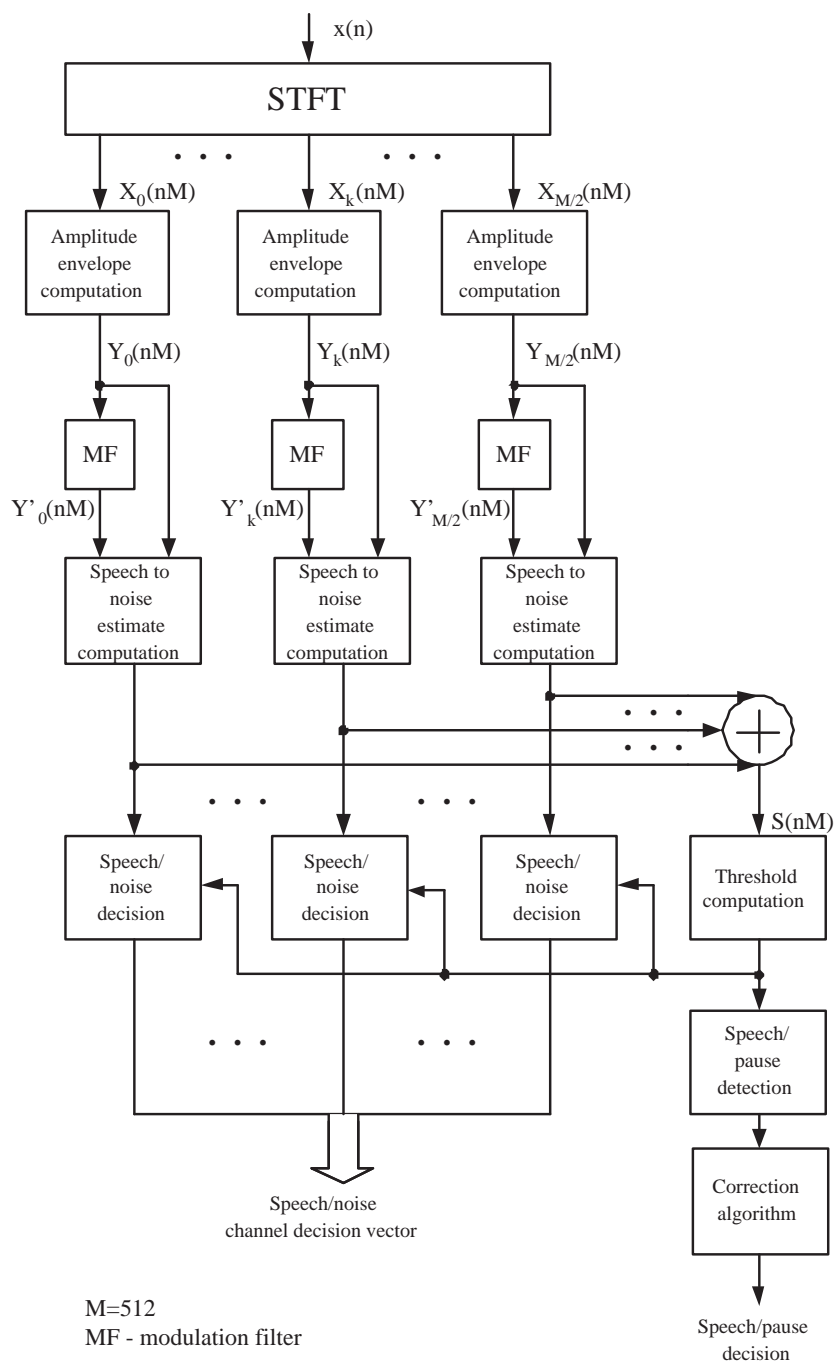


Fig. 2. Block diagram of the voice activity detector (VAD) based on modulation properties of speech

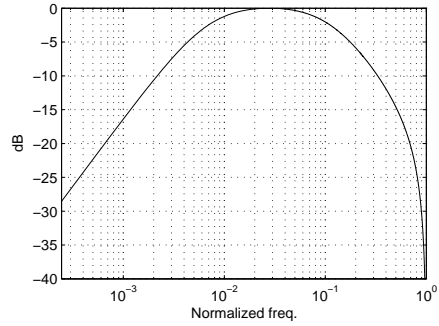


Fig. 3. Magnitude frequency response of modulation filter (MF)

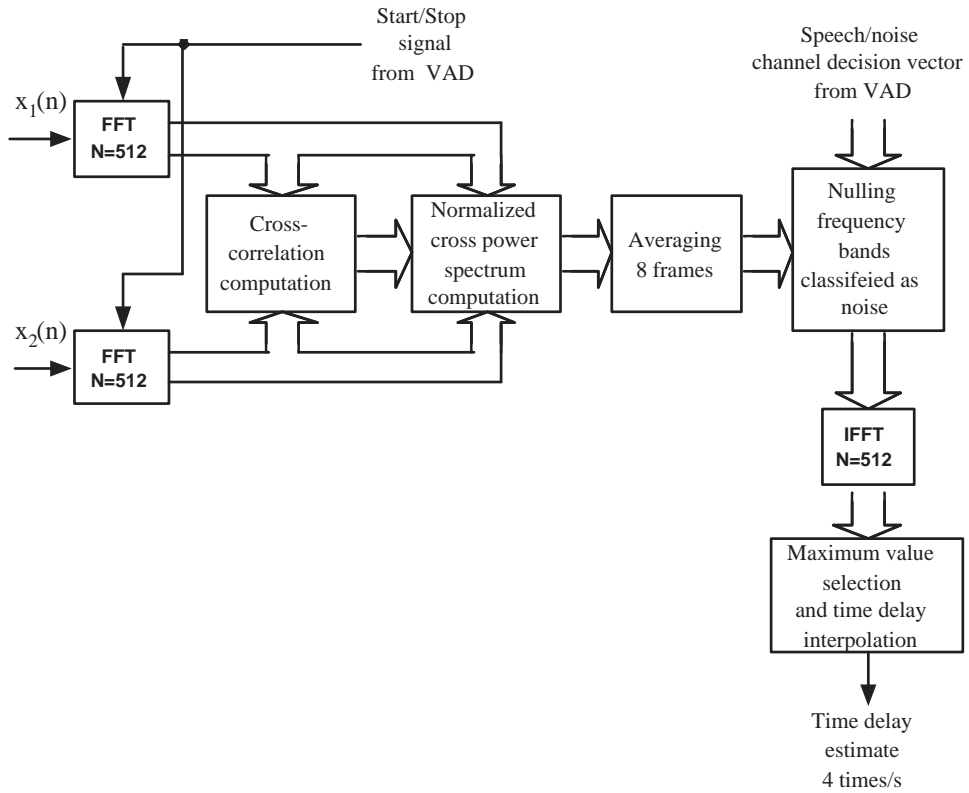


Fig. 4. The time delay estimation algorithm block diagram for two channels

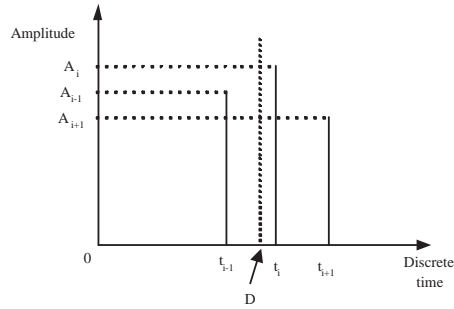


Fig. 5. The time delay interpolation method

5. Position calculation and results

The microphone array used for voice source localization was shown in Fig. 6. Two edge pair of microphones 1 and 3 are used for azimuth calculation by triangulation. Pairs 1,2, 2,3 and 2,5 are used for the depth calculation. Using the microphone pair 2,5 it is possible to distinguish between sources localized in front and behind the array. Experiment was carried out in room 7m x 5.5m x 2.8m. Sampling rate was 16 kHz, frame 512 samples, 8 frames were averaged to calculate time delay sample. In Table 1 measured distances for three positions was shown.

Table 1. Distance measurements and standard deviation for averaged 20 measurements

Distance (m)	Std. deviation	Azimuth (deg)	Elevation (deg)
2.21	0.05	-24.7	-15.4
3.08	0.2	20.3	-10.2
5.12	0.32	15.2	2.5

For elevation angle calculation pair 2,4 was used. For this pair some problems with strong reflections of the signal from the floor were observed. This situation can happen when a floor surface is made of terracotta tiles. In such situation time delay corresponding to the reflection of the source is longer than time delay of direct signal. To overcome this, in situation when two highest peaks of the cross-correlation function differ only on less than ten per cent, the peak closer to zero lag is chosen as the true lag.

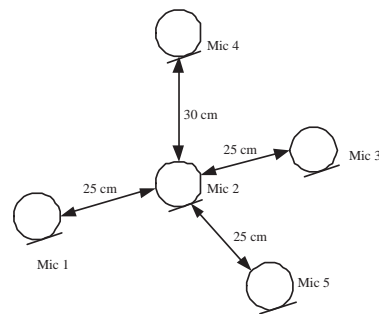


Fig. 6. The microphone array for sound source localization system

6. Conclusions

Passive sound source localization system for speech signals was developed and tested. Combination of time delay estimation algorithm with voice activity detector based on modulation properties of human speech gave the significant improvement in performance allowing to select precisely frames with speech but also to eliminate frequency bands without speech components and more accurately compute time delay. These future make possible to build reliable three dimensional speaker localization system using a small microphone array.

References

- [1] Baszun J., Voice Activity Detection for Speaker Verification Systems. Joint Rough Set Symposium, Toronto, Canada, (14-16 May, 2007), 181–186.
- [2] Benesty J., Chen J., Huang Y., Microphone Array Signal Processing, Springer Topics in Signal Processing Series, Vol. 1, Springer-Verlag, 2010.
- [3] Brutti A.B., Omologo M., Svaizer P., Comparision Between Different Sound Localization Techniques Based on a Real Data Collection IEEE HSCMA, (2008), 69–72.
- [4] Carter C.G., Nuttal A.H., Cable P.G., The Smoothed Coherence Transform, Proc. IEEE (Letter), Vol. 61, (Oct. 1973), 1497–1498.
- [5] Dmochowski J.P., Benesty J., Affes S., A Generalized Steered Response Power Method for Computationally Viable Source Localization, Audio, Speech and Language Processing, IEEE Trans. on, Vol. 15, I. 8, (Nov. 2007), 2510–2526.
- [6] Elhilali M., Chi T., Shamma S., A Spectro-temporal Modulation Index (STMI) for Assesment of Speech Intelligibility. Speech Communication, Vol. 41. (2003), 331–348.

- [7] El-Maleh K., Kabal P., Comparison of Voice Activity Detection Algorithms for Wireless Personal Communications Systems. Proc. IEEE Canadian Conference Electrical and Computer Engineering, (May 1997), 470–473.
- [8] Houtgast T., Steeneken H.J.M., A Review of the MTF Concept in Room Acoustics and its Use for Estimating Speech Intelligibility in Auditoria. J. Acoust. Soc. Am., Vol. 77, No. 3 (Mar. 1985), 1069–1077.
- [9] Huang Y.A., Benesty J., (Eds.), Audio Signal Processing for Next Generation Multimedia Communication Systems, Kluwer Academic Publishers, 2004.
- [10] Knapp C.H., Carter C., The Generalized Correlation Method for Estimation of Time Delay, IEEE Transaction on Acoustics, Speech, And Signal Processing, Vol. ASSP-24, No. 4 (Aug. 1976), 320–327.
- [11] Mesgarani N., Shamma S., Slaney M., Speech Discrimination Based on Multi-scale Spectro-Temporal Modulations. ICASSP, (2004), 601–604.
- [12] Moghaddam P.P., Amindavar H., Kirlin R.L., A New Time-Delay Estimation in Multipath, IEEE Transaction on Signal Processing, Vol. 51, (May 2003), 1129–1142.
- [13] Sovka P., Pollak P., The Study of Speech/Pause Detectors for Speech Enhancement Methods. Proc. of the 4th European Conference on Speech Communication and Technology, Madrid, Spain (Sep. 1994), 1575–1578.
- [14] Wang H., Chu P., Voice Source Localization for Automatic Camera Pointing System in Videoconferencing Proc. IEEE ASSP Workshop Applications on Signal Processing Audio Acoustics, (Oct. 1997), 1497–1498.

PASYWNY SYSTEM LOKALIZACJI ŹRÓDEŁ DŹWIEKU

Streszczenie Opracowano metodę lokalizacji akustycznych źródeł dźwięku zorientowaną na sygnału mowy. System zbudowano w oparciu o macierz pięciu mikrofonów. Obliczenia pozycji źródła w trzech wymiarach dokonano na podstawie estymacji różnicy czasu przybycia dla par mikrofonów. Zastosowany psychoakustycznie motywowany detektor mowy umożliwia ocenę aktywności mówcy w obecności zakłóceń. Dobrą efektywność uzyskano poprzez selekcję ramek z mową oraz zerowanie zakresów częstotliwości w których sygnał zakłócający maskuje sygnał mowy. Jego zaletą jest możliwość precyzyjnego obliczenia czasu opóźnienia. Eksperymenty w warunkach rzeczywistych pokazują dobrą odporność zaproponowanego algorytmu na szum i pogłos.

Słowa kluczowe: transformacja fazy, lokalizacja źródła, macierze mikrofonów.

Artykuł zrealizowano w ramach pracy badawczej S/WI/4/08.

DESIGN OF PSEUDO-EQUIVALENT MICROPROGRAM AUTOMATA ON PROGRAMMABLE LOGIC DEVICES

Irena Bulatowa, Mateusz Radziwoniuk

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

Abstract: In this paper, a new method of synthesis of microprogram automata from ASM specification is presented. This method allows converting pseudo-equivalent automaton to an equivalent one by eliminating the zero-value output sets appearing in additional internal states. The proposed method is based on a modified model of microprogram automaton, which permits changing the output signals only in the basic internal states, thereby eliminating the zero-value sets of output signals generated in additional states of pseudo-equivalent automata. This allows removing the adverse effects of introducing additional states and provides a wider application of numerous methods for the synthesis of pseudo-equivalent microprogram automata. The experimental results show that the cost of realization of the proposed structure in programmable logic devices increases insignificantly, but then it leads to extend the field of application synthesis methods based on the introduction of additional internal states.

Keywords: microprogram automaton, Algorithmic State Machine (ASM), pseudo-equivalent automaton, additional internal states, programmable logic devices (PLD)

1. Introduction

Developing effective methods for synthesis of microprogram automata on programmable logic devices (PLD) is a very important problem because the majority of control systems are based on the principle of microprogram control [7]. The behavior of microprogram automata is very often specified by Algorithmic State Machine (ASM) charts [2], which are very useful and convenient methods of control algorithm description. Lots of methods for the synthesis of microprogram automata from ASM specifications have been developed [2,3,5,1,6]. Some of these synthesis methods require the introduction of additional internal states for receiving special features of designed microprogram automata. For example, some methods based on additional internal states may allow simplifying the microprogram automata scheme

and reducing the cost of their realization [1], and other methods [1,6] allow including restrictions on the number of inputs of components used for microprogram automata realization.

However, the introduction of additional internal states is not always acceptable in microprogram automata design. This is due to the fact that the zero-value output sets are generated in additional internal states. It may result in damage to the functioning of the designed microprogram automata when the output signals must be maintained at a constant high level and any changes of the signal level are not permitted.

As a result of the introduction of additional internal states during the synthesis process, the pseudo-equivalent automaton will be received. Pseudo-equivalent automaton generates the same sequence of output signals as original automaton, but differs from it in that the zero-value output sets are generated in the output sequence of pseudo-equivalent automaton in additional states. That fact significantly reduces the application area of synthesis methods based on the introduction of additional internal states.

In this paper, a new method for the synthesis of microprogram automata from ASM specification is presented. This method allows converting the pseudo-equivalent automaton into an equivalent one by eliminating the zero-value output vectors appearing in additional internal states. The proposed method is based on a modified model of microprogram automaton, which allows triggering the output signals only in basic internal states thereby eliminating the zero-value sets on automaton outputs. The generation of an additional control signal in the proposed model leads to a slight increase in the complexity of automaton realization, but then it makes it possible to apply numerous methods for the synthesis of pseudo-equivalent microprogram automata, even in such applications in which it was previously impossible.

2. Synthesis of microprogram automata from ASM

Due to the principle of microprogram control [7], any complex operation executed by a digital device is represented as a sequence of elementary operations $Y = \{y_1, \dots, y_N\}$, called *microoperations*. The subset $Y^t \subseteq Y$ of microoperations executed in the same clock period forms a *microinstruction*. The order of microinstructions execution is determined by logical conditions $X = \{x_1, \dots, x_L\}$. The control algorithm specified in terms of microoperations and logical conditions is called a *microprogram* and the automaton, which realizes the microprogram, is called a *microprogram automaton* [6].

The Algorithmic State Machine (ASM) charts [2] (Fig.1) are widely used for control algorithm specification. Each operator vertex of ASM contains a microin-

struction $Y^t \subseteq Y$, $Y = \{y_1, \dots, y_N\}$ defined as a collection of microoperations which are executed in the same clock period, and $Y^t = \emptyset$ is acceptable. One of the logical conditions from the set $X = \{x_1, \dots, x_L\}$ is written in each conditional vertex and it is possible to write the same logical condition in different vertices [2].

The finite state machine (FSM) is used as a model of microprogram automata. Synthesis of FSM from the ASM chart begins from the construction of marked ASM, due to which the ASM chart is marked by labels $A = \{a_1, \dots, a_M\}$ corresponding to internal states of FSM [2]. Standard approaches to Moore and Mealy FSM synthesis are well known [2,3]. Labels and corresponding internal states introduced by these standard algorithms will be called *basic labels* and *basic internal states*.

Due to the standard algorithm for the synthesis of Mealy FSM from ASM chart [3], the input vertex following the initial vertex *Begin* and the input of vertex *End* are marked by the symbol a_1 (corresponding to the initial state of automaton), then the inputs of vertices following operator vertices are marked by symbols a_2, \dots, a_M . This algorithm of ASM marking allows building FSM in which the output functions y_1, \dots, y_N will depend on the current internal states and input variables x_1, \dots, x_L .

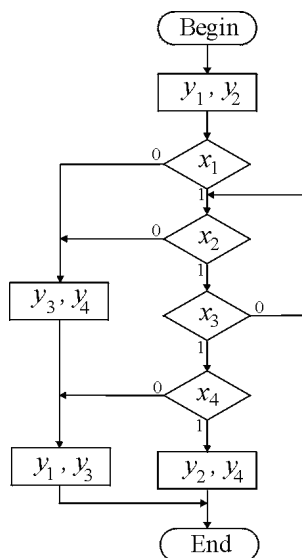


Fig. 1. An example of ASM chart

A graph (or transition table) of automaton is constructed from the marked ASM by defining all the transition paths between internal states: $a_m X(a_m, a_s) Y(a_m, a_s) a_s$,

where $X(a_m, a_s)$ – the product of logical conditions on the transition path from a_m to a_s , $a_m, a_s \in A$; $Y(a_m, a_s)$ – microinstruction generated on this transition.

For Moore automaton synthesis, the marked ASM is constructed as follows [2]: vertices *Begin* and *End* are marked by the same symbol a_1 , and all operator vertices are marked by different symbols a_2, \dots, a_M . This algorithm allows implementing the FSM with output functions y_1, \dots, y_N depending only on the current state of the automaton.

Many methods for FSM synthesis from ASM have been developed in which the additional internal states are introduced besides the basic internal states. In such methods, the additional symbols a_{M+1}, \dots, a_{M+K} are used for marking ASM that leads to the introduction of additional internal states of FSM.

Increasing the number of internal states allows the automata to acquire new properties. For example, in the synthesis method proposed in [1], additional states are used for minimizing the number of transitions between states that can reduce the complexity of automata realization.

In methods [1,6], the additional states are introduced to decrease the dependency of the transition and output functions on input variables. In these algorithms, after the standard marking of ASM, the additional labels a_{M+1}, \dots, a_{M+K} are placed at the inputs of conditional vertices. This makes it possible to reduce the rank of the conjunctions $X(a_m, a_s)$ defining the automaton transitions, which results in reducing the rank of the products in transition functions d_1, \dots, d_R and in output functions y_1, \dots, y_N . It may be important if there are restrictions on the number of inputs of components used for FSM realization.

As a result of the introduction of additional states, the pseudo-equivalent automaton will be obtained [6]. Let z_1, \dots, z_k be some sequence of input variables vectors on automaton inputs, and w_1, \dots, w_k will be the corresponding sequence output vectors generated on automaton outputs. Two automata S_1 and S_2 are called *equivalent*, if they generate the same output sequences w_1, \dots, w_k for each input sequence z_1, \dots, z_k . An automaton S_2 is called *pseudo-equivalent* to automaton S_1 , if it generates the same output sequence as S_1 , but in its output sequence zero-value output vectors may appear as a result of the introduction of additional states [6].

Zero-value output vectors correspond to paths in ASM which don't pass through an operator vertex (for Mealy automaton) or to paths which don't lead to an operator vertex (for Moore automaton). Such paths end in some additional label $a_j, j > M$. The appearance of zero-value vectors may be unacceptable in some applications when it is important to maintain the output signals at a constant high level.

Let us consider an example of ASM shown in Fig.2. At the beginning, the ASM has been marked by symbols a_1, \dots, a_5 according to the standard algorithm for the

synthesis of Moore FSM [2]. Then, the additional labels a_6 , a_7 and a_8 have been introduced to reduce the dependency of FSM transition functions from input variables [1]. According to this algorithm, the inputs of conditional vertices connected by the edge with the output of other conditional vertex are marked by additional labels, which leads to the introduction of additional internal states. As a result, each transition path of FSM will depend on no more than one logical condition, which allows limiting the maximum rank of conjunctions in transition functions of Moore FSM. This can be useful when there are hard restrictions on the number of inputs of elements used for circuit realization.

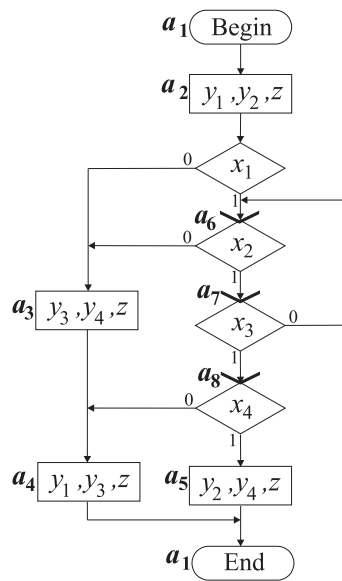


Fig. 2. Marked ASM for synthesis of Moore FSM with additional labels

However, the introduction of additional internal states may be unacceptable in some applications. Let us consider the transition path between basic states a_2 and a_5 on ASM presented in Fig.2. In both states, the output signal y_2 is generated. If in practical application, it is required to maintain the signal y_2 at a constant high level on the transition from a_2 to a_5 , the introduction of additional states between a_2 and a_5 will be unacceptable, because in additional states a_6 , a_7 and a_8 signal y_2 will be temporarily triggered to a low level. A similar situation is also possible for output signal y_1 on transition from a_2 to a_4 (Fig.2). This fact narrows the field

of application of synthesis methods based on the introduction of additional internal states and requires a preliminary inspection of the control algorithms before applying such synthesis methods.

In this paper, we propose a modified model of microprogram automaton, which allows eliminating the zero-value output sets in additional internal states and gives the ability of a wider application of synthesis methods of pseudo-equivalent automata.

3. Modified model of microprogram automaton

The modified model of microprogram automata is presented in Fig.3. The register RG stores the current automaton state code e_1, \dots, e_R , where $R = \lceil \log_2 M \rceil$ is the least integer greater than or equal to $\log_2 M$. The combinational logic circuit CL implements the output functions y_1, \dots, y_N and the transition functions d_1, \dots, d_R .

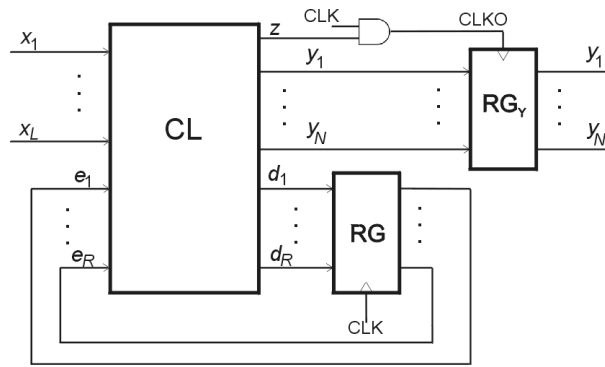


Fig. 3. Modified structure of microprogram automata

An additional register RG_Y is introduced in this model to store the output functions values y_1, \dots, y_N . The special pulse $CLKO$ is used to change the content of the RG_Y register. The signal $CLKO$ is generated on the basis of clock signal z formed by combinational circuit CL (Fig.4). The waveforms for $CLKO$ signal generation are shown in Fig.4, where the clock periods corresponding to the basic internal states are marked by arrows. An additional signal $z = 1$ is formed by combinational logic circuit CL only in the basic internal states, but in additional states $z = 0$.

This causes that the content of register RG_Y will be changed only in basic internal states, and the zero-value output vectors generated in the additional states will not be written to register RG_Y , so they never appear on the automaton outputs y_1, \dots, y_N .

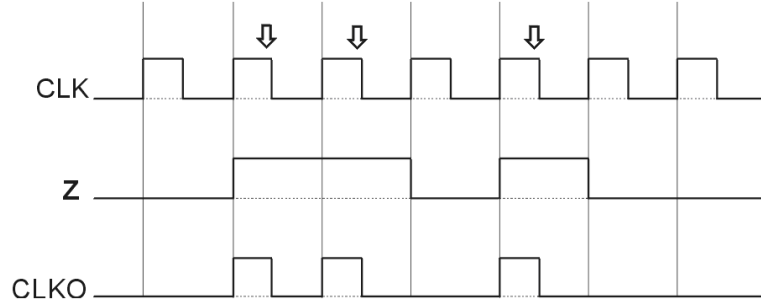


Fig. 4. Waveforms for CLKO signal forming

The implementation of additional register RG_Y in PLD structures does not increase the number of used macrocells for FSM realization, since the internal memory elements of macrocells are used for its implementation. The complexity of realization of the proposed structure has increased insignificantly; it is related to the implementation of only one additional function z .

To implement function z , all microinstructions generated in basic internal states should be expanded by one extra microoperation z that will correspond to forming the microoperation signal $z = 1$ only in the basic internal states.

In our example (Fig.3), the ASM was marked for synthesis of Moore FSM and then the additional states a_6 , a_7 and a_8 were introduced. According to the proposed synthesis method, an additional microoperation z must be inserted in all operator vertices of ASM, because each operator vertex corresponds to the basic state of Moore FSM.

The structure table of Moore automaton with additional microoperation z is shown in Table 1, where each transition is described by the following columns: a_m is the current FSM state, $K(a_m)$ is the code of the state a_m , a_s is the next state, $K(a_s)$ is the code of the state a_s , $X(a_m, a_s)$ is the conjunction of inputs determining the transition, $Y(a_m)$ is the microinstruction generated in the state a_m , $D(a_m, a_s)$ is a collection of transition functions for D-type memory elements.

On the basis of the structure table, the following expressions for output functions y_1, \dots, y_4 , transition functions d_1, \dots, d_3 and for additional function z are obtained:

$$\begin{aligned} y_1 &= \bar{e}_1 \bar{e}_2 e_3 + \bar{e}_1 e_2 e_3 \\ y_2 &= \bar{e}_1 \bar{e}_2 e_3 + e_1 \bar{e}_2 \bar{e}_3 \\ y_3 &= \bar{e}_1 e_2 \bar{e}_3 + \bar{e}_1 e_2 e_3 \\ y_4 &= \bar{e}_1 e_2 \bar{e}_3 + e_1 \bar{e}_2 \bar{e}_3 \end{aligned}$$

Table 1. Structure table of automaton

a_m	$K(a_m)$	a_s	$K(a_s)$	$X(a_m, a_s)$	$Y(a_m)$	$D(a_m, a_s)$
a_1	000	a_2	001	1	–	d_3
a_2	001	a_3	010	\bar{x}_1	y_1, y_2, z	d_2
		a_6	101	x_1		$d_1 d_3$
a_3	010	a_4	011	1	y_3, y_4, z	$d_2 d_3$
a_4	011	a_1	000	1	y_1, y_3, z	d_3
a_5	100	a_1	000	1	y_2, y_4, z	d_3
a_6	101	a_3	010	\bar{x}_2	–	d_2
		a_7	110	x_2		$d_1 d_2$
a_7	110	a_6	101	\bar{x}_3	–	$d_1 d_3$
		a_8	111	x_3		$d_1 d_2 d_3$
a_8	111	a_4	011	\bar{x}_4	–	$d_2 d_3$
		a_5	100	x_4		d_1

$$z = \bar{e}_1 e_2 \bar{e}_3 + \bar{e}_1 e_2 \bar{e}_3 + \bar{e}_1 e_2 e_3 + e_1 \bar{e}_2 \bar{e}_3$$

$$d_1 = \bar{e}_1 \bar{e}_2 e_3 x_1 + e_1 \bar{e}_2 e_3 x_2 + e_1 e_2 \bar{e}_3 \bar{x}_3 + e_1 e_2 \bar{e}_3 x_3 + e_1 e_2 e_3 x_4$$

$$d_2 = \bar{e}_1 \bar{e}_2 e_3 \bar{x}_1 + \bar{e}_1 e_2 \bar{e}_3 + e_1 \bar{e}_2 e_3 \bar{x}_2 + e_1 \bar{e}_2 e_3 x_2 + e_1 e_2 \bar{e}_3 x_3 + e_1 e_2 e_3 \bar{x}_4$$

$$d_3 = \bar{e}_1 \bar{e}_2 \bar{e}_3 + \bar{e}_1 \bar{e}_2 e_3 x_1 + \bar{e}_1 e_2 \bar{e}_3 + e_1 e_2 \bar{e}_3 \bar{x}_3 + e_1 e_2 \bar{e}_3 x_3 + e_1 e_2 e_3 \bar{x}_4$$

In our example, the complexity of FSM realization has increased slightly due to forming of an additional function z , which contains the same products as output functions of Moore FSM.

The zero-value output sets in additional internal states could also be eliminated in another way by simple repeating in additional states of the microoperations that are generated in the preceding basic state. However, such an approach leads to a significant increase in the complexity of output functions realization, so it will be less effective compared with the proposed method.

4. Experimental results

The proposed method was tested using the control algorithms from the ASM library of the Abelite EDA tool [4]. To perform the experiments, three methods using additional states have been implemented and compared: M1 – the method, in which the zero-value output vectors appear in additional states [1]; M2 – the proposed method, in which the additional signal z is formed to eliminate the zero-value output sets; M3 – the method, in which the zero-value output sets are eliminated by the repeating in additional states of the microoperations from the previous basic state. All these methods introduce the same number of additional states to separate all conditional

vertices on the ASM chart. For all the methods, the number of used macrocells of PLD were compared for the realization of automata on FLEX10K and MAX9000 devices of Altera, which are the typical representatives of two PLD classes: CPLD (MAX9000 device family) and FPGA (FLEX10K devices).

Table 2 shows the results of comparison of methods M1, M2 and M3 for the FLEX10K device family, where "ASM" is the name of the example from the ASM library, L, N, S, K are the numbers of inputs, outputs, states and additional states of automaton, respectively, C_{M1}, C_{M2}, C_{M3} are the numbers of macrocells of the FLEX10K device used for the realization automata for synthesis methods M1, M2 and M3, respectively. For methods M2 and M3, the values P_{M2} and P_{M3} have been calculated as: $P_{M2} = \frac{C_{M2}-C_{M1}}{C_{M1}}, P_{M3} = \frac{C_{M3}-C_{M1}}{C_{M1}}$, where P_{M2} and P_{M3} are the percent of growth of the number of used macrocells for methods M2 and M3, respectively, in comparison with method M1.

Analysis of the obtained results show that the number of used macrocells for the proposed method M2 increases on average 3.65% (1.38% in the best case) in comparison with method M1. The method M3, as it was expected, requires a significantly greater increase in the amount of hardware, on average 23.75% (even 38.19% in the worst case), so method M3 is much less effective in comparison with the proposed method M2.

The results presented in Table 3 show the dependency of value PM2 on such a parameter as the percentage of conditional blocks in ASM (B_X/B), where B is the whole number of blocks in ASM, B_X is the number of conditional blocks, L, N, S are the numbers of inputs, outputs, and states of automaton, respectively.

The results from Table 3, which show the correlation between the percentage of conditional block in ASM (B_X/B) and the growth of the number of macrocells used for FSM realization by method M2, are also presented in a scatter graph in Fig.5. The pattern of dots suggests the falling correlation between the parameters, thus for the tested examples the complexity growth rate for method M2 reduces with the increase of the percentage of conditional blocks in ASM.

Table 4 shows the results of comparison of methods M1, M2 and M3 for realization of automata on MAX9000 devices of Altera.

Analysis of the results presented in Table 4 shows that the number of used macrocells for the proposed method M2 increases on average 3.43% (0.55% in the best case) in comparison with method M1. And in the case of method M3, the complexity of realization increases significantly, on average 17.95% (even 39.78% in the worst case), so the proposed method M2 is a more effective approach to eliminating zero-value output sets.

Table 2. Comparison of synthesis methods for realization on FLEX 10K device family

ASM	L	N	S	K	C_{M1}	C_{M2}	C_{M3}	P_{M2}	P_{M3}
acdl	16	27	174	151	330	335	398	1.52%	20.61%
araf	25	65	124	48	230	240	279	4.35%	21.30%
ass13	5	25	38	19	89	91	110	2.25%	23.60%
berg	21	51	121	51	224	234	285	4.46%	27.23%
cpu	14	29	44	21	83	87	96	4.82%	15.66%
cyr	20	75	132	55	244	258	292	5.74%	19.67%
e1	12	13	114	90	201	205	274	1.99%	36.32%
e6	11	20	41	23	82	86	93	4.88%	13.41%
e15	13	20	85	67	163	167	198	2.45%	21.47%
klain	27	61	134	55	251	258	300	2.79%	19.52%
kobz	19	53	130	59	235	245	293	4.26%	24.68%
lcu	15	24	81	58	144	150	199	4.17%	38.19%
lior	24	31	116	79	188	198	255	5.32%	35.64%
max	26	41	105	56	189	197	233	4.23%	23.28%
micks	21	45	106	53	195	200	254	2.56%	30.26%
pilot	27	22	60	33	111	115	138	3.60%	24.32%
raz	23	72	131	60	238	248	300	4.20%	26.05%
sasi	19	54	129	54	240	251	293	4.58%	22.08%
structm	33	36	106	87	217	220	234	1.38%	7.83%
v16	14	18	89	72	164	167	211	1.83%	28.66%
oshr	19	72	144	53	257	266	307	3.50%	19.46%
e16	13	18	85	67	161	166	190	3.11%	18.01%
e8	13	20	85	67	163	167	198	2.45%	21.47%
bcomp	18	39	68	33	122	129	145	5.74%	18.85%
asm1	15	22	32	19	58	61	79	5.17%	36.21%
Average								3.65%	23.75%

5. Conclusions

The proposed method can be used for eliminating zero-value output vectors, which appear in additional internal states of microprogram automata. This method is based on a modified model of microprogram automata, which allows removing the adverse effects of introducing additional states in exchange for a slight increase in the amount of hardware. The experimental results show that the proposed method is more effective (on average 15.37% for realization on FLEX 10K devices and 19.11% for realization on MAX9000 devices) than the approach based on repeating output signals in additional states.

Table 3. Dependency of synthesis results on parameters of ASM and FSM

ASM	L	N	S	B	B _X	B _X /B	C _{M1}	C _{M2}	P _{M2}
acd1	16	27	174	194	171	88.14%	330	335	1.52%
alf	31	74	127	160	83	51.88%	241	262	8.71%
araf	25	65	124	134	58	43.28%	230	240	4.35%
ass13	5	25	38	52	33	63.46%	89	91	2.25%
bech	18	39	68	72	37	51.39%	119	128	7.56%
berg	21	51	121	132	62	46.97%	224	234	4.46%
bs	19	13	127	144	127	88.19%	214	219	2.34%
cat	11	22	30	37	20	54.05%	58	62	6.90%
cpu	14	29	44	49	26	53.06%	83	87	4.82%
cyr	20	75	132	140	63	45.00%	244	258	5.74%
e1	12	13	114	135	111	82.22%	201	205	1.99%
e6	11	20	41	50	32	64.00%	82	86	4.88%
e15	13	20	85	102	84	82.35%	163	167	2.45%
klain	27	61	134	153	74	48.37%	251	258	2.79%
kobz	19	53	130	141	70	49.65%	235	245	4.26%
lcu	15	24	81	99	76	76.77%	144	150	4.19%
lift	14	30	42	56	24	42.86%	79	86	8.86%
lior	24	31	116	134	97	72.39%	188	198	5.32%
max	26	41	105	113	64	56.64%	189	197	4.23%
micks	21	45	106	115	62	53.91%	195	200	2.56%
pilot	27	22	60	70	43	61.43%	111	115	3.60%

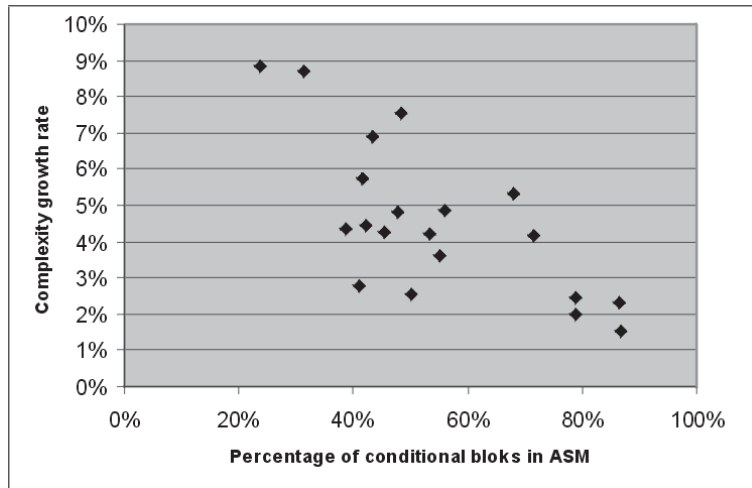


Fig. 5. Scatter graph for complexity growth rate in relation to percentage of conditional blocks

Table 4. Comparison of synthesis methods for realization on MAX9000 device family

ASM	L	N	S	K	C_{M1}	C_{M2}	C_{M3}	P_{M2}	P_{M3}
alf	31	74	127	50	146	153	166	4.79%	13.70%
araf	25	65	124	48	140	152	164	8.57%	17.14%
ass13	5	25	38	19	54	55	61	1.85%	12.96%
bech	18	39	68	33	69	73	74	5.80%	7.25%
berg	21	51	121	51	133	138	171	3.76%	28.57%
big	18	28	127	110	182	183	189	0.55%	3.85%
bs	19	13	127	110	165	168	184	1.82%	11.52%
e1	12	13	114	90	166	167	196	0.60%	18.07%
e15	13	20	85	67	110	111	125	0.91%	13.64%
klain	27	61	134	55	158	163	184	3.16%	16.46%
kobz	19	53	130	59	130	138	163	6.15%	25.38%
lcu	15	24	81	58	93	95	130	2.15%	39.78%
lift	14	30	42	10	50	51	53	2.00%	6.00%
max	26	41	105	56	128	130	151	1.56%	17.97%
micks	21	45	106	53	114	121	138	6.14%	21.05%
pp	20	28	89	72	120	121	137	0.83%	14.17%
pilot	27	22	60	33	62	67	82	8.06%	32.26%
sasi	19	54	129	54	136	144	175	5.88%	28.68%
oshr	19	72	144	53	150	154	174	2.67%	16.00%
e8	13	20	85	67	107	108	125	0.93%	16.82%
bcomp	18	39	68	33	73	77	86	5.48%	17.81%
asm1	15	22	32	19	38	39	47	2.63%	23.68%
asm2	15	22	31	17	40	41	44	2.50%	10.00%
Average								3.43%	17.95%

References

- [1] Baranov S., Sklarov V., Digital systems based on programmable circuits with matrix structure, Moscow: Radio i sviaz, 1986 (in Russian).
- [2] Baranov S., Logic Synthesis for Control Automata, Kluwer Academic Publishers, 1994.
- [3] Baranov S., Logic and System Design of Digital Systems, Tallinn: TTU Press and SiB Publishers, 2008.
- [4] Baranov S., High level synthesis in EDA tool "Abelite", Electronics and Telecommunications Quarterly, 2009, Vol.55, No.2, pp. 123-156.
- [5] Barkalov A., Titarenko L., Logic synthesis for compositional microprogram control units, Berlin: Springer-Verlag, 2008.
- [6] Salauyou V., Klimowicz A., Logical synthesis of digital devices in PLD structures, Bialystok: OWPB, 2010 (in Polish).
- [7] Wilkes M.V., The Genesis of Microprogramming, IEEE Annals of the History of Computing, 1986, V.8, No.2, pp.116-126.

PROJEKTOWANIE PSEUDOEKWIWALENTNYCH AUTOMATÓW MIKROPROGRAMOWALNYCH NA UKŁADACH PLD

Streszczenie Metody syntezy automatów mikroprogramowalnych oparte na wprowadzeniu dodatkowych stanów wewnętrznych prowadzą do otrzymania automatów pseudoekwiwalentnych. Sekwencja słów wyjściowych takich automatów naruszana jest pojawieniem się zerowych słów wyjściowych w stanach dodatkowych, co nie zawsze jest dopuszczalne w zastosowaniach praktycznych. W artykule została przedstawiona nowa metoda syntezy automatów mikroprogramowalnych, która pozwala przekształcić automat pseudoekwiwalentny na postać ekwiwalentną. Zaproponowana została zmodyfikowana struktura automatu mikroprogramowalnego, w której zmiana sygnałów wyjściowych jest możliwa wyłącznie w stanach podstawowych, tym samym eliminuje się słowa zerowe na wyjściach automatu. Badania eksperymentalne pokazały, że złożoność realizacji zaproponowanej struktury na układach programowalnych wzrasta w nieznacznym stopniu, natomiast takie podejście pozwala znacznie rozszerzyć obszar zastosowania metod syntezy automatów mikroprogramowalnych opartych na wprowadzeniu dodatkowych stanów wewnętrznych.

Słowa kluczowe: automat mikroprogramowalny, sieć działań, automat pseudoekwiwalentny, dodatkowe stany wewnętrzne, programowalne układy logiczne.

LOGOTYPE DETECTION FOR CHILD LOCK ON INTERNET TELEVISION

Marta Chodyka¹, Włodzimierz Mosorow²

¹ Institute of Computer Science, Pope John Paul II University in Biala Podlaska, Poland

² Computer Engineering Department, Technical University of Lodz, Poland

Abstract: Presently, Internet offers to all users easy and constant access to TV programmes through the Internet TV. These programmes are not always appropriate for all users (eg children) on account of presented content. There are diverse methods of TV programmes blocking in order to check TV programmes content broadcasted through the Internet. However, the problem of automatic blocking is not solved. It does not take a note of the method that consist in verification of the program mes through the identification of the image broadcasted from the video stream. The paper presents a method invented by the authors of the paper based on automatic identification of the provider's logo. The programme's provider reconnaissance will be realized on-line through the automatic identification of the static logo object together with the programme in a sequence of video images. The automatic identification of the provider's logo allows to block access to TV broadcast of the selected transmissions according to the transmission schedule. This method performs a temporal and spatial segmentation of the logo. In order to extract the regions of the logo's contours the Sobel operator is applied. Next, the averaged binerization of the image is obtained through the Otsu method, which identifies its threshold. The vector used in comparison process is calculated through the projection method. The findings received in this work confirm the effectiveness of that method. The method has been tested on transmissions available in the Internet TV. It allows to achieve over 98,7% correct results of the Internet TV programmes blocking on-line.

Keywords: logotype identyfication, child lock, contour image

1. Introduction

The problem of underage persons' easy access to the multimedia video with inappropriate content and its consequences is well known [3,15]. One of the sources enabling the access to such video programmes is the widely available Internet TV.

There are numerous methods which are used to control the content of the television programmes transmitted via the Internet. These include, among others, blocking video materials at certain hours [19-21] or filtering chosen IP addresses and keywords on web pages [18]. There are also parental control modules, which can be embedded in the anti-virus software, web browsers and operation systems. All these well-known methods do not, however, solve the above mentioned problem completely. Thus, in the case of a temporary access block on Internet TV, parents must be involved in the process of programme assessment and selection. With regard to IP address filtering, the problem concerns a rapidly growing number of keywords, which the filter should block, as well as easily made changes of the IP addresses by Internet providers.

Another solution is to do an analysis of the provider's logo transmitted together with the video stream. In a video production, logos are used to convey information about the provider's programme content, which can be used in the selection of age-appropriate programmes while broadcasting video. There are related applications which try to identify brand logotypes in video data [5, 6, 12, 16] by using the static character of the logo. In order to identify the logo, some logo detection algorithms use neural network and image analysis procedures [1,7,8,10,17]. However, the selection of an adequate neural network's models, their over-fitting capacity and the high computational cost of the methods limit their applications in practice.

The logo identification in the programme categorisation is presented by Cozar et al. 2007 [4]. This method performs a temporal and spatial segmentation by calculating the minimal luminance variance region of the set of frames and the non-linear diffusion filtering. However, 95% of correct identification has been achieved only when the analysis is conducted on-line. A different solution is presented by Ozay, Sankur [14]. This time, an algorithm performs a detection of the logo by morphological operations. Nevertheless, online tests for detection and recognition on running videos have achieved lower than 96% average accuracy. In [2] logo detection techniques have been used to differentiate advertisements from TV programmes. This approach assumes that a logo exists if a region with stable contours can be found in the image. No temporal information is used and the method has not been tested on video material in a real time transmission, which has resulted in many false detection cases.

Contrasting the aforementioned methods, the paper presents a more effective method for automatic identification of the provider's logo based on an original image of sequence analysis. The automatic identification of the provider's logo allows to block access to video programmes of the selected providers. It takes place regardless of the transmission time, IP address or the keywords used to find a required website.

The method has been tested on some transmitted video, achieving over 98,7% of correct identification.

The article consists of several parts. Section 2 contains a description of the logo detection algorithms based on spatial segmentation. It additionally presents the logo identification and its comparison with logo patterns. Section 3 concentrates on testing the presented method on chosen video streams and illustrating the results of its application. The article ends with Section 4, which includes main conclusions and presents plans for the further development of the above method.

2. Algorithm description

The video streaming Internet TV programme is a set of ordered frames through time. These frames can include one or several superimposed logos. Usually, a logo is defined as a small graphic or picture that appears behind the anchor person on the screen. Logo image areas show luminance variance values in narrower interval than other image areas, depending on the logo transparency. An important feature of a logo image is that the logo contours are stable, while the background varies during video broadcasting. Besides, during video broadcasting a logo can be present or absent, for instance during an interruption of the programme transmission. Logotypes are usually placed at any of the four corners of a frame. Therefore, four image corners should be considered as the regions of interest (ROIs). Moreover, their size is limited, since logos should not perturb video viewing (see Figure 1).

Furthermore, logo areas do not significantly change from frame to frame. A logo is a characteristic feature of any programme provider as well as its contents. Logo identification enables verification of various programmes providers, which makes it a tool of parental control, enabling blocking unsuitable programmes for underage viewers. A child's parent or guardian chooses logo patterns from a providers' base which are regarded as inappropriate for children. When a programme transmission takes place, its logo is identified and compared with the ones selected as unwelcome by the parent or guardian. Depending on the received information, the video signal is either blocked or allowed to flow.

Figure 2 illustrates a scheme of the proposed programme blocking system transmitted on-line.

When the logo of the transmitted on-line programme is not included in the data base, the system can add this new candidate logo to the logo patterns base. The new candidate undergoes a process of segmentation, yet it is not included in the currently transmitted logo identification process. Automatic logo adding to the logo data base can take place after it has been projected and recognised several times.

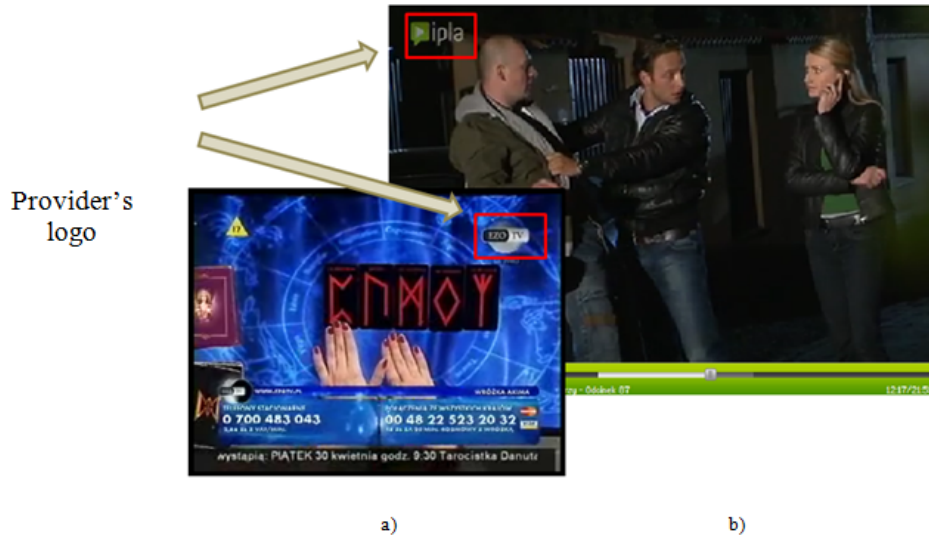


Fig. 1. Examples of the frames from broadcasting Internet video EZO (a) and IPLA (b) with the selected region of the provider's logo

Let a mathematical model of a logo image be a matrix, $\mathbf{I} = I(i, j)$, $i = 1..m$, $j = 1..n$, where m and n define the size of the logo image. Initially, the digital image \mathbf{I} of the analyzed logo region is converted to the monochrome image \mathbf{I}' . This operation includes the calculation of the brightness $I'(i, j)$, $0 \leq I'(i, j) \leq 255$, for each pixel of the RGB colour components $I'(i, j)$.

To extract contours of the logo regions of the monochromatic image \mathbf{I}' , the Sobel operator [9] is applied. Due to this operation an image of the logo contours is created \mathbf{I}^* . However, the extracted contours of the logo regions are often not salient because the result of the extraction depends considerably on the time variable background where logos appear. In order to achieve better quality of the contours, the adopted method averages the sequence of the logo contours \mathbf{I}^* :

$$\bar{I}(i, j) = \frac{1}{K} \sum_{k=1}^K I_k^*(i, j), \quad i = 1..m, \quad j = 1..n \quad (1)$$

where $I_k^*(i, j)$ is the k th logo contours image and K - is the number of images \mathbf{I}^* . As a result of these stages, the average image of the logo's contours $\bar{\mathbf{I}}$ - is created. It seems clear that in the sequence (see eq. 1) the number of the processed frames K depends mostly on the characteristics of the video stream. Thus, a video with a dynamic se-

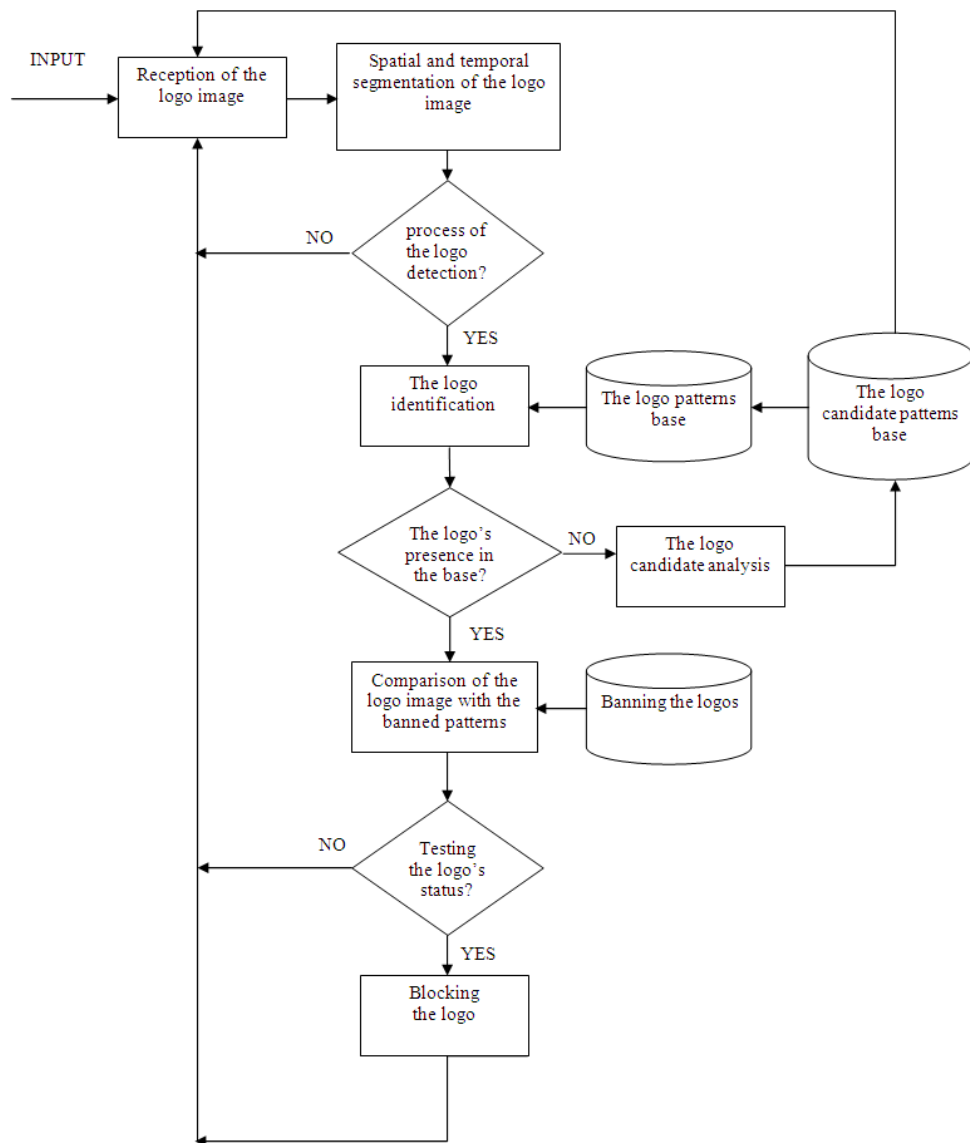


Fig. 2. An algorithm scheme for blocking programmes

quence of frames needs fewer frames to generate stable logo contours than the one with a static sequence. Therefore, K value should be chosen experimentally. It seems plausible that a large value K can guarantee better detection for logos which are static

for a long period of time. However, a wrong logo contour image is obtained if logo changes occur within the K frames. In this case, the number of frames K used for the logo extraction must be decreased.

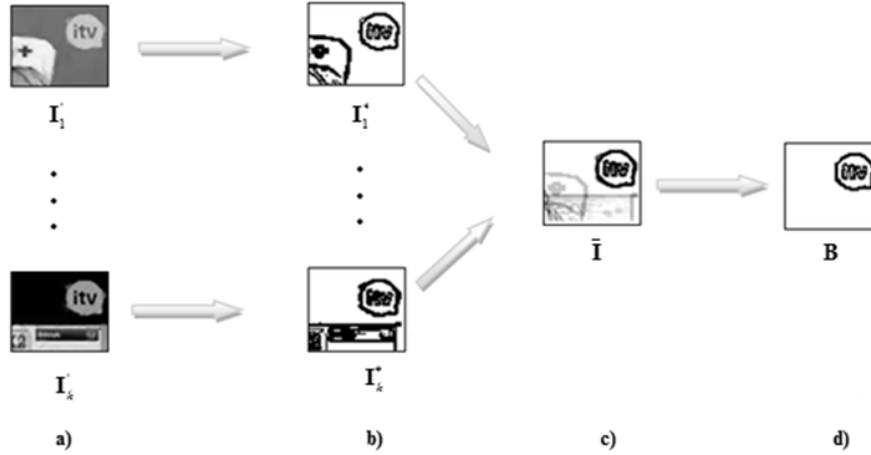


Fig. 3. Images obtained in each stage of the algorithm for a real sequence through time: logo monochrome image \mathbf{I} (a), logo contour image \mathbf{I}^* (b), average logo contour image $\bar{\mathbf{I}}$ (c), binary logo image \mathbf{B} (d)

In the next stage, a spatial segmentation of logo contours is conducted binarizing of \mathbf{I}^* :

$$\mathbf{B} = B(i, j) = \begin{cases} 0 & \text{for } \bar{I}(i, j) \geq p_1 \\ 1 & \text{for } \bar{I}(i, j) < p_1 \end{cases} \quad i = 1..m, j = 1..n \quad (2)$$

where \mathbf{B} is the binary image of the logo contours and the threshold level p_1 which are arbitrarily determined from a histogram. An appropriate choice of the threshold level p_1 is the basis of a proper process of identifying the logo contours from the image. In order to calculate the required level, average histograms of the logo contours are determined $\bar{\mathbf{I}}$, which, due to different backgrounds, vary considerably (see fig. 4).

The optimum level p_1 is calculated by means of the Otsu [13] method according to formula 3.

$$p_1 = \arg \max_p (\omega_0 \omega_1 (\mu_1 - \mu_0)^2) \quad (3)$$

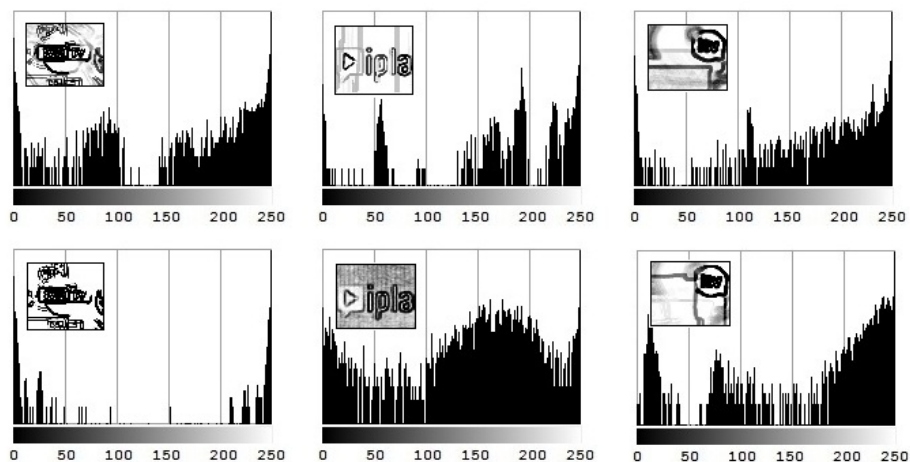


Fig. 4. Images of averaged logo contours \bar{I} and their respective histograms

where ω_0 - constitutes a standardised quantity of the logo contours (a quotient of the number of points belonging to the contours and the number of the image points), ω_1 is the standardised number of the background quality, μ_0 and μ_1 are the averaged qualities of the points brightness for the contours and background respectively, $0 \leq p \leq 255$.

Figure 5 presents example of an image and histogram before and after the application of the Otsu method.

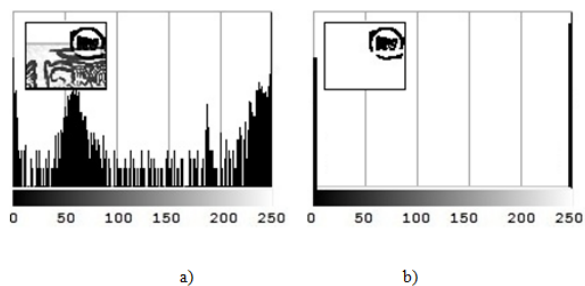


Fig. 5. An image and its histogram before (a) and after the application of the Otsu binarisation method (b)

Generally, there are cases when the analysed video stream does not comprise any logo, for instance, during commercial breaks. To recognise such a case the following procedure of logo histogram analysis is proposed. Examples of images without logo \bar{I} and their respective histograms are presented in figure 6. The next step includes cal-

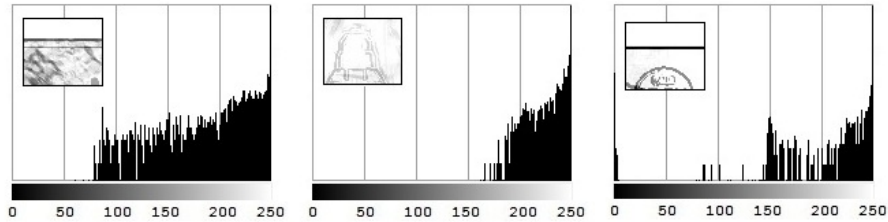


Fig. 6. Some images without logo \bar{I} and their respective histograms

culating sums S_1 and S_2 how often grey scale values $h(p)$ larger than $\frac{h_{max}}{2}$ appear into the two ranges $\langle 0..p_1 \rangle$ and $\langle p_1..255 \rangle$ respectively, where h_{max} indicates the maximum of a histogram. If $S_1 \leq S_2$, it may be inferred that the logo is not included in the image. A graphic representation of the idea is shown in figure 7. When images un-

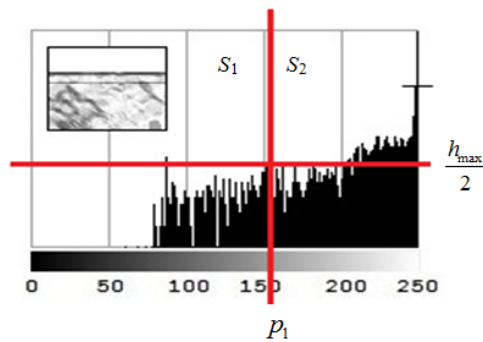


Fig. 7. An image without a logo and its histogram with a graphic presentation of calculating sums S_1 and S_2

dergo the analysis process, two kinds of errors may take place. The first one concerns a situation when the logo is present but has not been identified by algorithm. This

happens when algorithm reads incomplete logo contours, i.e. when it identifies light contours in a light background. The case is illustrated by figure 8. The other error

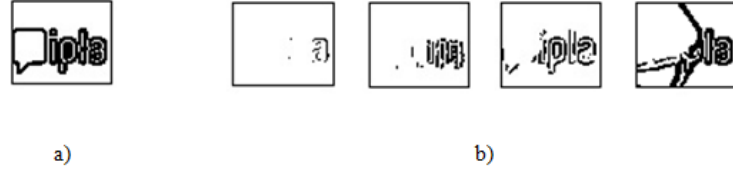


Fig. 8. Examples of binary image contours of the logo **B** presenting the logo of IPLA provider in real time sequences: full logo contours (a), and incomplete logo contours (b)

connected with the logo identification may take place when the logo is not present and algorithm identifies static contours of an object as the logo, and subsequently adds the identified contours to the data base as a new pattern. Some examples concerning such situations are presented in figure 9. The above situations may take place



Fig. 9. Examples of binary logo **B** contours identified inappropriately as potential candidates for new logos

due to the nature of the discussed problem. Proper recognition of such cases by algorithm is, however, difficult. To identify the logo, it is first of all necessary to define the logotype database as a set of the logo patterns representing different broadcast providers of the Internet TV programmes. Let $\{\mathbf{B}_r^z\}$, $r = 1..R$, be the reference set of the R logo patterns. Each pattern \mathbf{B}_r^z is obtained by the same procedure as the one described above, when the background is stable. A good descriptor of the binary image **B** of the logo contours is the shape itself, but a long feature vector would be created. An important reduction of the feature vector size, without a great loss of accuracy, can be achieved if the x -axis and y -axis shape projections are used. Let

$$w_i = \sum_{j=1}^n B(i, j), \quad i = 1..m, \quad k_j = \sum_{i=1}^m B(i, j), \quad j = 1..n$$

mean x -axis and y -axis shape projections of a binary image \mathbf{B} of the logo contours. Then, a good metric to compare the feature vectors $[\mathbf{w}, \mathbf{k}]$ and $[\mathbf{w}^z, \mathbf{k}^z]$ of \mathbf{B} and \mathbf{B}_r^z respectively is the distance given by the following expression:

$$\min_r \left(\sum_{i=1}^n |w_i^l - w_{r,i}^z| + \sum_{j=1}^m |k_j^l - k_{r,j}^z| \right), \quad r = 1..R \quad (4)$$

Algorithm enables an automatic supplementation of the pattern data base. A candidate analysis of a new pattern is conducted according to of the rank of correlative factors τ Kendalla [10] between the analysed image and patterns. The method enables qualifying if the logo included in the transmitted programme exists in the data base or whether it should be added as a potential candidate.

3. Method verification

"StopPlay", a novel application shown in Figure 3, has been written in the C# language. The application analyses a video stream of the selected Internet television programmes in on-line regime. In order to verify the correctness of the algorithm in the process of the logo recognition, a set of six patterns of the logo $\{ \mathbf{B}_r^z \}$, $r = 1..6$ (see Figure 10) of popular Internet televisions was defined. The Internet addresses of Internet television programmes used in the tests include Inter Alia: <http://www.itv.net.pl>, <http://www.ipla.pl>. The logo images of dimensions 60x50 pixels are automatically



Fig. 10. Set of chosen logotypes (a) main application window (b)

extracted from each frame in the video stream during the transmission of Internet television programme. The number of binary images needed to create an average contour image was set at $K=40$. As it is argued in Section 2, in order to rid the programme of disturbances and get clear contours of the logo in its background, qualities p_1 cannot be taken arbitrarily. Figure 11 presents examples of averaged contours of the logo and their respective binary images and histograms. The threshold levels p_1 are chosen according to the Otsu method and depend on the levels of grey shades in the image.

The use of such values allows to achieve approximately 99% of correct identification in the logo detection procedure. The only activity left for the user is to choose the

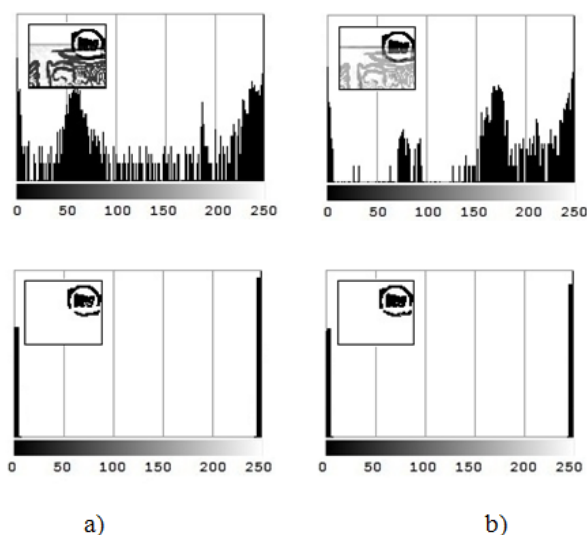


Fig. 11. Averaged images of the logo contours and their respective binary images with appropriate quality levels $p_1 = 35$ (a), $p_2 = 100$ (b) presented according to the Otsu method and their histograms

names of the provider (providers), whose logo should be recognised from a particular set of programmes. Figure 12 presents an analysis of the tested logos of the television programmes. The tests were conducted on an average of 20 000 video frames during approximately three hours' time on the three available TV sites: ITV, EZO, IPLA. The algorithm was tested during the TV programme transmission as well as during commercial breaks. The obtained results show that the presented algorithm detects the logo with an accuracy of over 99%.

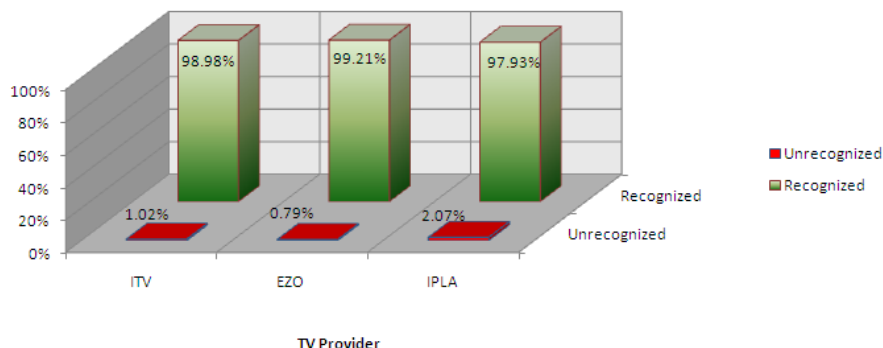


Fig. 12. Percentage chart for positive recognised logos in the selected broadcasting Internet video

The lack of proper recognition of the logo is due to cases when the logo and the background are in the same colours, i.e. without visible logo's contours, as well as cases when some permanent objects are present in the logo. When recording consecutive frames of video sequences these additional objects become regions identified in the algorithm as a logo. Figure 13 presents such tested logo patterns and examples of images of the logo contours I^* recognised (a) and unrecognised (b).

4. Conclusion

This article presents the logotype recognition algorithm and its application in the television programme providers in the on-line regime. The suggested method takes advantage of a multi-step segmentation of temporal and spacious logo, which enables detecting the image contours and eliminating the background objects from the on-line video images. A comparison of the achieved images of the logo with the patterns allows an automatic identification of the transmitted programme. The identification process takes into account situations when the logo is not present due to, for instance, an interruption of the transmission process. It has been proved that the implemented algorithm is capable of detecting images of the logo with an accuracy of over 98,7%. The cases which are problematic are due to situations when the logo and background images are in the same colours and when permanent objects appear in the logo region. Under such circumstances the algorithm identifies the entire regions as logos. However, in contrast to many other object recognition algorithms, the proposed algorithm does not require preparation of any learning set or application of any advanced methods for image processing. This allows for its practical and easy


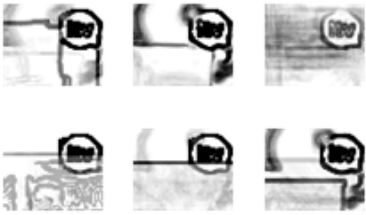







Logo patterns $\{B_r^i\}$	Images of the logo contours I^*	
	Recognised (a)	Unrecognised (b)
<p>ITV</p> 		
<p>EZO</p> 		
<p>IPLA</p> 		

Fig. 13. Examples of patterns $\{B_r^i\}$, and images of the logo contours I^* : recognised (a) and unrecognised (b)

use in the application of automatic identification of television programmes and minimises the potential negative effects of Internet television on children. Further studies will include refinement of the algorithm and propose solutions to these recognition problems which have not been identified or detected by the algorithm.

References

[1] Acharya T., Ray A.K., Image Processing: Principles and Applications, John Wiley, pp. 428, 2005.

- [2] Albiol A., Fulla M.J., Albiol A., Torres L., Detection of TV commercials, Proceedings of the International Conference on Acoustics, Speech and Signal Processing, vol. 3, pp. 541-544, 2004.
- [3] Bryant A., The Children's Television Community, Lawrence Erlbaum Associates, Mahwah, New Jersey, 2007.
- [4] Cozar J.R., Guil N., Gonzalez-Linares J.M., Zapata E.L., Izquierdo E., Logo-type detection to support semantic-based video annotation, Signal Processing: Image Communication 22, Elsevier B.V, pp. 669-679, 2007.
- [5] Duffner S., Garcia C., A neural scheme for robust detection of transparent logos in TV programs, Lecture Notes in Computer Science - II, vol. 4132, Springer, Berlin, pp. 14-23, 2006.
- [6] Ekin A., Braspenning R., Spatial detection of TV channel logos as outliners from the content, Proceedings of SPIE - The International Society for Optical Engineering, vol. 6077, 2006.
- [7] Gonzalez R.C., Woods R.E., Digital Image Processing, Prentice Hall, Upper Saddle River, New Jersey, 2001.
- [8] Handbook of Pattern Recognition and Computer Vision 4th Edition edited by C H Chen (University of Massachusetts Dartmouth, USA), 2009.
- [9] Jähne B., Digital Image Processing, Springer, Berlin, Heidelberg, New York, 2002.
- [10] Kendall, M.G., Rank Correlation Methods Edition 1. London: Charles Griffin, 1948.
- [11] Kim H., Loh W.Y., Classification Trees With Unbiased Multiway Splits, Journal of the American Statistical Association, pp. 598-604, 2001.
- [12] Meisinger, K., Troeger T., Zeller M., Kaup A., Automatic TV Logo Removal Using Statistical Based Logo Detection and Frequency Selective Inpainting, Proc. European Signal Processing Conference, 2005.
- [13] Otsu N., A threshold selection method from grey-level histograms, IEEE Trans. System Man Cybernet, 9(1), pp. 62-69, 1979.
- [14] Ozay N., Sankur B., Automatic TV logo detection and classification in broadcast videos, 17th European Signal Processing Conference (EUSIPCO 2009), Glasgow, Scotland, pp. 839- 843, 2009.
- [15] Kirsh S.J., Children, adolescents, and media violence: a critical look at the research, Sage, California, 2006.
- [16] Yan W.Q., Wang J., Kankanhalli M.S., Automatic video logo detection and removal, Multimedia Systems 10(5), pp. 379-391, 2005.
- [17] Young I.T., Gerbrands J.J., van Vliet L.J., Fundamentals of Image Processing, The Netherlands at the Delft University of Technology, 1998.

- [18] Patent: System and Method for Subscriber Controlled Signal Blocking, no CA2266982, 1999.
- [19] Patent: Method for child lock in Internet Television, no KR20010037415, 2001.
- [20] Patent: Method and apparatus for permitting a potential viewer to view a desired program, noUS2004015985, 2004.
- [21] Patent: Content control system, no US2005028191, 2005.

DETEKCJA LOGO JAKO NOWA METODA BLOKOWANIA NIEODPOWIEDNICH DLA DZIECI TRANSMISJI W TELEWIZJI INTERNETOWEJ

Streszczenie W obecnych czasach Internet oferuje wszystkim swoim użytkownikom łatwy i stały dostęp do programów telewizyjnych dzięki telewizji internetowej. Z uwagi na prezentowane treści, programy te nie zawsze są odpowiednie dla wszystkich użytkowników (np. dzieci). Istnieje wiele metod, które są używane do sprawdzania zawartości programów przekazywanych w programach telewizyjnych. Jednakże problem automatycznego blokowania programów na podstawie treści nie jest całkowicie rozwiązany. Nie istnieją metody polegające na sprawdzaniu programu poprzez automatyczną identyfikację obrazu logo ze strumienia wideo. W artykule przedstawiono autorską metodę polegającą na automatycznej identyfikacji logo nadawcy programu. Rozpoznawanie logo nadawcy będzie realizowane on-line poprzez identyfikację statycznego obiektu logo emitowanego wraz z programem w sekwencji obrazów wideo. Automatyczna identyfikacja nadawcy programu pozwoli na zablokowanie dostępu do wybranych transmisji telewizyjnych konkretnych nadawców. Metoda wykorzystuje czasowo - przestrzenną segmentację logo. W celu wyodrębnienia regionów konturów logo, stosowany jest operator Sobela, a następnie binaryzacja uśrednionego obrazu z progiem wyznaczonym metodą Otsu. Wyznaczanie zaś wektora do porównań wyznaczone jest metodą projekcji. Otrzymane w pracy wyniki potwierdzają skuteczność metody. Metoda została przetestowana na wybranych programach telewizji internetowej, osiągając ponad 98,7% poprawnych rezultatów blokowania programów telewizji internetowej on-line.

Słowa kluczowe: identyfikacja logo, kontrola rodzicielska, detekcja obrazu

FORECASTING STOCK INDEX MOVEMENT DIRECTION WITH *CPL* LINEAR CLASSIFIER

Jerzy Krawczuk

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

Abstract: Stocks, indexes, commodities, and precious metals price prediction is a difficult task where many approaches are used: traditional technical analysis, econometric time series or modern data mining techniques. One particular data mining technique - linear classifier is described in this article. Prediction based on linear classifier is done using current market state, which can be described by various data sets (attributes, features). The simplest form of this model could use data from yesterday's price movement. Advanced models are using more historical price movements. Very advanced models include various historical price movements for indexes from other countries and other instruments like currencies, commodities, etc. Using more features requires extended time to estimate model parameters. We build the linear classifier models by the minimisation of a convex and piecewise-linear function which is very efficient comparing to other functions. Computational costs for building the model are similar to linear programming. We also use feature selection method called RLS. Those techniques allow us to explore data with many features. Four scenarios are considered, in each scenario a different amount of market data is used to create a model. In the simplest scenario only one day's change in price is taken, in the most complicated one 421 historical prices of 43 different instruments are taken. Best results were achieved by using middle range of 52 attributes. In this scenario, the model was right 53.19% times. Meaning the directions of daily change in S&P500 index (up or down) were predicted correctly. This doesn't seem a lot, but if those predictions would have been used for investing, they could produce a total profit of 77% in the tested time period from November 2008 to March 2011 (2 years 4 months), or an average of 28% per year.

Keywords: market forecast, market prediction, linear classifier, convex and piecewise-linear function

1. Introduction

The first well known book written on analysis of the stock market was "Confusion of Confusions (1688)" by Joseph de la Vega, who described the way the Amsterdam

Stock Exchange worked and gave some hints for price analysis. In Asia during early 18th century, Homma Munehisa described the basics of candlestick techniques [20], which are nowadays a popular charting tool. Price chart analysis techniques, otherwise known as technical analysis [9], are very popular, and are used by traders on a daily basis. It focuses on searching for repeatable patterns in price charts, and on looking at some statistical indicators.

A more modern approach of describing the behaviour of market prices is known as econometric time series analysis [12]. The classic models assumes that current value of price is correlated with previous values (prices are autocorrelated). A stock's price is described as a linear equation of it is own previous values, such a model is called autoregressive. Box and Jenkins [6] describe the methodology to best fit such a model to data for a purpose of forecast. Different class of time series models are used for predicting not the expected value of process but the standard deviation of prediction. Main groups of such models are *ARCH* [11] and *GARCH* [5] also known as heteroskedasticity models. Those models play important role in risk analysis. Robert Engle work on time-varying volatility was awarded with Nobel price in 2003.

Data mining techniques that are developing quickly in recent years, are also being used for predicting market prices. They can be divided into models inspired by nature (neural networks [2], genetic algorithm [21]) and linear models (*SVM* [7], *CPL* [3]). In this article, linear models are presented with more detail, and we explain in an experiment how one such model can be used for market prediction, in this case the next day's move of S&P500 index. The model is built from one year of historical data, and then it is used to make predictions over the next half year. After each half year the model is rebuilt. Results are concluded at the end, and directions for future research are discussed.

2. Data mining techniques inspired by nature

2.1 Neural network

In general, a neural network or artificial neural network is a computer model whose architecture is inspired by human brain. It is build from elements called artificial neurons that process the information in a basic way. It transforms many input signals (real numbers) into one output number. For example if transformation is linear, and output is equal either 0 or 1, depending on some threshold, such *NN* is called a *perceptron* and neuron is a *linear classifier*. The learning power of *NN* lies in hierarchical structure of neurons which is called network. They can model complex non-linear relationships between inputs and outputs.

Nowadays neural networks (ANNs) have been popularly applied to finance problems including stock index prediction. Tokyo stock index was predicted by Kimoto [16] and Mizuno [19], Istanbul Stock Index by Egeli [10]. Other authors use different network architecture (topology), different methods of training and testing. We can find summary of such approaches in Zekic [24]. Authors claims that neural networks gives better results then buy-and-hold strategy.

2.2 Genetic algorithm *GA*, genetic programming *GP*

This group of methods is using an analogy of evolution processes in order to solve optimisation or search problem. With a help of evolution processes it transforms a set of population (mathematical objects *GA* or computer programmes *GP*) into a new population. Two key mechanism of biological evolution must be mapped in such transformation. First is a natural selection mechanism: those individuals from a population who can solve problem in most efficient way should have bigger chances to survive and reproduce in their environment. The second mechanism is a genetic drift which allow random changes in new population. Individuals are defined by they chromosomes. Transformations between populations is done through changes in chromosomes using genetic operations like inheritance, mutation, selection and crossover. Each population is called a generation. Usually the algorithm stops when either a maximum number of generations has been produced, or a best individual has a satisfactory fitness level.

Since the evolutionary algorithm is a general approach for solving optimisation problem, it can be used in many different ways for purpose of predicting the market. Three most common approaches are:

- finding optimal parameters for a model, usually using the technical analysis,
- feature selection usually with neural network,
- discover trading rules.

First approach is very common in the case of genetic algorithms. Very popular trading platform called MetaTrader is using the genetic algorithm for the purpose of finding optimal parameters for tested investing strategies. User can define a strategy in MQL4 programming language and he can choose from many built-in technical analysis indicators. Such strategy always have some parameters that need to be set before we can start using them. Strategy can be executed on historical data with different values for different parameters. Built-in strategy tester allow for many executions with different parameters. It can do so by Simple Search (searching whole parameters

space) or Genetic Algorithm. Some users claimed that with Genetic Algorithms it is possible to find solutions much faster than using other algorithms [14].

Second way of usage - the feature selection is also popular. If we want to describe the market with many features and use for example neural network to build a model, as a first step we may want to select only relevant features. But searching for all $2^N - 1$ subspaces is not possible for too many features N . Many authors argue that genetic algorithms can find very good subspace with reasonable time [23].

Third approach is to use genetic programming, where individuals in populations are represented by computer programmes. Each programme represents simple open or close price, or mathematical operator like add or technical analysis indicators like moving average [21]. From such elements the genetic algorithm is trying to build an optimal formula for the trading signal.

3. Linear models

Bobrowski [3] define linear classifier $LC(w[n], \theta)$ as decision rule:

$$LC(w[n], \theta) = \begin{cases} \text{if } w[n]^T x[n] \geq \theta, \text{ then } x[n] \text{ is located in class } \omega^+ \\ \text{if } w[n]^T x[n] < \theta, \text{ then } x[n] \text{ is located in class } \omega^- \end{cases} \quad (1)$$

where $w[n] = w[w_1, w_2, \dots, w_n]^T$ is a vector of weights $w_i \in R^1$ and θ is a threshold ($\theta \in R^1$). The creation of predictive rules (1) requires the calculation of the parameter values $w[n]$ and θ . Those parameters can be determined on the basis of learning sets G^+ and G^- containing examples of feature vectors $x_j[n]$ from class ω^+ ($j \in J^+$) and from class ω^- ($j \in J^-$).

$$G^+ = \{x_j[n] : j \in J^+\} \text{ and } G^- = \{x_j[n] : j \in J^-\} \quad (2)$$

In practice it is not always possible to find such $w[n]$ and θ where all feature vectors $x[n]$ are correctly classified. This is not always desirable as well, because of the danger of over-fitting to the data sets (2). The optimal parameters $w^*[n]$ and θ^* of the classification rule (1) can be determined in many ways. *SVM* and *CPL* approaches are introduced in this paper.

3.1 Support vector machine - SVM

SVM approach [7] is to find such optimal parameters $w^*[n]$ and θ^* that represents the largest separation, or margin, between objects from sets G^+ and G^- (2). Such margin could be represented as a two parallel hyperplanes (fig.1) that are in maximum

distance but still separating sets G^+ and G^- . They can be written as $wx - \theta = 1$ and $wx - \theta = -1$. It can be proven that a distance between these two hyperplanes is $2/||w||$. Optimisation problem could be defined as finding such optimal parameters $w^*[n]$ and θ^* that minimize $||w||$ and satisfy below constraints:

$$\begin{cases} w[n]^T x_i[n] - \theta \geq 1, \text{ for } x_i \in G^+ \\ w[n]^T x_i[n] - \theta < -1, \text{ for } x_i \in G^- \end{cases} \quad (3)$$

This problem can be solved by standard quadratic programming techniques. *SVM* approach described above has many extensions. Two main extensions are:

- soft margin: allow for misclassification of the data [7],
- non-linear classification: applying the kernel trick [1].

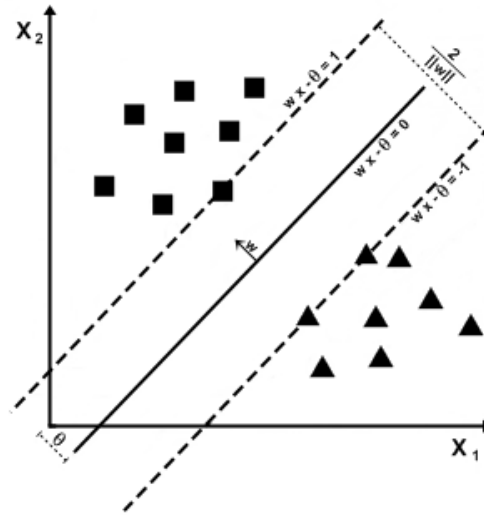


Fig. 1. Optimal hyperplane w constructed with two support vectors on the margin of sets G^+ and G^- . Source: own elaboration

One of the earliest studies on financial forecasting using support vector machines could be found in [15]. Author compared *SVMs* with other popular data mining techniques, including case-based reasoning and backpropagation neural networks. *SVM* outperformed other methods in prediction of future direction of Korea stock index. Results were in range of 50%-57%, depends on used parameters. In other study [13]

author predicted a weeks direction of change for a Japanese index NIKKEI with as high as 73% hit ratio for *SVM*.

3.2 Convex and piecewise-linear penalty function - CPL

Other way of finding optimal parameters $w^*[n]$ and θ^* of the classification rule (1) is proposed by Bobrowski [3] [17]. He define a convex and piecewise-linear (*CPL*) criterion functions in the below manner:

$$\phi_j^+(w[n], \theta) = \begin{cases} \theta + 1 - w[n]^T x_j[n] & \text{if } w[n]^T x_j[n] < \theta + 1 \\ 0 & \text{if } w[n]^T x_j[n] \geq \theta + 1 \end{cases} \quad (4)$$

$$\phi_j^-(w[n], \theta) = \begin{cases} \theta - 1 + w[n]^T x_j[n] & \text{if } w[n]^T x_j[n] > \theta - 1 \\ 0 & \text{if } w[n]^T x_j[n] \leq \theta - 1 \end{cases} \quad (5)$$

And the perceptron criterion function $F(w[n], q)$ as the weighted sum of the penalty functions (4) and (5):

$$\Phi(w[n], \theta) = \sum_{j \in J^+} \alpha_j \phi_j^+(w[n], \theta) + \sum_{j \in J^-} \alpha_j \phi_j^-(w[n], \theta) \quad (6)$$

where non-negative parameters α_j represent prices linked to particular feature vectors $x_j[n]$. The minimization of the criterion function $\Phi(w[n], q)$ (6) allow us to find the optimal parameters $w[n]^*$ and q^* of the prediction rule (1).

3.3 Feature selection method *RLS* (Relaxed Linear Separability)

It is *CPL embedded* feature selection method. Relaxed linear separability (*RLS*) methodology was introduced in [4][17]. It is defined by additional costs γ related to particular features x_i added to the penalty function (6) :

$$\Psi(w[n], \theta) = \Phi(w[n], \theta) + \lambda \sum_{i \in I} \gamma_i |w_i| \quad (7)$$

where λ is the cost level, and $I = \{1, \dots, n\}$. In accordance with the *RLS* method, a gradual increase of the cost level λ value in the criterion function (7) allows successive reduction of features x_i . In the result a descended sequence of feature subspaces can be generated. The quality of each subspace is measured and best subspace is selected. Quality measure is a classification accuracy calculated by the leave-one-out methodology. Each feature vector $x_j[n]$ is classified by the linear classifier (1) build on all other vectors except one which is classified. This method allow to reduce bias of the classifier accuracy estimation.

4. Experiment

The *CPL* linear classifier with *RLS* feature selection method was used in the experiment. One day move of S&P500 US stocks index was predicted. In this approach we do not predict the exact tomorrow's value for the index, we predict direction of change (either the index will move down or up). Forecast task is defined as a classification approach [8] not a regression.

Daily market data (open and close prices) was used, with data set starting from November 2007 till end of February 2011. Each day was described by the feature vector $x[n]$ and class ω . Vector $x[n]$ could be assigned to one of two classes. To class ω^+ if in the next day index rose, and ω^- if index fell. Four scenarios were used, in each scenario feature vector $x[n]$ describing current market situation was constructed in a different way. Each scenario has different number of features, starting from only 1 through 10, 52 to 421. For the purpose of this article we call them Simple (1), Normal (10), Big (52) and Huge(421). All features used in the Simple model were used in Normal, all used in Normal were used in Big, and all in Big were used in Huge. The features are:

- Simple: only overnight gap for SPY (change between yesterday close and today open).
- Normal: only historical prices for SPY:
 - open price,
 - gap, percent change from yesterday close,
 - daily change, percent change from yesterday open,
 - 2 days change, percent change from open to open,
 - 5 days change, percent change from open to open,
 - yesterday daily change, change from open to open at yesterday open,
 - 2 days back daily change,
 - 9 days moving average (close prices),
 - 12 days moving average (close prices),
 - 26 days moving average (close prices).
- Big: all features used for SPY in *normal* scenario, and also gaps of other 41 instruments plus VIX previous day close.
- Huge: all 10 features for all 42 symbols plus VIX level.

VIX is the Volatility Index. It measures the market's expectation of near term volatility based on options prices of S&P500 stock index.

Calculations were done using training and test data sets. All data was divided into 5 groups of corresponding training and test data sets in a way showed in (2).

The training set was build using 252 features vectors, one vector for each day. This is approximately 1 year of data. The test set was build using 126 vectors representing days following the training period. Such approach can be used by investor to trade on real markets. Starting point would be November 2008. At this time investor could use 1 year historical data and build a decision rule (1) to use it for a next 6 months.

Lets consider *Big* scenario to check what kind of results investor would see. After building a *CPL* linear classifier (1) with *RLS* feature selection procedure at November 2008 results would be as follow:

- *RLS* method will choose only 11 features,
- *CPL* linear classifier on those 11 features will have accuracy of classification measured by leave one out methodology of 67.98%.

If investor would be satisfied with those results on historical data, he could start using the model to trade on real markets. If so, in the next half year he would be successfully in 56.35%² cases predicting day index move (up or down). If he can bet on both market up and down his profit would be as high as 22.47%. Making profit on moving down market could be achieved by investing in futures contracts like E-mini³, or fund like SPY⁴.

After half year the investor could decide that his model is out of date and it has to be rebuilt using more up-to-date data. Construction of the new model could be again done using 1 year historical data, this time going back to May 2008. New model would use 42 features and have a little higher accuracy of 69.57% measured on training set, but in the next 6 months it would not produce a profit. Days on good position would be only 48.41% that would transfer to lose of 7.10%. All results till beginning of March 2011 are presented in 2, results for different scenarios are presented in 4..

5. Conclusion

These results show that it is hard to achieve a better prediction of the next day's market price change than 50% which in fact does not give better results than throwing a coin. Never the less the best results of 53.91% could give some advantage to the investor. Such advantage could actually return an average profit of 15% in 6 months,

¹ In other words *ex post* error of prediction is equal 43.65%

² [http://www.cmegroup.com/trading/equity-index/us-index/e-mini-sandp500_contract_specifications.html]

³ SPY - fund corresponds to the price and yield performance of the S&P 500 [<https://www.spdrs.com/product/fund.seam?ticker=spy>]

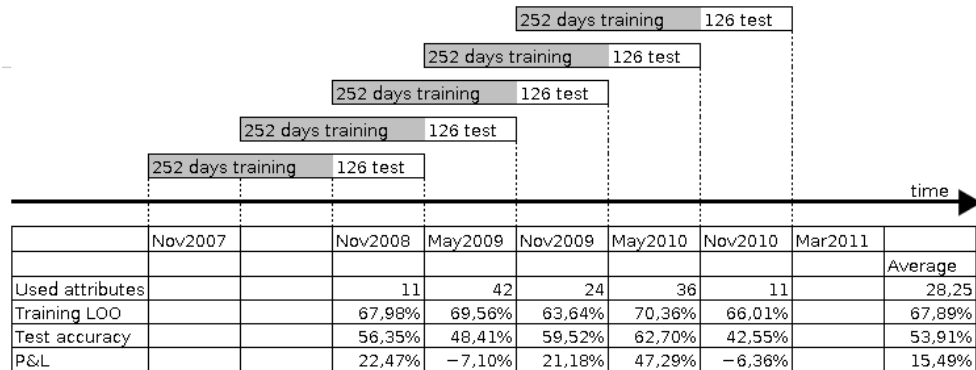


Fig. 2. Moving window of building model on 1 year of training data and testing it on next half year of the data from Nov 2007 till Feb 2011. Source: own work

Table 1. Results for the CPL classifier with Relax method of features selection on different set of attributes

Model	Simple	Normal	Big	Huge
No of attributes	1	10	52	421
Attributes after features selection	1	3.6	24.8	92
LOO accuracy on training sets	55.20%	55.41%	67.51%	91.86%
Average accuracy on test sets	51.07%	50.84%	53.91%	48.93%
Average profit or lose on test sets	-1.72%	5.01%	15.49%	4.90%

giving a total of 77% profit in the entire time period tested (November 2008 - March 2011). It could be argued that such a result from using the CPL linear classifier for predicting the one day movements of a market index is promising.

The practical aspect of this research is actual investing in the market. We could have a bad decision from 1 model resulting in different levels of loses. A loss could be as low as 0.1% or as high as 3.0%. In both cases the model will be wrong, but the consequences of each loss differ greatly. The question that could be asked here is: can we account for this variation during optimisation? It seems that the CPL penalty function $\Phi(w[n], \theta)$ could easily account for it by α_j parameters. More research in this area could be done in the future.

Other area of future research could be to look for explanation of observed differences between classification accuracy on training and test data sets. For example for biggest space with 421 attributes, accuracy measured by leave one out method was equal 91.86% but on test set it was only 48.93%. This difference get lower when less

features were used. Probably it could be explained by the features subset selection bias [18][22], but more research need to be done to verify this hypothesis.

References

- [1] Aizerman M., Braverman E., Rozonoer L., Theoretical foundations of the potential function method in pattern recognition learning, *Automation and Remote Control* 25, 1964.
- [2] Bishop M., *Neural networks for pattern recognition*, Oxford University Press, 2005.
- [3] Bobrowski L., *Eksploracja danych oparta na wypukłych i odcinkowoliniowych funkcjach kryterialnych*, Wydawnictwa Politechniki Białostockiej, 2005.
- [4] Bobrowski L., Łukaszuk T., Feature selection based on relaxed linear separability, *Biocybernetics and Biomedical Engineering*, Volume 29, Number 2, 2009, pp. 43-59.
- [5] Bollerslev T., Generalized Autoregressive Conditional Heteroskedasticity, *Journal of Econometrics*, 31, 1986, pp. 307-327.
- [6] Box G.E.P, Jenkins G.M., *Analiza szeregów czasowych*, Państwowe Wydawnictwo Naukowe, 1983.
- [7] Cortes C. Vapnik, V., Support-Vector Networks, *Machine Learning*, 20, 1995.
- [8] Duda O.R., Hart P.E., Stork D.G., *Pattern Classification*, J. Wiley, New York, 2001.
- [9] Edwards R.D., Magee J., *Technical Analysis of Stock Trends*, AMACOM, 7th edition, 1997.
- [10] Egeli B., Ozturan M., Badur B., *Stock Market Prediction Using Artificial Neural Networks*, *Proceedings of the 3rd International Conference on Business*, 2003.
- [11] Engle R.F., Autoregressive Conditional Heteroskedasticity with the Estimates of the Variance of U.K. Inflation, *Econometrica*, 50, No. 4, 1982, pp. 987-1007.
- [12] Hamilton J.D., *Time Series Analysis*, Princeton University Press, 1994.
- [13] Huang W., Nakamoria Y., Wangb S.Y., Forecasting stock market movement direction with support vector machine, *Computers & Operations Research* Vol. 32, Issue 10, 2005, pp. 2513-2522.
- [14] Khatimlianskii A., Genetic Algorithms vs. Simple Search in the MetaTrader 4 Optimizer, [<http://articles.mql4.com/361>]
- [15] Kim K.J., Financial time series forecasting using support vector machines, *Neurocomputing* Vol. 55, Issues 1-2, 2003, pp. 307-319.

- [16] Kimoto T., Asakawa K., Yoda M., Takeoka M., Stock market prediction system with modular neural network, Proceedings of the International Joint Conference on Neural Networks, 1990, pp. 1-6.
- [17] Krawczuk J., Bobrowski L., Short term prediction of stock indexes changes based on a linear classifier, Symulacja w badaniach i rozwoju, Vol. 1 nr 4/2010.
- [18] Miller A., Subset selection in regression, Chapman & Hall/CRC, 2002.
- [19] Mizuno H., Kosaka M., Yajima H., Komoda N., Application of Neural Network to Technical Analysis of Stock Market Prediction, Studies in Informatic and Control, Vol. 7, No. 3, 1998, pp. 111-120.
- [20] Nison S., Japanese Candlestick Charting Techniques, Second Edition, Prentice Hall Press, 2nd edition, 2001.
- [21] Poli R., Langdon W.B., McPhee N.F., A Field Guide to Genetic Programming, Lulu.com, 2008.
- [22] Singhi S.K., Liu H., Feature Subset Selection Bias for Classification Learning, Proceedings of the 23 rd International Conference on Machine Learning, Pittsburgh, PA, 2006, pp. 849-856.
- [23] Yang J., Honavar V., Feature subset selection using a genetic algorithm, Intelligent Systems and their Applications, IEEE, 1998.
- [24] Zekic M., Neural Network Applications in Stock Market Predictions - A Methodology Analysis, Proceedings of the 9th International Conference on Information and Intelligent Systems, 1998, pp. 255-263.

PROGNOZOWANIE KIERUNKU ZMIANY INDEKSÓW GIEŁDOWYCH ZA POMOCĄ KLASYFIKATORA LINIOWEGO TYPU CPL

Streszczenie Prognozowanie cen akcji i wartości indeksów giełdowych jest zadaniem trudnym, gdzie używanych jest wiele technik takich jak: analiza techniczna, ekonometryczne szeregi czasowe, techniki eksploracji danych. Artykuł ten przedstawia jedną z metod eksploracji danych - klasyfikator liniowy. Klasyfikator ten w przeprowadzonym eksperymencie został użyty do prognozowania wartości indeksu giełdy amerykańskiej. Prognozowanie takie oparte jest o dane opisujące obecny stan giełdy. Stan giełdy można opisać różną ilością danych (atrybutów, cech). W najprostszym przypadku może to być tylko jednodniowa zmiana ceny prognozowanego indeksu. W bardziej rozbudowanym modelu można użyć wielu cen historycznych. W modelu jeszcze bardziej rozbudowanym można użyć danych z innych giełd, kursów walut, cen towarów jak np. ropa. Użycie dużej ilości danych wymaga dłuższego czasu obliczeń parametrów modelu. W prezentowanym podejściu klasyfikator liniowy

budowany jest w oparciu o minimalizację wypukłej i odcinkowo-liniowej funkcji kryterialnej. Metoda ta jest bardzo wydajna o koszcie zbliżonym do programowania liniowego. Dodatkowo użyta została metoda selekcji cech RLS. Techniki te pozwoliły na efektywną eksplorację danych o wielu wymiarach. W artykule przedstawiono cztery scenariusze o różnej ilości danych opisujących giełdę. W najprostszym użytku tylko jednej danej, w najbardziej rozbudowanym 421 danych o 43 instrumentach finansowych. Najlepsze wyniki uzyskano dla pośredniego modelu o 52 cechach, w którym model przewidział prawidłowo 53.19% kierunków dziennych zmian indeksu S&P500. Otrzymany wynik nie wydaje się być wysoki, jednak gdyby inwestowano w indeks zgodnie z modelem zysk z takich inwestycji wyniósłby 77% w okresie od października 2008 do marca 2011, dając średnio 28% zysku rocznie.

Słowa kluczowe: klasyfikator liniowy, prognozowanie giełdy, funkcje wypukłe i odcinkowo-liniowe

AN IMPROVED GENETIC ALGORITHM FOR SOLVING THE SELECTIVE TRAVELLING SALESMAN PROBLEM ON A ROAD NETWORK

Anna Piwonska

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

Abstract: The Selective Travelling Salesman Problem (STSP) is a modified version of the Travelling Salesman Problem (TSP) where it is not necessary to visit all vertices. Instead of it, with each vertex a number meaning a profit is associated. The problem is to find a cycle which maximizes collected profit but does not exceed a given cost constraint. A direct application of the STSP, e.g. in Intelligent Transportation Systems, is finding an optimal tour in road networks. However, while the classic STSP is defined on a complete graph, a road network is in general not complete and often has a rather sparse edge set. This paper presents the STSP defined on a road network (R-STSP). Since the R-STSP is NP-hard, the improved genetic algorithm (IGA) is proposed which is the next version of our previous GA. The main aim of this paper is to investigate the role of the deletion mutation in the performance of the IGA.

Keywords: travelling salesman problem with profits, genetic algorithm, deletion mutation

1. Introduction

The TSP is an NP-hard problem studied in operations research and computer science [1]. The problem is formulated as follows. Given a list of cities and their pairwise distances, the task is to find the shortest possible tour that visits each city exactly once.

While in the TSP a salesman needs to visit all cities, some variant problems enforce to visit only selected ones, depending on a profit gained during visiting. This feature gives rise to a number of problems which are called in the literature the Travelling Salesman Problem with Profits (TSPwP) [3]. In this group of problems, usually one of n cities has a special meaning - it is considered as a depot. In one version of the TSPwP described in the literature, the problem is to find an elementary cycle

starting from a depot, that maximizes collected profit such that the tour length does not exceed a given constraint. This problem appears under the name "the orienteering problem" [13] or "the selective TSP" [12]. Since the TSPwP belongs to the class of NP-hard problems, many metaheuristic approaches have been proposed in the literature e.g. tabu search [6], ant colony optimization [8], genetic algorithms [7], neural networks [15] and harmony search [5].

The R-STSP was first formulated in the author's previous paper [11] and is some modification of the problem described above, with two important assumptions introduced. Firstly, a graph may not be complete: not every pair of vertices must be connected by an edge. In fact, a road network is in general not complete and often has a rather sparse edge set. This issue was considered by Fleischmann [4], who introduced the notion of the Travelling Salesman Problem on a Road Network (R-TSP). Despite the fact that we can transform such a not complete graph in a complete one by introducing dummy edges, such an approach seems to be ineffective. It increases the search space and has a direct impact on the execution time of the algorithm.

The second assumption is that we allow repeated visiting of a given city: a cycle we are looking for may not be an elementary one. This assumption results from the fact that a graph may not be complete. Moreover, in the real world returns are natural: one may want to travel using repeated fragments of a route. However, while a salesman can be in a given city more than once, a profit is realized only during first visiting. This assumption prevents from generating routes in which a city with the highest profit is continually visited while others are not. With these additional assumptions, the problem is more realistic and could have practical applications in logistics and shipping.

In [11] the GA with special crossover and mutation operators was proposed. In this paper we present the improved version of our previous GA (IGA) in which a new mutation is introduced as an additional operator, namely the deletion mutation. Moreover, the mutation which inserts a city to a tour is improved. The main aim of the paper is to investigate the role of the deletion mutation in the performance of the IGA. Experiments conducted on the real network of 160 cities in eastern and central Poland show that the deletion mutation significantly improves the quality of solutions generated by the IGA.

The rest of the paper is organized as follows. Section 2. presents formal definition of the R-STSP. Section 3. describes the details of the IGA, with particular focus on both mutations. Experimental results are reported in Section 4.. The last section contains conclusions and some remarks about future work.

2. Definition of the R-STSP

A network of cities is represented by a weighted, undirected graph $G = \langle V, E \rangle$. $V = \{1, 2, \dots, n\}$ is a set of n vertices and E is a set of edges. Each node in G corresponds to a city in a network. Vertex 1 has a special meaning and is interpreted as the depot. An undirected edge $\{i, j\} \in E$ means that there is a possibility to travel from the city i to the city j (and vice versa). The weight d_{ij} of the edge $\{i, j\}$ denotes a distance between cities i and j . Additionally, each vertex has assigned a non-negative number meaning a profit. Let $F = \{f_1, f_2, \dots, f_n\}$ be a vector of profits for all vertices. An important assumption is that a profit is realized only during first visiting of a given vertex. The exemplary graph is shown in Fig. 1.

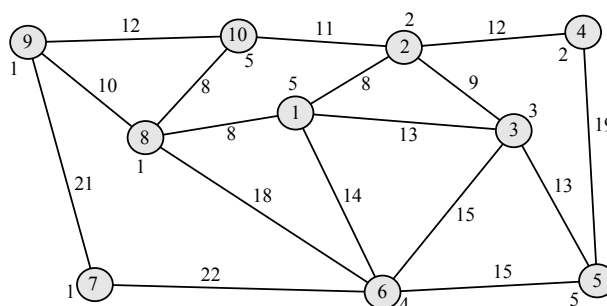


Fig. 1. A graph representation of a network of cities; the d_{ij} values are marked on the edges, the f_i values are marked next to the nodes

The R-STSP can be formulated as follows. The goal is to find a cycle starting from the depot that maximizes collected profit such that the tour length does not exceed a given constraint c_{max} .

3. The IGA for the R-STSP

During the last years several methods were proposed for handling constraints in GAs. Most of them are based on the concept of a penalty function [9]. In the IGA, as well as in our previous GA, a different approach is proposed. Due to the special way of generating the initial population and using specialized operators, the IGA searches solutions only in the feasible region.

The difference between our previous GA and the IGA is in a mutation operator: the IGA uses improved, heuristic mutation (hereafter called the insertion mutation)

and as an additional operator, the deletion mutation. The deletion mutation is presented in the IGA in two forms: the first tries to delete from a tour repeated cities (the deletion mutation I) and the second tries to delete from a tour a random city (the deletion mutation II).

The pseudocode of the IGA is presented as Algorithm 1.

Algorithm 1: the IGA for the R-STSP

```
Begin
  generate the initial population of individuals of size P;
  compute fitness function for each individual;
  for i:=1 to ng do
    begin
      select the population i from the population i-1
      by means of tournament selection
      with the group size equal to t_size;
      divide population into disjoint pairs;
      cross each pair if possible;
      apply the deletion mutation I to each individual;
      apply the deletion mutation II to each individual;
      apply the insertion mutation to each individual;
      compute fitness function for each individual;
    end
  choose the best individual from the population as the result;
End
```

When we want to solve a given problem by a GA, first we must encode a solution into a chromosome. Like in the most TSP-based problems, we decide to use the path representation [10]. In this approach, a tour is encoded as a sequence of vertices. For example, the tour 1 - 2 - 3 - 5 - 6 - 1 is represented by the sequence (1 2 3 5 6 1).

The IGA starts with a randomly generated population of P solutions. The initial population is generated in a special way. Starting at the depot, we choose a city to which we can travel from the depot with equal probability. We add the distance between the depot and the chosen city to the current tour length. If the current tour length is not greater than $c_{max}/2$, we continue, but instead of starting at the depot, we start at the chosen city. We again randomly select a city, but this time we exclude from the set of possible cities the city from which we have just arrived (the last city in a partial tour). This assumption prevents from continual visiting a given city but is relaxed if there is no possibility to choose another city. If the current tour length is greater than $c_{max}/2$, we reject the last city and return to the depot the same way. In this case the tour length does not exceed c_{max} therefore the constraint imposed by the problem is preserved. It is easy to observe that such an idea of generating the initial

population causes that individuals are symmetrical in respect of the middle city in the tour. However, experiments show that the IGA quickly removes these symmetries.

The next step is to evaluate individuals in the initial population by means of the fitness function. The fitness of a given individual is equal to collected profit under the assumption that a profit is gained only during first visiting of a given vertex.

Subsequently the IGA starts to improve the initial population through repetitive application of selection, crossover and mutation. In our experiments we use tournament selection: we select t_{size} individuals from the current population and determine the best one from the group. The winner is copied to the next population and the whole tournament group is returned to the old population.

In the first step of crossover, individuals are randomly coupled. Then, each couple is tested if crossover can take place. If two parents do not have at least one common gene (with the exception of the depot), crossover cannot be done.

Fig. 2 illustrates how the genetic material is swapped during crossover. First we randomly choose one common gene from the set of common genes in both parents: P1 and P2 (we exclude the depot from this set). This gene will be the crossing point. Then we exchange fragments of tours from the crossing point to the end of the chromosome in two parent individuals. If offspring individuals (O1 and O2) preserve the constraint c_{max} , they replace their parents in the new population. If one offspring individual does not preserve the constraint c_{max} , its position in the new population is occupied by fitter parent. If both children do not preserve the constraint c_{max} , they are replaced by their parents in the new population.

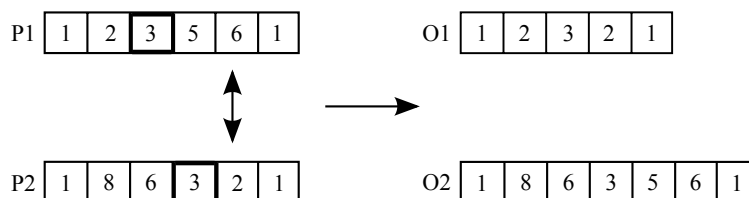


Fig. 2. An example of crossover operator

The last genetic operator is mutation. The role of mutations in a GA is to increase genetic diversity in a population [2]. In this paper we present two kinds of mutation adjusted to our problem: the deletion mutation and the insertion mutation.

The idea of using the deletion mutation comes from genetics. In living organisms, the deletion mutation takes place when a part of a chromosome or sequence of DNA is missing. In our problem, the deletion mutation I tries to eliminate from a

chromosome every appearance of repeated cities (with the exception of the first and the last gene which are established). They can appear in an individual after crossover (Fig. 2) and are also presented in chromosomes in the initial population. Such cities do not influence fitness function value and an attempt of removing them from a chromosome should be undertaken (obviously, this mutation leaves in a chromosome one appearance of a given city). Let us assume that a partial tour is $\dots i, j, k \dots$. The city j can be removed from a chromosome if there is the edge $\{i, k\} \in E$. The deletion mutation I is described as Algorithm 2.

```
Algorithm 2: the deletion mutation I
Begin
  for i:=2 to chromosome_length-1 do
    if a city in the position i is presented in a chromosome
      more than once then
        try to remove this city from a chromosome;
End
```

After removing repeated cities from chromosomes, a population undergo the deletion mutation II which tries to delete from each chromosome a random city (with the exception of the first and the last gene). The deletion mutation II is described as Algorithm 3.

```
Algorithm 3: the deletion mutation II
Begin
  choose a random city in a chromosome;
  try to remove this city from a chromosome;
End
```

The last mutation is the insertion mutation. The principle of operation of this mutation is presented as Algorithm 4.

```
Algorithm 4: the insertion mutation
Begin
  randomly choose in a chromosome two neighboring cities i and j;
  create a set S of possible cities which are not presented
  in a chromosome and which can be inserted between i and j;
  sort cities in S decreasingly according to profits;
  if profits are equal sort increasingly according to
  increment of a total length of a tour
  after inserting a given city;
  while S is not empty do
    begin
      take the first city in S, namely k, and remove k from S;
      if inserting city k between cities i and j does not violate
```

```
the constraint  $c_{max}$  then
begin
  insert city  $k$  between cities  $i$  and  $j$ ;
  break {a chromosome is mutated};
end;
end;
End
```

In the insertion mutation, a city is inserted into a tour only if it has not been inserted in a tour yet. The reason behind this is that inserting a city so far not presented in a tour improves fitness of a given individual. Moreover, the best city from the set of all possible cities is inserted. The best means the city with the highest profit and (if profits of cities are equal) the one city that will cause the least increment of the total length of a tour. If no city can be inserted into a chromosome without violating c_{max} , the mutation is not performed.

It is important that the insertion mutation is performed after both deletion mutations. If deletion mutations remove some cities from a tour, the total length of a tour decreases. This way the probability of successful application of the insertion mutation increases.

The IGA terminates when ng generations is reached.

4. Experimental Results

Computer experiments were performed on network of 160 cities in Poland. Twenty cities from each of eight provinces of eastern and central Poland were chosen. The network used in experiments was created from a real map, by including to a graph main segments of roads. Profits associated with cities were determined according to a number of inhabitants in a given city. As the depot, the capital of Poland, Warsaw, was chosen. Data concerning this network are accessible on the website <http://piwonska.pl/p/research/> in two text files: cities.txt and distances.txt.

The line number i in both files represents information about city number i . The number of lines in each file is equal to n . Format of the line i in cities.txt file is: i name-of-the-city f_i . Format of the line i in distances.txt file is: i j_1 d_{ij_1} ... j_k d_{ij_k} , where j_1 ... j_k are numbers of cities connected to the city i and d_{ij_1} ... d_{ij_k} are distances between them.

We performed experiments for 11 c_{max} values: $\{500, 600, 700, \dots, 1500\}$. We set $t_{size} = 3$ and $P = 300$. Higher values of P did not lead to an improvement in results. Since the algorithm converges quickly, setting $ng = 100$ was enough to obtain good results.

To investigate the role of the deletion mutation, three series of experiments were performed. In the first case, we turn off both deletion mutations. The only mutation performed during the IGA was the insertion mutation. In the second case, the deletion mutation I was turn on and in the third case the IGA performed both deletion mutations. Ten runs were performed for each case, what resulted in thirty runs for each c_{max} . Tab. 1 presents the average, the best and the worst (in brackets: the best; the worst) collected profits.

Table 1. The average, the best and the worst collected profits from 10 runs of the IGA

c_{max}	insertion mutation	insertion mutation + deletion mutation I	insertion mutation + deletion mutation I and II
500	55.8 (60; 48)	56.7 (60; 53)	56.8 (60; 50)
600	66.4 (73; 57)	67.0 (73; 60)	69.4 (73; 65)
700	72.9 (80; 66)	76.3 (80; 71)	77.7 (80; 71)
800	81.4 (86; 77)	84.1 (87; 80)	84.9 (87; 82)
900	88.4 (93; 81)	91.9 (101; 88)	92.3 (101; 90)
1000	94.4 (104; 85)	97.5 (109; 94)	105.1 (114; 95)
1100	103.0 (112; 95)	107.5 (117; 100)	113.1 (122; 106)
1200	107.0 (118; 99)	111.6 (119; 101)	120.5 (129; 109)
1300	119.5 (136; 106)	125.8 (141; 114)	133.3 (143; 110)
1400	126.9 (139; 117)	131.6 (146; 121)	139.7 (151; 131)
1500	131.3 (152; 115)	139.0 (158; 124)	147.4 (160; 133)

One can see that the average profits are the worst in the case of both deletion mutations turned off. Turning on the deletion mutation I improves the results, however the best improvement is gained when both deletion mutations are turned on. This effect is observed for all c_{max} values. In general, as c_{max} increases, the differences between the IGA without deletion mutations and the IGA with both deletion mutations start to deepen. The largest improvement rate is observed for $c_{max} = 1200$ and is equal to 12.6%.

Only for $c_{max} = \{500, 600, 700\}$ the best profits for the IGA without deletion mutations are the same as the best profits for the IGA with both deletion mutations. Starting from $c_{max} = 800$ the best results obtained by the IGA without deletion mutations are always worse than the best results in the case of both deletion mutations turned on.

Looking at the best and the worst collected profits one can see that all algorithms are not stable: the differences between the best and the worst results are relatively large. This issue will be investigate in our future research.

The chromosomes of the best individuals found by the IGA with both deletion mutations are presented in Tab. 2.

Table 2. The best individuals found by the IGA with the insertion mutation, the deletion mutation I and the deletion mutation II

c_{max}	profit	total distance	chromosome
500	60	487	1 11 72 71 70 69 79 61 80 62 66 64 63 73 15 16 1
600	73	592	1 10 5 6 12 70 69 79 61 80 62 66 68 64 63 73 72 71 11 1
700	80	684	1 11 71 72 73 67 63 64 68 66 65 62 80 61 79 69 70 12 5 10 1
800	87	798	1 10 5 6 12 70 69 79 61 80 62 65 66 64 68 119 117 118 67 63 73 72 71 11 1
900	101	898	1 16 18 97 91 93 81 92 90 89 107 102 106 105 118 67 63 64 66 62 80 61 79 69 70 71 72 11 1
1000	114	992	1 10 5 12 6 11 71 72 74 61 80 62 66 68 64 63 67 118 105 106 102 107 109 131 132 89 90 91 97 18 16 1
1100	122	1100	1 11 71 72 74 61 79 80 62 66 68 64 63 67 118 105 106 102 107 109 131 132 133 88 89 90 91 92 81 93 95 94 98 16 1
1200	129	1196	1 10 5 6 11 71 72 73 63 74 61 80 62 66 64 68 119 103 115 101 120 106 105 106 102 107 109 131 132 133 88 89 81 92 91 97 18 16 13 1
1300	143	1297	1 13 14 98 94 93 81 89 107 109 132 131 111 110 114 101 120 102 106 105 118 67 63 64 68 66 62 80 61 79 69 70 12 6 11 71 72 73 15 16 1
1400	151	1392	1 10 5 2 12 70 71 11 72 74 61 79 80 62 66 68 64 63 67 118 105 106 102 107 109 131 132 133 88 89 90 91 92 81 93 95 94 96 14 13 1
1500	160	1498	1 10 5 12 70 71 11 72 73 74 61 62 80 79 69 78 65 66 68 64 63 67 118 105 104 101 120 106 102 107 109 131 132 133 88 89 81 93 91 97 94 98 14 13 1

One can see that there are almost no repeated cities in these chromosomes. The exception is the individual for $c_{max} = 1200$ in which the city 106 occurs twice. However, this city can not be removed from the chromosome due to the lack of adequate edges.

As an example, Fig. 3 presents the best run of the IGA with both deletion mutations for $c_{max} = 1500$. One can see that the IGA converges quickly: before 50th generation. The potential reason of this problem could be the fact that the crossover and the mutation operators often can not produce modified individuals (e.g. due to the constraint c_{max}). For example, detailed analysis shows that in a typical run of the IGA

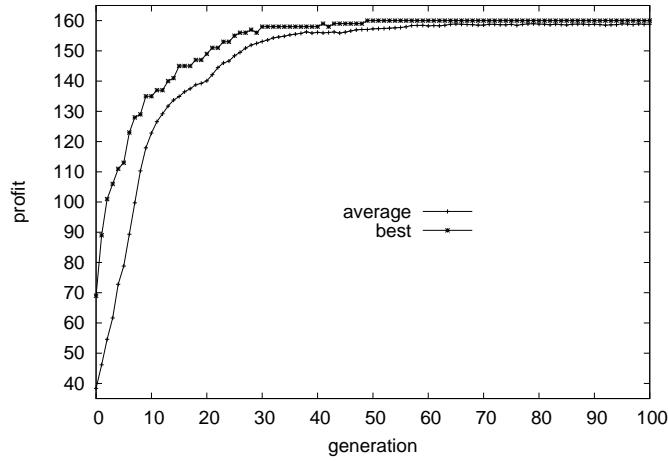


Fig. 3. The IGA run with the insertion mutation and both deletion mutations for $c_{max} = 1500$

on average only one out of every three individuals undergoes the deletion mutation II. That is why the population quickly fills with the copies of the same individual.

5. Conclusions

In this paper we presented the IGA for solving the R-STSP. Three mutation operators adjusted to the problem were proposed: the insertion mutation, the deletion mutation which tries to delete from a tour repeated cities (the deletion mutation I) and the deletion mutation which tries to delete from a tour a random city (the deletion mutation II).

The aim of the work was to investigate the role of deletion mutations on the performance of the IGA. Computer experiments conducted on the real network of 160 cities in Poland indicated that deletion mutations (as additional operators) improved the quality of obtained results.

Since the R-STSP is defined on the graph which in general is not complete, well known recombination operators for the classic TSP, e.g. inversion, mutual swap, cannot be used. Thus there is a need for designing specialized genetic operators adjusted to this problem. Also, there is a problem of premature convergence of the IGA which should be tackled. Another subject of future research is to investigate the effect of introducing elitism and niching techniques into the IGA.

The issue which must be carefully studied is another approach to handling constraint c_{max} . We plan to implement the IGA with the penalty function and compare

obtained results. Also, other heuristics will be tested, e.g. ant colony optimization, tabu search or harmony search.

An important issue is reusing discovered solutions. Optimal tours obtained for a given c_{max} can be potentially reused when solving problems for larger c_{max} . We think that such an idea could improve performance of the IGA. It will be the successive subject of our future research.

References

- [1] Applegate D.L., Bixby R.E., Chvátal V., Cook W.J., *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, 2006.
- [2] De Falco I., Della Cioppa A., Tarantino E., Mutation-based genetic algorithm: performance evaluation. *Applied Soft Computing*, vol. 1 (4), Elsevier, pp. 285-299, 2002.
- [3] Feillet D., Dejax P., Gendreau M., *Traveling Salesman Problems with Profits*, *Transportation Science*, Vol. 39, No. 2, pp. 188-205, 2005.
- [4] Fleischmann B., A new class of cutting planes for the symmetric travelling salesman problem. *Mathematical Programming*, vol. 40, pp. 225-246, 1988.
- [5] Geem Z.W., Tseng Ch.-L., Park Y., *Harmony Search for Generalized Orienteering Problem: Best Touring in China*. LNCS, vol. 3612, Springer, pp. 741-750, 2005.
- [6] Gendreau M., Laporte G., Semet F., A tabu search heuristic for the undirected selective travelling salesman problem. *European Journal of Operational Research*, vol. 106 (2-3), Elsevier, pp. 539-545, 1998.
- [7] Jozefowicz N., Glover F., Laguna M., Multi-objective Meta-heuristics for the Traveling Salesman Problem with Profits. *Journal of Mathematical Modelling and Algorithms*, vol. 7 (2), pp. 177-195, 2008.
- [8] Liang Y.-C., Smith A.E., An ant colony approach to the orienteering problem. Technical report. Department of Industrial and Systems Engineering, Auburn University, Auburn, USA, 2001.
- [9] Michalewicz Z., *Genetic Algorithms, Numerical Optimization and Constraints*. Proceedings of the 6th International Conference on Genetic Algorithms, Pittsburgh, July 15-19, pp. 151-158, 1995.
- [10] Michalewicz Z., *Genetic Algorithms+Data Structures=Evolution Programs*. WNT, Warsaw, 1996.
- [11] Piwonska A., Genetic algorithm finds routes in travelling salesman problem with profits. *Zeszyty naukowe Politechniki Białostockiej. Informatyka* (in Polish), Vol. 5, pp. 51-65, 2010.

- [12] Qin H., Lim A., Xu D., The Selective Traveling Salesman Problem with Regular Working Time Windows. *Studies in Computational Intelligence*, Vol. 214, pp. 291-296, 2009.
- [13] Sevkli Z., Sevilgen F.E., Variable Neighborhood Search for the Orienteering Problem. *LNCS*, Vol. 4263, pp. 134-143, 2006.
- [14] Souffriau W., Vansteenwegen P., Berghe G.V., Oudheusden D.V., A Greedy Randomised Adaptive Search Procedure for the Team Orienteering Problem. In proceedings of EU/MEeting 2008 - Troyes, France, 2008.
- [15] Wang Q., Sun X., Golden B.L., Jia J., Using artificial neural networks to solve the orienteering problem. *Annals of Operations Research*, vol. 61, Springer, pp. 111-120, 1995.

ULEPSZONY ALGORYTM GENETYCZNY DO ROZWIĄZANIA SELEKTYWNEGO PROBLEMU KOMIWOJAZERA W SIECI DROGOWEJ

Streszczenie Selektywny problem komiwojażera (STSP) jest zmodyfikowaną wersją problemu komiwojażera (TSP), w której nie jest konieczne odwiedzenie wszystkich wierzchołków. Zamiast tego, z każdym wierzchołkiem związana jest liczba oznaczająca zysk. Problem polega na znalezieniu cyklu w grafie, który maksymalizuje zysk, ale którego koszt nie przekracza zadanego ograniczenia. Bezpośrednim zastosowaniem problemu STSP, np. w Inteligentnych Systemach Transportowych, jest odnajdywanie optymalnej trasy w sieci drogowej. Jednakże, podczas gdy klasyczny problem STSP jest zdefiniowany na grafie pełnym, sieć drogowa zwykle nie jest grafem pełnym i często ma rzadki zbiór krawędzi. Artykuł przedstawia problem STSP zdefiniowany w sieci drogowej (R-STSP). Ponieważ R-STSP jest NP-trudny, zaproponowano ulepszony algorytm genetyczny (IGA), który jest rozszerzoną wersją poprzedniego algorytmu genetycznego. Głównym celem artykułu jest zbadanie roli mutacji usuwającej w w jakości wyników IGA.

Słowa kluczowe: selektywny problem komiwojażera na sieci drogowej, algorytm genetyczny, mutacja usuwająca

Artykuł zrealizowano w ramach pracy badawczej S/WI/2/2008

HIST - AN APPLICATION FOR SEGMENTATION OF HEPATIC IMAGES

Daniel Reska, Marek Krętowski

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

Abstract: HIST (Hepatic Image Segmentation Tool) is a Java-based application for segmentation and visualization of medical images, specialised for hepatic image analysis. This paper contains an overview of the application features, a description of adapted segmentation algorithms and their experimental validation. The application provides two main segmentation tools, based on region growing and active contour model methods, adapted for the case of liver segmentation. HIST also offers data visualization tools, including multiplanar reconstruction, volume rendering and isosurface extraction.

Keywords: liver segmentation, active contour, region growing, volume rendering, multiplanar reconstruction, isosurface extraction

1. Introduction

Medical imaging [17] is one of the fundamental tools of modern medicine. The ability of non-invasive exploration of internal aspect of various body parts is invaluable in research and clinical practice. One of the basic tasks in medical image analysis is the segmentation of interesting structure for further evaluation, such as diagnosis or surgery planning [16].

Image segmentation consists in partitioning an image into a set of separated regions that differ by some specific characteristic, such as intensity or texture. It is one of the most necessary but difficult tasks in computer vision, especially in the analysis of medical images. In this case, the variety of imaging methods and their applications imposes a necessity of customisation of the methods for each specific task, which makes the development of more robust techniques challenging.

Liver segmentation is a particularly difficult task, even for an expert [6]. Segmentation tools have to deal with the irregularity of the organ shape and of their boundaries, the intensity variation due to anatomical complexity, the pathologies and

neighboring of other organs - mainly the heart, stomach or rib cage structures (see Fig. 1). The interpersonal variance of the liver shape is also a problem for statistical model-based algorithms. All these factors make liver segmentation especially challenging.

In this paper, we present HIST (Hepatic Image Segmentation Tool) - an application for segmentation and visualization of medical images with tools adapted specifically for liver segmentation tasks. Many other software frameworks and complete applications with similar functionalities are available [19],[24],[4]. Most of them, however, implements only general-usage tools that are not suited for any specific purpose and usually require significant customisation. In the case of liver segmentation, many advanced methods have been developed and are proven effective [10]. In practical evaluation the methods should be examined in a consistent environment which would simulate their real-life usage.

HIST was developed with emphasis on providing out of the box tools for liver segmentation and further visualization and evaluation of the results. Furthermore, the application is aimed at being used by medical doctors, and therefore user experience in practice was also an important factor.

The rest of the paper is organised as follows. Section 2 describes the hepatic segmentation methods, visualization tools and other features of the application. Section 3 contains the results of experimental validation of the tools. Finally, Section 4 presents conclusions and directions of future research.

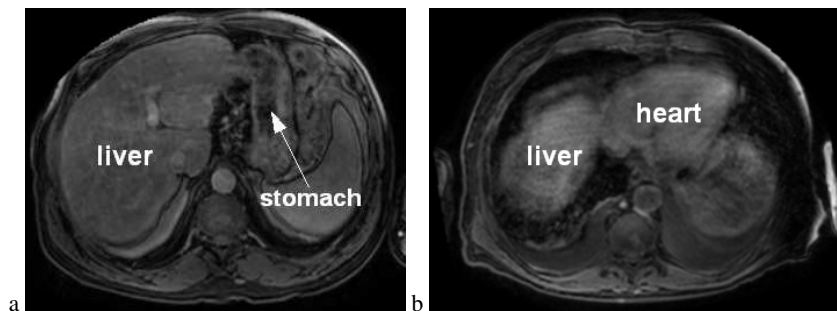


Fig. 1. Problematic liver position near other organs: stomach (a) and heart (b)

2. Hepatic segmentation and visualization

The main segmentation tools implemented in the application are based on two methods: region growing and active contour model. These techniques, commonly used in medical image analysis, were adapted to the specific task of hepatic MRI segmentation. The main goal in the adaptation process was to create usable real-time segmentation tools, that could be used and evaluated in a consistent environment.

2.1 Region growing

The first segmentation tool is seeded region growing [1] and merging algorithm. The base idea of this method is to initialise a set of pixels in the image domain and expand it by adding new pixels that meet specific criteria. Similar resulting regions are merged and post processed. Seed points initialisation can be performed manually or with specialised automated methods.

Seed points initialisation Region growing algorithms are particularly sensitive to the initial location of the start points. Manual initialisation is usually a tedious task, especially in the case of large data sets, therefore an automatic initialisation method was created. The method is based on split and merge algorithm [11], which divides the image into regions with uniform intensity and generates seed points from their centres. The dark regions around the body volume are excluded (see Fig. 2). These points can be used for segmentation of the whole image, but in the case of liver segmentation other constraints are also applied. The user can place a bounding box around the body volume and mark a uniform liver region with a cursor (see Fig. 3). The bounding box contains two regions with the highest probability of where the liver is located. The size and position of the region are based on proportions proposed in [6], although they were slightly modified as a result of experimental validation. A seed point of an area located in these regions and similar to the area pointed by a user is accepted as a potential liver point. This technique not only speeds up the initialisation process, but also increase its reproducibility.

Growing and merging The next phase of the method is growing of the seed points. Original regions (composed of a chosen pixel and its 3x3 neighbourhood) are expanding by adding new pixels that meet specific criteria. A set of new pixels P_{new} , added in a single grow cycle, can be described as:

$$P_{new} = \{p \in N_{region} : |I(p) - \mu| \leq k\sigma\},$$

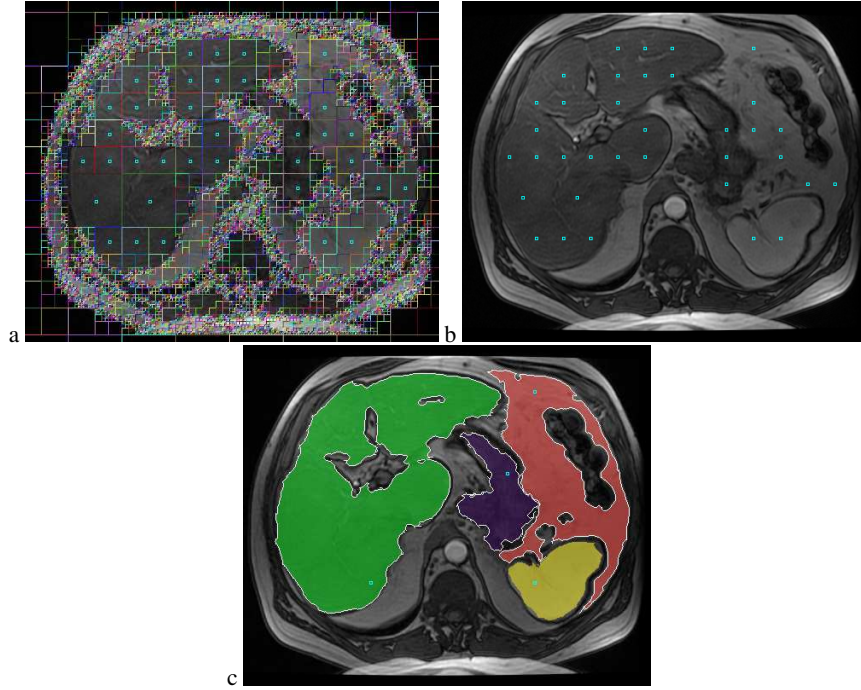


Fig. 2. Automated seed point initialisation: image divided into homogenous regions (a), generated seed points (b) and segmentation result (c)

where $I(p)$ is the intensity of pixel p , μ and σ are the mean and standard deviation of the intensity of the start region, k is a user-defined constant and N_{region} is a set of pixels adjacent to original region. In this method, a different approach was used. New points are added by analysing the neighbourhood of every existing region pixel p_{reg} . A new pixel p_{new} , adjoining to p_{reg} , can be added to the region only if it meet two conditions: the intensity difference between p_{reg} and p_{new} cannot exceed a given threshold and the intensity of p_{new} must be sufficiently similar to the start region intensity mean:

$$P_{new} = \{p_{new} \in N_{p_{reg}} : |I(p_{reg}) - I(p_{new})| \leq T_{adj} \wedge |I(p_{new}) - I_{start}| \leq T_{start}\},$$

where $I(p_{new})$ and $I(p)$ are the intensities of p_{new} and p_{reg} , T_{adj} is the intensity difference threshold, I_{start} is the original 3x3 region intensity mean, T_{start} is the start mean threshold and $N_{p_{reg}}$ is the neighbourhood of p_{reg} . The second condition prevents the "leakage" on a low gradient boundary of two regions, where the intensity difference

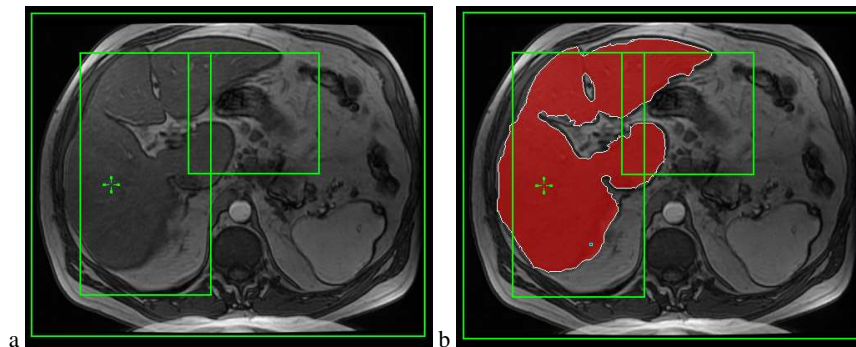


Fig. 3. Automated seed point initialisation with liver positioning constraint: position of bounding box and marked region (a) segmentation result (b)

of adjacent pixels is not sufficient to stop the growing. Algorithm iteration count can also be limited (see Fig. 4).

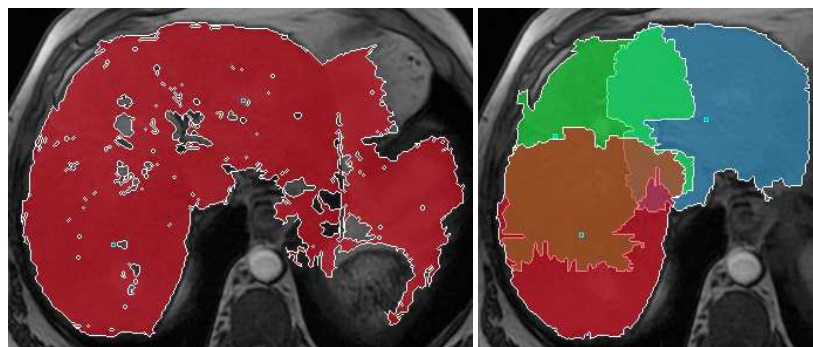


Fig. 4. Iteration limitation used to prevent leakage of a region: result of grow and merge of three seed without limitation (left) and result with decreased iteration count (right)

Merging is performed by calculating the overlap ratio. Two regions are merged if they contain a sufficient percent of shared pixels (see Fig. 5). The default merge ratio is 90%.

Postprocessing The regions segmented with region growing algorithms usually contain many internal discontinuities and border irregularities, therefore final enhance-

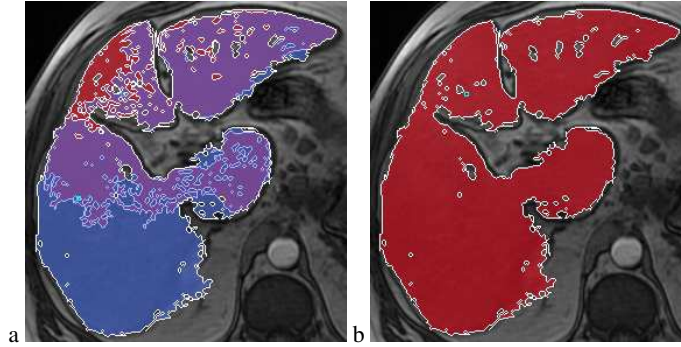


Fig. 5. Region growing and merging: result from three seed points (a) and merged regions (b)

ments are crucial. Apart from manual editing tools, three postprocessing methods were implemented:

- simple sealing by finding the pixels between the two outermost ones in every row and column of the region and filling the points present in both of these scanline runs - the simplest, fastest but least accurate method;
- morphological closing - more accurate, although with a tendency to make undesirable conjunctions in the region;
- classification of each pixel by analysing the count of region points in the pixel neighbourhood - the slowest but the most accurate technique.

The results of these methods are presented in Fig. 6.

2.2 Active contour

Active contour (snake) [12] is the second main segmentation tool. Its original representation is a parametric curve, which deforms under influence of internal and external forces. The goal of the evolution process is to minimise the total energy of the snake. With the contour defined as $v(s) = (x(s), y(s))$ where $s \in [0, 1]$, total snake energy could be written as:

$$E_{snake}^* = \int_0^1 E_{snake}(v(s)) ds = \int_0^1 E_{int}(v(s)) + E_{image}(v(s)) + E_{con}(v(s)) ds \quad (1)$$

where E_{int} is the internal energy (controlling bending and stretching), E_{image} is the image force (moving the snake towards desired features) and E_{con} represents other possible constraints.

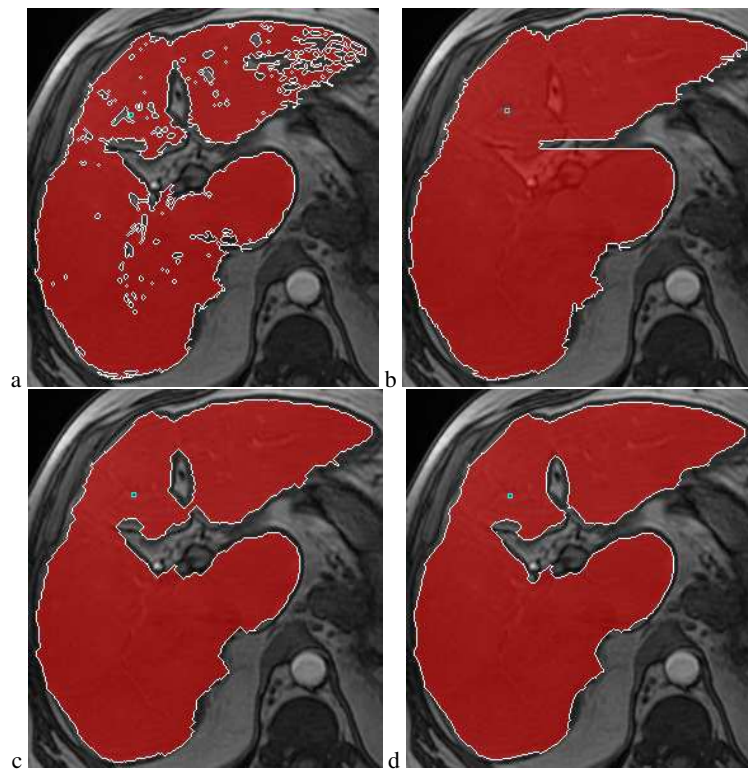


Fig. 6. Postprocessing of segmentation result: original region (a), scan fill with undesirable inner region inclusion (b), morphological closing with undesirable conjunction (c) and neighbourhood analysis (d)

The implemented model is a discrete form of the curve, composed of a set of points (snaxels). The total energy of the snake is minimised by moving each snaxel to a position of minimum local energy. The three main elements of the model are:

- balloon force, based on image properties and contour shape;
- image energy, minimising snake energy in the image domain;
- internal energies, responsible for flexibility, tension and topology of the curve.

Apart from a manual segmentation mode on a single image, this tool also has the ability to perform fast, semi-automatic segmentation on a series of images by contour propagation. The snake can operate directly on the source image or can use its gradient amplitude, calculated with custom liver-adapted filter.

Balloon force The first main element of the developed model is a balloon force [7] that inflates the contour and pushes it towards desired image elements. This method overcomes the limitations of the original snake model [12], which has a limited scope and have to be placed close to the segmented object boundary. A snake with the inflation force can be easily placed inside the segmented area without the need of precise initialisation. This feature is particularly helpful in liver segmentation, because of the its large surface and boundary length (see Fig. 7).

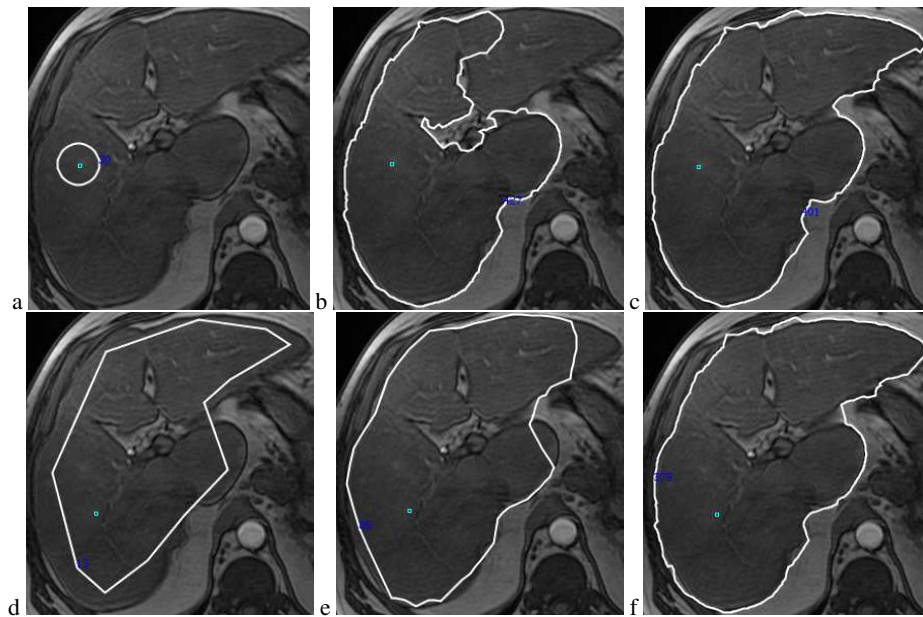


Fig. 7. Initialisation insensitivity: small snake placed inside the region (from a to c) gives similar result to more accurate initialisation (from d to f)

The balloon force needs a direction for moving the snake points. One of the most common approaches is to use the normal vector of each snaxel [7]. In the developed model, a different method was used. For a simple shape with a small points count (less than 30) the central point of the curve is used to calculate the growing directions. For more complicated curves, the vectors are based on the skeleton points [14] of an approximated simplified version of the curve. The vector of each point is constructed from its position and the closest skeletal points (see Fig. 8). In comparison with the normal vectors calculation, this approach results in a more uniform points

distribution during the growth and helps in preserving the correct topology. Along with the dynamic topology modification, it also speeds up the process of adaptation to complicated shapes of segmented regions (see Fig. 9).

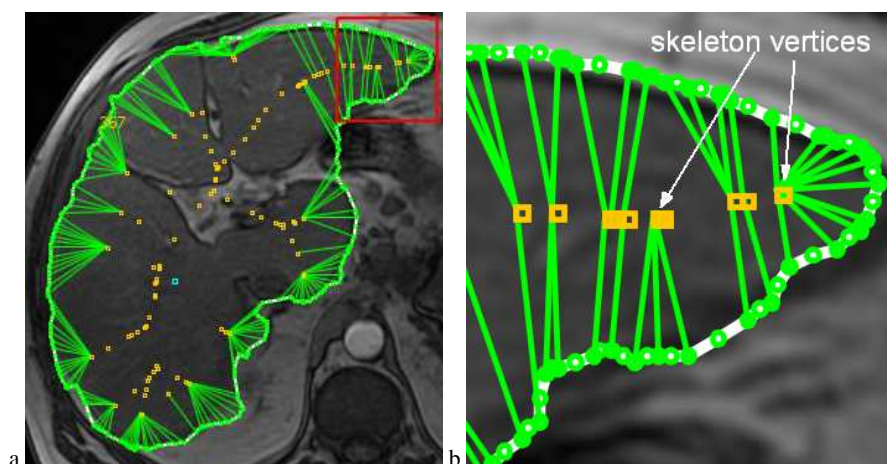


Fig. 8. Visualization of balloon force vectors: expansion directions are calculated from positions of each snaxel and the closest skeleton vertex (visible unconnected vertices were generated as a result of topology modification and will have their vectors calculated in the next iteration)

Movement of the snake points is constrained with conditions similar to the region growing method described earlier. Again, a snake point can advance along its directional vector when it meets the criteria of intensity difference between its current position, destination point and the start region. Additionally, contour points can not be moved into the area already covered by the snake and the maximal movement distance in one iteration can be limited. This force was customised to achieve fast expansion behaviour, taking into account the relatively large area of the liver. The high rate of the inflation can result in irregularities in the curve shape, therefore several topology optimisation procedures were developed.

Image energy Image energy is the second main element of the developed model. It puts a snake point in the position of the lowest energy within its local neighbourhood. The energy is measured by the image gradient amplitude value: a pixel with the highest gradient value has the lowest energy. The thickness of the liver boundaries often results in the occurrence of many highest valued pixels in the snake point boundary. In this case the pixel with the closest or furthest distance from the snake point, de-

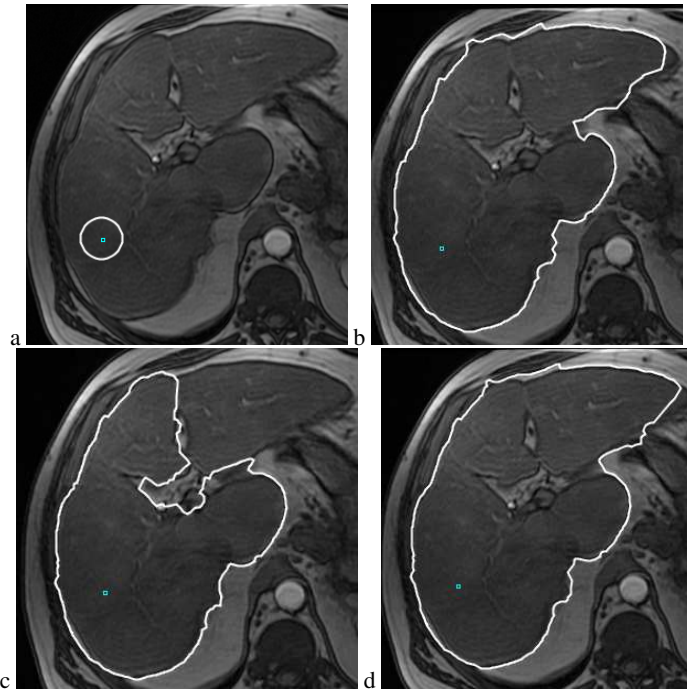


Fig. 9. Influence of the type of balloon force on the segmentation time: the evolution of the original snake (a) after 50 grow cycles using the skeleton-based balloon force (b) is significantly faster than the adaptation using the normal-based force (c), which needs another 50 iterations to equal the first method (d)

pending on user preference, will be chosen (see Fig. 10). This property allows the snake to contract or expand on the most significant boundaries, which is useful in the contour propagation process.

Topology optimisation Topology of the snake during the evolution process undergoes a constant optimisation. Internal constraints forbid the curve points from getting into undesirable locations. Points cannot move to a position already occupied by the snake surface, but can still share the same location. This could lead to snaxel redundancy and creation of unwanted loops, therefore the optimisation algorithm is searching for overlapping points and removes unnecessary snaxels between them.

Internal smoothing and tension forces also affect the contour shape. Fast inflation using the balloon force may cause significant irregularities in the shape of the curve and for that reason internal constraints are necessary. Specifically, these forces

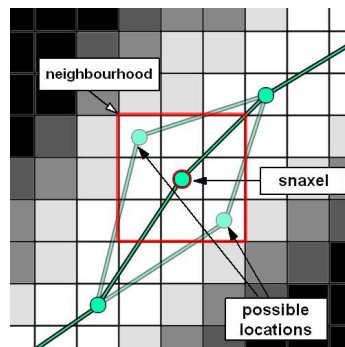


Fig. 10. Image force with possible snaxels positions: within a neighbourhood (marked by red square) with many image energy minimums, a snaxel can move to two outermost locations, resulting a small expansion or shrinkage of the curve

control the distance of a point from its neighbours points of the contour, preventing the point from moving too far, disrupting the snake smoothness and rigidity. These constraints can also be turned off for the points already positioned in their local minimum, affecting only the still-evolving snaxels and consequently speeding up the process.

New snake points are also added between existing snaxels, providing a simple subdivision scheme and allowing the curve to grow into further and more complex areas. Maximal point count can be limited, however usually the number of points quickly stabilises after the first few iterations and undergoes only minor oscillations during the evolution.

Contour propagation The presented model is capable of propagation over a series of images. This feature was used to implement a fast, semi-automatic segmentation tool. The user initialises a snake on one image. Then, the curve evolves and is copied to the next image, becoming the initial contour for the next snake. This process can be continued for the whole series of images (see Fig. 11). The deformation process in this case relies mainly on the image energy, because the pre-initialised snake from the previous image usually is already placed close to the boundaries. The process benefits from the image energy expansion/contraction preference, which can be set to correspond the actual change (growing/shrinking) of the liver region on consecutive slices.

The main limitation of this technique is the necessity of a small distance between slices in the data set. A relatively large spacing between subsequent images causes

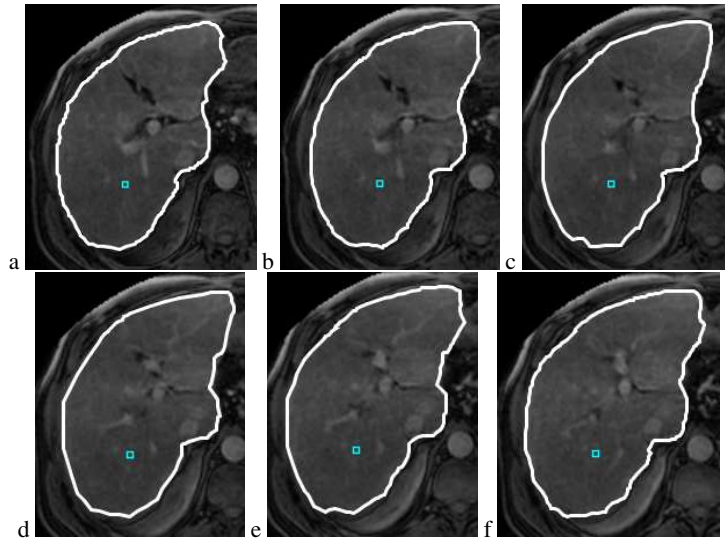


Fig. 11. Contour propagation over a series of images: manual initialisation on the first image (a) and a result of progression over 5 consecutive slices (from b to f)

abrupt changes in the segmented organ shape, making the reference contour less usable. In practice, this method was proven effective on sets with distance between two successive slices up to 3 mm.

2.3 Visualization

HIST contains three visualization tools: multiplanar reconstruction [20],[13], volume rendering [21] based on texture mapping [8],[5] and isosurface extraction with marching cubes algorithm [15]. All these methods can be used to present both segmentation result and the entire data set. These methods are commonly used in medical purposes and provide fast, real-time interaction.

Multiplanar reconstruction module enables visualization of the data set slices in other directions than the default one. The user can point a cursor to indicate the planes position and the reconstruction view will be instantly updated. An example with a segmentation result is presented in Fig. 12. Bilinear interpolation was used to achieve sufficient quality of low resolution sets.

Volume rendering of a loaded data set was achieved with texture mapping. The main idea of this method is to display a set of polygons with cross-sectioned images of the original data set mapped on. All three series of planes (transversal, coronal

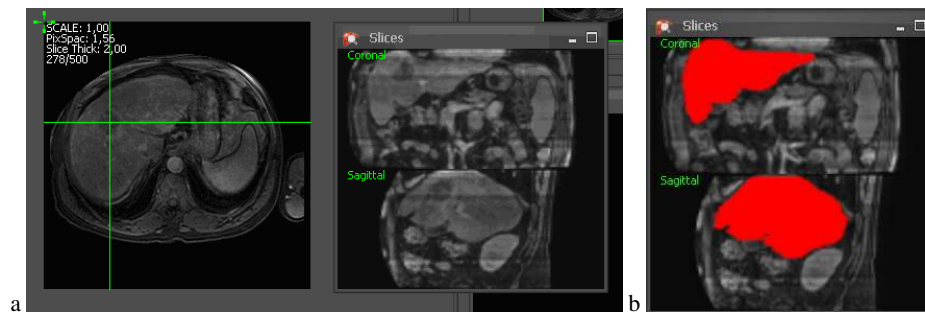


Fig. 12. Multiplanar reconstruction: original image with cross-section planes marker and reconstructed coronal and sagittal planes (a), reconstructed planes with marked segmentation (b)

and sagittal) are reconstructed with the multiplanar module. Combined planes are displayed using Java3D, which is an OpenGL wrapper. This enables fast hardware accelerated rendering and interaction. The set can be viewed from arbitrary angle and opacity characteristics can be instantly changed (see Fig. 13 and 14). Dynamic plane-switching was implemented to compensate the resolution differences between axes.

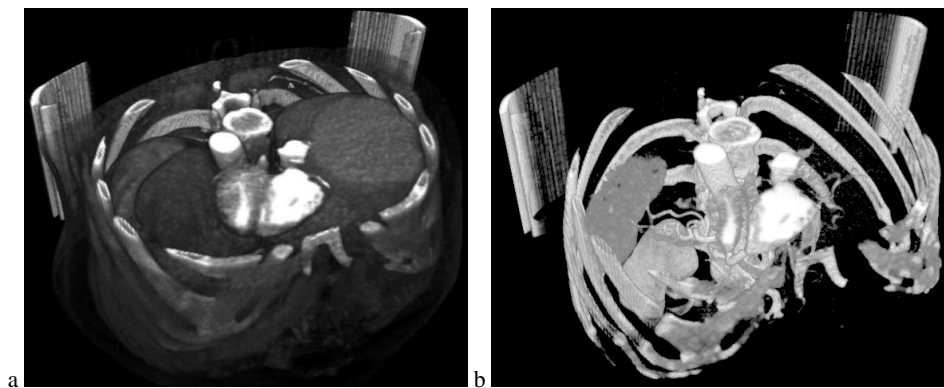


Fig. 13. Example of transparency modification usage: volume rendering of an entire abdominal CT (data courtesy of OsiriX [23]) (a) and a tissue separation achieved by dynamic transparency adjustment (b)

Isosurface extraction with marching cubes algorithm [15] is the last visualization method available in HIST. Unlike volume rendering, this method takes only a specific part of the data and converts it into a triangle mesh. The main challenge

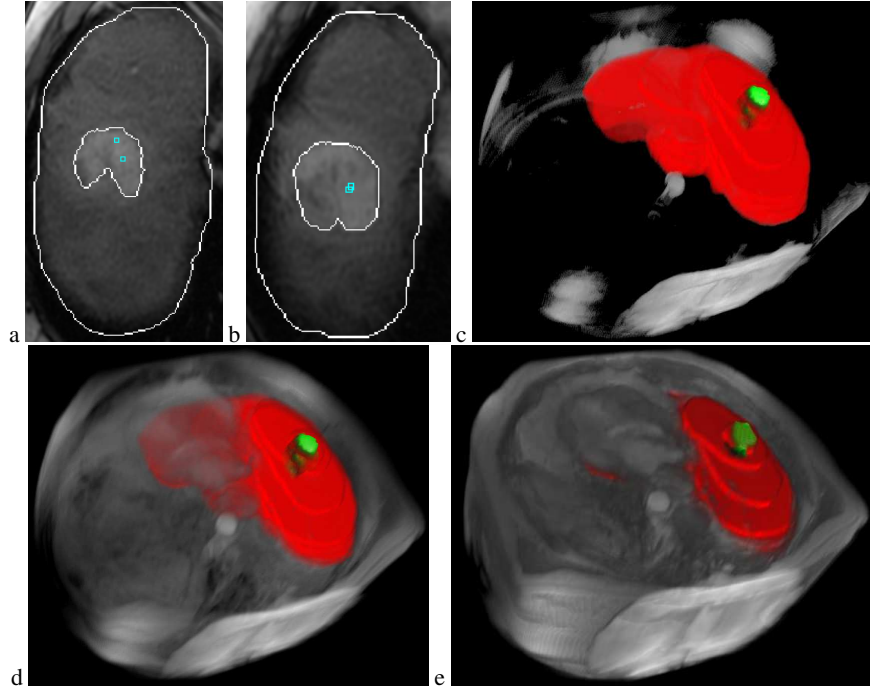


Fig. 14. Visualization of liver volume (red) and pathology (green): regions segmented on original images (a and b) and volume rendering in various transparency modes (from c to e)

of this method is to achieve a correct smoothing of the final mesh, which can be generated from the entire data set or only from the segmentation result. In the first case, smoothing can be easily performed by interpolating the values of each boundary voxels. Unfortunately, segmentation results are provided in the form of binary maps. Usage of specific smoothing algorithms is then necessary [18]. Furthermore, low spatial resolution of data sets can cause characteristic stairway artifacts in the final mesh. Moreover, the basic shape of the segmentation result has to be closely preserved and therefore simple smoothing algorithms are unacceptable from a medical point of view. The smoothing is performed with one iteration of morphological dilation, resulting in one pixel-wide smooth boundary around the region. Intensity $I_p(i)$ of each boundary pixel is calculated as:

$$I_p(i) = I_1 + I_2(s_i/s_{max}),$$

where I_1 and I_2 are predefined constants, s_i is the count of voxels adjacent to the i pixel, and s_{max} is the maximal count of possible adjacent voxels in $3 \times 3 \times 3$ neighbour-

hood cube. The binary format of the region map imposes that the $I_p(i)$ should take a value between 0 and 1, therefore $I_1 + I_2 = 1$. Sampling factor of the region can be also altered to create more consistent map. The results of the developed method are presented in Fig. 15.

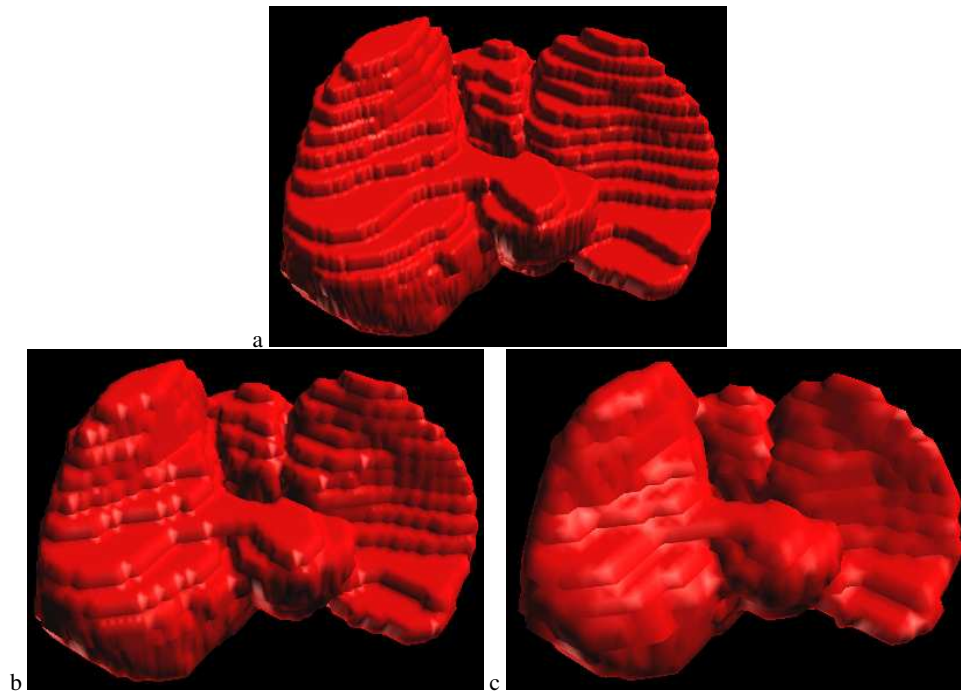


Fig. 15. Smoothing of a segmented liver isosurface: original result with visible artifacts (a), smoothing with factors of 4 (b) and 6 (c)

2.4 Application overview

The application was implemented in Java SE 6 platform, using Swing and Java3D libraries. This choice was based on availability of many useful, free libraries and cross-platform support.

The main tools proposed by our application are:

- browser for DICOM files (single/multi-frames, image directories and DICOMDIR files) and plain image files (BMP, JPEG, PNG);

- various image filters: bilateral filter [25], morphological closing, median filter, edge detection, histogram equalisation, thresholding, gray scale quantization, LUT colour palette applying and more;
- manual and automated segmentation with region growing, active contour and LiveWire [3] (on single and multiple images);
- segmentation results editing, grouping, saving, and merging;
- multiplanar reconstruction module;
- interactive 3D visualization of data sets and segmentation results with volume rendering and isosurface extraction;
- segmentation results analysis and comparison (described in Section 3);
- customisable user interface with many themes (see Fig. 16).

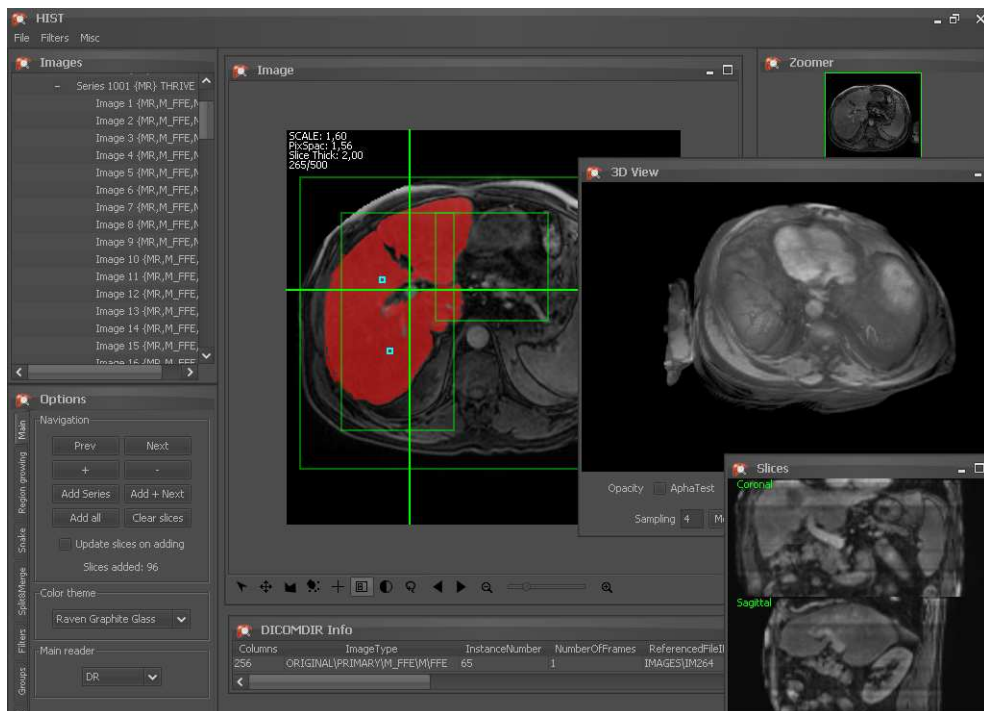


Fig. 16. Application main window with visible segmentation result, 2D multiplanar reconstruction and 3D visualization

2.5 User interface

The application had to be equipped with a comprehensible user interface. Many segmentation methods propose many parameters, making specific adjustments complex. Some works have shown that a user can simultaneously manipulate only up to four parameters [9] of solved task. This problem was taken into consideration and options of every implemented tool were divided into two sets: a small number of basic parameters and a set of advanced options, that could be understood during the usage of the application. Default values of these parameters were also adjusted to the specific task of liver segmentation.

3. Experimental validation

A preliminary validation of the implemented tools was performed, in which the segmentation time and quality was tested. Ideally, the usage of the tools should lead to a significant shortening of the segmentation time while maintaining the quality of the results.

3.1 Data sets

Quality and efficiency of the implemented tools were tested on various hepatic MRI sets, gathered at the Pontchaillou University Hospital, Rennes, France. The available data sets were generally divided into two groups: a smaller series of higher resolution images (usually over a dozen of 512x512 px) and smaller resolution sets of about 100 of images in series. Two representative sets were chosen:

- Set 1, containing 18 512x512 px images with 0,74x0,74x8,5 mm voxel size;
- Set 2, containing 100 256x256 px images with voxel size of 1,56x1,56x2 mm.

3.2 Evaluation

Segmentation quality was measured with two commonly used [10] error measures. The first one is the Overlap Error (OE), defined as:

$$OE(A, B) = 100(1 - (|A \cap B| / |A \cup B|)) \quad (2)$$

where A and B are two segmented pixel sets. The 0 value indicates that the two sets are identical and 100 that the sets do not overlap.

The second measure is the Relative Volume Difference (RVD), defined as:

$$RVD(A, B) = 100((|A| - |B|)/|B|) \quad (3)$$

where A is the tested segmentation and B is the reference. This measure can indicate a tendency to over- or undersegmentation of the method. It must be used along with other measures, because the actual sets overlap is not considered.

3.3 Results

Reference segmentations, used in evaluation of the examined methods, were performed made by the application author using the implemented manual tools. The intrapersonal variation was also an important factor, therefore several manual segmentations of each set were performed with 24-hour interval. Thereafter, the sets were evaluated with each other using the described measures. Set 1 was segmented using the automated region growing method and the propagating active contour was tested on Set 2. The averaged comparison of the results are enclosed in Table 1. Table 2 contains average time of manual and tool-guided segmentation. Figures 17 and 18 presents sampled tool-guided segmentation results along with the manually outlined referential areas.

Table 1. Manual and tool-guided segmentation quality

Data set	Manual		Tool-guided	
	OE	RVD	OE	RVD
Set 1	7,26 ± 4,46	-2,35 ± 7,59	10,94 ± 8,56	-3,14 ± 10,68
Set 2	7,84 ± 3,98	-1,35 ± 5,41	10,1 ± 4,29	-4,49 ± 5,56

Table 2. Segmentation time (in minutes)

Data set	Manual seg.	Tool-aided seg.
Set 1	25-30	5-10
Set 2	40-45	6-8

3.4 Outcome

Experimental results show a significant decrease in segmentation time. For instance, in the case of Set 2 from 45 minutes for manual segmentation to less than 10 minutes

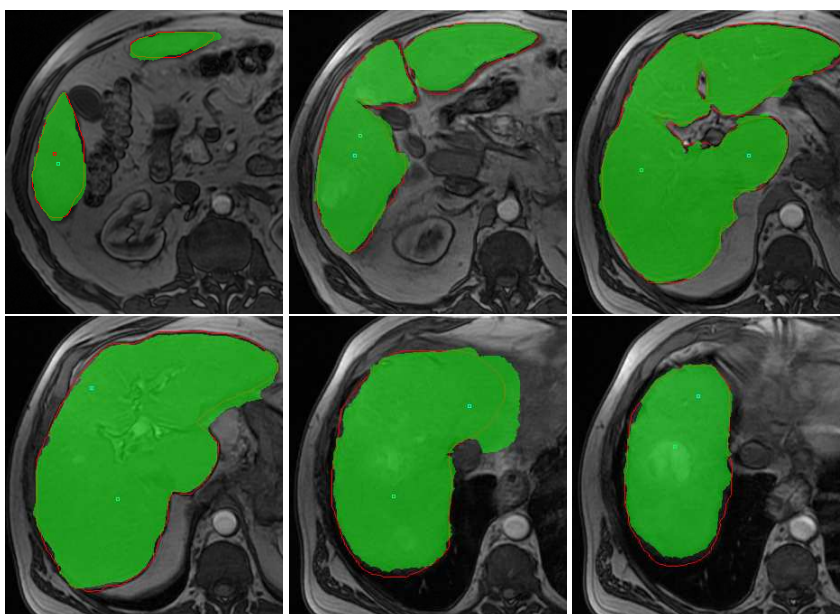


Fig. 17. Set 1 segmentation sample: the green regions are the result of the automated region growing segmentation and the referential region boundaries are marked in red

using the implemented tools. It has to be noted that these results depend heavily on personal experience and manual abilities, especially in the case of manual segmentation. As for the quality of the segmentation, the average Overlap Error was about 10% and the Relative Volume Difference was -2,69%. High values of the standard deviation of these results was caused mainly by the ambiguous cases in the outermost images of series, where the organ segmentation was particularly difficult because of the partial volume effect [2]. Obtained results were compared to the results of hepatic CT segmentation from MICCAI 2007 Grand Challenge [10]. For interactive methods, OE was $8\% \pm 2,6\%$ and RVD was equal $-2,81\% \pm 3,62\%$. Intrapersonal variation results was also remarkable, with 7,7% for OE and 1,8% for RVD.

A general tendency to undersegmentation in the developed methods was noted. Fortunately, the results of semi-automated methods are never conclusive and in the most cases can be easily corrected with available enhancement tools, preserving the time reduction.

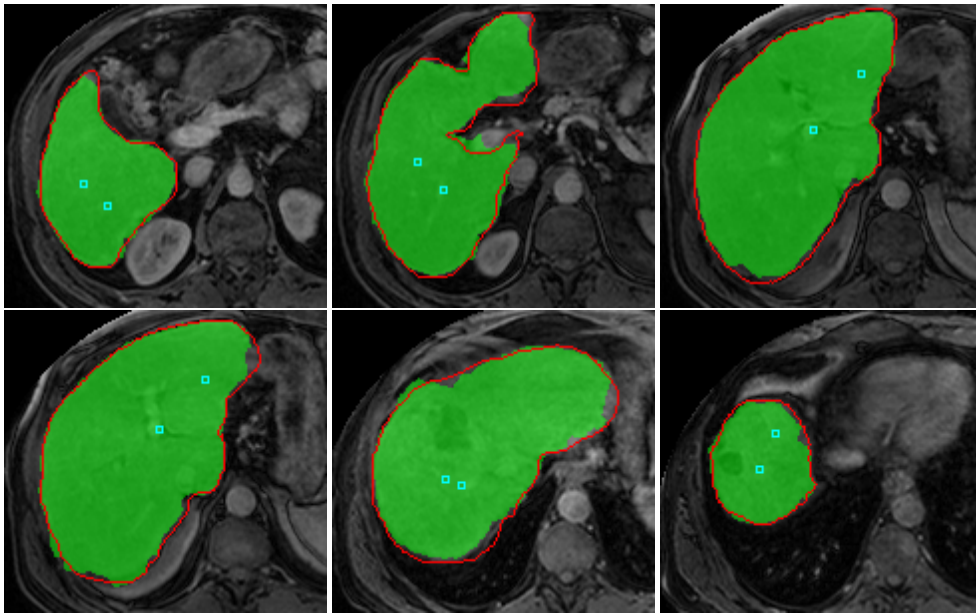


Fig. 18. Set 2 segmentation sample: the green regions are the result of the active contour segmentation and the referential region boundaries are marked in red

4. Conclusion and future work

Segmentation tools implemented in the presented application were proven to be useful in the challenging case of liver segmentation. Despite the early stage of their development, these methods have given promising results that encourage further study. Contour propagation was particularly effective in the case of large data sets and region growing was useful in the case of larger resolution images, that required more precise handling. However, the complexity of liver segmentation makes stronger generalisation of this kind difficult and is forcing the study of different methods. Currently, the highest priority task is the evaluation of the implemented tools on real-life cases in medical environment.

HIST provides a stable base for further improvement of segmentation and visualization algorithms. Current state-of-the-art techniques provide a wide variety of potentially effective methods [10]. Apart from expanding the current single-image based tools, implementation of methods working on full 3D data set is also planned. The visualization techniques can also benefit from greater usage of hardware acceleration on modern GPUs [22].

Acknowledgements

The authors are grateful to Prof. Johanne Bézy-Wendling for inspiring discussion and useful comments.

References

- [1] Adams R., Bischof L., Seeded region growing, *Pattern Analysis and Machine Intelligence*, IEEE Transactions on, 16(6):641–647, 1994.
- [2] Ballester M.A., Zisserman A.P., Brady M., Estimation of the partial volume effect in MRI, *Medical Image Analysis*, 6(4):389–405, 2002.
- [3] Barrett W., Mortensen E.N., Interactive live-wire boundary extraction, *Medical Image Analysis*, 1(4):331–341, 1997.
- [4] Brigham and Women’s Hospital, 3D Slicer, <http://www.slicer.org> Accessed at 2011-07-04.
- [5] Cabral B., Cam N., Foran J., Accelerated volume rendering and tomographic reconstruction using texture mapping hardware, In *Proceedings of the 1994 symposium on Volume visualization, VVS ’94*, pages 91–98, 1994.
- [6] Campadelli P., Casiraghi E., Esposito A., Liver segmentation from computed tomography scans: A survey and a new algorithm, *Artificial Intelligence in Medicine*, 45:185–196, 2009.
- [7] Cohen L.D., On active contour models and balloons, *CVGIP: Image Underst.*, 53:211–218, 1991.
- [8] Cullip T.J., Neumann U., Accelerating volume reconstruction with 3D texture hardware, Technical report, Chapel Hill, NC, USA, 1994.
- [9] Halford G.S., Baker R., McCredden J.E., Bain J.D., How many variables can humans process?, 2004, University of Queensland, Brisbane, Australia, and Griffith University, Brisbane, Australia.
- [10] Heimann T., van Ginneken B., Styner M.A., et al, Comparison and evaluation of methods for liver segmentation from CT datasets, *IEEE Transactions on Medical Imaging*, 28:1251–1265, 2009.
- [11] Horowitz S.L., Pavlidis T., Picture segmentation by a directed split and merge procedure, In *International Conference on Pattern Recognition*, pp. 424–433, 1974.
- [12] Kass M., Witkin A., Terzopoulos D., Snakes: Active contour models, *International Journal of Computer Vision*, 1(4):321–331, 1988.
- [13] Kramer D.M., Kaufman L., Guzman R.J., Hawryszko C., A general algorithm for oblique image reconstruction, *Computer Graphics and Applications*, IEEE, 10(2):62–65, 1990.

- [14] Lee D.T., Medial axis transformation of a planar shape, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-4(4):363–369, 1982.
- [15] Lorensen W.E., Cline H.E., Marching cubes: A high resolution 3D surface construction algorithm, *Computer Graphics*, 21(4):163–169, 1987.
- [16] Meinzer H.P., Thorn M., Cardenas C.E., Computerized planning of liver surgery - an overview, *Computer and Graphics*, 26(4):569–576, 2002.
- [17] Mudry K.M., Plonsey R., Bronzino J.D., *Biomedical imaging*, CRC Press, 2003.
- [18] Neubauer A., Forster T.M., Wegenkittl R., Mroz L., Bühler K., Interactive display of background objects for virtual endoscopy using flexible first-hit ray casting, *VisSym (Joint EG - IEEE TCVG Symp. on Visualization)*:301–304, 2004.
- [19] Rasband W., ImageJ - image processing and analysis in Java, <http://rsbweb.nih.gov/ij/> Accessed at 2011-07-04.
- [20] Rhodes M.L., Glenn W.V., Azaawi Y.M., Extracting oblique planes from serial ct sections, *J. Comput. Assist Tomogr.*, 4(5):649–657, 1980.
- [21] Sabella P., A rendering algorithm for visualizing 3D scalar fields, In *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '88, pp. 51–58, New York, 1988.
- [22] Shihao Ch., Guiqing H., Chongyang H., Rapid texture-based volume rendering, In *Environmental Science and Information Application Technology, 2009, ESIAT 2009, International Conference on*, volume 2, pp. 575–578, 2009.
- [23] OsiriX Software, DICOM sample image sets, <http://pubimage.hcuge.ch:8080/> Accessed at 2011-08-21.
- [24] ITK-SNAP Team, ITK-SNAP home page, <http://www.itksnap.org> Accessed at 2011-07-04.
- [25] Tomasi C., Manduchi R., Bilateral filtering for gray and color images. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, IEEE Computer Society, 1998.

HIST - APLIKACJA DO SEGMENTACJI OBRAZÓW WĄTROBY

Streszczenie HIST (ang. Hepatic Image Segmentation Tool – narzędzie do segmentacji obrazów wątroby) jest napisaną w języku Java aplikacją do segmentacji i wizualizacji obrazów medycznych, wyspecjalizowaną w segmentacji obrazów wątroby. Artykuł ten zawiera

przeгляд możliwości aplikacji, opis zaadaptowanych algorytmów segmentacji i wizualizacji oraz ich eksperymentalną walidację. Aplikacja oferuje dwie główne metody segmentacji, oparte o algorytmy rozrostu regionów i aktywnego konturu, dostosowane do segmentacji wątroby. Narzędzia wizualizacyjne aplikacji wykorzystują rekonstrukcję multiplanarną, rendering wolumetryczny oraz ekstrakcję izopowierzchni.

Słowa kluczowe: segmentacja wątroby, aktywny kontur, rozrost regionów, rendering wolumetryczny, rekonstrukcja multiplanarna, ekstrakcja izopowierzchni.

Artykuł zrealizowano w ramach pracy badawczej S/WI/2/08