Volume 23 Number 4

ACTA CYBERNETICA

Szeged, 2018

ACTA CYBERNETICA

Information for authors. Acta Cybernetica publishes only original papers in the field of Computer Science. Manuscripts must be written in good English. Contributions are accepted for review with the understanding that the same work has not been published elsewhere. Papers previously published in conference proceedings, digests, preprints are eligible for consideration provided that the author informs the Editor at the time of submission and that the papers have undergone substantial revision. If authors have used their own previously published material as a basis for a new submission, they are required to cite the previous work(s) and very clearly indicate how the new submission offers substantively novel or different contributions beyond those of the previously published work(s). There are no page charges. An electronic version of the published paper is provided for the authors in PDF format.

Manuscript Formatting Requirements. All submissions must include a title page with the following elements: title of the paper; author name(s) and affiliation; name, address and email of the corresponding author; an abstract clearly stating the nature and significance of the paper. Abstracts must not include mathematical expressions or bibliographic references.

References should appear in a separate bibliography at the end of the paper, with items in alphabetical order referred to by numerals in square brackets. Please prepare your submission as one single PostScript or PDF file including all elements of the manuscript (title page, main text, illustrations, bibliography, etc.).

When your paper is accepted for publication, you will be asked to upload the complete electronic version of your manuscript. For technical reasons we can only accept files in LaTeX format. It is advisable to prepare the manuscript following the guidelines described in the author kit available at `http://www.inf.u-szeged.hu/kutatas/acta-cybernetica/information-for-authors#AuthorKit` even at an early stage.

Submission and Review. Manuscripts must be submitted online using the editorial management system at `http://cyber.bibl.u-szeged.hu/index.php/actcybern/submission/wizard`. Each submission is peer-reviewed by at least two referees. The length of the review process depends on many factors such as the availability of an Editor and the time it takes to locate qualified reviewers. Usually, a review process takes 6 months to be completed.

Subscription Information. Acta Cybernetica is published by the Institute of Informatics, University of Szeged, Hungary. Each volume consists of four issues, two issues are published in a calendar year. Subscription rates for one issue are as follows: 5000 Ft within Hungary, €40 outside Hungary. Special rates for distributors and bulk orders are available upon request from the publisher. Printed issues are delivered by surface mail in Europe, and by air mail to overseas countries. Claims for missing issues are accepted within six months from the publication date. Please address all requests to:

Acta Cybernetica, Institute of Informatics, University of Szeged
P.O. Box 652, H-6701 Szeged, Hungary
Tel: +36 62 546 396, Fax: +36 62 546 397, Email: `acta@inf.u-szeged.hu`

Web access. The above information along with the contents of past and current issues are available at the Acta Cybernetica homepage `https://www.inf.u-szeged.hu/en/kutatas/acta-cybernetica` .

László Lovász
Department of Computer Science
Eötvös Loránd University
Budapest, Hungary
lovasz@cs.elte.hu

Dana Petcu
Department of Computer Science
West University of Timisoara, Romania
petcu@info.uvt.ro

Heiko Vogler
Department of Computer Science
Dresden University of Technology
Dresden, Germany
Heiko.Vogler@tu-dresden.de

Gerhard J. Woeginger
Department of Mathematics and
Computer Science
Eindhoven University of Technology
Eindhoven, The Netherlands
gwoegi@win.tue.nl

Acta Cybernetica **23** (2018) 981–993.

On Derivation Languages of a Class of Splicing Systems

Kalpana Mahalingam[ab], Prithwineel Paul[b], and Erkki Mäkinen[c]

Abstract

Derivation languages are language theoretical tools that describe halting derivation processes of a generating device. We consider two types of derivation languages, namely Szilard and control languages for splicing systems where iterated splicing is done in non-uniform way defined by Mitrana, Petre and Rogojin in 2010. The families of Szilard (rules and labels are mapped in a one to one manner) and control (more than one rule can share the same label) languages generated by splicing systems of this type are then compared with the family of languages in the Chomsky hierarchy. We show that context-free languages can be generated as Szilard and control languages and any non-empty context-free language is a morphic image of the Szilard language of this type of system with finite set of rules and axioms. Moreover, we show that these systems with finite set of axioms and regular set of rules are capable of generating any recursively enumerable language as a control language.

Keywords: Splicing systems, Szilard languages, Control languages

1 Introduction

The information regarding terminal derivation processes of a generative device is well studied in the literature. Each rule of the generative system in question is labeled and the given sequence of labels is considered as the output of the computation. The set of all such words constitute a language. When the labelling is done in a one to one fashion, the set of all labeled sequences is called a Szilard language. Szilard languages have been defined for a variety of generative mechanisms (for Chomsky grammars [8, 9, 14, 12], for regulated rewritings [5, 18] and for grammar systems [6, 10], to name a few) and their closure and decidability properties and

[a] corresponding author

[b] Department of Mathematics,Indian Institute of Technology, Madras, Chennai - 36. E-mail: kmahalingam@iitm.ac.in, prithwineelpaul@gmail.com

[c] Faculty of Natural Sciences/Computer Science, University of Tampere, Finland. E-mail: em@sis.uta.fi

DOI: 10.14232/actacyb.23.4.2018.1

complexity ([2, 3]) have been studied extensively. The study of the derivation process has also been extended in the context of P systems ([15, 16, 21, 20]). Since P systems are parallel computing devices and several rules can be used in a single computation step, one to one mapping of the labels may lead to complications (see [1]). To overcome this, all rules that are used in a computation step are labeled with the same symbol or some of them are labeled with the empty symbol λ. The set of all sequences of labels that lead to a halting computation is called the control language. The characterization of such control languages in terms of Chomsky hierarchy has been discussed for various P systems [15, 16, 21, 20]. Note that we use the terms control word and control language in the sense of [15, 16, 21, 20] which differs from their original use [19].

In this paper, we extend the study of derivation languages to a particular type of splicing systems, namely to $EGenSS$'s defined in [11]. In $EGenSS$, iterated splicing is done in non-uniform way. More specifically, at any step splicing is done between a string generated in the previous step and axioms. Splicing systems were introduced by Head [7] as a theoretical model to study the recombinant behaviour of DNA molecules. Splicing operation between two strings is defined to be a cut and paste operation where the both strings are cut at particular sites and the first component of the first string is pasted with the second component of the second string, and the second component of the first string is pasted with the first component of the second string to obtain two new strings. It is well known that if in a splicing system, the set of axioms and the set of rules are finite, the system cannot generate beyond regular languages [4, 13]. Different versions of such finite splicing systems are capable of generating recursively enumerable languages [13]. One such version is the concept of extended H system. Extended H systems can be thought of as the set of all DNA strings satisfying a particular property. However, it is not clear when this desired set of strings is obtained. In order to understand this, a labelling of rules is done. The sequence of labels of the applied rules that leads to a terminal derivation is included in the language of the system. In this paper we consider the derivation languages of a variant of splicing system defined in [11].

The paper is organized as follows: Section 2 presents some basic notations. Section 3 defines the Szilard language of splicing systems and shows that there exist regular and context-free languages that are Szilard languages. It is well known that $\{aa\}$ and $\{a^{4n} \mid n \geq 1\}$ cannot be a Szilard language of a Chomsky grammar ([12]). However, we show that $\{aa\}$ can be the Szilard language of $EGenSS$ with finite set of axioms and rules where splicing is done in non-uniform way. The language $\{a^{4n} \mid n \geq 1\}$ cannot be Szilard language of this type of system with finite set of axioms and rules but it can be a Szilard language if the system contains regular set of axioms and finite set of rules. Also we will show that every non-empty context-free language is a morphic image of the Szilard language of an $EGenSS$ with finite set of axioms and rules. In Section 4, we define the control language of a splicing system. We show that both the families of regular and context-free languages are proper subsets of the family of control languages generated by the $EGenSS$'s with finite set of axioms and rules. Also we show that any recursively enumerable language is a control language of this type of splicing system with finite set of axioms and regular set of rules when the rules can also be labeled with λ. We end the paper with a few concluding remarks.

2 Preliminaries

For basic notations and results of formal language theory we refer the reader to [13, 17, 19]. Let V be an alphabet and let V^* denote the set of all strings over V. The empty string is denoted by λ. If F is a family of languages, then $F \setminus \{\lambda\}$ denotes the λ-free family of languages. By FIN, REG, CF, CS, RE we denote the families of finite, regular, context-free, context-sensitive and recursively enumerable languages, respectively.

A word u is a prefix (resp. suffix) of a word v if v is of the form $v = uw$, $w \in V^*$ (resp. $v = wu$). The set of all prefixes (resp. suffixes) of v is denoted as $pref(v)$ (resp. $suff(v)$). The length of a string w is denoted by $|w|$.

A morphism is a mapping from $h : \Sigma^* \to \Delta^*$ such that $h(xy) = h(x)h(y)$ where $x, y \in \Sigma^*$. A morphism $h : \Sigma^* \to \Delta^*$ is called non-erasing, if $h(x) \neq \lambda$ for all $x \in \Sigma$.

A splicing rule over V is a string of the form $r = u_1\#u_2\$u_3\#u_4$, where $u_i \in V^*$, $1 \leq i \leq 4$ and $\#, \$ \notin V$. The maximum of $|u_i|, 1 \leq i \leq 4$, is the radius of the splicing rule r.

An extended generating H system is a 4-tuple $H = (V, T, A, R)$, where V is the alphabet, $T \subseteq V$ is the terminal alphabet, $A \subseteq V^*$ is the set of axioms, and $R \subseteq V^*\#V^*\$V^*\#V^*; \#, \$ \notin V$ is the set of splicing rules. For a splicing rule $r = u_1\#u_2\$u_3\#u_4$ and an ordered pair of words $x, y \in V^*$, denote, $\sigma_r(x, y) = \{u = x_1u_1u_4y_2, v = y_1u_3u_2x_2$ where $x = x_1u_1u_2x_2,\ y = y_1u_3u_4y_2$, for some $x_1, x_2, y_1, y_2 \in V^*\}$. We also write $(x, y) \vdash_r (u, v)$, where u and v are referred to as the first and the second components obtained when r is applied to x and y. Let R be a set of splicing rules and L a language, then $\sigma_R(L)$ is defined as

$$\sigma_R(L) = \bigcup_{r \in R} \bigcup_{w_1, w_2 \in L} \sigma_r(w_1, w_2).$$

If L_1, L_2 are any two languages, then $\sigma_R(L_1, L_2)$ is denoted as

$$\sigma_R(L_1, L_2) = \bigcup_{x_1 \in L_1} \bigcup_{x_2 \in L_2} \sigma_R(x_1, x_2),$$

where

$$\sigma_R(x_1, x_2) = \bigcup_{r \in R} \sigma_r(x_1, x_2).$$

A non-uniform variant for extended generating splicing system is defined in [11]. The system is an extended generating H system, $H = (V, T, A, R)$ with the additional requirement that splicing at any step occurs between a generated word in the previous step and an axiom:

$$\tau_R^0(A) = A,\ \tau_R^{i+1}(A) = \sigma_R(\tau_R^i(A), A), i \geq 0\ ,\ \tau_R^*(A) = \textstyle\bigcup_{i \geq 0} \tau_R^i(A).$$

The system is denoted as $EGenSS\ H$. The language generated by an $EGenSS\ H$ is defined as $L_n(H) = T^* \cap \tau_R^*(A)$. The family of languages generated by $EGenSS$'s in non-uniform way is denoted by $\mathscr{L}_n(EGenSS)$.

The class of languages generated by non-uniform extended generating splicing systems with finite set of axioms and finite set of rules equals REG [11].

3 Szilard language associated with splicing systems

In this section we extend the concept of Szilard languages to splicing systems. We define Szilard languages of $EGenSS$'s and compare them with the family of languages in the Chomsky hierarchy. We also show that the language $\{aa\}$ which is not the Szilard language of a Chomsky grammar [12] is indeed the Szilard language of an $EGenSS$.

A labeled extended generating H system is a construct of the form $\gamma = (V_1, T_1, A_1, R_1, Lab)$, where $H = (V_1, T_1, A_1, R_1)$ is an extended generating splicing system as defined in Section 2, and Lab, $Lab \cap V_1 = \emptyset$ is a set of labels that are used to uniquely name the rules. Since the splicing in the system works in the non-uniform manner, we call this type of splicing systems non-uniform labeled extended generating splicing systems. A derivation in the splicing system is terminal if it obeys one of the following two patterns:
(1) $(x_0, y_0) \vdash^{a_1} (x_1, y_1^0), (x_1, y_1) \vdash^{a_2} (x_2, y_2^0), (x_2, y_2) \vdash^{a_3} (x_3, y_3^0), \cdots$
$(x_{n-1}, y_{n-1}) \vdash^{a_n} (x_n, y_n^0)$, or
(2) $(y_0, x_0) \vdash^{a_1} (x_1, y_1^0), (y_1, x_1) \vdash^{a_2} (x_2, y_2^0), (y_2, x_2) \vdash^{a_3} (x_3, y_3^0), \cdots$
$(y_{n-1}, x_{n-1}) \vdash^{a_n} (x_n, y_n^0)$,
where $x_i \in V_1^*$, $y_i \in A_1$, for $0 \le i \le n-1$, $x_n \in T_1^*$, $y_i^0 \in V_1^*$, and $a_i \in Lab$, for $1 \le i \le n$.

The set of all such label sequences $a_1 a_2 \cdots a_n$ of the applied rules that leads to a terminal derivation constitute $SZ(\gamma)$, the Szilard language of the non-uniform labeled extended generating H system γ. The family of Szilard languages generated by the non-uniform labeled extended generating splicing systems is denoted by $\mathscr{S}\mathscr{Z}_{LEGenSS_n}(FL_1, FL_2)$, with axioms from the family FL_1 and rules from the family FL_2 .

In the following we show that there exist an infinite regular and a non-regular context-free language which is the Szilard language of a finite labeled $EGenSS$.

Theorem 1. *$REG \cap \mathscr{S}\mathscr{Z}_{LEGenSS_n}(FIN, FIN) \neq \emptyset$.*

Proof. We construct a labeled H system such that $SZ(\gamma) = \{a^n \mid n \ge 1\}$. Let $\gamma = (V_1, T_1, A_1, R_1, Lab)$ be an labeled $EGenSS$ where $V_1 = \{X, S_1, Y, Z\}, T_1 = \{X, Y\}, A_1 = \{XS_1Y, ZS_1Y, ZY\}$, $R_1 = \{a : \#S_1Y\$Z\#\}$ and $Lab = \{a\}$. If the strings XS_1Y and ZS_1Y are spliced, then

$$(X \mid S_1Y,\ Z \mid S_1Y) \vdash^a (XS_1Y,\ ZS_1Y).$$

The rule can be applied iteratively to XS_1Y and ZS_1Y to obtain XS_1Y and ZS_1Y. A terminal derivation is obtained if XS_1Y is spliced with ZY:

$$(X \mid S_1Y,\ Z \mid Y) \vdash^a (XY,\ ZS_1Y).$$

Any other possibility does not lead to a terminal derivation and, hence, $SZ(\gamma) = \{a^n \mid n \geq 1\}$.

□

It was shown in [12] that the language $\{aa\}$ cannot be a Szilard language of any type-0 grammar. We show that there exists a splicing system γ such that $SZ(\gamma) = \{aa\}$.

Theorem 2. *The language $\{aa\}$ is a Szilard language of a finite labeled EGenSS.*

Proof. We construct a labeled splicing system $\gamma = (V_1, T_1, A_1, R_1, Lab)$ such that $SZ(\gamma) = \{aa\}$. We define, $V_1 = \{X_1^1, Y_0^1, u_1^1, u_3^1, \alpha, \beta, \beta_1, X_0^2, Y_0^2\}$, $T_1 = \{X_1^1, u_1^1, \beta, \beta_1, Y_0^2\}$, $A_1 = \{X_1^1u_1^1\alpha u_1^1\alpha u_1^1\beta X_0^2, Y_0^1u_3^1\beta\beta_1Y_0^2\}$, $Lab = \{a\}$ and
$R_1 = \{a : u_1^1\#\alpha u_1^1\beta\$u_3^1\#\beta\beta_1\}$. There is a terminal derivation

$$\begin{gathered}(X_1^1u_1^1\alpha u_1^1 \mid \alpha u_1^1\beta X_0^2\ ,\ Y_0^1u_3^1 \mid \beta\beta_1Y_0^2) \vdash^a (X_1^1u_1^1\alpha u_1^1\beta\beta_1Y_0^2\ ,\ Y_0^1u_3^1\alpha u_1^1\beta X_0^2)\\ (X_1^1u_1^1 \mid \alpha u_1^1\beta\beta_1Y_0^2\ ,\ Y_0^1u_3^1 \mid \beta\beta_1Y_0^2) \vdash^a (X_1^1u_1^1\beta\beta_1Y_0^2\ ,\ Y_0^1u_3^1\alpha u_1^1\beta\beta_1Y_0^2).\end{gathered}$$

It is easy to verify that no other derivation is possible and, hence, $SZ(\gamma) = \{aa\}$.

□

In the following we show that there exists a regular language that cannot be a Szilard language of any finite labeled *EGenSS*.

Theorem 3. $REG \setminus \mathscr{SZ}_{LEGenSS_n}(FIN, FIN) \neq \emptyset$.

Proof. Let $L = \{a^{4n} \mid n \geq 1\}$. To derive a contradiction, suppose a finite labeled *EGenSS*, $\gamma = (V_1, T_1, A_1, R_1, Lab)$ such that $SZ(\gamma) = \{a^{4n} \mid n \geq 1\}$. The system contains only one rule, $a : u_1^1\#u_2^1\$u_3^1\#u_4^1$, where $u_1^1, u_2^1, u_3^1, u_4^1 \in V_1^*$. Hence, there exists a terminal derivation with label sequence a^4 as follows:
$(x_0, y_0) = (x_0^1u_1^1 \mid u_2^1x_0^2\ ,\ y_0^1u_3^1 \mid u_4^1y_0^2) \vdash^a (x_0^1u_1^1u_4^1y_0^2\ ,\ y_0^1u_3^1u_2^1x_0^2)$
$(x_1, y_1) = (x_1^1u_1^1 \mid u_2^1x_1^2\ ,\ y_1^1u_3^1 \mid u_4^1y_1^2) \vdash^a (x_1^1u_1^1u_4^1y_1^2\ ,\ y_1^1u_3^1u_2^1x_1^2)$
$(x_2, y_2) = (x_2^1u_1^1 \mid u_2^1x_2^2\ ,\ y_2^1u_3^1 \mid u_4^1y_2^2) \vdash^a (x_2^1u_1^1u_4^1y_2^2\ ,\ y_2^1u_3^1u_2^1x_2^2)$
$(x_3, y_3) = (x_3^1u_1^1 \mid u_2^1x_3^2\ ,\ y_3^1u_3^1 \mid u_4^1y_3^2) \vdash^a (x_3^1u_1^1u_4^1y_3^2\ ,\ y_3^1u_3^1u_2^1x_3^2)$,
where $x_3^1u_1^1u_4^1y_3^2 \in T_1^*, x_i^j, y_i^j \in V_1^*$, $0 \leq i \leq 3$, $1 \leq j \leq 2$ such that $x_0^1u_1^1u_4^1y_0^2 = x_1^1u_1^1u_2^1x_1^2$, $x_1^1u_1^1u_4^1y_1^2 = x_2^1u_1^1u_2^1x_2^2$ and $x_2^1u_1^1u_4^1y_2^2 = x_3^1u_1^1u_2^1x_3^2$. Then we have the following cases:

1. $x_2^1 = x_3^1u_1^1u_2^1\alpha_1$, $\alpha_1 \in pref(x_3^2)$

2. $x_2^1 = x_3^1u_1^1\alpha_1$, $\alpha_1 \in pref(u_2^1)$

3. $x_2^1 = x_3^1\alpha_1$, $\alpha_1 \in pref(u_1^1)$

4. $x_2^1 = \alpha_1$, $\alpha_1 \in pref(x_3^1)$.

If $x_2^1 = x_3^1 u_1^1 u_2^1 \alpha_1$, then $x_2 = x_3^1 u_1^1 u_2^1 \alpha_1 u_1^1 u_2^1 x_2^2$ and $(x_2, y_3) \vdash^a (x_4, y_4)$ where $x_4 \in T_1^*$ generating a^3. The cases $x_2^1 = x_3^1 \alpha_1$, $\alpha_1 \in pref(u_1^1)$ and $x_2^1 = \alpha_1$, $\alpha_1 \in pref(x_3^1)$ lead to the same contradiction. If $x_2^1 = x_3^1 u_1^1 \alpha_1$ where $\alpha_1 \in pref(u_2^1)$, then $x_2 = x_2^1 u_1^1 \alpha_1 u_1^1 u_2^1 x_2^2$. Note that $\alpha_1 \notin T_1^*$, otherwise, the rule a can be applied to x_2 and y_3 which leads to a terminal derivation generating a^3.

Now from $x_0^1 u_1^1 u_4^1 y_2^2 = x_1^1 u_1^1 u_2^1 x_1^2$, we have the following possibilities:
$x_0^1 \in pref(x_1^1), x_0^1 \in x_1^1\ pref(u_1^1), x_0^1 \in x_1^1 u_1^1\ pref(u_2^1)$ and $x_0^1 \in x_1^1 u_1^1 u_2^1\ pref(x_1^2)$. Similarly, from $x_1^1 u_1^1 u_4^1 y_1^2 = x_2^1 u_1^1 u_2^1 x_2^2$, we obtain $x_1^1 \in pref(x_2^1)$, $x_1^1 \in x_2^1\ pref(u_1^1)$, $x_1^1 \in x_2^1 u_1^1\ pref(u_2^1)$ and $x_1^1 \in x_2^1 u_1^1 u_2^1\ pref(x_2^2)$. If $x_0^1 = x_1^1$ or $x_1^1 = x_2^1$, then the system will generate strings $a^i \notin L$.

All other cases mentioned will either increase in the length of u_1^1 or u_2^1 or both or increase in the number of independent occurrences of prefixes of u_1^1 and prefixes of u_2^1. Since L is infinite and x_0 is a finite string, this is not possible and, hence, L is not a Szilard language of any finite splicing system.

□

In the following we construct a labeled $EGenSS$ with regular set of axioms such that $\{a^{4n} \mid n \geq 1\} \in \mathscr{S}\mathscr{Z}_{LEGenSS_n}(REG, FIN)$.

Example 1. We construct a labeled $EGenSS$, $\gamma = (V_1, T_1, A_1,\ R_1, Lab)$ such that $SZ(\gamma) = \{a^{4n} \mid n \geq 1\}$. Let $V_1 = \{X, u_1^1, \beta, Y, Z\}, T_1 = \{X, \beta, Y\}, A_1 = \{X u_1^{4n} \beta Y \mid n \geq 1\} \cup \{Z\beta Y\}, R_1 = \{a : \# u_1 \beta Y \$ Z \# \beta Y\}$, and $Lab = \{a\}$. It is clear that any derivation of the above system reaches a terminal derivation only after applying the rule a four times. Since the set A_1 is regular, $\{a^{4n} \mid n \geq 1\} \in \mathscr{S}\mathscr{Z}_{LEGenSS_n}(REG, FIN)$.

Next, we show that there exists a context-free language that is the Szilard language of a finite labeled $EGenSS$. From [11] we know that this type of splicing systems cannot generate non-regular languages.

Theorem 4. *$CF \cap \mathscr{S}\mathscr{Z}_{LEGenSS_n}(FIN, FIN) \neq \emptyset$.*

Proof. Let $\gamma = (V_1, T_1, A_1, R_1, Lab)$ be a labeled $EGenSS$ where $V_1 = \{X_1, Y, Y_1, a, Z, A_1, A_2\}$, $T_1 = \{X_1, Y_1\}$, $A_1 = \{X_1 Y, Z a A_1 Y, Z A_2 Y_1\}$, and $Lab = \{1, 2, 3, 4, 5\}$. The system contains the rules $R_1 = \{\mathbf{1} : X_1 \# Y \$ Z \# a A_1 Y$, $\mathbf{2} : a \# A_1 Y \$ Z \# a A_1 Y$, $\mathbf{3} : aa \# A_1 Y \$ Z \# A_2 Y_1$, $\mathbf{4} : a \# a A_2 Y_1 \$ Z \# A_2 Y_1$, $\mathbf{5} : X_1 \# a A_2 Y_1 \$ Z A_2 \# Y_1\}$.

Initially, the rule 1 can be applied to the strings $X_1 Y$ and $Z a A_1 Y$. After applying rule 1, the string $X_1 a A_1 Y$ is produced. Then rule 2 can be applied iteratively $(n-1)$ times, to generate a string of the form $X_1 a^n A_1 Y$. Then rule 3 can be applied to obtain the string $X_1 a^n A_2$. After application of rule 3, only rules 4 or 5 are applicable. If rule 4 is applied $(n-1)$ times, $X a A_2 Y$ is produced. Finally, rule 5 is applied to obtain the terminal string $X_1 Y_1$. If a derivation does not start with 1, it does not lead to a terminal derivation. Thus, $SZ(\gamma) = \{12^n 34^n 5 \mid n \geq 1\}$. □

It was shown in [9] that context-free languages can be represented as a morphic images of Szilard languages associated with the left most derivations of context-free grammars. However, the class of all languages obtained by taking the morphic image of Szilard languages of the context-free grammars in general are incomparable with context-free languages [9]. In the following we show a result similar to that in [9], that each context-free languages can be expressed as a morphic image of the Szilard language of a finite labeled *EGenSS*.

Theorem 5. *Every non-empty context free language is a morphic image of the Szilard language of a finite labeled EGenSS.*

Proof. Let L be a non-empty context-free language and let $G = (N, T, P, S)$ be a grammar in Greibach normal form such that $L = L(G)$. The rules in P are of the form, $D_i \to a\alpha$ and $D_i \to a$, where $\alpha \in N^+, D_i \in N, a \in T$ and $N = \{D_1, D_2, \ldots, D_n\}$. We show that there exists a finite splicing system γ such that $L = h(SZ(\gamma))$ where h is a non-erasing morphism from Lab^* to T^*.

We construct a labeled splicing system $\gamma = (V_1, T_1, A_1, R_1, Lab)$ such that $L = h(SZ(\gamma))$ where

- $V_1 = \{Y\} \cup N \cup \Delta_1$, for $\Delta_1 = \{Y_a \mid D_i \to a\alpha, \alpha \in N^+, D_i \in N, a \in T\} \cup \{Y_a \mid D_i \to a \in P, D_i \in N, a \in T\}$;
- $T_1 = \{Y\}$;
- $A_1 = \{YSY\} \cup \Delta_2 \cup \Delta_3$, where
  $\Delta_2 = \{Y\alpha Y_a \mid D_i \to a\alpha, D_i \in N, a \in T, \alpha \in N^+\}$ and
  $\Delta_3 = \{YY_a \mid D_i \to a, D_i \in N, a \in T\}$.
- $R_1 = \{(a^i_{\alpha_k} : Y\alpha_k\#Y_a\$YD_i\#) \mid D_i \to a\alpha_k \in P\} \cup\{(a^i : Y\#Y_a\$YD_i\#) \mid D_i \to a \in P\}$ that is, for every rule of the form $D_i \to a\alpha_k$ in G, where $a \in T, \alpha_k \in N^+$, k a positive integer, a splicing rule $(a^i_{\alpha_k} : Y\alpha_k\#Y_a\$YD_i\#)$ is constructed. Similarly, if there exists a rule $D_i \to a$, then a splicing rule $(a^i : Y\#Y_a\$YD_i\#)$ is constructed.
- $Lab = \{a^i_{\alpha_k} \mid D_i \to a\alpha_k \in P\} \cup \{a^i \mid D_i \to a \in P\}$.

Finally, we define the morphism $h : Lab^* \to T^*$ such that $h(a^i_{\alpha_k}) = h(a^i) = a$ where $a_{\alpha^i_k}, a^i \in Lab$ and $a \in T$.

We first prove that $L(G) \subseteq h(SZ(\gamma))$. Any computation in G starts from S and after sequential application of the rules in P, a string over T is generated. The splicing rules simulating the rules $D_i \to a\alpha_k$, $D_i \to a, D_j \to a\alpha_k$, and $D_j \to a$ in P are labeled with $a^i_{\alpha_k}, a^i, a^j_{\alpha_k}$, and a^j, respectively. Suppose a terminal string, say, w is generated in G. If the corresponding labeled rules are also applied in γ, a terminal derivation can be obtained. If the labels of the applied splicing rules are concatenated, a string over Lab, say, w_1 is generated. But if the morphism h is applied w, each occurrence of $a^i_{\alpha_k}, a^i, a^j_{\alpha_k}$, and a^j is replaced by a. Hence, if $w \in L(G)$, we have $w = h(w_1) \in h(SZ(\gamma))$ where $w_1 \in SZ(\gamma)$.

Next we prove the inclusion $h(SZ(\gamma) \subseteq L(G)$. Let $w = h(w_1)$ where $w_1 \in SZ(\gamma)$. Let $w_1 = a_1a_2 \ldots a_n \in SZ(\gamma)$, i.e., there exists a terminal derivation in γ with which w_1 is generated. In G, computations starts from S. If the rules in

G are applied in the same sequence as the (simulated) labeled rules are applied in γ, a terminal string is generated. So, a terminal string in G and a terminal derivation in γ is obtained at the same time. Again, $h(a^i_{\alpha_k}) = h(a^i) = a$, and hence, $h(w_1) = w \in L(G)$. So, we can conclude $h(SZ(\gamma)) \subseteq L(G)$. □

4 Control languages of splicing systems

In the previous section we discussed the Szilard languages associated with splicing systems. In this section we define control languages associated with splicing systems and compare the family of control languages generated by the labeled *EGenSS* with the family of languages in the Chomsky hierarchy. Control languages have already been discussed for several variants of, for example, tissue P systems, spiking neural P systems, and P systems with isotonic array grammars ([15, 16, 21, 20]) to name a few. We extend the concept of control languages to splicing systems and show that all non-empty regular and context-free languages are indeed control languages of finite labeled *EGenSS*.

We conisder a labeled extended generating H system $\gamma = (V_1, T_1, A_1, R_1, Lab)$, working in non-uniform manner, where V_1, T_1, A_1, R_1, and Lab are as defined in Section 3 except that multiple rules in R_1 can be assigned with the same label. Also a single rule cannot be mapped with different labels. The rules can also be labeled with the empty string λ. The concatenation of the labels of the applied splicing rules in any terminal derivation will form a string over *Lab*. It is called a control word of the labeled *EGenSS*. The set of all control words constitute the control language of the labeled *EGenSS* γ. It is denoted by $CTL(\gamma)$.

The family of control languages generated by any labeled extended generating splicing system $\gamma = (V_1, T_1, A_1, R_1, Lab)$ with $card(A) \leq n$ and $rad(R) \leq m$, where $n, m \geq 1$, is denoted by $\mathscr{RL}_{CTL}([n],[m])$. When no restriction on the number n of axioms or on the maximal radius m are considered but n and m are still finite, they are simply replaced with FIN. If empty labels are allowed then the family is denoted by $\mathscr{RL}_{CTL_\lambda}([n],[m])$. If the system contains axioms from F_1 and rules from F_2, for some families of languages F_1 and F_2, then the family of control languages generated by the systems is denoted by $\mathscr{RL}_{CTL}(F_1, F_2)$. When the system contains λ-labeled rules, we denote it by $\mathscr{RL}_{CTL_\lambda}(F_1, F_2)$.

$\mathscr{L}_n(EGenSS)$ with finite set of axioms and finite set of rules [11] with no restriction on the radius of the splicing rules equals the class of regular languages. In the next theorem, we show that the class of non-empty regular languages are contained in $\mathscr{RL}_{CTL}(FIN,[1])$.

Theorem 6. *$(REG \setminus \{\lambda\}) \subseteq \mathscr{RL}_{CTL}(FIN,[1])$.*

Proof. Let L be a λ-free regular language. Then there exists a regular grammar $G = (N, T, P, S)$ such that $L = L(G)$. Suppose the non-terminals N of G are $D_i, 1 \leq i \leq n$, where $D_1 = S$ is the start symbol. We now construct a finite labeled *EGenSS* γ such that $L = L(G) = CTL(\gamma)$. The rules in P are of the form $D_i \to aD_i$, $D_i \to aD_j (i \neq j)$, and $D_i \to a$, $D_i, D_j \in N$, and $a \in T$.

Let $\gamma = (V_1, T_1, A_1, R_1, Lab)$ be a labeled *EGenSS*, where

- $V_1 = \{X, Y, Y_1, D_1, D_2, \ldots, D_n\} \cup \Delta_1$, for $\Delta_1 = \{Y_a \mid D_i \to aD_i\} \cup \{Y_a \mid D_i \to aD_j(i \neq j)\} \cup \{Y_a \mid D_i \to a \in P, a \in T\}$;
- $T_1 = \{Y\} \cup \Delta_1$;
- $A_1 = \{XD_1Y\} \cup \Delta_2 \cup \Delta_3 \cup \Delta_4$, where
  1. $\Delta_2 = \{YD_jY_a \mid D_i \to aD_j \in P,\ a \in T\}$,
  2. $\Delta_3 = \{YD_iY_a \mid D_i \to aD_i \in P,\ a \in T\}$,
  3. $\Delta_4 = \{Y_aY_1 \mid D_i \to a \in P,\ a \in T\}$;
     (The set $(\Delta_2 \cap \Delta_3)$ may or may not be disjoint.)
- The rules in R_1 are of the form
  $(a : D_i\#Y_a\$D_i\#Y)$, for $D_i \to aD_i, a \in T$,
  $(a : D_j\#Y_a\$D_i\#Y)$, for $D_i \to aD_j, i \neq j, a \in T$,
  $(a : Y_a\#Y_1\$D_i\#Y)$, for $D_i \to a, a \in T$, where $1 \leq i, j \leq n$;
- $Lab = T$.

Every rule in G is simulated by a corresponding splicing rule with the required label a that corresponds to the grammar rule under consideration. Thus, every $w \in L(G)$ can be simulated by a terminal derivation in γ and vice versa. The sequence of splicing rules reach a terminal derivation only when the rule $(a : Y_a\#Y_1\$D_i\#Y)$ corresponding to the rule $D_i \to a, a \in T$, is applied. Thus, $L(G) = CTL(\gamma)$.

□

In the next theorem we show that, every non-empty context-free language can be a control language of a finite labeled *EGenSS*.

Theorem 7. $(CF \ \setminus \{\lambda\}) \subseteq \mathscr{RL}_{CTL}(FIN, FIN)$.

Proof. Let L be any non-empty context-free language such that $\lambda \notin L$. Then let $G = (N, T, P, S)$ be a context-free grammar in Greibach normal form such that $L = L(G)$. We construct a finite labeled *EGenSS*, $\gamma = (V_1, T_1, A_1, R_1, Lab)$ such that $L = L(G) = CTL(\gamma)$. Let $\gamma = (V_1, T_1, A_1, R_1, Lab)$ be a labeled splicing system where:

- $V_1 = \{X, Y, Y_1\} \cup N \cup \Delta_1$, for $\Delta_1 = \{Y_a \mid D \to a\alpha, \alpha \in N^+, a \in T\} \cup \{Y_a \mid D \to a \in P, a \in T, , D \in N\}$;
- $T_1 = \{Y\} \cup \Delta_1$;
- $A_1 = \{XSY\} \cup \Delta_2 \cup \Delta_3$, where
  1. $\Delta_2 = \{Y\alpha Y_a \mid D \to a\alpha \in P, a \in T, \alpha \in N^+, D \in N\}$,

2. $\Delta_3 = \{YY_a, Y_aY_1 \mid D \to a \in P, a \in T, D \in N\}$;

- R_1 contains the following rules :
  For $D \to a\alpha \in P$, we have, $\{(a : Y\alpha\#Y_a\$XD\#Y), (a : Y\alpha\#Y_a\$YD\#Y)\} \cup$
  $\{(a : Y\alpha\#Y_a\$YD\#\beta_1\beta_2 \ldots \beta_iY) \mid \beta_i \in N, 1 \le i \le (n-1)\}\cup$
  $\{(a : Y\alpha\#Y_a\$YD\#\beta_1\beta_2 \ldots \beta_n) \mid \beta_i \in N\}$
  For $D \to a \in P$, we have, $\{(a : Y\#Y_a\$YD\#\beta_1\beta_2 \ldots \beta_iY) \mid \beta_i \in N, 1 \le i \le (n-1)\}\cup$ $\{(a : Y_a\#Y_1\$XD\#Y)\}$ $\cup\{(a : Y\#Y_a\$YD\#\beta_1\beta_2 \ldots \beta_n) \mid \beta_i \in N\} \cup \{(a : Y_a\#Y_1\$YD\#Y)\}$
  where $n = Max\{|\alpha| \mid D \to a\alpha \in P\}$;
- $Lab = T$.

Corresponding to each rule of the form $D \to a\alpha \in P$ there exist rules in γ labeled with a, $(a : Y\alpha\#Y_a\$XD\#Y)$, $(a : Y\alpha\#Y_a\$YD\#\beta_1Y)$, $(a : Y\alpha\#Y_a\$YD\#\beta_1\beta_2Y)$, $(a : Y\alpha\#Y_a\$YD\#\beta_1\beta_2\beta_3Y)$, $\ldots$, and $(a : Y\alpha\#Y_a\$YD\#\beta_1\beta_2 \ldots \beta_n)$. These rules can be applied to the pairs of strings XDY, $Y\alpha Y_a$ and $YDQY$, $Y\alpha Y_a$, where $Q \in N^*$, respectively. At first, $Y\alpha Y_a$ and XSY are spliced and $Y\alpha Y$ and XSY_a are produced. No rule is applicable to XSY_a, but $Y\alpha Y$ can be spliced further with the rules in the system. If XSY and Y_aY_1 are spliced together, it will produce XSY_1 and Y_aY. Strings of the form $YDQY$, where $Q \in N^*$, can be spliced with the strings $Y\alpha Y_a$ and YY_a to obtain YDY_a and $Y\alpha QY$ or YQY. After the application of the rule $(a : Y_a\#Y_1\$YD\#Y)$ to YDY and Y_aY_1, the strings YDY_1 and Y_aY are produced. The string Y_aY is a terminal string and the strings of labels of the rules applied are in the control language. The above construction of γ simulates the rules of P in R. The splicing rules in γ are applied in the same sequence as the rules are applied in the derivation $S \Rightarrow^* x$, for $x \in L(G)$. Thus $x \in L(G)$ iff there exist a terminal derivation in γ generating x. Whenever the rules $D \to a\alpha$, and $D \to a$ are applied to a non-terminal in G, the corresponding splicing rule labeled with a is applied in the system γ and vice versa. Thus, $L(G) = CTL(\gamma)$. □

In the following we show that there exists a context-sensitive language that cannot be the control language of any finite labeled *EGenSS*.

Theorem 8. *$CS \setminus \mathscr{RL}_{CTL\lambda}(FIN, FIN) \neq \emptyset$.*

Proof. Let $L = \{a^{2^n} \mid n \ge 0\}$ be a context-sensitive language. Assume that there exists a finite labeled *EGenSS*, $\gamma = (V_1, T_1, A_1, R_1, Lab)$ such that $CTL(\gamma) = \{a^{2^n} \mid n \ge 0\}$.
Since $a \in CTL(\gamma)$, there exist an 'a' labeled rule (say r_1) and $x_0, y_0 \in A_1$ such that $(x_0, y_0) \vdash^a_{r_1} (x_t, y'), x_t \in T_1^*$. Since $x_t \in T_1^*$, x_t cannot be spliced further and hence it is not possible to generate a^2, from the strings x_0, and y_0 and just by using the rule r_1. Therefore there exists an 'a' labeled rule r_2 such that $r_1 \neq r_2$ and $(x_0, y_0) \vdash^a_{r_2} (x_1, y')$, $(x_1, y_1) \vdash^a_{r_1} (x_t, y'')$, $x_t \in T_1^*$. Thus to generate a^{2^n}, for some n starting with x_0, y_0, there must exist 'a' labeled rules $r_1, r_2, \cdots r_k$ such that $k \le 2^n$. Since the number of rules in the system are finite, some of these rules are repeated recursively which will end up in generating strings of the type a^i such that $i \neq 2^n$. Hence $CTL(\gamma) \neq L$. □

The following theorem shows that the family of control languages generated by the labeled *EGenSS* with rules from a regular set where some of the rules are labeled with λ is equal to the family of recursively enumerable languages.

Theorem 9. $\mathscr{R}\mathscr{L}_{CTL_\lambda}(FIN, REG) = RE$.

Proof. The inclusion $\mathscr{R}\mathscr{L}_{CTL_\lambda}(FIN, REG) \subseteq RE$ follows from the Church-Turing Thesis. We have to prove only the inclusion $RE \subseteq \mathscr{R}\mathscr{L}_{CTL_\lambda}(FIN, REG)$. Let $G = (N, T, P, S)$ be a type-0-grammar in Kuroda normal form. We construct a labeled *EGenSS*, $\gamma = (V_1, T_1, A_1, R_1, Lab)$ such that $CTL_\lambda(\gamma) = L(G)$. Let $\gamma = (V_1, T_1, A_1, R_1, Lab)$ is a labeled *EGenSS* with $U = N \cup T \cup \{E\}$, where

- $V_1 = N \cup \{E, X, X^{'}, Y, Z\} \cup \{Y_\alpha \mid \alpha \in U\}$;
- $T_1 = \{X, Y, E\}$;
- $A_1 = \{XESY, XZ, ZY\} \cup \{ZY_\alpha \mid \alpha \in U\} \cup \{X^{'}\alpha Z \mid \alpha \in N \cup E\} \cup \{ZBCY \mid A \to BC \in P\} \cup \{ZCDY \mid AB \to CD \in P\} \cup \{ZY_aY \mid A \to a \in P\}$;
- R_1 contains the following rules:
  1. $(\lambda : Xw\#AY\$Z\#BCY)$, for $A \to BC \in P, w \in (N \cup \{E\})^*$
  2. $(\lambda : Xw\#ABY\$Z\#CDY)$, for $AB \to CD \in P, w \in (N \cup \{E\})^*$
  3. $(a : XwE\#AY\$ZY_a\#Y)$, for $A \to a \in P, w \in N^*$
  4. $(\lambda : XwE\#AY\$Z\#Y)$, for $A \to \lambda \in P, w \in N^*$
  5. $(\lambda : Xw\#\alpha Y\$Z\#Y_\alpha)$, for $\alpha \in N \cup E, w \in (N \cup \{E\})^*$
  6. $(\lambda : X^{'}\alpha\#Z\$X\#wY_\alpha)$, for $\alpha \in N \cup E, w \in (N \cup \{E\})^*$
  7. $(\lambda : X^{'}w\#Y_\alpha\$Z\#Y)$, for $\alpha \in N \cup E, w \in (N \cup \{E\})^*$
  8. $(\lambda : X\#Z\$X^{'}\#wY)$, for $w \in (N \cup \{E\})^*$;
- $Lab = T \cup \{\lambda\}$.

The above system is constructed in the same manner as in any standard proof of extended H system with finite set of axioms and regular set of rules that can generate *RE* languages. The splicing rules are labeled with the terminal symbol a that simulates the rule $A \to a$ in G and the rest of the rules are labeled with λ. The grammar G is in Kuroda normal form and any element $w \in L(G)$ can be generated by the application of the recursive rules $A \to BC$ and $AB \to CD$ rules in P in any manner and then by application of the terminating rules $A \to \lambda$ and $A \to a$ in the leftmost manner. The splicing rules (1) and (2) simulate the non-terminal recursive rules and the splicing rules in (3) and (4) simulate the terminating rules $A \to a$ and $A \to \lambda$, respectively. The splicing rules (3) and (4) are applicable only to the non-terminal symbol present between E and the right hand marker Y and by doing this the left-most derivation is simulated, since the terminating rules are applied in leftmost manner. The rules in γ from (1) to (4) simulate the rules in P. Rules from (5) to (8) are used to rotate the string inside the markers X and Y.

Note that the λ-labeled splicing rules in (1) and (2) can be applied any number of times. Also, the rotation rules from (5) to (8) are λ-labeled and they can also be applied any number of times. But the rules in (3) and (4) are applicable to the non-terminals in between E and Y. The a-rule in (3) eliminates one non-terminal (adjacent to E) from the string inside the markers X and Y in γ. It simulates the application of the rule $A \to a$ to the left most non-terminal in any derivation of G . The λ-labeled rule in (4) also works in the same manner. Thus, $w \in L(G)$ iff there exists a derivation $S \Rightarrow^* w$ and the system γ generates the string $XEY \in T_1^*$ in the first component of a step, i.e., a terminal derivation is obtained. Thus, $w \in L(G)$ iff $w \in CTL_\lambda(\gamma)$. □

5 Conclusion

We have defined the derivation languages of non-uniform variant of generating splicing systems and have compared them with the families of languages in the Chomsky hierarchy. We have shown that infinite regular and non-regular context-free languages can be Szilard languages of finite splicing systems. and that every non-empty context-free language is a morphic image of the Szilard language of a finite splicing system. Also we showed that the family of infinite regular and non-regular context-free languages are properly contained in the family of control languages of finite splicing systems. We also have shown that if the set of axioms are finite and the set of rules are regular and λ labeled rules are allowed, any recursively enumerable language can be generated as a control language of a non-uniform labeled extended generating splicing system. It will also be interesting to explore power of the derivation languages of other variants of splicing system.

References

[1] Ciobanu, G., Păun, G., Stefanescu, G. : Sevilla carpets associated with P systems. in BWMC 2003, Tarragona Univ., TR 26/03 (2003)

[2] Cojocaru, L., Mäkinen, E., Tiplea, F. L. : Classes of Szilard languages in $\mathscr{NC}$. In: 11th International Symposium on Symposium on Symbolic and Numeric Algorithms for Scientific Computing, pp. 299–306 (2009)

[3] Cojocaru, L., Mäkinen, E. : On some derivation mechanisms and the complexity of their Szilard languages. Theoretical Computer Science. 537, 87–96 (2014)

[4] Culik, K. II, Harju, T. : Splicing semigroups of dominoes and DNA. Discrete Appllied Matematics. 31, 261–277 (1991)

[5] Dassow, J., Păun, G. : Regulated rewriting in formal language theory. Springer, Berlin (1989)

[6] Dassow, J., Mitrana, V., Păun, G. : Szilard languages associated to co-operating distributed grammar systems. Stud. Cercet. Mat. 45, 403–413 (1993)

[7] Head, T. : Formal language theory and DNA : an analysis of the generative capacity of specific recombinant behaviours. Bulletin of Mathematical Biology. 49(6), 737–759 (1987)

[8] Mäkinen, E. : On context-free and Szilard languages. BIT. 24, 164–170 (1984)

[9] Mäkinen, E. : On homomorphic images of Szilard languages. International Journal of Computer Mathematics. 18, 239–245 (1986)

[10] Mihalache, V. : Szilard languages associated to parallel communicating grammar systems. Developments in Language Theory II, At the Crossroads of Mathematics, Computer Science and Biology, Magdeburg, Germany, July 1995, World Scientific, Singapore, 247–256 (1996)

[11] Mitrana, V., Petre, I., Rogojin, V. : Accepting splicing systems. Theoretical Computer Science. 411, 2414–2422 (2010)

[12] Păun, G. : On some families of Szilard languages. BULL. MATH. de la Soc. Sci. Math. de la R. S. de Roumanie Tome 27(75), 259–265 (1983)

[13] Păun, Gh., Rozenberg, G., Salomaa, A. : DNA Computing: New computing paradigms, Springer-Verlag, Berlin (1998)

[14] Penttonen, M. : On derivation language corresponding to context-free grammars. Acta Informatica. 3, 285–291 (1974)

[15] Ramanujan, A., Krithivasan, K. : Control words of transition P systems. BICTA 2012, Advances in Intelligent Systems and Computing. 145–155 (2012)

[16] Ramanujan, A., Krithivasan, K. : Control languages associated with spiking neural P systems. Romanian Journal of Information Science and Technology. 15(4), 301–318 (2012)

[17] Rozenberg, G., Salomaa, A. (Eds.) : Handbook of formal languages, vol. I-III, Springer-Verlag, Berlin (1997)

[18] Salomaa, A. : Matrix grammars with a left most restriction. Information Control. 20(2), 143–149 (1972)

[19] Salomaa, A. : Formal languages, Academic Press, New York (1973)

[20] Sureshkumar, W., Rama, R. : Chomsky hierarchy control on isotonic array P systems. International Journal of Pattern Recognition and Artificial Intelligence. 30(2), 10.1142/S021800141650004X (2016)

[21] Zhang, X., Liu, Y., Luo, B., Pan, L. : Computational power of tissue P systems for generating control languages. Information Sciences. 278, 285–297 (2014)

Received 23rd March 2017

Acta Cybernetica **23** (2018) 995–1015.

Constructive Model of the Natural Language

Viktor Shynkarenko[a] and Olena Kuropiatnyk[a]

Abstract

The paper deals with the natural language model. Elements of the model (the language constructions) are images with such attributes as sounds, letters, morphemes, words and other lexical and syntactic components of the language. Based on the analysis of processes of the world perception, visual and associative thinking, the operations of formation and transformation of images are pointed out. The model can be applied in the semantic NLP.

Keywords: image, image operations, constructive-synthesizing structure, natural language, language construct

1 Introduction

Information is one of the most important resources of the last decades. Considerable part of it is presented in verbal form in the natural languages (NL) and requires systematizing and automated processing to enable further acquisition of knowledge with the possibility of quick access to such knowledge. Systematization and further automation require formalization of the language concept and its components.

The problem of the natural language processing is conditioned by its features, such as permanent development including growth of the vocabulary and rules, redundancy, polysemy, and diversity of the forms of presentation.

There are many directions in the processing of texts in the NL (NLP) [8]: static and corps methods of NLP, usage of linguistic bases [22], [18], [33], finite state machines [19], [1] which is actively used, for example, in Nooj components [31], [32], regular expressions (in particular, in Semantic Tagger ANNIE Gate), and hidden Markov models [7].

Language as a set of constructions is represented: in the form of a neural probability model [2]; in the form of a tree-bank [17]; with the use of formulas of functional languages [11]; with the use of n-grams including those based on classes [5], and vector representation [14]. The latter approach is also used to create semantic models [15]. A graph presentation of semantics [22], [10], including semantic networks

[a] Department of Computer Informational Technologies, Dnipropetrovsk National University of Railway Transport named after Academician V. Lazaryan, Dnipro, Ukraine, E-mail: `shinkarenko_vi@ua.fm, olena.kuropiatnyk@gmail.com`

DOI: 10.14232/actacyb.23.4.2018.2

[6], thesauri based systems, tensor models [16], is also used for processing the texts in NL.

The paper is aimed to formalize the concept of language using the means of constructive-synthesizing structures [26]. The language will be represented in the form of the certain construction (in the broadest sense) which is the result of the corresponding constructive process of formation of images and words (phrases, sentences) as the attributes of images. As a result of this process, a constructive model of the NL is formed.

The paper represents development of the direction of mathematical and algorithmic constructivism which has already found its application in modelling the processes of alternatives ranking by the AHP method [24], adaptation of data structures in the operative memory [29], construction of a graph model of the text [28], etc.

2 State of the art

The main objective of NLP is the improvement of artificial intelligence (AI) systems. Today there are many sub-objectives aimed at improving the user interface with technical systems, quality of texts in the natural language, interaction between people, increasing the effectiveness of search engines and anti-plagiarism systems, etc.

Achieving the objectives involves solving of a number of problems: construction of models for representing the language and its constructions; development of methods for their processing and analysis, including specific applications (analysis of social media profiles for advisory systems, analysis of text messages, automatic translation and annotating...).

In the course of solving these tasks and achieving the objectives, the following questions arise: NL coverage (vocabulary, syntax, and semantics); possibilities of expanding the language model; ability to work with polysemous words, synonyms, homonyms (homographs); possibility of oral speech modelling (including recognition of homophones), taking into account personality of a speaker, approaching the natural thinking processes of an individual.

Models based on n-grams and tree-banks cover vocabulary and syntax and find their usage in Stanford Parser [21]; semantics is represented in graph and vector models [22], [15], [10]. All the models considered allow scaling the model. Modification of n-gram models due to probabilities [2] reduces their dimensionality. The possibility of modeling non-written speech is assumed in the models proposed in the works of Krak [12], [13]. There are no models used in NLP which take into account peculiarities of an information source. Models allowing classification of the language [9] also work regardless of its carrier.

A model that is close to the processes of human mental activity is considered in the work [4]; it is based on the figurative analysis and synthesis. Also, there is an approach proposed for constructing a conceptual model of the figurative analysis and synthesis of NL structures on the basis of psycho-physiological phenomena [3].

Studies have shown that NLP models work with texts, without taking into account specific features of their authors. At the same time, they are aimed at studying the concepts of the language and text as its construction. They are the most complete and close in its semantic nature to the studies carried out by the authors.

The proposed language model is based on the model of the human image system, and operations introduced for its construction and expansion do not contradict the above mentioned ones. Therefore, we have an opportunity to cover both the vocabulary and semantics of the oral and written speech, as well as to take into account phonetic and personal characteristics of the language and the particular individual, respectively.

3 Generalized constructive-synthesizing structure

The following triple [26] is called the generalized constructive-synthesizing structure (GCSSt):

$$C_G = \langle M,\ \Sigma,\ \Lambda \rangle\,,$$

where M is inhomogeneous structure medium (the main set of the elements), Σ is the signature comprising a set of relations and linking, substitution and output operations, as well as operations on attributes, Λ is the constructive axiomatics. GCSSt axiomatics is presented in the paper [26].

The constructive-synthesizing structure (CSSt) is intended for the formation of a plurality of structures using operations and relations of signatures, the rules for implementation of which are given in the axiomatics.

To form the structures, it is necessary to perform a number of transformations of CSSt: specialization, interpretation, and concretization [27]. Implementation of CSSt consists in formation of constructions (in this context, language constructs) of elements of CSSt medium by performing CSSt algorithms related to operations of the signature.

4 Specialization of CSSt of human images

Everything that surrounds an individual, the real and the virtual things (processes, entities, events and phenomena), as well as the individual him/herself, including material (tissues, organs) and nonmaterial (emotion, feelings) components, will be called the prototype, as a certain integral part of the world which is considered in isolation.

Image sensitivity is a characteristic feature of any individual. We understand the image as a representation of the prototype, its properties on some physical medium. Such a medium can be an individuals memory as a part of the nervous system, animals memory, computer, and computer networks.

Specialization involves determining the application environment, i.e., semantic nature of the CSSt medium, a finite set of operations and their semantics, operation attributes, as well as the order of their performing. Let us consider CSSt specialization of the human image system:

$$C = \langle M,\, \Sigma,\, \Lambda \rangle_{\,S} \mapsto {}_{S}C_h = \langle M_h,\, \Sigma_h,\, \Lambda_h \rangle\,, \tag{1}$$

where $M_h = T \bigcup N$ is a heterogeneous scalable medium, T is a set of terminals – images, N is a set of non-terminals, Σ_h is the signature of relations and operations performed on elements of the medium, Λ_h is the constructive axiomatics containing updates, additions and restrictions for media elements, operations and signature relations, on the basis of which the construction is performed.

4.1 Partial axiomatics of the medium

The image ${}_{\bar{w}}m_i \in M_h$ has a set of attributes $\bar{w} = \{w_1,\, w_2,\, \ldots,\, w_n\}$. Heterogeneous multiset of elements with attributes is meant by the set. Belonging of the attribute w_j to the image m will be denoted as $w_j \mathrel{\underset{\leftarrow}{\rfloor}} m$. All attributes are the images.

The images may change over time. Each image has an attribute of the time of creation or last modification ($t \mathrel{\underset{\leftarrow}{\rfloor}} m_i$). The given attribute is changed in the course of operation on the image and depends on the time of its execution.

The world image ${}_tP \in M_h$ will be called a continuous representation of the human environment presented in the form of dynamic flow of images, sounds, tactile, gustatory, olfactory and spatial-temporal sensations, feelings and emotions, reflected by the nervous system of an individual under the influence of physical stimuli. This image is a controlled one and depends on any particular individual (it is not essential in the context of this paper). At any given time t the certain world representation exists. Further, this attribute is not specified.

The form (l) is a set of elements M_h connected by the relationships of Σ_h.

Sentential form is a form obtained at any time as a result of inference from the initial non-terminal symbol according to the rules of inference from concretized CSSt.

The construction (K) is a sentential form at the current time, comprising only the terminals [26]. Constructions and relations are images as well.

The set of images is a construction based on some relation of similarity with the properties of reflexivity and symmetry [23].

4.2 Partial axiomatic of operations and relations

The signature Σ_h consists of the set of operations $\Sigma_h = \langle \Xi,\, \Theta,\, \Phi,\, \{\rightarrow\} \rangle \bigcup \Psi$, where Ξ – relations and homonym operations, operations of linking and transformation of the medium elements $\{\cdot,\, \vec{\in},\, \circ,\, \bar{\circ},\, \wedge,\, \vee,\, \Diamond, \uparrow,\, \Uparrow,\, >>,\, <<,\, \exists\} \subset \Xi$, $\Theta = \{\Rightarrow,$ $\mid \Rightarrow,\, \| \Rightarrow\}$ – the operations of substitution and inference, $\Phi = \{:=\}$ – the relations and homonym operations on the attributes, $\{\rightarrow\}$ – substitutive relation. $\Psi = \{\psi_i : \langle s_i,\, g_i \rangle\}$ is a set of substitution rules, s_i – a sequence of substitutive relations, g_i – set of operations on the attributes.If the operations on attributes are not performed, the substitution rule will take the form $\langle s_i,\, \varepsilon \rangle$, where ε is a null character. Relations from Ξ are applied in the inference rules, and operations corresponding to relations are applied during implementation of CSSt.

Execution time is an essential attribute of any operation $\tau \leftarrow\!\rfloor *$, where $*$ means any operation of Σ_h. Time attribute of each image after operation can be determined as $t = t_{start} + \tau$, with t_{start} representing the time of the operation start. The value of this attribute is determined by abilities of the performer. Further, it is not specified.

Image concatenation operation $\cdot({}_{\bar{w}_1}m_1,\, {}_{\bar{w}_2}m_2,\, P)$ involves linking of images ${}_{\bar{w}_1}m_1,\, {}_{\bar{w}_2}m_2$ under the influence of the world image P. The result is an image m – a sequence of the images ${}_{\bar{w}_1}m_1,\, {}_{\bar{w}_2}m_2$.

Element inclusion operation $\vec{\in}(\bar{m},\, {}_{\bar{w}}m)$ involves adding the image ${}_{\bar{w}}m$ into a set $\bar{m}$, $\bar{m}$ is the operand and result of the operation.

Image explication operation [25] is a selection of the part of the world and formation of the individual object-image with its own set of attributes. Result of the operation $\circ({}_{\bar{w}_1}m_1,\, {}_{\bar{w}_2}m_2)$ is the image ${}_{\bar{w}}m$ explicated from the image ${}_{\bar{w}_1}m_1$ under the influence of ${}_{\bar{w}_2}m_2$. The images ${}_{\bar{w}_1}m_1,\, {}_{\bar{w}_2}m_2$ can be any images of the medium M_h, construction or image of the world ${}_{focus,\,t}P$ at some moment in time t, on which attention of an individual is focused (it is indicated by the attribute $focus$). Further these attributes of the image P will be used as needed. Modification of the operation $\bar{\circ}({}_{\bar{w}_1}m_1,\, {}_{\bar{w}_2}m_2)$ is an explication of the relation ${}_{\bar{w}}m$ from ${}_{w}m_1$ under the influence of the image ${}_{\bar{w}_2}m_2$.

Inheritance operation with specification $\wedge({}_{w_1}m_1,\, {}_{w_2}m_2)$ involves the creation of a new image ${}_{w^*}m^*$ that repeats the image ${}_{\bar{w}_1}m_1$ and has $\bar{w}_1$ and ${}_{\bar{w}_2}m_2$ as the attributes.

Inheritance operation with the modification $\vee({}_{\bar{w}_1}m_1,\, {}_{\bar{w}_2}m_2 \leftarrow\!\rfloor\, m_1,\, {}_{\bar{w}_3}m_3)$ involves creation of the new image ${}_{w^*}m^*$ that repeats the image ${}_{\bar{w}_1}m_1$ with substitution of the attribute ${}_{\bar{w}_2}m_2$ for ${}_{\bar{w}_3}m_3$.

The finite set of linking operations of the images ${}_{\bar{w}_i}m_i,\, {}_{\bar{w}_j}m_j - \diamondsuit_k({}_{\bar{w}_i}m_i,\, {}_{\bar{w}_j}m_j)$. The result of the operation is m', composite image or relation image. Each of the given operations is the image of relation and belongs to the medium and signature $(\diamondsuit_i \in M_h,\, \diamondsuit_i \in \Sigma_h)$.

Generalization operation $\uparrow(\bar{m})$, where $\bar{m}$ is a set of images involves selection of the set with one or more identical attributes and formation of the result of operation m as a new generalized image with the same attributes and similar transformed images of the original images.

Unification operation ${}_{c}\Uparrow(\bar{m},\, {}_{\bar{w}_1}m_1,\, {}_{\bar{w}_2}m_2)$ allow creating some set of images $\bar{m}$ with adding of the image ${}_{\bar{w}_1}m_1$, provided that each element $\bar{m}$ and image ${}_{\bar{w}_1}m_1$ have similar attribute ${}_{\bar{w}_2}m_2$.

Image transfer operation ${}_{ch}>>({}_{\bar{w}}m,\, P)$ involves transmission of the image ${}_{\bar{w}}m$ to the external world through the channel ch in some encoded form. In the course of forming the language, it is visual (s) and auditory (h) form. Upon that, the world image P is changed, being supplemented by a new image of the operation performer image of the code: word, sentence, gesture, etc.

Image reception operation ${}_{ch}<<(P^*)$ involves determining (obtaining) certain code of the image ${}_{\bar{w}}m$ using the channel ch from the part of the external world P^*.

In this image of the world P is not changed, and the medium of CSSt images of performer is supplemented by a new image.

Operation to verify existence of the attribute $\exists({}_{\bar{w}_1}m_1,\ {}_{\bar{w}_2}m_2)$ determines presence of the attribute ${}_{\bar{w}_2}m_2$ in the image ${}_{\bar{w}_1}m_1$, i.e. ${}_{\bar{w}_2}m_2 \ \underset{\leftarrow}{\rfloor}\ {}_{\bar{w}_1}m_1$. The result of the operation is the logical value of truth in the presence of the required attribute ${}_{\bar{w}_2}m_2$, otherwise it is false.

Substitutive relation is a binary relation with the attributes ${}_{w_i}l_{iw} \rightarrow {}_{w_j}l_j$, where l_i, l_j are the sentential forms [26]. The sequence of the substitutive relations s_n is written as $s_m = \langle l_i \rightarrow l_j | l_k \rangle$, where l_i, l_j, l_k are the sentential forms. The substitutive relation can be written in short form $s_m = \langle l_i \rightarrow l_j | l_k \rangle$, where l_i, l_j, l_k are the forms, and it is equivalent to $s_m = \langle l_i \rightarrow l_j \rangle$, $s_n = \langle l_i \rightarrow l_k \rangle$.

For the given form ${}_{w_l}l = {}_{w_0} \otimes ({}_{w_1}l_1,\ {}_{w_2}l_2,\ \ldots,\ {}_{w_h}l_h,\ \ldots,\ {}_{w_k}l_k)$ and available substitutive relation ${}_{w_p} \rightarrow ({}_{w_h}l_h,\ {}_{w_q}l_q)$, where ${}_{w_h}l_h$ is a subform ${}_{w_l}l$ $({}_{w_h}l_h \prec {}_{w_l}l)$, the result of the triple substitution operation ${}_{w_l^*}l^* = {}_{w_p} \Rightarrow ({}_{w_h}l_h, {}_{w_q}l_q,\ {}_{w_l}l)$ will be the form ${}_{w_l^*}l^* = {}_{w_0} \otimes ({}_{w_1}l_1,\ {}_{w_2}l_2,\ \ldots,\ {}_{w_q}l_q,\ \ldots,\ {}_{w_k}l_k)$, where $\Rightarrow \in \Theta$, $\otimes$ is any operation of linking from Ξ.

Double operation of partial output ${}_{w_l^*}l^* = {}_{w_p}| \Rightarrow (\Psi,\ {}_{w_l}l)$ $(| \Rightarrow \in \Theta)$ consists in:

- selection of one of the available substitution rules $\psi_r\ :\ \langle s_r,\ g_r \rangle$ with the substitutive relations s_r;
- performance of substitution operations on the basis of it;
- performance of on the attributes g_r in the predetermined sequence.

Binary operation of full output or simply output ($|| \Rightarrow (\Psi,\ {}_{w_l}l)$, $|| \Rightarrow \in \Theta$) resents step-by-step transformation of forms, starting from the initial non-terminal and ending with the construction satisfying the condition of the output ending, which implies cyclical performance of the partial output operations.

Operation $:= (a,\ b)$ consists in assigning the value of operand b to the operand a.

5 Interpretation of CSSt of human images

To determine the performance algorithms of possible operations and relations on images, let us interpret the structure (1):

$$\begin{aligned} \langle C_h = \langle M_h,\ \Sigma_h,\ \Lambda_h \rangle\,,\ C_A = \langle M_A,\ \Sigma_A,\ \Lambda_A \rangle \rangle\ {}_I \mapsto {}_{I,\,C_A} \\ \mapsto {}_{I,\,C_A}C_h = \langle M_h,\ \Sigma_h,\ \Lambda_1,\ Z \rangle\,, \end{aligned} \tag{2}$$

where $M_A \supset V_A$, $V_A = \{A_i^0|_{X_i}^{Y_i}\}$ – a set of basic algorithms [30], X_i, Y_i – sets of determinations and values of the algorithm $A_i^0|_{X_i}^{Y_i}$, $\Lambda_1 = \Lambda_h \bigcup \Lambda_A \bigcup \Lambda_2$, Z – a set of possible CSSt performers which are able to implement all algorithms C_A; $\Lambda_A = \{M_A = \bigcup\limits_{A_i^o \in V_A} (X(A_i^0) \bigcup Y(A_i^0)) \bigcup \Omega(C_h)\}$ – inhomogeneous medium, $\Omega(C_h)$ – a set of constructions of the images which satisfy C_h.

Performer ${}_{\bar{k}_i}z_i$ of the structure (2) has a set of attributes, and we shall distinguish some of them $\bar{k} = \{location,\ occupation,\ l\_condition,\ p\_characters\}$, where *location* means the locality (residence), *occupation* means profession (activity), *l_condition* means the living conditions, *p_characters* are psycho-physiological characteristics, including those connected with perception and processing of information.

The structure ${}_{I,\,C_A}C_h$ includes algorithms of performing the operations:

- A_1^0 – composition of algorithms $A_1^0|_{A_i,\,A_j}^{A_i\cdot\,A_j}$, $A_i \cdot A_j$ – sequential execution of the algorithm A_j after algorithm A_i;
- A_2^0 – conditional execution $A_2^0\,|_b^{A_1}$: algorithm A_i is performed, if the condition b is true (execution is allowed);
- $A_3|_{m_1,\,m_2,\,P}^{m_1\cdot m_2}$ – images concatenation;
- $A_4|_{m,\,\bar{m}}^{\bar{m}}$ – image inclusion;
- $A_5|_{m_1,\,m_2}^{m^*}$ – image explication;
- $A_6|_{m_1,\,m_2}^{m^*}$ – explication of relation image;
- $A_7|_{m_1,\,m_2}^{m^*}$, $A_8|_{m_1,\,m_2,\,m_3}^{m^*}$ – inheritance of the image with specification and modification;
- $A_9|_{m_i,\,m_j}^{m'}$ – linking the images (establishment of relations);
- $A_{10}|_{\bar{m}}^{\bar{m}}$ – images generalization;
- $A_{11}|_{\bar{m},\,m}^{\bar{m}}$ – images unification;
- $A_{12}|_{m,\,P}^{P}$, $A_{13}|_{m,\,P}^{P}$ – transfer of image using audio and visual channel;
- $A_{14}|_{P}^{m}$ – receiving the image using audio and visual channel;
- $A_{16}|_{m_1,\,m_2}^{c}$ – checking for presence of the attribute m_2 in the image m_1;
- $A_{17}|_{l_h,\,l_q,\,f_i}^{f_i}$ – substitution;
- $A_{18}|_{f_i,\,\Psi}^{f_j}$, $A_{19}|_{\sigma,\,\Psi}^{\bar{\Omega}}$ – partial and full output, where f_i, f_j – forms, σ – axiom, $\bar{\Omega}$ – a set of the formed constructions;
- $A_{20}|_{a,\,b}^{b}$ – assignation.

Axiomatics of linking the operations and algorithms is as follows:
$\Lambda_2 = \{(A_1^0|_{A_i,A_j}^{A_i\cdot A_j} \leftarrow\!\rfloor\ \cdot),\ (A_2^0\,|_b^{A_1} \leftarrow\!\rfloor :), (A_3|_{m_1,\,m_2,\,P}^{m_1\cdot m_2} \leftarrow\!\rfloor\ \cdot), A_4|_{m,\,\bar{m}}^{\bar{m}} \leftarrow\!\rfloor\ \vec{\in}),$
$(A_5|_{m_1,\,m_2}^{m^*} \leftarrow\!\rfloor\ \circ), (A_6|_{m_1,\,m_2}^{m^*} \leftarrow\!\rfloor\ \bar{\circ}), (A_7|_{m_1,\,m_2}^{m^*} \leftarrow\!\rfloor\ \wedge), (A_8|_{m_1,\,m_2,\,m_3}^{m^*} \leftarrow\!\rfloor\ \vee),$
$(A_9|_{m_i,\,m_j}^{m'} \leftarrow\!\rfloor\ \lozenge), (A_{10}|_{\bar{m}}^{\bar{m}} \leftarrow\!\rfloor\!\uparrow), (A_{11}|_{\bar{m},\,m}^{\bar{m}} \leftarrow\!\rfloor\!\Uparrow), (A_{12}|_{m,\,P}^{P} \leftarrow\!\rfloor\ h >>),$

$(A_{13}|^{P}_{m,\,P} \hookleftarrow {}_{s} >>), (A_{14}|^{m}_{P} \hookleftarrow {}_{h} <<), (A_{15}|^{m}_{P} \hookleftarrow {}_{s} <<), (A_{16}|^{c}_{m_1,\,m_2} \hookleftarrow \exists)$, $(A_{17}|^{f_i}_{l_h,\,l_q,\,f_i} \hookleftarrow \Rightarrow), (A_{18}|^{f_j}_{f_i,\,\Psi} \hookleftarrow | \Rightarrow), (A_{19}|^{\bar{\Omega}}_{\sigma,\,\Psi} \hookleftarrow || \Rightarrow), (A_{20}|^{b}_{a,\,b} \hookleftarrow :=)\}$.

These algorithms are specific to each internal performer; they are based on the chemical and biological processes associated with the work of the human nervous system (partially they are highlighted in the attributes). Since the features of these operations performance depend on the performer, the multiple interpretation supposing different algorithms for one and the same operation is possible.

6 Concretization of CSSt of the human images

To clarify the input operations, let us perform concretization of the structure (2):

$$ {}_{I,\,C_A}C_h = \langle M_h,\, \Sigma_h,\, \Lambda_1,\, Z\rangle \; {}_{K}\mapsto_{K,I,\,C_A} C_h = \langle M_h,\, \Sigma_h,\, \Lambda_2,\, Z\rangle\,, \qquad (3)$$

where $\Lambda_2 = \Lambda_1 \bigcup \Lambda_3$, $\Lambda_3 \supset \{M_h = T \bigcup N,\ T = \{K,\, P,\, K_{pw},\, K_s,\, K_{aw}\}\}$ – a set of terminals, K – construction in the form of the set of images, K_s – construction of the ordered images of sounds, K_{pw} – construction of the ordered images of the written construction, K_{aw} – construction of the images received during observation of actions, glances, facial expressions and so on, $N = \{\sigma,\, \eta,\, \alpha,\, \beta,\, \delta,\, \chi,\, \gamma,\, \kappa\,\mu,\, \theta,\, \nu,\, \lambda\}$ – a set of non-terminals, σ – initial non-terminal.

6.1 Axiomatics of substitution rules

Let us consider the operations associated with imaginative thinking of an individual.

Rules of substitution $s_1 - s_3$ allow generating a new image based on the explication operation:

$$s_1 = \langle \sigma \to K\rangle\,, s_2 = \big\langle K \to \vec{\in}(K, \chi)\big\rangle\,, s_3 = \langle \chi \to \circ(P,\, \varepsilon)| \circ (K,\, P)| \circ (P,\, K)\rangle\,.$$

Rules of substitution $s_4 - s_5$ allow performing inheritance of the image with the specification:

$$s_4 = \big\langle K \to \vec{\in}(K, \wedge(\chi,\, \gamma))\big\rangle\,, s_5 = \big\langle \gamma \to \circ(P,\, K)| \circ (K,\, P),\, K \to \vec{\in}(K, \gamma)\big\rangle\,.$$

Rules of substitution $s_5 - s_7$ allow inheriting the image with the modification

$$s_6 = \Big\langle K \to \vec{\in}(K, \vee(\chi,\, \beta \hookleftarrow \chi,\, \gamma)\Big\rangle\,, s_7 = \langle \beta \to \circ(\chi,\, P)| \circ (\chi,\, K)\rangle\,.$$

Rules of substitution s_8, s_9 can be used for generalization of images:

$$s_8 = \big\langle K \to \vec{\in}(K,\, \uparrow(\alpha)\big\rangle\,, s_9 = \langle \alpha \to_c \Uparrow (\alpha,\, \gamma,\, \beta)|\varepsilon\rangle\,, g_9 = \langle c := \exists(\gamma,\, \beta)\rangle\,.$$

Image detailing is the operation being the reverse of generalization. It is performed by adding new attributes to generalized image using the specification operation. The following substitutive relations make it possible to link two images:

$$s_{10} = \big\langle K \to \vec{\in}(K,\, \mu),\, \mu \to \Diamond(\chi,\, \gamma),\, \Diamond \to \bar{\circ}(P,\, K)\big\rangle\,,$$

$$s_{11} = \left\langle K \to \vec{\in}(K,\, \mu),\, \mu \to \Diamond(\chi,\, \gamma),\, \Diamond \to \bar{\circ}(K,\, P) \right\rangle,$$

where $\Diamond \to \bar{\circ}(P,\, K)$, $\Diamond \to \bar{\circ}(K,\, P)$ is the selection of the relation image from the external world or construction, $\chi,\, \gamma \in M_h$.

Further we shall consider the operations related to the associative thinking and transmission of information.

Substitutive relations $s_{12} - s_{15}$ can be used to determine the attribute of the code (η) for the image χ:

$$s_{12} = \left\langle K \to \vec{\in}(K,\, \wedge(\chi,\, \eta)),\, \eta \to \wedge(\eta,\, \chi) \right\rangle, s_{13} = \langle \eta \to \circ(K_s,\, K) | \circ (K_s,\, P) \rangle,$$

$$s_{14} = \left\langle K \to \vec{\in}(K,\, K_s),\ K_s \to \vec{\in}(K_s,\, \nu) \right\rangle,$$

$$s_{15} = \Big\langle \nu \to \cdot(\delta,\, \nu,\, P),\, \delta \to \circ(\mathit{sound} \leftarrow\!\rfloor P,\, P) | \circ (\mathit{sound} \leftarrow\!\rfloor P,\, K)$$

$$| \circ (\mathit{sound} \leftarrow\!\rfloor K,\, K) | \circ (\mathit{sound} \leftarrow\!\rfloor K,\, P) | \varepsilon,\ \nu \to \circ(\mathit{sound} \leftarrow\!\rfloor P,\, P)$$

$$| \circ (\mathit{sound} \leftarrow\!\rfloor P,\, K) | \circ (\mathit{sound} \leftarrow\!\rfloor K,\, K) | \circ (\mathit{sound} \leftarrow\!\rfloor K,\, P) \Big\rangle.$$

The code can be used for transmission of information and in the process of thinking. The code may be represented as the image of sound (s_{15}) or picture (s_{19}) originally selected from the image of the external world. The sound image is a construction built on single atomic sounds, i.e. phonemes.

The visual image (the letter) can be put in correspondence with the image of the phoneme; the language construction (LC), such as word, word combination, sentence, etc., can be associated with the image of sound.

The written LC can be constructed as follows:

$$s_{16} = \left\langle K \to \vec{\in}(K,\, \wedge(\chi,\, \kappa)),\, \eta \to \wedge(\kappa,\, \chi),\, \kappa \to \wedge(\kappa,\, \eta), \eta \to \wedge(\eta,\, \kappa) \right\rangle,$$

$$s_{17} = \langle \kappa \to \circ(K_{pw},\, K) | \circ (K_{pw},\, P) \rangle,$$

$$s_{18} = \left\langle K \to \vec{\in}(K,\, K_{pw}),\, K_{pw} \to \vec{\in}(K_{pw},\, \kappa) \right\rangle,$$

$$s_{19} = \Big\langle \kappa \to \cdot(\delta,\, \kappa,\, P),\, \delta \to \circ(\mathit{img} \leftarrow\!\rfloor P,\, P) | \circ (\mathit{img} \leftarrow\!\rfloor P,\, K) | \circ (\mathit{img} \leftarrow\!\rfloor K,\, K) |$$

$$\circ(\mathit{img} \leftarrow\!\rfloor K,\, P) | \varepsilon,\, \kappa \to \circ(\mathit{img} \leftarrow\!\rfloor P,\, P) | \circ (\mathit{img} \leftarrow\!\rfloor P,\, K) | \circ (\mathit{img} \leftarrow\!\rfloor K,\, K) |$$

$$\circ(\mathit{img} \leftarrow\!\rfloor K,\, P) \Big\rangle,$$

where $\mathit{img} \leftarrow\!\rfloor P$ are the pictures included in the image of the world.

As a result of implementation of the rules $s_{12} - s_{19}$, the image of the external world P is put in correspondence with each completed language and visual construction. The image of the external world is complemented by the LC attribute identifying the same.

In addition to speech and written LC the character constructions of images (K_{aw}), such as gestures, glances, facial expressions, special fonts and scripts (for example, Braille script) and other actions and sensations can be formed:

$$s_{20} = \left\langle K \to \vec{\in}(K,\, \theta),\, \theta \to \circ(K_{aw}, K) | \circ (K_{aw}, P) \right\rangle,$$

$$s_{21} = \left\langle K \to \vec{\in}(K,\, K_{aw}),\, K_{aw} \to \vec{\in}(K_{aw},\, \lambda)\right\rangle,$$

$$s_{22} = \Big\langle \lambda \to \cdot(\theta,\, \lambda,\, P),\, \theta \to \circ(imgd \underset{\leftarrow}{\rfloor} P, P)| \circ (imgd \underset{\leftarrow}{\rfloor} K, P)| \circ (imgd \underset{\leftarrow}{\rfloor} P, K)|$$
$$\circ(imgd \underset{\leftarrow}{\rfloor} K, K)|\varepsilon,\ \lambda \to \circ(imgd \underset{\leftarrow}{\rfloor} P, P)| \circ (imgd \underset{\leftarrow}{\rfloor} K, P)| \circ (imgd \underset{\leftarrow}{\rfloor} P, K)|$$
$$\circ(imgd \underset{\leftarrow}{\rfloor} K, K)\Big\rangle,$$

where *imgd* is dynamic image associated with some human activities.

The substitutive relation s_{23} allows transfer images:

$$s_{23} = \langle \chi \to {}_{ch} >> (\eta,\, P)\rangle\,.$$

Substitutive relations $s_{24} - s_{25}$ allow receiving the image by supplementing the CSSt medium of LC images of a performer (an individual) in the following forms:

- written form $s_{24} = \langle K \to \vec{\in}(K,\, K_{pw}),\, K_{pw} \to \vec{\in}(K_{pw},\, \kappa),$ $\kappa \to {}_{s} << (img \underset{\leftarrow}{\rfloor} P),\ K \to \vec{\in}(K,\, \wedge(\chi,\, \kappa)),\, \chi \to \circ(P,\, \varepsilon)| \circ (K,\, P)| \circ (P,\, K),$ $K_{pw} \to \vec{\in}(K_{pw},\, \wedge(\kappa,\, \chi))\rangle;$

- speech form $s_{25} = \langle K \to \vec{\in}(K,\, K_s),\, K_s \to \vec{\in}(K_s,\, \nu),$ $\nu \to {}_{h} << (sound \underset{\leftarrow}{\rfloor} P),\ \ K \to \vec{\in}(K,\, \wedge(\chi,\, \nu)),\ \chi \to \circ(P,\, \varepsilon)| \circ (K,\, P)| \circ$ $(P,\, K),\ K_s \to \vec{\in}(K_s,\, \wedge(\nu,\, \chi))\rangle;$

- other form $s_{26} = \langle K \to \vec{\in}(K,\, K_{aw}),\ \ K_{aw} \to \vec{\in}(K_{aw},\, \theta),\, \theta \to_s << (imgd \underset{\leftarrow}{\rfloor}$ $P),\, K \to \vec{\in}(K,\, \wedge(\chi,\, \theta)),\, \chi \to \circ(P,\, \varepsilon)| \circ (K,\, P)| \circ (P,\, K),$ $K_{aw} \to \vec{\in}(K_{aw},\, \wedge(\theta,\, \chi)).$

7 Application of CSSt for constructive modeling of images

Let us consider the example of receiving and constructing images for a specific language construct – sentence 1 – “The branch operator is an operator that ensures the performing of certain commands only if a certain logical expression is true”.

This sentence is perceived as a construction of visual images-symbols consisting of images of words. To process this construction it is necessary to:

1. obtain all images of words using the operation ${}_{s} << (img \underset{\leftarrow}{\rfloor} P)$, where P is the image of the world, which includes the sentence under consideration;

2. compare the images of words with the images of prototypes named by them. Words-articles do not have any significant influence on the meaning of constructions, hence their semantic images will be omitted;

3. build a construction of images expressing the meaning of the language construct.

Let us implement the second paragraph:

$$K \overset{24(1)}{\Rightarrow} \vec{\in}(K,\, K_{pw}) \overset{24(2)}{\Rightarrow} \vec{\in}(K,\, \vec{\in}(K_{pw},\, \kappa_i)) \overset{24(3)}{\Rightarrow} \tag{4}$$

$$\overset{24(3)}{\Rightarrow} \vec{\in}(K, \vec{\in}(K_{pw},\, {}_{s} << (img \underset{\hookleftarrow}{\rfloor}\, {}_{f_i,\,t_i}P) = K_i, i = \overline{1, C}$$

where κ_i is the image of the word added to the performer's images, K_i is the construction of images – result of fulfillment of the relations, C is the number of words in the sentence, 24(1) is the application of the first relations from the set of relations s_{24} (likewise for similar records).

Let us establish the connection "meaning-word":

$$K_i \overset{24(4)}{\Rightarrow} \vec{\in}(K_i,\, \wedge(\chi_i,\, \kappa_i)) \overset{24(5)}{\Rightarrow} \vec{\in}(K_i,\, \wedge(\circ(K_i,\, {}_{f_i,t_i}P)_i,\, \kappa_i)), \tag{5}$$

where κ_i is the image of the word that is an independent part of speech.

The connection "word-meaning" (rule 24(6)) will be established if the image of the word is received by performer for the first time. Conclusions similar to (4-5) can be further omitted.

Let us construct the images corresponding to the sentence under consideration, using the formula (4). We shall form the images of words $\kappa_1 - \kappa_{21}$. The time and focus corresponding to this operation are given below (Table 1).

Table 1: Images of words of the sentence 1.

Image / word	Focus	Time	Image /word	Focus	Time
1/the	1	1	11/of	14	14
2/branch	2	2	12/certain	15	15
3/operator	3	3	13/commands	16	16
4/is	5	5	14/if	21	21
5/an	6	6	15/a	22	2
6/operator	7	7	16/certain	23	23
7/that	9	9	17/logical	24	24
8/ensures	10	10	18/expression	25	25
9/the	11	11	19/is	28	28
10/the	12	12	20/true	29	29

For all words of speech, we explicate images-meanings χ_i, $i = \overline{1,\, 20}$ and connect them with the words using the formula (5). The time and focus will coincide with the corresponding indicators when receiving word images. When constructing the images (complex and composite images), images of non-independent parts of speech can be omitted or interpreted as images of relations. Let us consider construction of composite images (Table 2). The constructed images can be enriched by adding attributes and their specification.

Table 2: Constructing the composite image constructs.

Conclusion	Image	Focus	Time	Prototype
$K \overset{4}{\Rightarrow} \vec{\in}(K,\, \wedge(\chi_2,\, \chi_1))$		4	4	the branch operator
$K \overset{11(1)}{\Rightarrow} \vec{\in}(K,\, \mu_1) \overset{11(2)}{\Rightarrow} \vec{\in}(K,\, \Diamond_1(\wedge(\chi_2,\, \chi_1), \chi_6)) \overset{11(3)}{\Rightarrow} \vec{\in}(K,\, \bar{\circ}(K,\, {}_{f_8,\, t_8}P)(\wedge(\chi_2,\, \chi_1), \chi_6))$	μ_1	8	8	the branch operator is an operator
$K \overset{4}{\Rightarrow} \vec{\in}(K,\, \wedge(\chi_{13},\, \chi_{12}))$		17	17	certain commands
$K \overset{11(1)}{\Rightarrow} \vec{\in}(K,\, \mu_2) \overset{11(2)}{\Rightarrow} \vec{\in}(K,\, \Diamond_2(\chi_{10}, \wedge(\chi_{13},\, \chi_{12}))) \overset{11(3)}{\Rightarrow} \vec{\in}(K,\, \bar{\circ}(K,\, {}_{f_{18},\, t_{18}}P)(\chi_{10}, \wedge(\chi_{13},\, \chi_{12})))$	μ_2	18	18	the performing of certain commands
$K \overset{11(1)}{\Rightarrow} \vec{\in}(K,\, \mu_3) \overset{11(2)}{\Rightarrow} \vec{\in}(K,\, \Diamond_3(\chi_8, \mu_2)) \overset{11(3)}{\Rightarrow} \vec{\in}(K,\, \bar{\circ}(K,\, {}_{f_{19},\, t_{19}}P)(\chi_8, \mu_2))$	μ_3	19	19	ensures the performing of certain commands
$K \overset{11(1)}{\Rightarrow} \vec{\in}(K,\, \mu_4) \overset{11(2)}{\Rightarrow} \vec{\in}(K,\, \Diamond_4(\chi_6, \mu_3)) \overset{11(3)}{\Rightarrow} \vec{\in}(K,\, \bar{\circ}(K,\, {}_{f_{20},\, t_{20}}P)(\chi_6, \mu_3))$	μ_4	20	20	operator that ensures the performing of certain commands
$K \overset{4}{\Rightarrow} \vec{\in}(K,\, \wedge(\chi_{18},\, \chi_{17}))$		26	26	logical expression
$K \overset{4}{\Rightarrow} \vec{\in}(K,\, \wedge(\wedge(\chi_{18},\, \chi_{17}),\, \chi_{16}))$		27	27	certain logical expression
$K \overset{11(1)}{\Rightarrow} \vec{\in}(K,\, \mu_5) \overset{11(2)}{\Rightarrow} \vec{\in}(K,\, \Diamond_1(\wedge(\wedge(\chi_{18},\, \chi_{17}), \chi_{16}), \chi_{20})) \overset{11(3)}{\Rightarrow} \vec{\in}(K,\, \bar{\circ}(K,\, {}_{f_{30},\, t_{30}}P)(-//-))$	μ_5	30	30	certain logical expression is true
$K \overset{11(1)}{\Rightarrow} \vec{\in}(K,\, \mu_5) \overset{11(2)}{\Rightarrow} \vec{\in}(K,\, \Diamond_6(\chi_8, \mu_1')) \overset{11(3)}{\Rightarrow} \vec{\in}(K,\, \bar{\circ}(K,\, {}_{f_{31},\, t_{31}}P)(\chi_8, \mu_5))$	μ_6	31	31	ensures only if a certain logical expression is true

Each received image is added to the carrier of the performer . If such image already exists, then it can be redefined or extended by inheritance operation.

All images are “assembled” into a construction describing the branch operator, a definition derived from the written language construct-sentence.

In the same manner, a similar image described in the sentence 2 ("Conditional construct is an operator that allows performing certain actions if a certain condition is true") can be generated.

Let us construct the images corresponding to the considered sentence using the formula (4). We form the images of words $\kappa_{22} - \kappa_{37}$. The time and focus corresponding to this operation are given below (Table 3).

Table 3: Images of the words of sentence 2.

Image/word	Focus	Time	Image/word	Focus	Time
22/ conditional	32	32	30/certain	41	41
23/ consrtuct	33	33	31/action	42	42
24/ is	34	34	32/if	47	47
25/ an	35	35	33/a	48	48
26/ operator	36	36	34/certain	49	49
27/ that	38	38	35/condition	50	50
28/ allows	39	39	36/is	52	52
29/ performing	40	40	37/true	53	53

For all words of speech, we explicate the images-meanings χ_i, $i = \overline{22, 37}$and connect them with words using the formula (5). Let us consider the construction of composite images for the given sentence (Table 4).

For graphical representation of the performed operations and the structure of resulting constructions, let us construct the graphs (Fig. 1), vertices of which are the images (χ_i) corresponding to the words and the arcs are the images of relations ($\Diamond_i(\chi_m, \gamma_n)$, $\wedge(\chi_k,\, \chi_l)$).

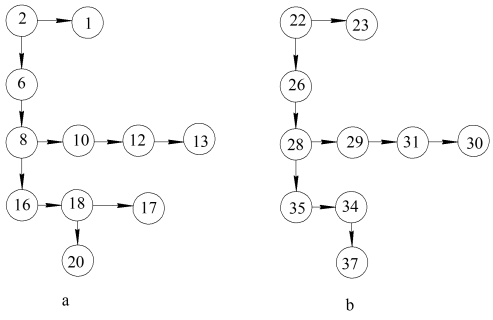


Figure 1: Structure of construction in the graph representation

Table 4: Constructing the composite image constructs of the sentence 2.

Conclusion	Image	Focus	Time	Prototype
$K \overset{4}{\Rightarrow} \vec{\in}(K, \wedge(\chi_{23}, \chi_{22}))$		33	33	conditional construct
$K \overset{11(1)}{\Rightarrow} \vec{\in}(K, \mu_7)$ $\overset{11(2)}{\Rightarrow} \vec{\in}(K, \Diamond_1(\wedge(\chi_{23}, \chi_{22}), \chi_{26}))$ $\overset{11(3)}{\Rightarrow} \vec{\in}(K, \bar{\circ}(K, {}_{f_{37}, t_{37}}P)$ $(\wedge(\chi_{23}, \chi_{22}), \chi_{26}))$	μ_7	37	37	conditional construct is an operator
$K \overset{4}{\Rightarrow} \vec{\in}(K, \wedge(\chi_{31}, \chi_{30}))$		43	43	certain actions
$K \overset{11(1)}{\Rightarrow} \vec{\in}(K, \mu_8)$ $\overset{11(2)}{\Rightarrow} \vec{\in}(K, \Diamond_8(\chi_{29}, \wedge(\chi_{31}, \chi_{30})))$ $\overset{11(3)}{\Rightarrow} \vec{\in}(K, \bar{\circ}(K, {}_{f_{44}, t_{44}}P)$ $(\chi_{29}, \wedge(\chi_{31}, \chi_{30})))$	μ_8	44	44	performing certain actions
$K \overset{11(1)}{\Rightarrow} \vec{\in}(K, \mu_9)$ $\overset{11(2)}{\Rightarrow} \vec{\in}(K, \Diamond_9(\chi_{28}, \mu_8))$ $\overset{11(3)}{\Rightarrow} \vec{\in}(K, \bar{\circ}(K, {}_{f_{45}, t_{45}}P)(\chi_{28}, \mu_8))$	μ_9	45	45	allows performing certain actions
$K \overset{11(1)}{\Rightarrow} \vec{\in}(K, \mu_{10})$ $\overset{11(2)}{\Rightarrow} \vec{\in}(K, \Diamond_{10}(\chi_{26}, \mu_9))$ $\overset{11(3)}{\Rightarrow} \vec{\in}(K, \bar{\circ}(K, {}_{f_{46}, t_{46}}P)(\chi_{26}, \mu_9))$	μ_{10}	46	46	operator that allows performing certain actions
$K \overset{4}{\Rightarrow} \vec{\in}(K, \wedge(\chi_{35}, \chi_{34}))$		51	51	certain condition
$K \overset{11(1)}{\Rightarrow} \vec{\in}(K, \mu_{11})$ $\overset{11(2)}{\Rightarrow} \vec{\in}(K, \Diamond_1(\wedge(\chi_{35}, \chi_{34}), \chi_{37}))$ $\overset{11(3)}{\Rightarrow} \vec{\in}(K, \bar{\circ}(K, {}_{f_{52}, t_{52}}P)(-//-))$	μ_{11}	52	52	certain condition is true
$K \overset{11(1)}{\Rightarrow} \vec{\in}(K, \mu_{12})$ $\overset{11(2)}{\Rightarrow} \vec{\in}(K, \Diamond_{12}(\chi_{28}, \mu_{11}))$ $\overset{11(3)}{\Rightarrow} \vec{\in}(K, \bar{\circ}(K, {}_{f_{53}, t_{53}}P)(\chi_{28}, \mu_{11}))$	μ_{12}	31	31	allows if a certain condition is true

8 Language realization of CSSt of human and community images

The result of realization of the structure (3) is the set of images of LC as a whole and its parts ${}_t\Omega(C_h({}_{\bar{k}_i}z_i))$:

$${}_t\Omega(C_h({}_{\bar{k}_i}z_i)) \supset ({}_t\Omega_s(C_h({}_{\bar{k}_i}z_i)) \bigcup {}_t\Omega_{pw}(C_h({}_{\bar{k}_i}z_i)) \bigcup {}_t\Omega_{aw}(C_h({}_{\bar{k}_i}z_i))),$$

where the sets ${}_t\Omega(C_h({}_{\bar{k}_i}z_i))$ – all the images formed by the performer ${}_{\bar{k}_i}z_i$ at the moment of time t, ${}_t\Omega_{pw}(C_h(z_i))$ – all images of the written LC, ${}_t\Omega_s(C_h(z_i))$ mean verbal constructions (including the ones corresponding to the ${}_t\Omega_{pw}(C_h(z_i))$), ${}_t\Omega_{aw}(C_h(z_i))$) represent other images of the LC.

Constructions, ${}_t\Omega(C_h({}_{\bar{k}_i}z_i))$, the elements of communication inherent in a specific performer ${}_{\bar{k}}z_i \in Z$ will be called an individual language. Free language is a set of potentially possible constructions that any performer (individual) can recognize (understand) and use to transfer the information.

We assume that there is some subset $\bar{Z} \subseteq Z$ form n of the CSS performers C_h. The language of the community of performers $\bar{Z}$ will be considered as a set of constructions built on the structure (3) medium, as a result of its implementation: $L(t) = \bigcup_b({}_t\Omega^*(C_h({}_{\bar{k}_i}z_i)) \bigcap_t \Omega^*(C_h({}_{\bar{k}_j}z_j)))$, where $b = ({}_{\bar{k}_i}z_i, {}_{\bar{k}_j}z_j \in \bar{Z}, {}_{\bar{k}_i}z_i \neq {}_{\bar{k}_j}z_j)$, ${}_t\Omega^*(C_h({}_{\bar{k}_i}z_i)) = {}_t\Omega_{pw}(C_h({}_{\bar{k}_i}z_i)) \bigcup_t \Omega_s(C_h({}_{\bar{k}_i}z_i)) \bigcup \Omega_{aw}(C_h({}_{\bar{k}_i}z_i))$. Language exists at some point in time t. LC belongs to the language if there are two or more of its carriers, capable of receiving and transmitting it $({}_t\Omega^*(C_h({}_{\bar{k}}z_i)) \bigcap {}_t\Omega^*(C_h({}_{\bar{k}}z_j)) \neq \emptyset, {}_{\bar{k}_i}z_i \neq {}_{\bar{k}_j}z_j)$. $L_{pw}(t) = \bigcup_b({}_t\Omega_{pw}(C_h({}_{\bar{k}_i}z_i)) \bigcap \ {}_t\Omega_{pw}(C_h({}_{\bar{k}_j}z_j)))$ is the written language of the community of performers, $L_s(t) = \bigcup_b({}_t\Omega_s(C_h({}_{\bar{k}_i}z_i)) \bigcap {}_t\Omega_s(C_h({}_{\bar{k}_j}z_j)))$ is the oral one.

The community includes groups of people which satisfy a certain relation of similarity. Presence of these groups allows distinguishing various sublanguages: languages of peoples, professional language, dialect languages, jargon, etc. Territorial characteristic, sphere of activity, standard of living, habitat, etc. can be used as an attribute on which a similarity relation is specified to distinguish the groups. For example, the professional language of programmers can be determined as $L_p(t) = \bigcup_b({}_t\Omega_p(C_h({}_{\bar{k}_i}z_i)) \bigcap \ {}_t\Omega_p(C_h({}_{\bar{k}_j}z_j)))$ where $b = ({}_{\bar{k}_i}z_i, {}_{\bar{k}_j}z_j \in \bar{Z}, {}_{\bar{k}}z_i \neq {}_{\bar{k}}z_j$, $\mathit{occupation} \lrcorner\, z_i = \mathit{occupation} \lrcorner\, z_j)$. For complete understanding and interaction of performers, it is desirable to draw a sample according to several characteristic, for example, occupation and territory (*location="Ukraine, Dnipro", occupation="programmer C#, senior"*), since in addition to generally accepted documented terms one can use definitions which represent, for example, transliteration or inexact translation of generally accepted English words. To select the professional LC of the chosen performers it is necessary to determine a set of images ${}_t\Omega_p(C_h({}_{\bar{k}_i}\bar{z}_i))$ as a set formed of elements with the same attribute for generalization ${}_t\Omega_p(C_h({}_{\bar{k}_i}\bar{z}_i)) = \{\omega_i : \forall\omega_i, \omega_j \in \ {}_t\Omega_{pw}(C_h({}_{\bar{k}_i}z_i)) \bigcup \ {}_t\Omega_s(C_h({}_{\bar{k}_i}z_i)) \,\exists\, \omega_{pw,\,i} : w_i \lrcorner\, \omega_i = w_j \lrcorner\, \omega_j\}$. For example, images cycle, variable, recursion have the common attribute-image programming by which they can be generalized to the programming term.

Carriers of the language and speech can be dynamic and static ones. Dynamic carriers can both store and generate constructions, for example, an individual, AI systems. Static carriers include those that cannot independently generate the language constructions; such carriers can be permanent books, audio discs, videotapes, and editable text computer files, soundtracks.

9 Analysis and identification of the similarity of language constructs

Working with static language carrier is useful in the tasks of information search and detection of plagiarism. To establish the fact of plagiarism, it is necessary to determine the matching content. To identify the matching content in the texts TXT_i and TXT_j it is necessary to distinguish the sets of printed word images of specific performers associated with this text $TXT_i \longrightarrow \Omega_{pw}(C_h({}_{\bar{k}_i}z_i))$ (the semantic content of some author's text), where $\longrightarrow$ is the display operation (can be implemented using s_{23}, s_{26}). Result of the operation $(TXT_i \longrightarrow \Omega_{pw}(C_h({}_{\bar{k}_i}z_i)))$ is a set of images $\{\omega_{pw,\,i}(C_h({}_{\bar{k}_i}z_i)) \in \Omega_{pw}(C_h({}_{\bar{k}_i}z_i)) : \forall txt_i \in TXT_i\, \exists txt_i \hookleftarrow\!\rfloor\, \omega_{pw,\,i}\}$, the attributes of which are the elements of the text $txt_i \in TXT_i$. The matching content (common fragments of texts) are defined as $(TXT_i \longrightarrow \Omega_{pw}(C_h({}_{\bar{k}_i}z_i))) \bigcap\limits_{b} (TXT_j \longrightarrow \Omega_{pw}(C_h({}_{\bar{k}_j}z_j)))$, $b = ({}_{\bar{k}_i}z_i,\, {}_{\bar{k}_j}z_j \in \bar{Z},\, {}_{\bar{k}_i}z_i \neq {}_{\bar{k}_j}z_j)$. The display operation is performed by the same performers for two (and more) texts.

Let us construct the correspondence table on the basis of the relations' images and their semantic similarity (Table 5). The obtained correspondences are based on the semantic similarity of the concepts considered in the field of programming. As it can be seen from the Fig. 1 and Table 5, the constructs have structural similarity and some complete coincidences of parts.

The constructs mentioned in the table have incomplete correspondence ( ). To reveal the similarity of concepts, analysis of similarity is carried out by one and the same performer. Part of the images considered is identical, and they are expressed by the same language constructs. The other ones have certain semantic similarity (finding of the same depends on the level of the basic programming concepts knowledge of the model performer). Lets give some explanations for selected images and their constructs (in the lines of Table 5):

1. the branch operator is the means of implementing a conditional construct, i.e. one may talk of similarity of concepts (the latter is broader);
2. some commands imply certain actions which can be realized by the programming language;
3. performance of commands involves performance of actions; both precedents can be reduced to one concept and result;
4. provision (guarantee, assurance) represents more strict form of the permission;

5. the logical impression implies description of a certain condition;
6. similar ones, on the basis of two previous lines.

This approach can be used for the automated search for matching content in the tasks of anti-plagiarism of the natural language constructions-texts at the semantic level. The usage of image approach to the semantics representation allows reducing a few words to a single image-sense that solves the problem of synonyms in the anti-plagiarism systems.

Table 5: Correspondence of the image constructions.

#	Text 1		Text 2		Comment on the similarity of concepts
	Image/ construction	Text fragment	Image/ construction	Text fragment	
1	$\wedge(\chi_2, \chi_{21})$	branch operator	$\wedge(\chi_{23}, \chi_{22})$	conditional construct	In fact, the same name found in different literature sources
2	$\wedge(\chi_{13}, \chi_{12})$	certain commands	$\wedge(\chi_{31}, \chi_{30})$	certain action	Commands and actions are close concepts, because they are realized by the programming language operators
3	μ_2	performing of certain commands	μ_8	performing certain actions	Participial construction with similar meaning
4	μ_4	operator that ensures the performing of certain commands	μ_{10}	operator that allows performing certain actions	Participial construction with similar meaning
5	μ_5	certain logical expression is true	μ_{11}	certain condition is true	Logical expression describes a condition
6	μ_6	ensures if a certain logical expression is true	μ_{12}	allows if a certain condition is true	It imposes the same condition on execution of an action

10 Conclusions

The constructed model of the NL is based on the figurative perception of the world by an individual. The model basis is represented by the formal grammars, which is widely recognized method of calculations. Formalization is provided for the thinking processes which are inextricably connected with encoding and transmission of thoughts using the elements of communication, such as gestures, facial expressions, speech, and writing. The latter are the basis for determining free and individual languages of people which are relevant to the concepts of the objective and subjective languages [20].

The presented model, in contrast to well-known ones:

- uses a single constructive approach for modeling all components and operations;
- is applicable to different forms of presentation of the language constructs;
- covers various aspects (syntactic, semantic ones) of the language;
- is closer to the natural processes;
- unlike the models that collect statistics, construct matrices, etc., the observed model already has a basis, i.e. the extensible, dynamic carrier of the performer, on the elements of which the relations are constructed and operations are performed;
- in contrast to the models used in NLP, for example, n-gram ones, it uses the meaning, not numeric attributes.

The model makes it possible to:

- consider NL as a set of communicative abilities of an individual which takes into account his/her language features and a person performer of the given model;
- consider the language as a constructive process which can be used as the basis for creating a methodology for building the systems with high degree of intellectuality;
- formally represent classifications of the language (areal classification, classification by the sphere of use (common, professional)), taking into account the characteristics of its speaker/carrier;
- improve the semantic NLP, in particular, in the tasks of comparing and identifying matching semantic content in texts, thus significantly reducing the influence of synonyms, homonyms, paraphrases, and translation.

The scope of the presented model covers NLP-components of robots and applications, including the systems of translation and anti-plagiarism, as well as expert systems.

References

[1] Beaufort, Richard, Roekhaut, Sophie, Cougnon, Louise-Amélie, and Fairon, Cédrick. A hybrid rule/model-based finite-state framework for normalizing sms messages. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 770–779. Association for Computational Linguistics, 2010.

[2] Bengio, Yoshua, Ducharme, Réjean, Vincent, Pascal, and Jauvin, Christian. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137 – 1155, 2003.

[3] Bisikalo, O. V. Conceptual model of imagery analysis and synthesis of natural language constructions. *Mathematical Machines and Systems*, (2):184 – 187, 2013.

[4] Bisikalo, O. V. *Formal methods imagery analysis and synthesis of natural language constructions: monograph.* Vinnitsa: VNTU, 2013.

[5] Brown, Peter F, Desouza, Peter V, Mercer, Robert L, Pietra, Vincent J Della, and Lai, Jenifer C. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467 – 479, 1992.

[6] Cambria, Erik and White, Bebo. Jumping nlp curves: a review of natural language processing research. *IEEE Computational Intelligence Magazine*, 9(2):48 – 57, 2014.

[7] Choudhury, Monojit, Saraf, Rahul, Jain, Vijit, Mukherjee, Animesh, Sarkar, Sudeshna, and Basu, Anupam. Investigation and modeling of the structure of texting language. *International Journal on Document Analysis and Recognition*, 10(3-4):157 – 174, 2007.

[8] Chowdhury, G. Natural language processing. *Annual Review of Information Science and Technology*, 37(1):51–89, 2003.

[9] Eger, Steffen, Schenk, Niko, and Mehler, Alexander. Towards semantic language classification: Inducing and clustering semantic association networks from europarl. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 127 – 136, 2015.

[10] Franco-Salvador, Marc, Gupta, Parth, and Rosso, Paolo. Cross-language plagiarism detection using a multilingual semantic network. In *European Conference on Information Retrieval*, pages 710–713. Springer, 2013.

[11] Kohan, Ya. O. On the possibilities of formalizing natural languages. In *TAAPSD*, volume 3, pages 137 – 143, 2016.

[12] Krak, Iu.V., Barmak, O.V., and Romanyshyn, S.O. The method of generalized grammar structures for text to gestures computer-aided translation. *Cybernetics and Systems Analysis*, 50(1):116 – 123, 2014.

[13] Krak, Iu.V., Kuznetsov, V.A., and Ternov, A.S. Facial expressions modeling based on a parametric models of a human head. *Artificial Intelligence*, pages 154 – 170, 2013.

[14] Le, Quoc and Mikolov, Tomas. Distributed representations of sentences and documents. *ICML*, pages 1188 – 1196, 2014.

[15] Liddy, Elizabeth D, Paik, Woojin, and Yu, Edmund Szu-li. Natural language processing system for semantic vector representation which accounts for lexical ambiguity, February 1999. US Patent 5,873,056.

[16] Marchenko, OO. A method for automatic construction of ontological knowledge bases. i. development of a semantic-syntactic model of natural language. *Cybernetics and Systems Analysis*, 52(1):20–29, 2016.

[17] Mazzei, Alessandro and Lombardo, Vincenzo. Building a large grammar for italian. In *LREC*, 2004.

[18] Micol, Daniel, Muñoz, Rafael, and Ferrández, Óscar. Investigating advanced techniques for document content similarity applied to external plagiarism analysis. In *Proceedings of Recent Advances in Natural Language Processing*, pages 240 – 246, Hissar, Bulgaria, September 2011.

[19] Mohri, M. On some applications of finite-state automata theory to natural language processing. *Natural Language Engineering*, 2(1):61 – 80, 1996.

[20] Nekipelova, I. M. Zarifullina, E. G. Language system as a natural multi-level classification of a high degree of reliability. *Modern studies of social problems*, (6 (26)), 2013.

[21] Nivre, Joakim, de Marneffe, Marie-Catherine, Ginter, Filip, Goldberg, Yoav, Hajic, Jan, Manning, Christopher D, McDonald, Ryan T, Petrov, Slav, Pyysalo, Sampo, Silveira, Natalia, et al. Universal dependencies v1: A multilingual treebank collection. In *LREC*, 2016.

[22] Osman, Ahmed Hamza, Salim, NAOMIE, Binwahlan, S, Hentabli, H, and Ali, ALBARAA M. Conceptual similarity and graph-based method for plagiarism detection. *Journal of Theoretical and Applied Information Technology*, 32(2):135 – 145, 2011.

[23] Schrader, Yu. A. *Equality, resemblance, order.* M .: Publishing house Science, 1971.

[24] Shynkarenko, V. and Vasetska, T. Modeling of ranking alternatives methods of analysis hierarchy of means of constructive-synthesizing structures. *Mathematical machines and systems*, (1):39 – 47, 2016.

[25] Shynkarenko, V. I. Qualification characteristics of natural, artificial and hybrid intelligence. *Artificial intelligence*, 67(1 – 2):9 – 19, 2015.

[26] Shynkarenko, V. I. and Ilman, V. M. Constructive-synthesizing structures and their grammatical interpretations. part i. generalized formal constructive-synthesizing structure. *Cybernetics and Systems Analysis*, 50(5):665 – 672, 2014.

[27] Shynkarenko, V. I. and Ilman, V. M. Constructive-synthesizing structures and their grammatical interpretations. part ii. refining transformations. *Cybernetics and Systems Analysis*, 50(6):829 – 841, 2014.

[28] Shynkarenko, V. I. and Kuropiatnyk, O. S. Constructive productive model of graph representation of text. *Problems of Programming*, (2 – 3):63 – 67, 2016.

[29] Shynkarenko, V. I. and Zabula, G. V. Constructive model of adaptation of data structures in operational memory. part i. constructing texts of programs. part ii. designers of scenarios and adaptation processes. *Science and Transport Progress. Bulletin of Dnipropetrovsk National University of Railway Transport*, 61, 62(1, 2):109 – 121, 88 – 97, 2016.

[30] Shynkarenko, V.I., Ilman, V.M., and Skalozub, V.V. Structural models of algorithms in problems of applied programming. i. formal algorithmic structures. *Cybernetics and Systems Analysis*, 45(3):329 – 339, 2009.

[31] Silberztein, Max. *Formalizing Natural Languages: The NooJ Approach.* John Wiley & Sons, 2016.

[32] Silberztein, Max. A new linguistic engine for nooj: Parsing context-sensitive grammars with finite-state machines. In *International Conference on Automatic Processing of Natural-Language Electronic Texts with NooJ*, pages 240–250. Springer, 2017.

[33] Tsatsaronis, George, Varlamis, Iraklis, Giannakoulopoulos, Andreas, and Kanellopoulos, Nikolaos. Identifying free text plagiarism based on semantic similarity. In *Proceedings of the 4th International Plagiarism Conference*, 2010.

Received 9th June 2017

Acta Cybernetica **23** (2018) 1017–1038.

Sector Based Linear Regression, a New Robust Method for the Multiple Linear Regression

Gábor Nagy[a]

Abstract

This paper describes a new robust multiple linear regression method, which based on the segmentation of the N dimensional space to N+1 sector. An N dimensional regression plane is located so that the half (or other) part of the points are under this plane in each sector. This article also presents a simple algorithm to calculate the parameters of this regression plane. This algorithm is scalable well by the dimension and the count of the points, and capable to calculation with other (not 0.5) quantiles. This paper also contains some studies about the described method, which analyze the result with different datasets and compares to the linear least squares regression.

Sector Based Linear Regression (SBLR) is the multidimensional generalization of the mathematical background of a point cloud processing algorithm called Fitting Disc method, which has been already used in practice to process LiDAR data. A robust regression method can be used also in many other fields.

Keywords: linear regression, robust regression, quantile regression

1 Introduction

The linear regression is an important component in a lot of calculation in the science and the engineering practice. This tool makes a relationship between one or more independent and one dependent variables by a linear function according to a given dataset.

The most popular method of the linear regression uses the least squares approach for fitting a line (or a plane in higher dimensions) to the given dataset. The outlier points makes remarkable impact in the result of the least squares based regression method.

There are some robust method of the linear regression [18, 21, 17, 22], for example, the Random Sample Consensus (RANSAC) method [6, 4, 7] and the Theil-Sen estimator [19, 23].The complexity of the RANSAC method is increased

[a] Óbuda University, Alba Regia Technical Faculty, Institute of Geoinformatics, E-mail: nagy.gabor@amk.uni-obuda.hu

DOI: 10.14232/actacyb.23.4.2018.3

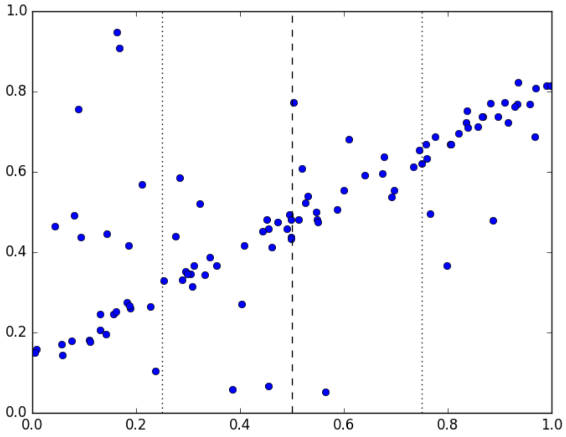


Figure 1: The sectors in case of $N = 1$. (The $N = 1$ is the number of the independent values, the total dimension of the space is $N + 1 = 2$, because the dependent value increases the dimension.) The area is divided to two parts (by the dashed line). The centres of this sectors are displayed by dotted lines.

highly with the dimension in the multiple linear regression, because $\binom{N}{M}$ different planes can be fitted to M given points in an N dimensional space. Both of these methods are not suitable for using with different quantiles.

This article describes the Sector Based Linear Regression (SBLR), a new robust method for the multiple linear regression. The SBLR method runs $O\left(MN^3\right)$ time, where M the number of the points of the dataset, and N is the number of the independent variables. The dimension of the space will be $N + 1$ with the one dependent variable. The SBLR can be used with different quantiles, for example a regression line over the 10 percent ($q = 0.1$) of the points, as other quantile regression methods [12, 23, 1].

2 Principles of the method

In the simple linear regression (one independent variable and one dependent variable, $N = 1$), the regression line has two parameters, for example the a and the b in the $y = ax + b$ equation. The plane can be divided into two parts (in the following: sectors) by a line parallel to the y axis (Figure 1.). A regression line are searched where the half ($q = 0.5$) or the other portion of the points are under the line in both sectors (Figure 2.).

In case of the regression planes (two independent variables and one dependent variable, $N = 2$), the plane of the two independent variable can be divided to three 120 degrees angles as sectors (see Figure 3.). The division can be performed

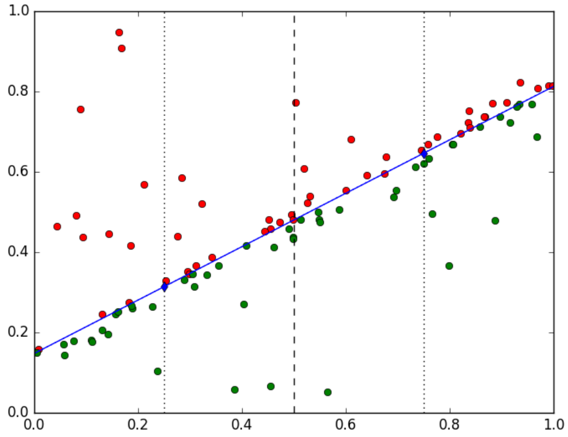


Figure 2: The principle of the SBLR method in case of $N = 1$ (one independent and one dependent values). The half of the points (displayed by green dots) are under, and the other half (displayed by red dots) are over the regression line in both sectors.

by the azimuth, which can be calculate from the two independent variables. (The `atan2()` function can calculate the azimuth in many programming languages.) The determined portion of the points are under the regression plane in all of these three sectors.

3 Extension to N independent variables

The method can be extended to any independent variables, the number of these variables is denoted by N. The dimension of space will be $N+1$ with the dependent variable.

The division of the sectors can be performed by the distances from the centres of the sectors, the points are classified to the sector, whose centre is the closest to the point. (The coordinates of the point are the independent variables of the regression.) This method is usable in any dimension, if the centres of the sectors are known.

The $N + 1$ centres of the sectors are the vertices of a regular N dimensional hyper-tetrahedron (N-simplex), whose centre is the origin of the N dimensional Cartesian coordinate system. The coordinates of the vertices (denoted $v^N_{i,j}$, where i is the index of the vertex from 0 to $N + 1$, j is the index of the coordinate from 1 to N, and N is the dimension of the space) can be calculated by the following recursive function:

- if $N = 0$, the result is `[[]]` (a list which contains an empty list)

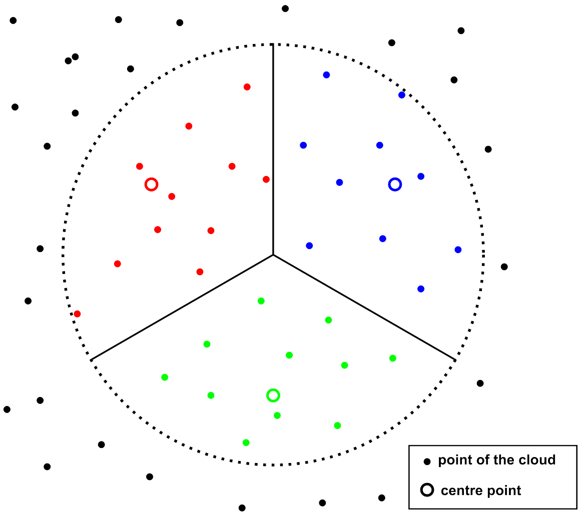


Figure 3: The sectors in case of $N = 2$. This figure represents the plane of the two independent variables, the coordinate of the dependent variable is perpendiculat to this plane. The half (or other quantile) of the points are under the regression plane in all sectors. (The points of the different sectors are displayed by different colors) This case is used in the LiDAR data processing where the points are the points of the LiDAR point cloud, the independent values are the horizontal coordinates of the points and the dependent coordinate is the vertical coordinate.

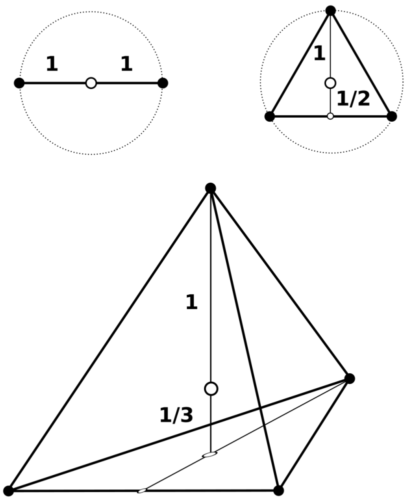


Figure 4: Calculate the coordinates of the vertices of an N-dimensional hyper-tetrahedron. (where $1 \le N \le 3$)

- if $N > 0$, the coordinates of the vertices are calculated by this expression:

$$v_{i,j}^{N} = \begin{cases} v_{i,j}^{N-1}\sqrt{1-\frac{1}{N}} & \text{if } i < N+1 \,\text{and}\, j < N \\ -\frac{1}{N} & \text{if } i < N+1 \,\text{and}\, j = N \\ 0 & \text{if } i = N+1 \,\text{and}\, j < N \\ 1 & \text{if } i = N+1 \,\text{and}\, j = N \end{cases} \tag{1}$$

If $N = 1$ then $v_{1,1}^{1} = -1$ and $v_{2,1}^{1} = 1$. If $N = 2$ then $v_{1,1}^{2} = -\frac{\sqrt{2}}{2}$, $v_{1,2}^{2} = -\frac{1}{2}$, $v_{1,1}^{2} = \frac{\sqrt{2}}{2}$, $v_{1,2}^{2} = -\frac{1}{2}$, $v_{1,1}^{2} = 0$ and $v_{1,2}^{2} = 1$. (Figure 4.)

These vertices are at 1 unit distance from the origin of the coordinate system. The sectors centres are $\frac{N}{N+1}$ units from the origin, because this point is the nearest to the centres of the sector. The sectors are indexed from 0 to N. The coordinates of the sector centres are:

$$s_{i,j}^{N} = \frac{N}{N+1} v_{i+1,j}^{N} \tag{2}$$

The $N+1$ dimensional regression hyperplane can be specified by $N+1$ value in two ways. One of them is a linear expression:

$$h = l_0 + l_1 x_1 + l_2 x_2 + \cdots + l_j x_j + \cdots + l_N x_N \tag{3}$$

where x_j is the coordinates of the position (the independent values, j indexed from 1 to N), and l_j is the $N+1$ coefficients of the N dimensional hyperplane (j indexed from 0 to N) in a $N+1$ dimensional space.

The other way to define the independent values (the elevations of the plane) in the $N+1$ centres of the sectors (the vertices of the N dimensional regular hyper-tetrahedron), which are denoted c_i, where i is the index of the vertex from 0 to N. The vector of c_i values (denoted $\underline{c}$) can be calculated simply from the vector of l_j values (denoted $\underline{l}$):

$$\underline{c} = \underline{\underline{Q}} \cdot \underline{l} \tag{4}$$

And the $\underline{l}$ can be calculated from the $\underline{c}$, if both sides of (4) are multipled left-hand side by $\underline{\underline{Q}}^{-1}$:

$$\underline{l} = \underline{\underline{Q}}^{-1} \cdot \underline{c} \tag{5}$$

The 4 and the 5 link between heights of sector's centres and coefficients of the linear equation of the hyperplane.

The $\underline{\underline{Q}}$ is an $N+1 \times N+1$ size matrix:

$$\underline{\underline{Q}} = \begin{bmatrix} 1 & s^N_{0,1} & \cdots & s^N_{0,j} & \cdots & s^N_{0,N-1} & s^N_{0,N} \\ 1 & s^N_{1,1} & \cdots & s^N_{1,j} & \cdots & s^N_{1,N-1} & s^N_{1,N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 1 & s^N_{i,1} & \cdots & s^N_{i,j} & \cdots & s^N_{i,N-1} & s^N_{i,N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 1 & s^N_{N-1,1} & \cdots & s^N_{N-1,j} & \cdots & s^N_{N-1,N-1} & s^N_{N-1,N} \\ 1 & s^N_{N,1} & \cdots & s^N_{N,j} & \cdots & s^N_{N,N-1} & s^N_{N,N} \end{bmatrix}$$

The $\underline{\underline{Q}}^{-1}$ is the inverse of $\underline{\underline{Q}}$, and can be calculated in $O\left(N^3\right)$ time. Because the $\underline{\underline{Q}}$ contains only constant values (the coordinates of the sector centres, and 1 values), the program has to calculate the matrix inversion only once. The multiplications (in (4) and (5)) need $O\left(N^2\right)$ time.

4 The calculation method

There is a given dataset, which contains M points. Each point contains N independent values (the coordinates in an N dimensional space) and one dependent value (which is an extra dimension). In the following, $p_{k,j}$ notation is used for the independent variables of the points, where k is the index of the point from 0 to $M-1$, and the j is the index of the coordinates from 1 to N. The $p_{k,0}$ values are the dependent variables.

4.1 Normalization

The first step is the normalization of the coordinates to the $[-1, +1]$ interval by the $x_j = a_j X + b_j$ expression. If one regression will be calculated for all points, calculate the normalized coordinates with $a_j = \frac{2}{\max(x_j) - \min(x_j)}$ and $b_j = -1 - \min(x_j)\, a_j$.

In another case, the regression will be calculated a selected part of the dataset. The points, which are nearest to a specified position (specified an $\underline{r}$ vector, whose elements are r_j) than a defined R radius ($R^2 \leq \sum_{j=1}^{N} (x_j - r_j)^2$). In this case, the $a_j = \frac{1}{R}$ and the $b_j = -r_j$.

In the following steps, the program uses these normalized coordinates.

4.2 Separating into sectors

In the next step, the points will be separated into the sectors, and calculate the initial value of the sector centres (c_i). Each points put the sector whose centre is the closest to the point. I use $p_{i,k,j}$ notation in the separated dataset, where i is the index of the sector (from 0 to N) and k is the number of the point in the sector from 1 to m_i.

All of the sectors have to contain at least one point ($\forall\, i\; m_i > 0$). If any sector does not contain any point ($\exists\, i\; m_i = 0$), the method can not work. This can happen, when the number or the dispersion of the points is not suitable. The probability of the any empty sector, when the dispersion is random (the $P\,(\text{point in the sector}) = \frac{1}{N+1}$ in all of the sectors) is $P\,(\text{any empty sector}) = 1 - \left(1 - \left(\frac{N}{N+1}\right)^{M}\right)^{N+1}$.

The initial values of the sector's centres (c_i) are the defined quantile (q) of the dependent variables of the sector's points:

$$c_i = \text{quantile}\left(\left[p_{i,1,0}, p_{i,2,0}, \ldots, p_{i,m_i,0}\right], q\right) \tag{6}$$

These values determines the initial regression plane. (See the Figure 5. in case of $N = 1$.)

4.3 The iteration steps

The key element of the method is an iteration step. The program goes from sector to sector and calculates the new values of the sector's centre.

Many $N + 1$ dimensional hyperplanes can be calculated, which are fitted to the centres of the other sectors and each points of the sector. The row of the sector's centre in the $\underline{\underline{Q}}$ matrix has to be changed to the coordinates of the point ($[p_{i,k,1}, p_{i,k,2}, \ldots, p_{i,k,N}]$), and the c_i value has to be changed to the $p_{i,k,0}$ (k is the index of the point in the sector) in the $\underline{c}$ vector, and use this modified (5) to calculate the parameters of the hyperplane. After calculating of the hyperplane parameters (l_j), calculate and store the the elevation of this plane in the sector centre by the (3):

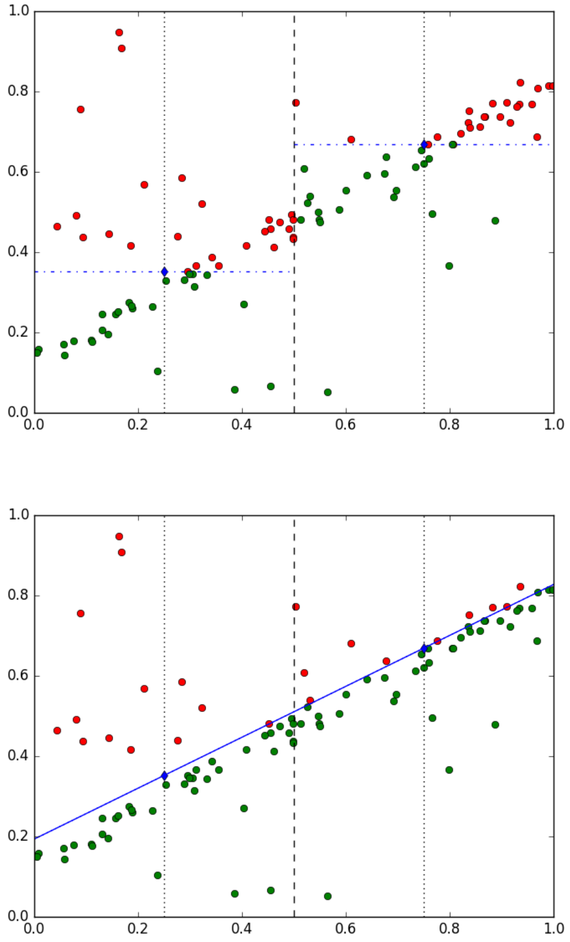


Figure 5: The initial step in case of $N = 1$. The median values are calculated for both sectors. (upper figure) These values (displayed by diamonds) will be the height of the initial regression line in the center of the sectors (dotted line). The points are displayed by red dots over and green dots under the lines (the height of the median, and the initial regression line). The initial regression line is fitted to the centre points. (lower figure)

$$h_k = l_0 + l_1 p_{i,k,1} + l_2 p_{i,k,2} + \cdots + l_j p_{i,k,j} + \cdots + l_N p_{i,k,N} \tag{7}$$

The new value of the sector's centre is the defined quantile (q) of these values:

$$c_i^{new} = \text{quantile}\left(\left[h_1, h_2, \ldots, h_{m_i}\right], q\right)$$

The program continues this process in the sector number $(i+1) \mod N+1$, and check the difference between the new and the old c_i values. If the difference less than a specified value ($\left|c_i^{old} - c_i^{new}\right| < \varepsilon$), a counter is increased one, otherwise the counter set to zero. The iteration loop is repeated while this counter is less than $N+1$. (The first two step in case of $N = 1$ is presented in Figure 6.)

The changes of the heights of the sector's centres typically will be less in the iterations. This ensures convergence.

4.4 Completion

Finally, the parameters of the regression plane are calculated by the (5) from the centres of the sectors. The received parameters are in a normalized coordinate system. (See 4.1)

If only the elevation of the plane is needed in the origin of the normalized coordinate system, the l_0 is this. If the plane equation is needed in the original coordinate system, the $l_i a_i$ expression can be used.

5 Studying the SBLR algorithm

Some simple Python [20, 16, 14] programs were made to test the SBLR algorithm. The `sblr.py` module is a simple implementation of the SBLR method. The test programs use this module.

The test programs use random datasets, which are created by the `random` Python module. This module can generate random numbers with several distribution. In the following studies the test programs use the $y = 3x - 5$ linear base function. The independent values (x) are generated by a uniform random value between 0 and 10 (`random.uniform(0,10)`). The dependent values are calculated by the $y = 3x - 5 + error$ equation. The *error* is various random number with 1 standard deviation and 0 median. A specific part of the points are outlier; the dependent variable of this points is a uniform random value between -7 and 27.

The test programs use different random numbers for the *error* value based on the `random` Python module. The uniform distribution error is a random number between $-\sqrt{3}$ and $\sqrt{3}$ by the `random.uniform(-1,1)*1.7320508075688772` expression. The normal distribution uses `random.normalvariate(1,0)`, the lognormal distribution uses `random.lognormalvariate(1,0)-1` and the exponential distribution uses `random.expovariate(1)-0.6931471805599453`. The minus 1 and minus $0.6931471805599453 \simeq \ln(2)$ need for the 0 median.

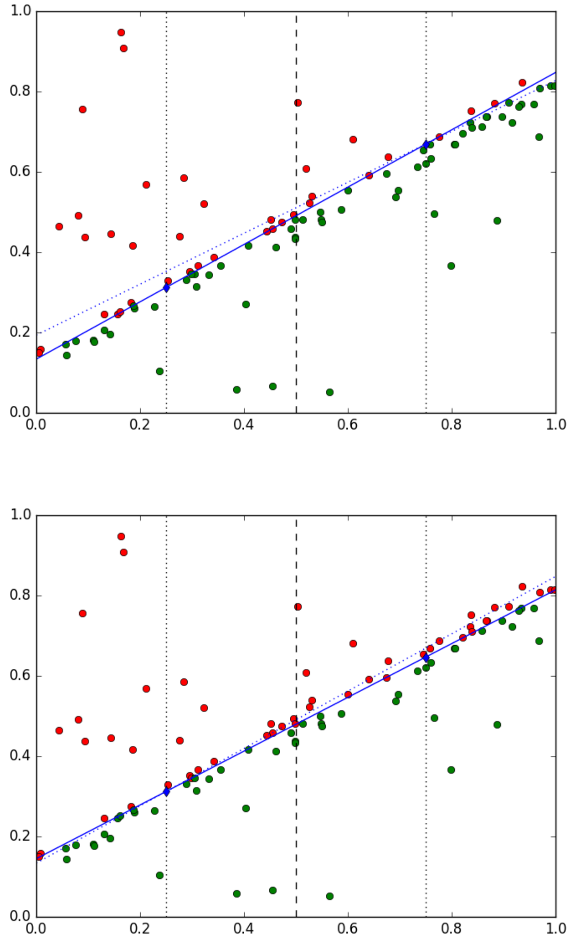


Figure 6: The iteration steps in case of $N = 1$. The new values of the sector's centres are determined so that the half (or other quantile) of the sector's points will be under the line, which is fitted the new centre of this sector and the other sector's centre. The new line is continuous, the line of the last iteration is dotted. The iteration is repeated until the change of the values are less than a limit (denoted ε) in both sectors.

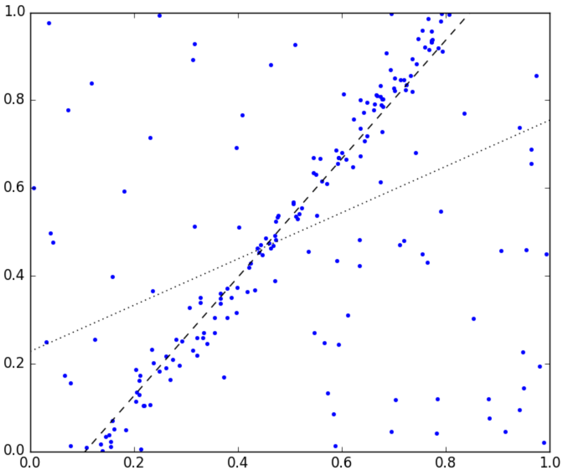


Figure 7: The result of the SBLR method (dashed line) and the least squares linear regression (dotted line) in a dataset with many outlier points.

5.1 Comparsion to the least squares linear regression

The least squares method is the most common regression tool, but the any outlier measurements can indicate significant difference in the result. (Figure 7.) A regression line can be calculated by the least squares method, the sum of squares of the differences between the points and the regression line will be the smallest with this regression line.

A test program generated random datasets with different portion of outlier points (from 0 to 75 percent). The test program generated 5000 datasets in all outlier portion (0%, 1%, 2%, ... 75%) and calculated regression lines in each dataset by the SBLR and the least squares methods. The two regression lines were compared to the original line, and calculate the averages of the distance from this line in the $[0, 10]$ interval. This number was the metrics of the fitting in these studies.

In each outlier portion, the test program stored 5000 fitting value; and another program calculated the averages of these values in both methods in each outlier portion. The Figure 8. shows the result of these studies with different number of points.

Another studies compare the average distance with different count of the points and different distribution of errors. The point numbers were the elements from an arithmetic sequence from 50 to 2000 with step of 50. The studies made with different errors (normal, uniform, lognormal and exponential) and different percentage of outliers (0 and 2). The program generates 5000 random datasets in each case. The result of these studies are seen in the Figure 9. and Figure 10.

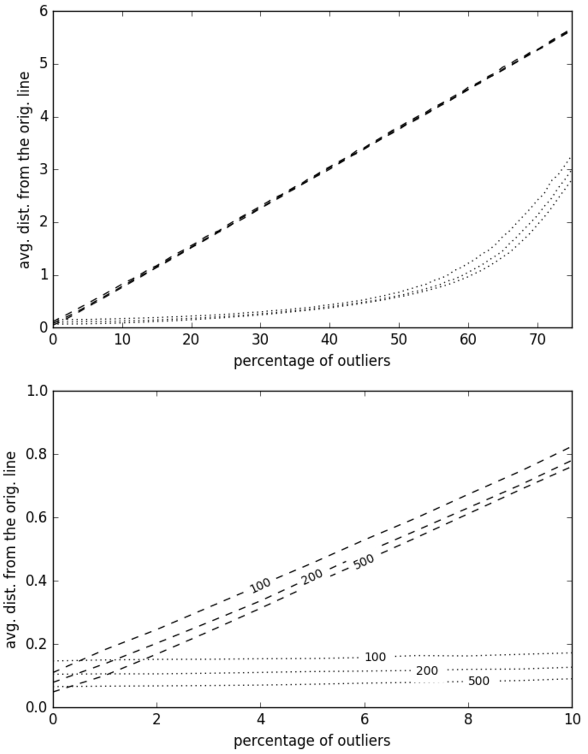


Figure 8: The average distance from the original line and the percentage of outliers with different number of points (100, 200 and 500) by least squares (dashed line) and SBLR method (dotted line). The range between 0 and 10 percent is zoom in on the lower figure.

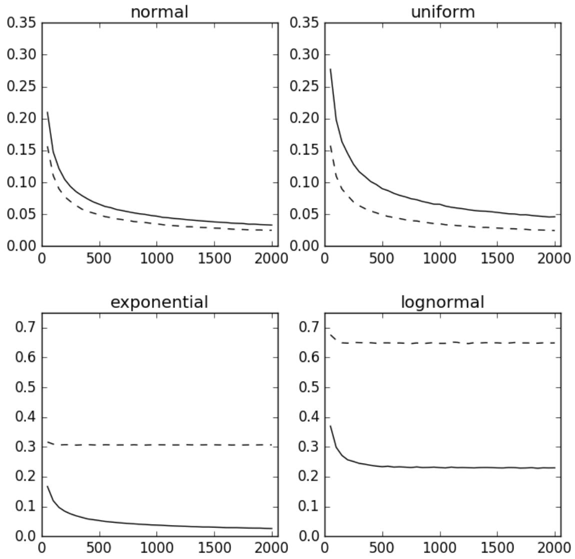


Figure 9: The average distance from the original line (vertical axes) with different number of points (horizontal axes) by least squares (dashed line) and SBLR (continuous line) methods. The datasets do not contain outlier points.

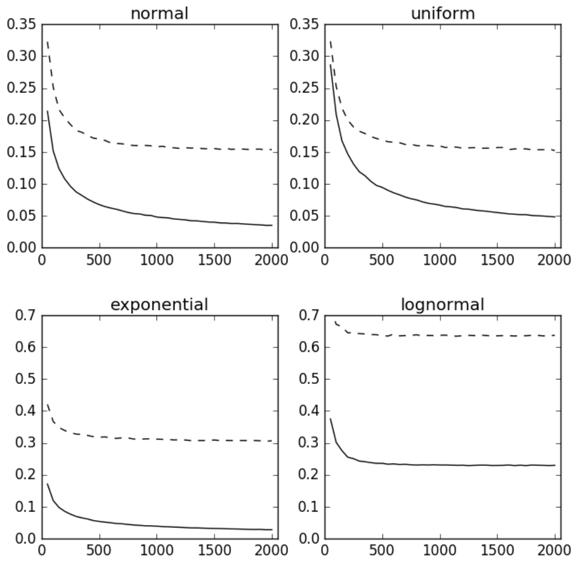


Figure 10: The average distance from the original line (vertical axes) with different number of points (horizontal axes) by least squares (dashed line) and SBLR (continuous line) methods. The datasets contain 2 percent of outlier points.

In the asymmetric error distributions (exponential and lognormal), the SBLR created better result than the least squares method without outlier points. If the dataset has 2 percent of outlier points, the SBLR made better result in all of the examined error distributions.

5.2 Examining the iteration steps

The computation time of the SBLR method grows linearly with the count of the points (denoted M in this article). This computation time may be increased if the iteration steps of the method grows with M.

Some test programs were created to study the correlation between the number of the points and the iteration steps. The M was different values according to a geometric sequence. The initial value of this sequence is 100 and the common ratio is $\sqrt[4]{2} \simeq 1.1892$ (the result was rounded). The largest datasets had 102400 points.

The test programs created 1000 different random datasets with each M and each distributions, calculated the regression lines and store the number of the iteration steps with $\varepsilon = 10^{-5}$. Another program analyzed the stored data and calculate the means of the iteration steps. (Table 1.)

The number of the iteration step does not grow, moreover a little decrease, when the M increased. The computation time of the SBLR method is $O\left(MN^3\right)$. The result is same when the 30 percent of the points are outlier. Other parameters were not changed. (Table 2.)

5.3 The limits and possible errors of the SBLR method

The SBLR method can calculate only linear regression, and only one regression in a dataset. The Ref. [10] presents a method, which can be found more linear regression from one dataset.

The method can work if all sectors have at least one point. The good result needs more points in all sectors to eliminate the impact of the outliers.

An outlier point may result wrong sector layout. The normalization step (see in 4.1) create a wrong result, where the outlier point is in a sector, and all the rest in the other sector. This problem can be avoided, if the sector centre is determined as the median of the values. In the practical applications (in the author's practice), this mistake has not occurred, because the points are selected from a bigger dataset (see in 6.1), therefore it did not have far points in the independent coordinates.

6 Application possibilities

The SBLR has a lot of application possibilities. This method may be used in projects, where need a robust linear regression. The SBLR may be useful, when quantile regression are needed in any dimension spaces.

Table 1: The average number of iteration steps with different distribution of errors and different number of points (without outliers)

Count of points	distribution of the errors			
	normal	lognormal	exponential	uniform
100	9.369	9.561	9.453	9.473
119	9.305	9.394	9.273	9.281
141	9.148	9.299	9.293	9.237
168	9.252	9.255	9.176	9.265
200	9.177	9.374	9.234	9.305
238	9.010	9.199	8.909	9.087
283	8.893	9.098	8.879	9.004
336	8.891	9.011	8.869	9.022
400	8.718	9.030	8.764	8.829
476	8.526	8.887	8.624	8.842
566	8.572	8.833	8.566	8.772
673	8.351	8.773	8.459	8.609
800	8.394	8.725	8.346	8.564
951	8.221	8.578	8.319	8.485
1131	8.164	8.477	8.239	8.280
1345	8.056	8.391	8.145	8.272
1600	8.028	8.437	8.149	8.188
1903	7.921	8.317	8.008	8.023
2263	7.842	8.206	7.949	7.982
2691	7.748	8.230	7.863	7.855
3200	7.730	8.152	7.868	7.898
3805	7.600	8.071	7.763	7.634
4525	7.474	8.003	7.627	7.593
5382	7.407	7.966	7.615	7.632
6400	7.371	7.888	7.560	7.511
7611	7.255	7.868	7.544	7.382
9051	7.059	7.768	7.453	7.301
10763	7.070	7.811	7.363	7.170
12800	6.997	7.730	7.307	7.127
15222	6.952	7.654	7.175	7.034
18102	6.833	7.521	7.174	7.008
21527	6.720	7.501	7.152	6.921
25600	6.629	7.468	7.067	6.828
30444	6.611	7.441	6.973	6.704
36204	6.553	7.329	6.926	6.603
43054	6.433	7.311	6.929	6.618
51200	6.294	7.234	6.815	6.481
60887	6.215	7.146	6.827	6.384
72408	6.201	7.079	6.752	6.275
86108	6.093	7.046	6.662	6.172
102400	6.002	6.990	6.685	6.128

Table 2: The average number of iteration steps with different distribution of errors and different number of points (with 30 percent outliers)

Count of points	distribution of the errors			
	normal	lognormal	exponential	uniform
100	9.738	9.776	9.740	9.729
119	9.678	9.601	9.583	9.614
141	9.484	9.502	9.472	9.492
168	9.570	9.557	9.439	9.611
200	9.409	9.498	9.338	9.482
238	9.275	9.349	9.438	9.453
283	9.264	9.277	9.195	9.305
336	9.157	9.146	9.184	9.251
400	9.142	9.068	9.279	9.262
476	9.118	9.121	9.001	9.203
566	8.919	8.926	9.012	9.096
673	8.908	8.907	8.865	9.005
800	8.841	8.755	8.932	9.019
951	8.800	8.781	8.815	8.804
1131	8.744	8.786	8.848	8.787
1345	8.617	8.725	8.794	8.816
1600	8.732	8.547	8.827	8.736
1903	8.545	8.552	8.690	8.570
2263	8.494	8.517	8.670	8.481
2691	8.524	8.525	8.648	8.417
3200	8.395	8.527	8.640	8.328
3805	8.290	8.357	8.572	8.265
4525	8.221	8.352	8.439	8.179
5382	8.210	8.355	8.429	8.236
6400	8.116	8.359	8.334	8.065
7611	8.016	8.191	8.327	7.996
9051	7.988	8.209	8.253	7.844
10763	7.959	8.147	8.114	7.873
12800	7.817	8.087	8.141	7.705
15222	7.887	8.051	8.134	7.715
18102	7.804	8.005	8.057	7.586
21527	7.681	7.909	8.056	7.523
25600	7.727	7.925	7.938	7.451
30444	7.680	7.912	7.948	7.373
36204	7.506	7.843	7.880	7.291
43054	7.509	7.786	7.851	7.218
51200	7.354	7.733	7.800	7.107
60887	7.336	7.746	7.707	7.076
72408	7.320	7.646	7.712	6.973
86108	7.257	7.631	7.652	6.918
102400	7.144	7.549	7.622	6.894

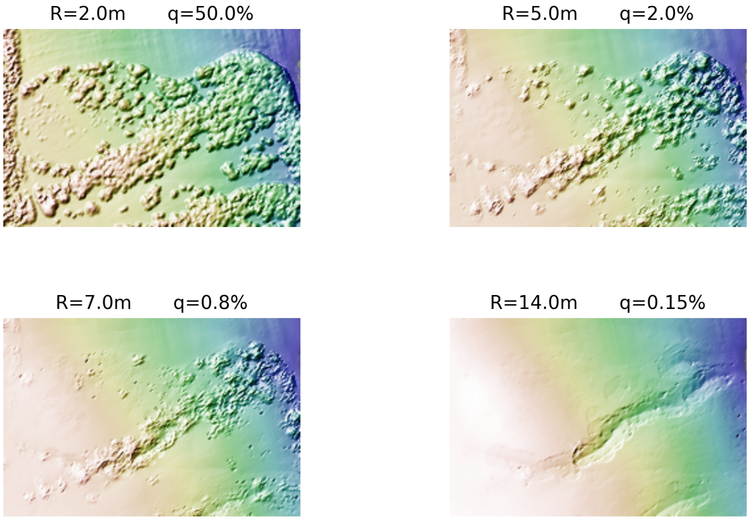


Figure 11: The application of the SBLR method in LiDAR data processing with different R and q values.

6.1 LiDAR data processing

SBLR can be used in any application, where a robust linear regression method is required. If the distribution of the measurement error is skewed, the method can use a different q value than 0.5.

This method has been used for processing the LiDAR point clouds. In this case ($N = 2$), the two independent value are the horizontal coordinates, the dependent variable is the elevation, and the measurements are the points of the LiDAR point cloud. (See the Figure 3.) The classical X, Y, and Z coordinates of the points are denoted x_1, x_2 and h in this case in the equation of a fitting plane, and $p_{k,1}$, $p_{k,2}$ and $p_{k,0}$ in the point of the cloud.

The regression plane is fitted to a part of the total LiDAR cloud, which is cut by a circle shape with R radius. The regression plane fits to this part of the cloud, because this method is called "Fitting Disc" method. [15] This principle may be used in other cases, where the connection is not linear between the independent and the dependent values: select the points, which are nearest than a radius (R) from an examined position, and fit a linear, N dimensional plane to this part, which is approximately linear. (See in the Figure 12., in a two-dimensional illustration.) The Fitting Disc method is a local application of the Sector Based Linear Regression.

Digital Elevation Models can be created, if the SBLR based Fitting Disc method is applied in each point of the DEM grid. The result depends from R and q values, for example the Figure 11. In the forest areas the appropriate result needs very low

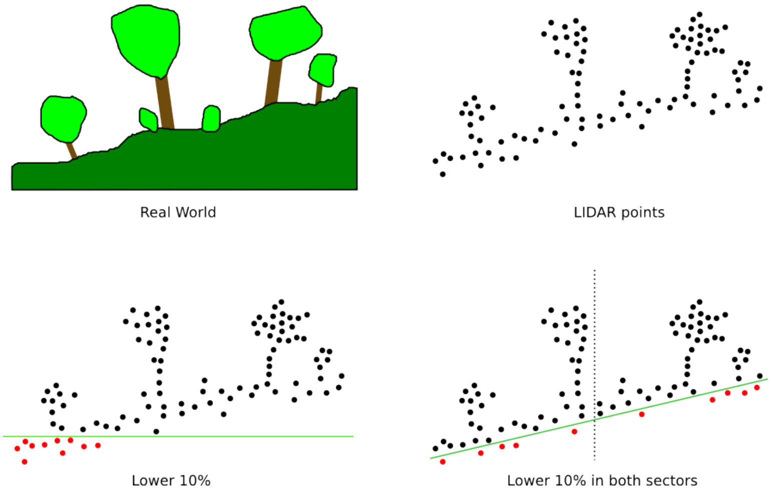


Figure 12: The LiDAR data processing with SBLR in a two-dimensional illustration. The ground surface is evaluated by $q = 0.1$ parameter, because the majority of the points are in the trees and bushes, over the ground surface.

q values; and the very low q values need long R radius, because some points must be under the plane. If the intention is at least on average n points under the plane in each sectors, the radius is $R \geqq \sqrt{\frac{3n}{qd\pi}}$, where d is the density of the LiDAR point cloud in $^{\text{points}}/_{\text{m}^2}$.

The SBLR based Fitting Disc method can be applied to recognize planes in a point cloud, for example the roofs of the buildings. In these cases the plane of the detected object (for example a roof) can be calculated by SBLR from a segment of the point cloud.

6.2 Other possibilities

A linear regression plane can be fitted to the data of the pixels of a picture near a position (like the LiDAR data processing) and calculated a filtered color by this regression plane. This filtering method is same as the Two-Dimensional Median Filtering Algorithm [8].

The SBLR can be used for any data processing task, where a linear regression is needed in an N-dimensional space. This method can be used well with a lot of outlier data or a random error with asymmetric distribution.

The SBLR is a linear case of the quantile regression [13, 12, 11]. The quantile regression is used in different disciplines, for example ecology [3] or economy [2, 5].

A robust linear regression method can provide a robust method to determine the parameters an affine transformation by control points. This calculation needs two independent linear regression for the two coordinates (in case of the two-dimensional affine transformation), because each equations of the affine transformation are a linear regression, where the independent variables are the coordinates of the reference system one, and the dependent variable is a coordinate of the reference system two.

7 Conclusions and future work

The Sector Based Linear Regression is a robust method for fitting an N dimensional hyperplane to a dataset which has N independent and 1 dependent variables. The studies of this article focused to the simple $N = 1$ case, and the practical application (LiDAR data processing) uses the $N = 2$ case, but the method can be applied in any dimension. This method provides quantile regression, it is useful in some cases (for example the LiDAR data processing, when the majority of the points are over the ground surface).

The processing time of the SBLR method is increased only linear with the size of the input data (the number of the points, denoted by M in this article). This advantage makes it ideal for big data processing applications.

This article presents the principle of the method, an algorithm for the SBLR, and some studies and application possibilities of the method. A simple implementation of the SBLR method has been made. The source code of this Python 3 module is attached to this article. In the future, i would like to implement the method in other programming languages, and improve the efficiency of the program.

The principle of the Sector Based Linear Regression can be adapted to non-linear regressions. The area must be divided more sectors in these cases, because the non-linear curves need more parameters.

8 Acknowledgement

This research was supported by the project number TÁMOP-4.2.2.B-15/1/KONV-2015-0010, titled "Tudományos képzés műhelyeinek fejlesztése az Alba Regia Műszaki Karon" (in English: Developing workshops of the scientific education in the Alba Regia Technical Faculty).

The Figure 1., Figure 2., Figure 5., Figure 6, Figure 7., Figure 8., Figure 9., Figure 10. and Figure 11. were created by Matplotlib [9].

9 Additional files

This article contains two animated GIF files. The `sblr.gif` shows the SBLR method during operation in case of $N = 1$. The `fitdisc.gif` presents the test area of the Figure 11. in many other cases of R and q parameters of the Fitting Disc method.

The implemented SBLR algorithm is already attached `sblr.py` Python 3 module. This module provides the SBLR calculations in any Python 3 program.

References

[1] Bertsimas, Dimitris and Mazumder, Rahul. Least quantile regression via modern optimization. *The Annals of Statistics*, pages 2494–2525, 2014.

[2] Buchinsky, Moshe. Changes in the us wage structure 1963-1987: Application of quantile regression. *Econometrica: Journal of the Econometric Society*, pages 405–458, 1994.

[3] Cade, Brian S and Noon, Barry R. A gentle introduction to quantile regression for ecologists. *Frontiers in Ecology and the Environment*, 1(8):412–420, 2003.

[4] Choi, Sunglok, Kim, Taemin, and Yu, Wonpil. Performance evaluation of ransac family. *Journal of Computer Vision*, 24(3):271–300, 1997.

[5] Coad, Alex and Rao, Rekha. Innovation and firm growth in high-tech sectors: A quantile regression approach. *Research policy*, 37(4):633–648, 2008.

[6] Fischler, Martin A and Bolles, Robert C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[7] Hast, Anders, Nysjö, Johan, and Marchetti, Andrea. Optimal ransac-towards a repeatable algorithm for finding the optimal set. 2013.

[8] Huang, T, Yang, G, and Tang, G. A fast two-dimensional median filtering algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27(1):13–18, 1979.

[9] Hunter, J. D. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.

[10] Isack, Hossam and Boykov, Yuri. Energy-based geometric multi-model fitting. *International journal of computer vision*, 97(2):123–147, 2012.

[11] Jurečková, Jana. Robust quantile regression. *Encyclopedia of Environmetrics*, 2006.

[12] Koenker, Roger. *Quantile regression.* Number 38. Cambridge university press, 2005.

[13] Koenker, Roger and Bassett Jr, Gilbert. Regression quantiles. *Econometrica: journal of the Econometric Society*, pages 33–50, 1978.

[14] Millman, K Jarrod and Aivazis, Michael. Python for scientists and engineers. *Computing in Science & Engineering*, 13(2):9–12, 2011.

[15] Nagy, Gábor, Tamás, Jancsó, and Chen, Chongcheng. The fitting disc method, a new robust algorithm of the point cloud processing. *ACTA POLYTECHNICA HUNGARICA*, 14(6):59–73, 2017.

[16] Oliphant, Travis E. Python for scientific computing. *Computing in Science & Engineering*, 9(3), 2007.

[17] Rousseeuw, Peter J and Hubert, Mia. Regression depth. *Journal of the American Statistical Association*, 94(446):388–402, 1999.

[18] Rousseeuw, Peter J and Leroy, Annick M. *Robust regression and outlier detection*, volume 589. John Wiley & Sons, 2005.

[19] Theil, Henri. A rank-invariant method of linear and polynomial regression analysis. In *Henri Theils Contributions to Economics and Econometrics*, pages 345–381. Springer, 1992.

[20] Van Rossum, Guido et al. Python Programming Language. In *USENIX Annual Technical Conference*, volume 41, 2007.

[21] Wilcox, Rand R. *Introduction to robust estimation and hypothesis testing.* Academic press, 2011.

[22] Wilcox, Rand R and Keselman, HJ. Modern regression methods that can substantially increase power and provide a more accurate understanding of associations. *European journal of personality*, 26(3):165–174, 2012.

[23] Zhou, Weihua and Serfling, Robert. Multivariate spatial u-quantiles: A bahadur–kiefer representation, a theil–sen estimator for multiple regression, and a robust dispersion estimator. *Journal of Statistical Planning and Inference*, 138(6):1660–1678, 2008.

Received 11th November 2017

Acta Cybernetica **23** (2018) 1039–1054.

An Efficient Method to Reduce the Size of Consistent Decision Tables

János Demetrovics[a] Hoang Minh Quang[b]
Vu Duc Thi[c] and Nguyen Viet Anh[b]

Abstract

Finding reductions from decision tables is one of the main objectives in information processing. Many studies focus on attribute reduct that reduces the number of columns in the decision table. The problem of finding all attribute reducts of consistent decision table is exponential in the number of attributes. In this paper, we aim at finding solutions for the problem of decision table reduction in polynomial time. More specifically, we deal with both the object reduct problem and the attribute reduct problem in consistent decision tables. We proved theoretically that our proposed methods for the two problems run in polynomial time. The proposed methods can be combined to significantly reduce the size of a consistent decision table both horizontally and vertically.

Keywords: attribute reduct, object reduct, consistent decision table, rough set theory, relational database theory

1 Introduction

Rough set theory was first represented by Pawlak in 1982. Since then, rough set theory [9] has found many interesting applications in areas such as knowledge acquisition, decision analysis, knowledge discovery from databases, expert systems, inductive reasoning and so on. The theory seems to be particularly important when applied for *information systems* (sometimes called data tables, decision tables, attribute-value systems, or condition-action tables) for knowledge representing, knowledge reduction, dependency reasoning and many other research problems.

Knowledge reduction [6, 17] is considered one of the most fundamental and important research tasks when working with information systems. The knowledge

[a] Computer and Automation Institute Hungarian Academy of Sciences, E-mail: `demetrovics@sztaki.mta.hu`

[b] Institute of Information Technology - Vietnam Academy of Science and Technology, E-mail: `{hoangquang,anhnv}@ioit.ac.vn`

[c] The Information Technology Institute (ITI) - Vietnam National University, Hanoi, E-mail: `vdthi@vnu.edu.vn`

DOI: 10.14232/actacyb.23.4.2018.4

reduction problem relates to the general concept of independence and knowledge core. It is about removing redundant attributes from the information system in such a way that the set of remaining attributes preserves only part of knowledge that is really useful. However, as proved in [11], the problems of generating the minimal reduct and minimal dependency are both NP-hard. Thus, the problem of finding all reducts as well as finding the minimal reduct using rough set theory may only be effective on small data sets. Knowledge reduction can be categorised into to finding reducts or relative reducts [7]. The concept of a reduct is built based on the idea of information systems and is easy to apply in applications. In contrast, the concept of a relative reduct is built based on decision systems. In fact, there are various definitions of reducts. The positive region reduct [8] is the most popular one, which is defined based on lower and upper approximation sets for defining an attribute reduct, for examples, Shannon's information entropy based reduct [8], classical rough set model based reduct[10], reduct based on discernibility matrix and discernibility function [16], reduct based on heuristic search [17].

Most of reducts implied that they are attribute reducts and not object reducts. [7] defines a reduct based on a distance measure with three evaluation metrics: *finding optimal*, *maximal exceeding*, *average exceeding*. Some authors introduce the concept of *variable precision rough set* [6] in which the concept of β lower distribution reduct and β upper distribution reduct are used. In these works, the equivalent definitions are given and the relationships among β lower, β upper distribution reducts and alternative types of knowledge reduction in inconsistent systems [16] are investigated. Moreover, with some special threshold, β lower and β upper distribution reducts are equivalent to the maximum distribution and the possible reduct, respectively. The authors use discernibility matrices associated with the β lower and upper distribution reducts from which the approaches to knowledge reduction in variable precision rough set can be obtained. [4] propose to use fuzzy rough set and information granulation for finding attribute reduct, and obtain different semantics in numerical attribute reduct and categorical attribute reduct. [4] derive several attribute significance measures based on the proposed fuzzy-rough model. From this, authors construct a greedy forward algorithm to find attribute reduct as feature subset selection. [4] also propose two strategies in attribute subset selection such as wrapper and filter for granular computing by using fuzzy information granules from numerical features and transforming numerical attribute into fuzzy linguistic variables. Some studies find attribute reducts according to the definition of concept lattices [1]. By combining rough set theory and formal concept analysis, these studies obtained reduction of a context by deleting rows (object oriented concept lattice) or columns (attribute oriented concept lattice) or both. Then, based on granular computing theory, they use the information granules or discernibility matrix and discernibility function to explore the attribute reduct. Therefore, the relationships between the attribute reducts of the concept lattices and the attribute reducts of the information system in rough set theory are found. Because of non-polynomial time complexity, most of algorithms mentioned above have to use a heuristic approach to search for reducts.

In this paper we propose two methods for dealing with the problem of finding all reduct attributes (or non-redundant attributes), columns of the consistent decision table are involved at least in one of attribute reducts, and the problem of finding an object reduct that removes redundant objects, rows of the consistent decision table are no effect to finding set of all attribute reducts over decision attributes. The proposed methods are proved theoretically having polynomial complexity in running time. Moreover, by combining the two methods, we can obtain a consistent decision table that its size is reduced in both horizontal and vertical dimensions. Our ideas are based on some basis concepts of relational database theory [2, 3, 12] and rough set theory [8, 11]. In relational dabase theory, the basis important concept is the concept of minimal keys and antikeys. They form the so-called Sperner-systems. We consider decision tables that can be regarded as relation tables in relational database theory. Decision tables and relation tables are tables containing rows and columns. A decision table has an attribute set that can be divided into the condition set and the decision set. It is obvious that there is a correspondence between function dependencies in a relation and dependencies in a decision table. By applying methods of finding keys and antikeys, we construct keys and antikeys for a consistent decision table. Some results in relation about keys and antikeys have polynomial time complexity. By using these results of minimal keys and antikeys and based on the maximal equality set definition, we build an algorithm for finding a reduct of consistent decision table in polynomial time. To the best of our knowledge, this is the first time some interesting results in the relational database theory are directly applied in efficiently finding reducts from decision tables.

The rest of the paper is organized as following. In Section 2, we give necessary notions and definitions regarding the relational database theory and rough set theory which will be later used in the paper. In Section 3, we describe our proposed methods for object reduct and attribute reduct from a consistent decision table. In Section 4, we give a case study to illustrate our proposed methods. We summarize the paper in Section 5.

2 Preliminaries

In this section we show some basis concepts of relational database theory [2, 3, 12] and rough set theory [8, 9, 11, 13].

2.1 Relational database theory

Definition 1. *Let $R = \{a_1, ..., a_n\}$ be a finite set of attributes and let $D(a_i)$ be the set of all possible values of attribute a_i, the* relation *r over R is the set of tuples $\{h_1, ..., h_m\}$ where $h_j : R \rightarrow \bigcup_{a_i \in R} D(a_i), 1 \leq j \leq m$, is a function that $h_j(a_i) \in D(a_i)$.*

Definition 2. *Let $r = \{h_1, ..., h_m\}$ be a relation over $R = \{a_1, ..., a_n\}$. Any pair of attribute sets $A, B \subseteq R$ is called* functional dependency *(FD) over R, and it is* *denoted by* $A \to B$ *if and only if*

$$(\forall h_i, h_j \in r)((\forall a \in A)(h_i(a) = h_j(a)) \Rightarrow (\forall b \in B)(h_i(b) = h_j(b))).$$

Definition 3. *The set* $F_r = \{(A, B) : A, B \subset R, A \to B\}$ *is called a* full family *of functional dependencies in* r*. Let* $P(R)$ *be the power set of attribute set* R*. A family* $F \subseteq P(R) \times P(R)$ *is called a* f-family *over* R *if and only if for all subsets of attributes* $A, B, C, D \subseteq R$ *the following properties hold:*
1) $(A, A) \in F$.
2) $(A, B) \in F, (B, C) \in F \Rightarrow (A, C) \in F$.
3) $(A, B) \in F, A \subseteq C, D \subseteq B \Rightarrow (C, D) \in F$.
4) $(A, B) \in F, (C, D) \in F \Rightarrow (A \cup C, B \cup D) \in F$.

Clearly, F_r is an f-family over R. It is also known that if F is an f-family over R, then there is a relation r such that $F_r = F$. Let us denote by F^+ the set of all FDs, which can be derived from F by using rules 1)-4).

Definition 4. *A pair* $s = \langle R, F\rangle$*, where* R *is a set of attributes and* F *is a set of FDs on* R*, is called a* relation schema*. For any* $A \subseteq R$*, the set* $A^+ = \{a : A \to \{a\} \in F^+\}$ *is called the* closure *of* A *on* s*. It is clear that* $A \to B \in F^+$ *if and only if* $B \subseteq A^+$*. Similarly,* $A_r^+ = \{a : A \to \{a\} \in F^+\}$ *is called the closure of* A *on relation* r*.*

Definition 5. *Let* r *be a relation,* $s = \langle R, F\rangle$ *be a relation scheme and* $A \subseteq R$*. Then* A *is a* key *of* r *(a key of* s*) if* $A \to R$ *(*$A \to R \in F^+$*).* A *is a* minimal key *of* r *(*s*) if* A *is a key of* r *(*s*) and any proper subset of* A *is not key of* r *(*s*). The set of all minimal keys of* r *(*s*) is denoted by* K_r *(*K_s*). A family* $K \subseteq P(R)$ *is a* Sperner-system *on* R *if for any* $A, B \in K$ *implies* $A \not\subseteq B$*. It is clear that* K_r *(*K_s*) are Sperner-systems.*

Definition 6. *Let* K *be a Sperner-system over* R *as the set of all minimal keys of* s*. We defined the set of* antikeys *of* K*, denoted by* K^{-1}*, as follows:*

$$K^{-1} = \{A \subset R : (B \in K) \Rightarrow (B \not\subset A) \text{ and if } (A \subset C) \Rightarrow (\exists B \in K)(B \subseteq C)\}.$$

It is easy to see that K^{-1} is the set of subsets of R, which does not contain the element of K and which is maximal for this property. They are the maximal non-keys. Clearly, K^{-1} is also a Sperner-system.

Definition 7. *Let* r *be a relation over* R*. Denote* $E_r = \{E_{ij} : 1 \le i \le j \le |r|\}$*, where* $E_{ij} = \{a \in R : h_i(a) = h_j(a)\}$*. Then* E_r *is called an* equality set *of* r*.*

For $A_r \in R, A_r^+ = \cap E_{ij}$, if there exists $E_{ij} \in E_r : A \subseteq E_{ij}$, otherwise $A_r^+ = R$.

Definition 8. *Let* $r = \{h_1, ..., h_m\}$ *be a relation over* R*,* E_r *is the equality set of* r*. Let*

$$M_r = \{E_{ij} \in E_r : \forall E_{st} \in E_r : E_{ij} \subseteq E_{st}, E_{ij} \neq E_{st}\}$$

where $1 \le i < j \le m$*,* $1 \le s < t \le m$*.* M_r *is called the* maximal equality system *of* r*.*

Definition 9. *Let $s = \langle R, F \rangle$ be a relation scheme over R and $a \in R$. The set*

$$K_a^s = \{A \subseteq R : A \to \{a\}, \nexists B : (B \to \{a\})(B \subset A)\}$$

is called a family of minimal sets of the attribute a over s. Similarly, the set

$$K_a^r = \{A \subseteq R : A \to \{a\}, \nexists B \subseteq R : (B \to \{a\})(B \subset A)\}$$

is called a family of minimal sets of the attribute a over r.

Definition 10. *If K is a Sperner-system over R as the family of minimal sets of the attribute a over r (or s); in other words $K = K^r$ (or $K = K^s$), then $K^{-1} = (K_a^r)^{-1}$ (or $K^{-1} = (K_a^s)^{-1}$) is the family of maximal subsets of R which are not the family of minimal sets of the attribute a, defined as:*

$$(K_a^r)^{-1} = \{A \subseteq R : A \to \{a\} \notin F_r^+, A \subset B \Rightarrow B \to \{a\} \in F_r^+\},$$
$$(K_a^s)^{-1} = \{A \subseteq R : A \to \{a\} \notin F^+, A \subset B \Rightarrow B \to \{a\} \in F^+\}.$$

It is clear that $R \notin K_a^s$, $R \notin K_a^r$, $\{a\} \in K_a^s$, $\{a\} \in K_a^r$ and K_a^s, K_a^r are Sperner-systems over R.

2.2 Rough set theory

Definition 11. *An* information system *S is an order quadruple $S = (U, A, V, f)$ where U is a finite set of* objects, *called the* universe*; A is a finite set of* attributes*; $V = \bigcup_{a \in A} V_a$ and V_a is the* domain *of attribute a; $f : U \times A \to V$ is a total function, such that $f(x, a) \in V_a$ for every $a \in A$ and $x \in U$ called the* information function. *The function $f_x : A \to V$ such that $f_x(a) = f(x, a)$ for every $a \in A$ and $x \in U$ will be called* information about *x in S. We denote $a(x) = f_x(a)$. If $B = \{b_1, b_2, ..., b_k\} \subseteq A$ is subset of attributes, then the set of $b_i(x)$ is denoted as $B(x)$. Therefore, if x, y are two objects in U, then $B(x) = B(y)$ if and only if $b_i(x) = b_i(y), \forall i = 1, ..., k$.*

Definition 12. Decision table *is an information system $S = (U, A, V, f)$, where $A = C \cup D$ and $C \cap D = \emptyset$. Without loss of generality, suppose that D consists of only one decision attribute d. Therefore, from this time we consider the decision table $DS = (U, C \cup \{d\}, V, f)$, where $\{d\} \notin C$.*

Definition 13. *Let* decision table *$DS = (U, C \cup \{d\}, V, f)$, $U = \{u_1, ..., u_m\}$ be a* relation *over $C \cup \{d\}$. A decision table DS is* consistent *if and only if the functional dependency $C \to \{d\}$ is true; it means that for any $x, y \in U$ if $C(x) = C(y)$ then $d(x) = d(y)$. Conversely, DS is inconsistent.*

Definition 14. *Every attribute subset $P \subseteq C \cup D$ determines an* indiscernibility relation
$IND(P) = \{(u, v) \in U \times U | \forall a \in P, f(u, a) = f(v, a)\}$
$IND(P)$ determines a partition *of U which is denoted by U/P.*
Any element $[u]_P = \{v \in U | (u, v) \in IND(P)\}$ in U/P is called an equivalent class.

- B-upper approximation *of X is the set* $\overline{B}X = \{u \in U | [u]_B \cap X \neq \emptyset\}$,
- B-lower approximation *of X is the set* $\underline{B}X = \{u \in U | [u]_B \subseteq X\}$ *with* $B \subseteq C$, $X \subseteq U$,
- B-boundary *is the set* $BN_B(X) = \overline{B}X \backslash \underline{B}X$,
- B-positive region *of D is the set* $POS_B(D) = \bigcup_{X \in U/D} (\underline{B}X)$

Definition 15. *Let* $DS = (U, C \cup \{d\}, V, f)$ *be a decision table. If* $B \subseteq C$ *satisfies*
1) $POS_B(D) = POS_C(D)$
2) $\forall b \in B, POS_{B-\{b\}}(D) \neq POS_C(D)$
then B is called attribute reduct *of C.*

If DS is a consistent decision table, B is an attribute reduct *of C if B satifies* $B \rightarrow \{d\}$ *and* $\forall B' \subset B$, $B' \not\rightarrow \{d\}$. *Let* $RED(C)$ *be the set of all reducts of C. From definition 15 and formula* K_a^r *in definition 9 we have* $RED(C) = K_d^r - \{d\}$ *where* K_d^r *is the family of all minimal set of the attribute* $\{d\}$ *over* $r = \langle U, C \cup \{d\} \rangle$

3 Object reduct and attribute reduct

In this section, we construct some methods to finding all non-redundant attributes, an object reduct and an attribute reduct and all of them have complexity in polynomial. A lot of existing approaches try to find all attribute reducts first, and then select the most *suitable* one. Unfortunately, the problem of finding all attribute reducts of consistent decision table is exponential in the number of attributes [5]. Because of exponential computational time, many research using heuristic methods to find an attribute reduct [1, 4, 6, 7, 10, 16, 17]. The method of finding an attribute we propose is not a heuristic algorithm. First, we eliminate all redundant attributes, that are not involved in any attribute reduct of consistent decision table, by using the algorithm 1. After that, we build two algorithms that one find an object reduct, the algorithm 2, and another find an attribute reduct, the algorithm 4. The combination of these methods generate a consistent decision table that is reduced in size in both vertical and horizontal dimensions. These results will reduce cost of storage data, specially for massive dataset, and the object reduct completely preserve information for finding all attribute reducts.

Lemma 1. *Let* $DS = (U, C \cup \{d\}, V, f)$ *be a consistent decision table where* $C = \{c_1, c_2, ..., c_n\}, U = \{u_1, u_2, ..., u_n\}$. *Let us consider* $r = \{u_1, u_2, ..., u_m\}$ *on the attribute set* $R = C \cup \{d\}$.
We set $E_r = \{E_{ij} : 1 \leq i < j \leq m\}$ *where* $E_{ij} = \{a \in R : a(u_i) = a(u_j)\}$.
We set $M_d = \{A \in E_r : d \notin A, \nexists B \in E_r : d \notin B, A \subset B\}$.
Then we have $M_d = (K_d^r)^{-1}$ *where* K_d^r *is a family of minimal sets of the attribute* $\{d\}$ *over relation r.*

The lemma 1 is proved in [13].

Theorem 1. *[2] Let K be a Sperner-system over Ω. Then*

$$\bigcup_{A \in K} A = \Omega - \bigcap_{B \in K^{-1}} B$$

Definition 16. *Given a consistent decision table $DS = (U, C \cup \{d\}, V, f)$, let DS be relation $U = \{u_1, ..., u_m\}$ over attribute set $R = C \cup \{d\}$, from definition 15 we have $RED(C) = K_d^r - \{d\}$, if denote $REAT(C)$ a set of all non-redundant attributes or reduct attributes of C then:*

$$REAT(C) = \bigcup_{A \in RED(C)} A = \left(\bigcup_{A \in K_d^r} A \right) - \{d\}$$

Algorithm 1 Finding the set of all reduct attributes of C

```
Function REAT($DS = (U, C \cup \{d\}, V, f)$, $POS_C(\{d\}) = U$, $C = \{c_1, ..., c_n\}$, $U = \{u_1, ..., u_m\}$)
    Consider the relation $r = \{u_1, ..., u_m\}$ over the attribute set $R = C \cup \{d\}$.
    Step 1: Compute $E_r = \{A_1, ..., A_t\}$
    Step 2: Compute $M_d = \{A \in E_r : d \notin A, \nexists B \in E_r : d \notin B, A \subset B\}$.
    Step 3: Construct $N = R - \bigcap_{B \in M_d} B$
    Step 4: Set $REAT(C) = N - \{d\}$
```

Theorem 2. *$REAT(C)$ is set of all reduct attributes of C.*

Proof. The theorem 2 is proved in [14]. It is restated as follows:
By lemma 1 $M_d = (K_d^r)^{-1}$. At step 3, combine with definition 6, $(K_d^r)^{-1}$ and (K_d^r) are Sperner-systems, with theorem 1 we have:

$$N = R - \bigcap_{B \in M_d} B = R - \bigcap_{B \in \left(K_d^r\right)^{-1}} B = \bigcup_{A \in K_d^r} A$$

At step 4 we have:

$$REAT(C) = N - \{d\} = \left(\bigcup_{A \in K_d^r} A \right) - \{d\} = \bigcup_{A \in RED(C)} A$$

Thus, by definition 16, $REAT(C)$ is the set of all reduct attributes of C, $REAT(C)$ is the set of all non-redundant attributes of C. □

It can be seen that the number of computational steps of E_r is not greater than $|U|^2$ and the number of computational steps of M_d is not greater than $|E_r|^2$. Thus, the worst case time computational complexity of the algorithm is $O(|U|^4 + |C \cup \{d\}|)$ which is polynomial by number of rows and columns of decision table DS.

Definition 17. *An object reduct of a consistent decision table $DS = (U, C \cup \{d\}, V, f)$ is a consistent decision table, $DS' = (U', C \cup \{d\}, V, f)$, where $RED(C) = RED_U(C)$ and:*
1) $U' \subseteq U$,
2) $RED_U(C) = RED_{U'}(C)$,
3) $RED_U(C) \neq RED_{U'-\{u\}}(C)$, $\forall u \in U'$.

Algorithm 2 Finding an object reduct over consistent decision table

```
Function ObjectReduct($DS = (U, C \cup \{d\}, V, f)$)
1: Step 1: Compute $E_r = \{A_1, ..., A_t\}$
2: Step 2: Compute $M_d^U = \{A \in E_r : d \notin A, \nexists B \in E_r : d \notin B, A \subset B\}$.
3: Step 3: Set $T(0) = U = \{u_1, ..., u_m\}$
4: Step 4: Set
       $T(i+1) = \begin{cases} T(i) - u_{i+1}, & \text{if } M_d^{T(i)-u_{i+1}} = M_d^U \\ T(i), & \text{otherwise} \end{cases}$
5: Then we set $U' = T(m)$.
```

Theorem 3. *$T(m)$ satisfies the two conditions 1), 2) and 3) in definition 17.*

Proof. We prove the theorem by induction. At basis step $T(0) = U$, clearly, $U' = U$, $RED_{U'}(C) = RED_U(C)$ thus the two conditions 1), 2) are satisfied. At inductive step, assume that we have $T(i) = U(i)$ satisfies two conditions 1), 2) in definition 17. We have to prove that $T(i+1) = U(i+1)$ satisfies the two conditions.

- In the first case: If $T(i+1) = T(i)$ then it is obvious that $U(i+1) = U(i)$, $RED_{U(i+1)}(C) = RED_{U(i)}(C) = RED(C)$ by induction hypothesis. Thus, $T(i+1)$ satisfies the two conditions 1), 2) in definition 17.

- In the second case: If $T(i+1) = T(i) - \{u_{i+1}\}$ then $M_d^U = M_d^{U(i+1)}$. By lemma 1, $M_d^U = \left(K_d^U\right)^{-1}$ where ($U = \{u_1, ..., u_m\}$) $\Rightarrow M_d^{U(i+1)} = \left(K_d^{U(i+1)}\right)^{-1} \Rightarrow \left(K_d^U\right)^{-1} = \left(K_d^{U(i+1)}\right)^{-1}$. By definition 6 and 10 (K and K^1 are uniquely determined by one another), it can see that $\left(K_d^U\right) = \left(K_d^{U(i+1)}\right)$. From definition 15 and the result of definition 15, we have $RED_U(C) = \left(K_d^U\right) - \{d\}$ and $RED_{U(i+1)}(C) = \left(K_d^{U(i+1)}\right) - \{d\} \Rightarrow$ (ii1) $RED_U(C) = RED_{U(i+1)}(C)$. From induction hypothesis, we have (ii2) $RED_U(C) = RED_{U(i)}(C)$. From (ii1), (ii2) we obtain $RED_U(C) = RED_{U(i)}(C) = RED_{U(i+1)}(C)$. Because $RED_U(C) = RED(C)$ is a Sperner-system (by definition K_d^U is a Sperner-system and $\Rightarrow K_d^U - \{d\}$ is a Sperner-system), $RED_{U(i)}(C)$ and $RED_{U(i+1)}(C)$ are Sperner-systems. Finally, the two conditions in definition 17 are satisfied at step $i+1$ as follow:

1) $U(i+1) \subseteq U(i)$,
2) $RED_{U(i+1)}(C) = RED_{U(i)}(C) = ... = RED_U(C) = RED(C)$

When $i+1=m$ then algorithm 2 stops. Now we need to show that $U(m)$ satisfies the condition 3) in definition 17 which means that $RED_{U(m)-u}(C) \neq RED_U(C)$ where $\forall u \in U(m)$. Assume that there exists $u = u_{i+1}$, $u \in U(m)$ such that $RED_{U(m)-u_{i+1}}(C) = RED_U(C)$ $(ii3)$. By definition 15, $RED_{U(m)-u_{i+1}}(C) = K_d^{U(m)-u_{i+1}} - \{d\}$ and $RED_U(C) = K_d^U - \{d\}$, thus

$$(ii3) \Leftrightarrow K_d^{U(m)-u_{i+1}} - \{d\} = K_d^U - \{d\} \Leftrightarrow K_d^{U(m)-u_{i+1}} = K_d^U\,(ii4)$$

By definition 6, 10 and lemma 1 (K and K^{-1} are uniquely determined by one another), it means that

$$(ii4) \Leftrightarrow \left(K_d^{U(m)-u_{i+1}}\right)^{-1} = \left(K_d^U\right)^{-1} \Leftrightarrow M_d^{U(m)-u_{i+1}} = M_d^U\,(ii5)$$

By above proving induction, if $M_d^{U(m)-u_{i+1}} = M_d^U$ then u_{i+1} will be removed, thus $u_{i+1} \notin U(m)$ contradicts with hypothesis $u = u_{i+1} \in U(m)$. Hence, the condition 3) in definition 17 is satisfied. The theorem is proved. □

It is clear that the number of steps computing E_r by definition 7 is less than $|U|^2$. The number of steps computing M_d is less than $|E_r|^2$ and $|E_r| \leq \frac{|U|(|U|-1)}{2}$. Thus, the worst-case time complexity of algorithm 2 is not greater than $O(|U|^5)$. If we change the order of the universe set U, we can find another object reduct.

Algorithm 3 Finding the minimal key from a set of antikeys

```
Function MinimalKey(Let $K$, $H$ be Sperner-systems and $C = \{c_1, ..., c_n\} \subseteq U$
    such that $H^{-1} = K$ and $\exists B \in K : B \subseteq C$)
  Step 1: We set $A(0) = C$
  Step $i+1$: Set
      $A(i+1) = \begin{cases} A(i) - \{c_{i+1}\}, & \text{if } \forall B \in K : A(i) - \{c_{i+1}\} \not\subseteq B \\ A(i), & \text{otherwise} \end{cases}$
  Then we set $D = A(n)$.
```

Lemma 2. *[12] If K is a set of antikeys, then $A(n) \in H$.*

Algorithm 4 Finding an attribute reduct from a consistent decision table

```
Function OneAttributeReduct($DS = (U, C \cup \{d\}, V, f)$)
Step 1: Compute $E_r = \{A_1, ..., A_t\}$
Step 2: Compute $M_d = \{A \in E_r : d \notin A, \nexists B \in E_r : d \notin B, A \subset B\}$.
Step 3: Set $H(0) = C = \{c_1, ..., c_n\}$
Step 4: Set
    $$H(i+1) = \begin{cases} H(i) - c_{i+1}, & \text{if } \nexists B \in M_d : H(i) - c_{i+1} \subseteq B \\ H(i), & \text{otherwise} \end{cases}$$
Then we set $D = H(n)$.
```

Theorem 4. *$H(n) \in RED(C)$, where $H(n)$ in algorithm 4.*

Proof. The algorithm 4 is based on the algorithm 3. By lemma 1, $(K_d^r)^{-1} = M_d$. By lemma 2, $H(n) \in K_d^r$ (1). By the result of definition 15, $RED(C) = K_d^r - \{d\}$ (2). At step 3 of algorithm 4 we set $C = \{c_1, ..., c_n\}$ then $d \notin C$. Thus, in algorithm 4 we have $d \notin H(n)$ (3). From (1) and (3) we have $H(n) \in K_d^r - \{d\}$ (4). From (2) and (4) we obtain $H(n) \in RED(C)$. The theorem is proved. □

Similar to the algorithm 2, the time complexity of algorithm 4 is not greater than $O(|C| \times |U|^4)$. If we change the order of the set C in step 3 we can get another attribute reduct of the consistent decision table DS. Thus, the problem of finding all attribute reducts is exponential time complexity in the number of attributes [5].

In order to reduce the size of consistent decision table in both vertical and horizontal dimensions, the first step in our method is to use the algorithm 1 to determine $REAT(C)$ and then use the algorithm 2 to get an object reduct. So $REAT(C)$ is the set of all reduct attributes, we obtain reduction in the horizontal dimension, reducing number of columns, of the consistent decision table. After that the object reduct is reduction in the vertical dimension, reducing number of rows, of the consistent decision table. It is easy to prove that our method run in polynomial time because the algorithm 1 and 2 are polynomial time complexity. It is obvious that the consistent decision table that is reduced both vertically and horizontally occupies much less capacity of storage than the original, but it preserves all necessary information for finding all attribute reducts. In addition, our method applies the algorithm 4 to find an attribute reduct that run in polynomial time and the attribute reduct help more and more efficiently and effectively in learing process.

4 A case study

Example 1. Given a consistent decision table $DS = (U, C \cup \{d\}, V, f)$
where $U = \{u_1, ..., u_{14}\}$ $(\{1, ..., 14\})$,

d is decision attribute "Play Golf",
$C = \{Outlook, Grass, Temperature, Humidity, Windy, NumberHoles\}$
($\{o, g, t, h, w, n\}$ or $\{ogthwn\}$),
$R = C \cup \{d\} = \{ogthwnd\}$.
$V_{Outlook} = \{Sunny, OverCast, Rain\}$,
$V_{Temperature} = \{High, Middle, Low\}$,
$V_{Humidity} = \{High, Middle\}$,
$V_{Grass} = \{Wet, Dry\}$,
$V_{Windy} = \{Weak, Strong\}$,
$V_{NumberHoles} = \{20, 10\}$,
$V_d = \{No, Yes\}$, $V = V_{Outlook} \cup V_{Grass} \cup V_{Temperature} \cup V_{Humidity} \cup V_{Windy} \cup V_{NumberHoles} \cup V_d$,
and function $f : U \times C \cup \{d\} \to \bigcup_{a \in C} V_a$ as table 1.

Table 1: A consistent decision table

No.	O	G	T	H	W	N	d
1	Sunny	Wet	High	High	Weak	10	No
2	Sunny	Dry	High	High	Strong	20	No
3	Overcast	Wet	High	High	Weak	10	Yes
4	Rain	Dry	Middle	High	Weak	10	Yes
5	Rain	Wet	Low	Middle	Weak	20	Yes
6	Rain	Wet	Low	Middle	Strong	20	No
7	Overcast	Dry	Middle	Middle	Strong	20	Yes
8	Sunny	Wet	Low	High	Weak	10	No
9	Sunny	Wet	Middle	Middle	Weak	10	Yes
10	Rain	Dry	Middle	Middle	Weak	20	Yes
11	Sunny	Dry	Middle	Middle	Strong	20	Yes
12	Overcast	Dry	Middle	High	Strong	10	Yes
13	Overcast	Dry	High	Middle	Weak	20	Yes
14	Rain	Dry	Middle	High	Strong	10	No

Example 2. (continue the example 1) By applying algorithm 1 we have that E_r contains all $E_{i,j}$ as follows:

$E_{1,2} = othd$, $E_{1,3} = gthwn$, $E_{1,4} = hwn$, $E_{1,5} = gw$, $E_{1,6} = gd$, $E_{1,8} = oghwnd$, $E_{1,9} = ogwn$, $E_{1,10} = w$, $E_{1,11} = o$, $E_{1,12} = hn$, $E_{1,13} = tw$, $E_{1,14} = hnd$, $E_{2,3} = th$, $E_{2,4} = gh$, $E_{2,5} = n$, $E_{2,6} = wnd$, $E_{2,7} = gwn$, $E_{2,8} = ohd$, $E_{2,10} = gn$, $E_{2,12} = ghw$, $E_{2,13} = gtn$, $E_{2,14} = ghwd$, $E_{3,4} = hwnd$, $E_{3,5} = gwd$, $E_{3,6} = g$, $E_{3,7} = od$, $E_{3,8} = ghwn$, $E_{3,9} = gwnd$, $E_{3,10} = wd$, $E_{3,11} = d$, $E_{3,12} = ohnd$, $E_{3,13} = otwd$, $E_{4,5} = owd$, $E_{4,7} = gtd$, $E_{4,9} = twnd$, $E_{4,10} = ogtwd$, $E_{4,12} = gthnd$, $E_{4,14} = ogthn$, $E_{5,8} = gtw$, $E_{5,10} = ohwnd$, $E_{6,10} = ohn$, $E_{7,9} = thd$, $E_{7,11} = gthwnd$, $E_{7,13} = oghnd$, $E_{9,10} = thwd$, $E_{9,12} = tnd$, $E_{9,13} = hwd$, $E_{9,14} = tn$,

$E_{10,13} = ghwnd$, $E_{10,14} = ogt$, $E_{11,12} = gtwd$, $E_{11,13} = ghnd$, $E_{12,13} = ogd$

$M_d = \{gthwn, ogthn, ogwn\} = \{M_1, M_2, M_3\}$

$G = \bigcap_{M \in M_d} = M_1 \cap M_2 \cap M_3 = \{gthwn\} \cap \{ogthn\} \cap \{ogwn\} = \{gn\}$

$REAT(C) = N - \{d\} = R - G - \{d\} = \{ogthwnd\} - \{gn\} - \{d\} = \{othw\}$

Thus, the two attributes "Grass" and "NumberHoles" are redundant, by removing these two attributes, we obtain non-redundant attributes consistent decision table $NOREDS = (U, \{o,t,h,w\} \cup \{d\}, V, f)$ of the consistent decision table 1. From this example, we consider $C = \{o,t,h,w\}$ instead of $C = \{o,g,t,h,w,n\}$.

Example 3. (continue example 2) By defintion 14 we find $RED(C)$.
$POS_o(\{d\}) = \{3, 7, 12.13\}$,
$POS_t(\{d\}) = \emptyset, POS_h(\{d\}) = \emptyset, POS_w(\{d\}) = \emptyset$,
$POS_{ot}(\{d\}) = \{1, 2, 3, 7, 8, 9, 11, 12, 13\}$,
$POS_{oh}(\{d\}) = \{1, 2, 3, 7, 8, 9, 11, 12, 13\}$,
$POS_{ow}(\{d\}) = \{3, 4, 5, 6, 7, 12, 13, 14\}$,
$POS_{th}(\{d\}) = \{7, 8, 9, 10, 11, 13\}$,
$POS_{tw}(\{d\}) = \{2, 4, 6, 9, 10\}$,
$POS_{hw}(\{d\}) = \{5, 9, 10, 13\}$,
$POS_{oth}(\{d\}) = \{1, 2, 3, 7, 8, 9, 10, 11, 12, 13\}$,
$POS_{otw}(\{d\}) = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}$,
$POS_{ohw}(\{d\}) = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}$,
$POS_{thw}(\{d\}) = \{2, 4, 5, 6, 7, 8, 9, 10, 11, 13\}$.
We see that $POS_{\{otw\}}(\{d\}) = POS_{\{ohw\}}(\{d\}) = POS_C(\{d\})$. By definition 15, the set of all reducts of $C = \{othw\}$ over the consistent decision table $DS = \{U, C \cup \{d\}, V, f\}$ in example 2 is $RED(C) = \{otw, ohw\}$

Example 4. (continue example 2) In step 1 in algorithm 2, from example 2 and definition 7, for each pair of rows (i, j), we construct the sets E_{ij}. We have:

$E_{1,2} = \{othd\}$, $E_{1,3} = \{thw\}$. By doing the same thing with pairs (1, 4), ..., (1, 14), (2, 3), (2, 4), ..., (13, 14) we obtain the set E_r containing sets A_i as follows:

$A_1 = \{othd\}$, $A_2 = \{thw\}$, $A_3 = \{hw\}$, $A_4 = \{w\}$, $A_5 = \{d\}$, $A_6 = \{ohwd\}$, $A_7 = \{ow\}$, $A_8 = \{o\}$, $A_9 = \{h\}$, $A_{10} = \{tw\}$, $A_{11} = \{hd\}$, $A_{12} = \{th\}$, $A_{13} = \{wd\}$, $A_{14} = \{ohd\}$, $A_{15} = \{t\}$, $A_{16} = \{hwd\}$, $A_{17} = \{od\}$, $A_{18} = \{otwd\}$, $A_{19} = \{owd\}$, $A_{20} = \{td\}$, $A_{21} = \{twd\}$, $A_{22} = \{thd\}$, $A_{23} = \{oth\}$, $A_{24} = \{oh\}$, $A_{25} = \{thwd\}$, $A_{26} = \{ot\}$

$$E_r = \{A_1, ..., A_{26}\} = E_r^U$$

Example 5. (continue example 4) In step 2 of algorithm 2, we construct the set M_d^U being the maximal equality system of E_r that do not have decision attribute d. We obtain:

$$M_d = M_d^U = \{thw, oth, ow\} = \{B_1, B_2, B_3\}$$

Example 6. (continue example 5) In step 3 and step 4 of algorithm 2 we have: $T(0) = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}$, by using definition 7 and formula at

step 2 in algorithm 2 we compute:
$M_d^{T(0)-\{1\}} = \{thw, oth, ow\} = M_d^U \Rightarrow T(1) = T(0) - \{1\}$
$M_d^{T(1)-\{2\}} = \{thw, oth, ow\} = M_d^U \Rightarrow T(2) = T(1) - \{2\}$
$M_d^{T(2)-\{3\}} = \{thw, oth, ow\} = M_d^U \Rightarrow T(3) = T(2) - \{3\}$
$M_d^{T(3)-\{4\}} = \{thw, oth, ow\} = M_d^U \Rightarrow T(4) = T(3) - \{4\}$
$M_d^{T(4)-\{5\}} = \{thw, ow, oh, ot\} \neq M_d^U \Rightarrow T(5) = T(4)$
$M_d^{T(5)-\{6\}} = \{thw, ow, ot\} \neq M_d^U \Rightarrow T(6) = T(5)$
$M_d^{T(6)-\{7\}} = \{thw, oth, ow\} = M_d^U \Rightarrow T(7) = T(6) - \{7\}$
$M_d^{T(7)-\{8\}} = \{thw, oth\} \neq M_d^U \Rightarrow T(8) = T(7)$
$M_d^{T(8)-\{9\}} = \{thw, oth\} \neq M_d^U \Rightarrow T(9) = T(8)$
$M_d^{T(9)-\{10\}} = \{thw, oth, ow\} = M_d^U \Rightarrow T(10) = T(9) - \{10\}$
$M_d^{T(10)-\{11\}} = \{thw, oth, ow\} = M_d^U \Rightarrow T(11) = T(10) - \{11\}$
$M_d^{T(11)-\{12\}} = \{oth, ow, tw\} \neq M_d^U \Rightarrow T(12) = T(11)$
$M_d^{T(12)-\{13\}} = \{thw, oth, ow\} = M_d^U \Rightarrow T(13) = T(12) - \{13\}$
$M_d^{T(13)-\{14\}} = \{oth, ow, tw\} \neq M_d^U \Rightarrow T(14) = T(13)$
Set $U' = T(14) = \{5, 6, 8, 9, 12, 14\}$ then $OBREDS = (\{5, 6, 8, 9, 12, 14\}, C \cup \{d\}, V, f)$ is the object reduct of the consistent decision table $NOREDS$

Example 7. (continue example 2, 3 and 6) Based on the object reduct of the consistent decision table $OBREDS = DS' = (U', C \cup \{d\}, V, f)$ from example 6, we use definition 14 to find $RED_{U'}(C)$.
$POS'_o(\{d\}) = \{12\}$
$POS'_t(\{d\}) = \emptyset, POS'_h(\{d\}) = \emptyset, POS'_w(\{d\}) = \emptyset$
$POS'_{ot}(\{d\}) = \{8, 9, 12, 14\}$
$POS'_{oh}(\{d\}) = \{8, 9, 12, 14\}$
$POS'_{ow}(\{d\}) = \{5, 6, 12, 14\}$
$POS'_{th}(\{d\}) = \{8, 9\}$
$POS'_{tw}(\{d\}) = \{6, 9\}$
$POS'_{hw}(\{d\}) = \{5, 6, 8, 9, 12, 14\}$
$POS'_{oth}(\{d\}) = \{8, 9, 12, 14\}$
$POS'_{otw}(\{d\}) = \{5, 6, 8, 9, 12, 14\}$
$POS'_{ohw}(\{d\}) = \{5, 6, 8, 9, 12, 14\}$
$POS'_{thw}(\{d\}) = \{5, 6, 8, 9\}$
We see that $POS'_{\{hw\}}(\{d\}) = POS'_{\{otw\}}(\{d\}) = POS'_{\{ohw\}}(\{d\}) = U'$. Let the set $P = \{POS'_B(\{d\})\} = \{hw, otw, ohw\}$. Because U' is an object reduct of U according to the definition of the maximal equality system of attribute $\{d\}$, the set of all reducts of C is a Sperner-system.
Thus, $RED_{U'}(C) = \{B \in P, \nexists A \in P, A \subset B\}$. In P, $\{hw\} \subset \{ohw\}$, we remove $\{hw\}$ and P becomes a Sperner-system. It is obvious that $RED_{U'}(C) = P - \{hw\} = \{otw, ohw\} = RED_U(C)$. Clearly, $RED_{U'}(C)$ generated by object reduct in the consistent decision table DS' equals to $RED_U(C)$ generated by the original consistent decision table DS.

Example 8. (continue example 5) By applying algorithm 4 we find only one attribute reduct on consistent decision table from example 2, $C = \{othw\}$
$temp = H(0) - \{o\} = \{thw\} = B_1 \in M_d \Rightarrow H(1) = H(0)$
$temp = H(1) - \{t\} = \{ohw\} \not\subseteq \{B \in M_d\} \Rightarrow H(2) = temp$
$temp = H(2) - \{h\} = \{ow\} = B_3 \in M_d \Rightarrow H(3) = H(2)$
$temp = H(3) - \{w\} = \{oh\} \subseteq B_2 \in M_d \Rightarrow H(4) = H(3)$
Set $H = H(4) = \{ohw\}$ and algorithm 4 stops and $H \in RED(C)$. We have table $ATREDS = (U, \{o, h, w\} \cup \{d\}, V, f)$.

Example 9. Combining algorithm 1 and 2 on examples 2 and 6, we obtain the consistent decision table which are reduced in both vertical and horizontal dimensions. In addition to attribute reduct that is obtained by algorithm 4 on example 8, the relation $r_1 = \{5, 6, 8, 9, 12, 14\}$ over $\{ohw\}$ is a consistent decision table as table 2 for learning process.

Table 2: Table r_1 is combination of NOREDS, OBREDS and ATREDS

No.	Outlook	Humidity	Windy	d
5	Rain	Middle	Weak	Yes
6	Rain	Middle	Strong	No
8	Sunny	High	Weak	No
9	Sunny	Middle	Weak	Yes
12	Overcast	High	Strong	Yes
14	Rain	High	Strong	No

A decision tree that is generated from the consistent decision table r_1 (table 2) as Fig 1. The decision tree (Fig 1) is also one of the decision trees that are generated from the consistent decision table 1 by algorithm ID3 (or C4.5).

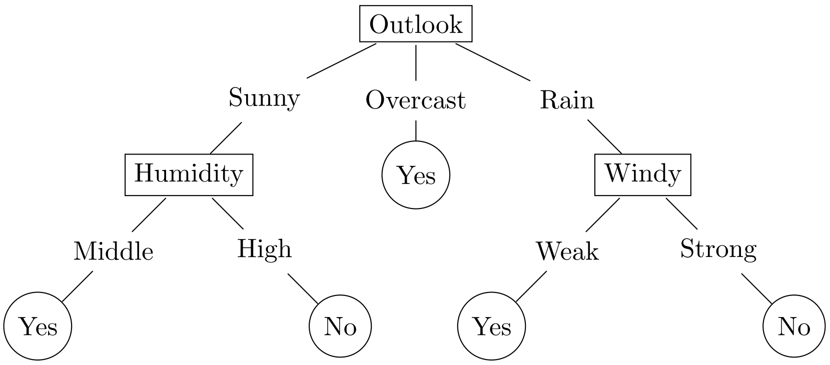


Figure 1: The decision tree generated from combination reducts table 2

5 Conclusion

In this paper, we have proposed some novel methods to reduce the consistent decision tables in both horizontal and vertical dimensions. Our ideas are based on some results from relational database theory and rough set theory. The algorithm of finding all reduct attributes and the algorithm of finding an object reduct run in polynomial time complexity. The algorithm of finding attribute reducts may be either polynomial time complexity in the case of finding only one attribute reduct or exponential time complexity [5] in the case of finding all attribute reducts of consistent decision table. The learning decision trees [15] that are generated from the reduced decision table are obtained from those generated from the original decision table. Thus, our methods can help to facilitate the learning process from larger decision tables compared with existing methods.

References

[1] Cornejo, Ma Eugenia, Medina, Jesús, and Ramírez-Poussa, Eloisa. Attribute reduction in multi-adjoint concept lattices. *Information Sciences*, 294:41–56, 2015.

[2] Demetrovics, János and Thi, Vu Duc. Keys, antikeys and prime attributes. In *Annales Univ. Sci. Budapest, Sect. Comp*, volume 8, pages 35–52, 1987.

[3] Demetrovics, János and Thi, Vu Duc. Algorithms for generating an armstrong relation and inferring functional dependencies in the relational datamodel. *Computers & Mathematics with Applications*, 26(4):43–55, 1993.

[4] Hu, Qinghua, Xie, Zongxia, and Yu, Daren. Hybrid attribute reduction based on a novel fuzzy-rough model and information granulation. *Pattern recognition*, 40(12):3509–3521, 2007.

[5] Janos, Demetrovics, Thi, Vu Duc, and Giang, Nguyen Long. On finding all reducts of consistent decision tables. *Cybernetics and Information Technologies*, 14(4):3–10, 2014.

[6] Mi, Ju-Sheng, Wu, Wei-Zhi, and Zhang, Wen-Xiu. Approaches to knowledge reduction based on variable precision rough set model. *Information sciences*, 159(3):255–272, 2004.

[7] Min, Fan, He, Huaping, Qian, Yuhua, and Zhu, William. Test-cost-sensitive attribute reduction. *Information Sciences*, 181(22):4928–4942, 2011.

[8] Pawlak, Zdzisław. Rough sets. *International Journal of Computer & Information Sciences*, 11(5):341–356, 1982.

[9] Pawlak, Zdzisław and Skowron, Andrzej. Rough sets and boolean reasoning. *Information sciences*, 177(1):41–73, 2007.

[10] Qian, Yuhua and Liang, Jiye. Combination entropy and combination granulation in rough set theory. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 16(02):179–193, 2008.

[11] Skowron, Andrzej and Rauszer, Cecylia. The discernibility matrices and functions in information systems. In *Intelligent Decision Support*, pages 331–362. Springer, 1992.

[12] Thi, Vu Duc. The minimal keys and antikeys. *Acta Cybernetica*, 7(4):361–371, 1986.

[13] Thi, Vu Duc and Giang, Nguyen Long. A method to construct decision table from relation scheme. *Cybernetics and Information Technologies*, 11(3):32–41, 2011.

[14] Thi, Vu Duc and Giang, Nguyen Long. Some problems concerning condition attributes and reducts in decision tables. In *Proceeding of the fifth National Symposium Fundamental and Applied Information Technology Research*, pages 142—152. FAIR, Dong Nai, Vietnam, 2012.

[15] Vens, Celine, Struyf, Jan, Schietgat, Leander, Džeroski, Sašo, and Blockeel, Hendrik. Decision trees for hierarchical multi-label classification. *Machine Learning*, 73(2):185–214, 2008.

[16] Yao, Yiyu and Zhao, Yan. Discernibility matrix simplification for constructing attribute reducts. *Information sciences*, 179(7):867–882, 2009.

[17] Zheng, Kai, Hu, Jie, Zhan, Zhenfei, Ma, Jin, and Qi, Jin. An enhancement for heuristic attribute reduction algorithm in rough set. *Expert Systems with Applications*, 41(15):6748–6754, 2014.

Acta Cybernetica **23** (2018) 1055–1069.

An Approximative and Semi-automated Method to Create MPEG-4 Compliant Human Face Models[*]

Ákos Tóth[a] and Roland Kunkli[a]

Abstract

In this paper, we introduce our method to facilitate the process of creating an MPEG-4 compliant face model based on a simple 3D mesh. The presented method is semi-automatic, and the user needs to choose several points on the model as a preprocessing step. We use a cage based deformation technique to approximate an input model with a generic one which already contains the required MPEG-4 parameters. In the paper, we also show how the cage can be constructed to surround the model completely and be close enough to the head to get better deformation results. After these steps, the resulting model can be used in any MPEG-4 based facial animation player.

Keywords: MPEG-4, talking head, facial animation, automation, cage based deformation

1 Introduction

The usage of virtual avatars—also called talking heads in the case of eliminating the body part—is widespread in HCI (Human-Computer Interaction), and they may play an essential role in the future as well [9].

Methods for creating a talking head based on the face of a real or a fictive person typically consist of two main steps. The first one is the creation of the model itself, while the second step is to prepare our model for further facial animations. There are several different approaches for this preparation from complex manual parameterizations [18, 26] all the way up to automatic motion capture based techniques [4].

Some of the main challenges are the portability and the reusability of the model prepared by the previously mentioned methods. Their principles can be very different; therefore, in most cases, the conversion between the outputs is nearly impossible. For solving this problem, many earlier systems [2, 7] support the well-known

[*] This work was supported by the construction EFOP-3.6.3-VEKOP-16-2017-00002. The project was supported by the European Union, co-financed by the European Social Fund.

[a] University of Debrecen, Faculty of Informatics, E-mail: {toth.akos, kunkli.roland}@inf.unideb.hu

DOI: 10.14232/actacyb.23.4.2018.5

MPEG-4 facial and body animation standard [16]. Using MPEG-4, we can guarantee that the desired animation can be attached to any standard model automatically, but unfortunately, the creation of an MPEG-4 compliant face model requires a lot of manual user interactions. With all this in mind, our goal in this paper is to give a possible solution to this problem by reducing the amount of required interaction in a semi-automatic way. In the next section, we give a short overview related to MPEG-4 facial animation and its usage in science. Then, in Section 3, we discuss some previous works related to the area of MPEG-4 facial animation. We highlight their main advantages and disadvantages as well. In Section 4, we present our method, which can provide a solution to some of these mentioned disadvantages, based on a prepared generic model and a cage based mesh deformation technique. We demonstrate the results of our algorithm in Section 5. Then, in the last section, we discuss the conclusions and our future ideas as well.

2 Facial animation in MPEG-4

In March of 1999, the Moving Picture Experts Group announced MPEG-4 as an ISO standard also for facial animation. Nowadays, it is the only widely accepted standard for this kind of application; furthermore, the industry also pays close attention to it [23]. The standard defines the parameters, which control the animation of a human face, in detail. All possible natural facial expressions are available using these parameters.

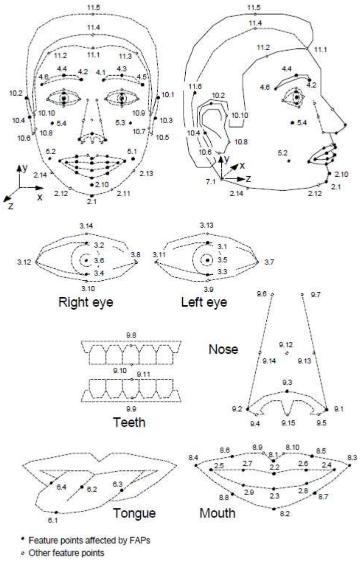


Figure 1: The MPEG-4 Feature Points [16].

MPEG-4 defines 84 Feature Points (FPs) on a neutral face as shown in Figure 1.

Besides that, the zone of the influence of each FP must be set for a realistic and believable animation. These FPs are described in <*group*>.<*index*> format, where *group* stands for the feature type like eyes, cheek, or mouth, and *index* is just a label. The FPs are employed to supervise the animation, but they have to be controlled at the same time in order to reach the desired facial expression properly. The 68 FAPs (Facial Animation Parameters) are used to manipulate these FPs on the face to produce natural facial expressions. FAPs are universal parameters, but we have to calibrate them before their usage. This calibration is feasible by using the so-called Face Animation Parameter Units (FAPU). FAPUs provide the interpretation of FAPs on any facial models. Each unit corresponds to the fraction of distances between fundamental facial features (e.g., the distance between the mouth and the nose). So a 3D model together with its FPs and FAPUs is perfectly enough to be able to animate the model in any MPEG-4 compatible facial animation player, e.g., [2, 7, 20, 19]. Owing to the previously mentioned advantages of the standard, researchers like to use MPEG-4 facial animations in several different projects, e.g., for supporting psychosocial treatments [10] or for speech synthesis frameworks [11].

3 Previous work

Nowadays, there are standards and descriptions, which can help in the parameterization based facial animation process, like FACS [17] (Facial Action Coding System) or the MPEG-4 facial and body animation standard. The latter one is often used because it provides a foundation not only for facial animation but also for speech animation by defining control points in the mouth region as well. Therefore, the many parameters on the lips create exact and granular shapes and also provide a useful toolkit for speech articulation systems. Also, owing to the parameters, we can reuse previous facial expressions or animation sequences. There are numerous projects [2, 6, 7, 20] with which we can play an MPEG-4 based facial animation or create the parameterization of the face model.

The face model adaptation (parameterization) heavily depends on the topology of the considered mesh. In the case of simple models, which consist of a few hundred vertices, it can be executed easily manually. However, if the models are highly detailed, relatively complex, and include thousands of vertices, then the calibration is not practical. Unfortunately, the adaptation of an existing parameterization to a new facial model is not possible. To solve some of these problems (e.g., the parameterization of the models), Sheng et al. have been released a PDE (Partial Differential Equations) based method [21], but unfortunately it does not support the MPEG-4 standard.

MPEG-4 compliant deformation-based methods have been also published. Escher et al. have been proposed a solution [8] which can create standard faces by using a generic model. If all the feature points of the calibration model are available, then the method fits the generic model with Dirichlet Free Form Deformation. In other cases, a pre-processing step is required to compute the missing feature points using a cylindrical projection and a Delaunay triangulation. Lavagetto et al. introduced a method [13] which can help in the calibration process. Based on the feature points, they used an RBF (Radial Basis Function) to reshape the geometry of the neutral face. However, due to the limitations of the used deformation algorithm (i.e., RBF and Free Form Deformation [22]), the techniques cannot provide a satisfactory deformation of the generic model, especially in the case of the nose, the eyes, and the ears (see Figure 2).

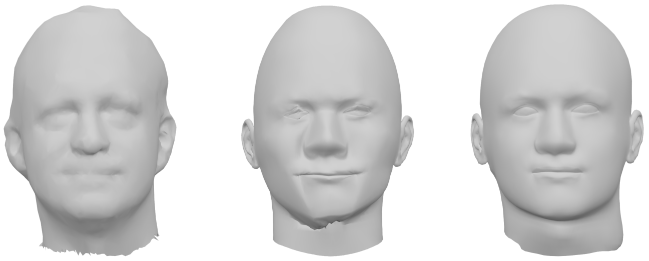


Figure 2: Input model (left), the results of the deformation using Surface-Oriented Free-Form Deformation (middle) and cage based deformation technique (right) which is the basis of our approach.

As we discussed in Section 1, our primary goals are to ease the adaptation and avoid possible errors caused by the users. We suggest a semi-automatic method based on a generic model which should be deformed in order to approximate the input one. MPEG-4 parameters are already specified on the generic model. Thus, the face model calibration can be skipped. For the manipulation of the generic model, we used a deformation method called cage based deformation technique (see Figure 3). One of the main advantages of these methods against other deformation solutions is that using them we can work in real-time, and we have an easy to use, smooth, and intuitive control over the mesh with the defined cage [15].

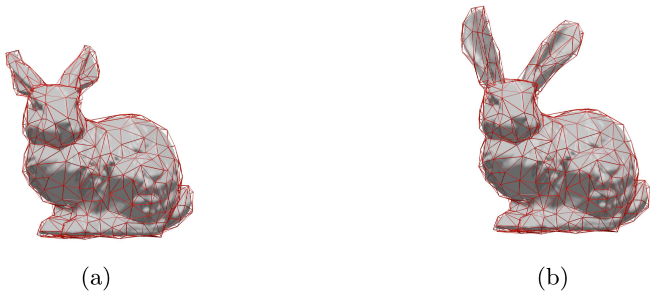


Figure 3: (a) An original mesh and its surrounding cage. (b) We can move the vertices of the cage, thereby generating a smooth deformation of the model.

4 Our method for automating the face model calibration

As we mentioned in Section 1, the creation of an MPEG-4 compliant face model requests a lot of time if we do it manually because we need to define all of the 84 feature points and the zone of influence for each FP. Thus, we suggest a semi-automatic solution to create a standard face if its simple 3D mesh is given. At first, we consider a specified generic model together with all its predefined standard points, and then a cage based method is used to approximate the input model with the generic one. Therefore, the whole procedure of the face model adaptation does not need to be executed.

4.1 Generic model

In the area of facial animation and human modeling, the generic model has to satisfy some requirements. In order to get a realistic talking head animation, the generic mesh must be fine triangulated in high-curvature regions, while low-curvature areas are allowed to contain larger triangles. Naturally, if the goal is a cartoon talking head, the usage of a less detailed generic mesh is more practical. Besides that, the lower and upper lips must be separated from each other so that the speech can be animated as well. We integrated the generic model (see Figure 4) into our system from an open-source application called Xface [2]. We calibrated it to be MPEG-4 compliant, and we modified the geometry of it to get a more neutral face, and we removed the hair, the tongue, and the teeth in order to handle it easily for future deformations. Thus, using this generic model and its FDP (Face Definition Parameters) file we can execute any deformations, and the resulting model will be usable in any MPEG-4 based facial animation player.

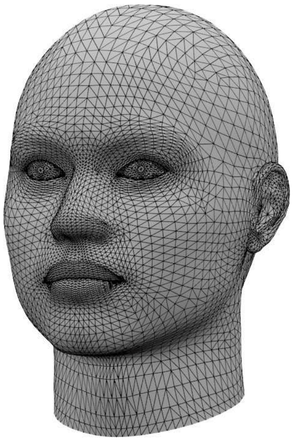


Figure 4: Our generic model that contains 5570 vertices and 11032 triangles.

4.2 Input model

The input head model can be any human-like head model (e.g., realistic models from 3D reconstructions or animation characters). Compared with the generic model, input models are allowed to have lower resolution or defects as well.

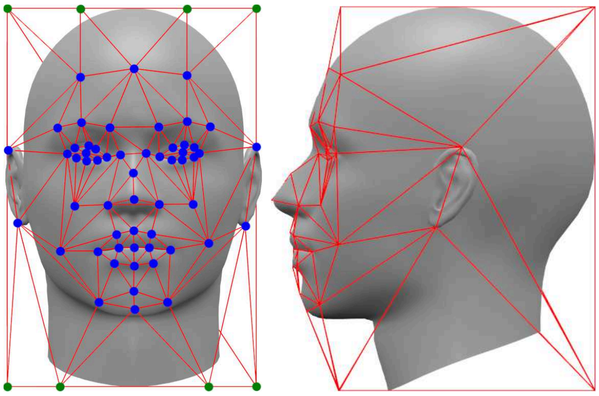


Figure 5: The control cage which has been created for the generic model. Blue dots are the facial feature points of the model, which need to be marked manually, while the green ones are the auxiliary points. These latter points (top four and bottom four on the left image) are calculated from the bounding box of the model automatically.

4.3 Our deformation method

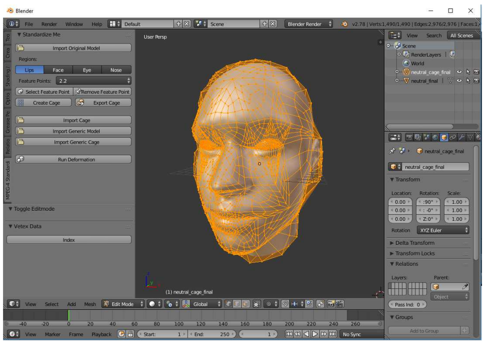
The necessary modifications of the generic model are achieved by a cage based deformation technique called harmonic coordinates [12]. As we mentioned earlier, a topologically flexible cage (or also known as control mesh) is used to control the deformation of the interior object. If we move the cage vertices C_i to the new positions C_i', an interior point p moves to the new location p', and it is computed as

$$p' = \sum_i h_i(p) C_i',$$

where $h_i(p)$ is the harmonic coordinate of p respect to C_i. Therefore, at first, we define simple control cages (shown in Figure 5) surrounding the generic and the input models in the same way, based on pre-marked facial feature points on the heads. These cages are responsible for the deformation. Then, we modify the cage of the generic model by translating each point to the corresponding position on the input model's cage separately. As a result, the generic model sufficiently approximates the input one, and it still remains MPEG-4 compliant.

4.4 Generating the control cages

In order to generate the cages of the models efficiently, we have created a Blender [3] add-on called *Standardize Me* [25] (see Figure 6).


v2.78 | Verts:1,490/1,490 | Edges:2,976/2,976 | Faces:1,

Feature Points: 2.2

Location: 0.00 0.00 0.00

Rotation: :90° : -0° Z:0°

Scale: 1.00 1.00 1.00

Rotation XYZ Euler

Pass Ind: 0

Start: 1 End: 250 1

Figure 6: The user interface of our *Standardize Me* Blender add-on.

With our Blender add-on, the users can load a 3D head model and mark the vertices of it; these are required for the generation of its control cage. The vertices to be marked are derived from the feature points of the MPEG-4 standard because the characteristic of the human head can be defined properly with these points. Using the add-on, users have to mark 56 positions on the model to generate its control cage automatically.

Because of the harmonic coordinates method requires that all points of the object to be deformed must lie within the cage, we use 12 additional auxiliary points—beyond the user-defined points—for the cage construction as well, which are automatically found by the add-on. The position of these points is based on the minimum and maximum coordinates of the model. We assume that the line of the centers of the eyes is orthogonal to the plane of y and z axes, and the origin is located at the center of mass of the model.

After the above-mentioned points have been defined, we have to apply a triangulation on them to construct the cage itself. The used one (see in Figure 5) is uniform for any head models. Thus, we can define a one-to-one correspondence between the points of the generic cage and the input one. Then, in the last step of the cage generation, we scale the cage to avoid intersections with the input model itself.

4.5 Improvement in the usage of harmonic coordinates

In our view, the method of harmonic coordinates has two important properties. First, the quality of the deformation depends on the number of control cage vertices. So, if we increase the number of these vertices, we can get smoother results, and this is essential in areas where facial animation plays a central role, such as virtual reality, computer games, or the animation film industry. Therefore, we apply a subdivision technique—which can handle a triangulated mesh—on both cages to improve the influence of harmonic coordinates. The used scheme is the following: In each iteration step, an original face of the model is divided into new triangle faces by connecting the midpoints of its edges. It is important to note that in the used subdivision method, the positions of the original cage vertices do not change.

The other good property is that the reduction of the distance between the control mesh and the 3D model results a much more efficient deformation. To this end, we reduce this distance using the following technique. We define rays from the origin (which is at the model's center of mass) through the cage vertices. Then, in the case of the generic model, we translate all cage vertices with equal length vectors on its ray to the maximum level, while keeping the whole model within the cage.

In the case of an input model, the positions of its cage vertices will be the intersection points of the corresponding rays and the model. It follows from this construction that the resulting cage will intersect the input model, but this is not a problem for us because we do not need to apply the method of harmonic coordinates on the input model, unlike the generic one.

The above-mentioned distance reduction technique may give undesirable results around the ears of the models (especially, when the ears are highly detailed). The reason is that the intersections are not located evenly, so the structure of the cage may be changed. Thus, the new positions of the cage vertices, which are close to the ears, will be the minimum or maximum x coordinates of the model, so the smooth deformation of the model's ears can be achievable.

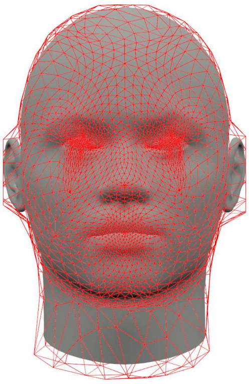


Figure 7: The new control cage of the generic model after applying two iterations of subdivision and the distance reduction technique.

After these modifications, we are able to get a cage which conforms the small changes in the model's geometry as well, and it also fulfils the required conditions. The new cages (see Figure 7) contain approximately 1500 vertices and 3000 triangles, which are achieved by two iterations of subdivision, while the harmonic coordinates and the deformation itself still can be computed in real-time.

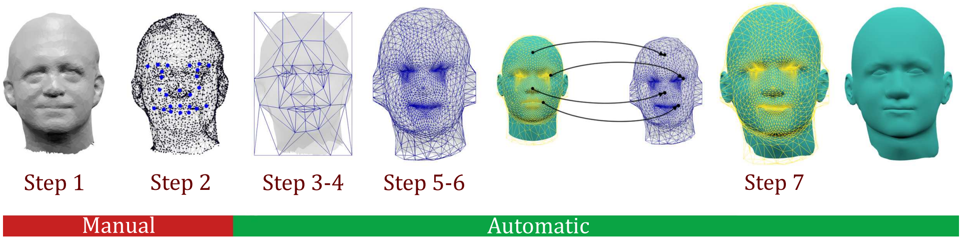


Figure 8: Our semi-automatic MPEG-4 calibration workflow. Each subfigure shows the result of the actual step. Only the first two steps are manual, the user has to define an input model, then mark the necessary points on the model. The further steps are automatic; the algorithm generates the proper control cage for the input model in steps 3–6. Then it translates the vertices of the generic model's control cage to the corresponding ones on the input model's control cage. Because of the applied harmonic coordinates on the generic model, the algorithm deforms the generic model (step 7) which approximates the input one.

4.6 The step by step process of our semi-automatic calibration

We can summarize our method in the following steps. Only the first two steps require user interactions; the remaining ones are automatically executed. Before running the algorithm, we must have a generic model with its control cage and FDP file.

1. Creating a 3D head model of a person or a fictive character, whose talking head we want to create.

2. Marking all necessary points on the 3D head model.

3. Generating the control cage for the input model from the marked and the auxiliary points by using a triangulation.

4. Applying a scaling transformation to the generated control cage in order to avoid possible intersections.

5. Applying two iterations of subdivision on the control cage.

6. Minimizing the distance between the control cage and the 3D head model using the previously mentioned distance reduction technique.

7. Translating the vertices of the generic model's control cage to the corresponding ones on the input model's control cage.

The above-mentioned steps can be seen in Figure 8.

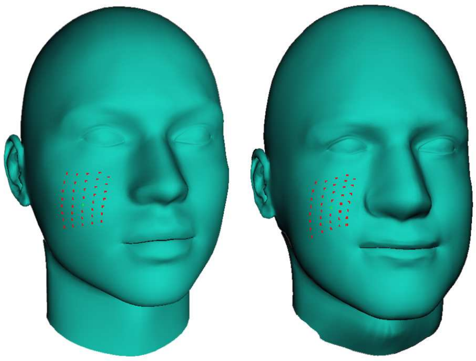


Figure 9: The generic and the deformed model. The green dot marks the 5.4 MPEG-4 FP, while the red ones mark the zone of influence of it.

As we mentioned above, at the end of the process, the cage of the generic model will be transformed into the cage of the input one. The deformation of the generic model depends on the positions of the pre-marked facial features on the input model. The size of the input and the deformed generic model may be a bit different, but we can eliminate this problem by using the dimensions of the bounding box of the original input model. Therefore, we receive a deformed generic model which is similar to the person's face; and all MPEG-4 standard points are already defined on it (see Figure 9).

5 Results

5.1 Implementation

We created the necessary input models in three different ways. We used Microsoft's *Kinect* sensor, an open-source 3D modeling software called *MakeHuman* [24], and a photo based reconstruction application called Autodesk® 123D Catch® [1]. We tried our method on twenty models which are originated from the previously mentioned sources. Considering these systems, *MakeHuman*'s models gave the best results, because they were detailed enough to mark the expected positions of facial features easily, without any defects. In *MakeHuman*, there is a race-related setting, which let us test our solution thoroughly. In the case of the other two, reconstruction based methods, the resolution of the models became a serious limitation. Therefore, we employed a subdivision technique on the meshes before the generation of their cages.

As we mentioned above, our solution is implemented in our self-developed *Blender* add-on, called *Standardize Me*, which is a Python script. The script can be installed in *Blender* as an add-on, and it can be used for executing the whole process of creating an MPEG-4 compliant face model.

5.2 Validation

The comparison of the models (the input model and the resulting one) played a serious role in the improvement of our algorithm. For measuring and visualising the difference between the two mentioned models we used the *one-sided Hausdorff distance* in Meshlab [5] with the following formula:

$$h(\mathcal{A},\mathcal{B}) \equiv \max_{a\in\mathcal{A}}\left(\min_{b\in\mathcal{B}} d(\mathbf{a},\mathbf{b})\right),$$

where $\mathcal{A}$ is the resulting model, while $\mathcal{B}$ is the input one.

It returns both numerical and visual evaluations of meshes' likeness (see in Table 1). We also computed the *two-sided Hausdorff distance* (see in the last row of Table 1) between the two models by the following way:

$$H(\mathcal{A},\mathcal{B}) \equiv \max(h(\mathcal{A},\mathcal{B}), h(\mathcal{B},\mathcal{A})).$$

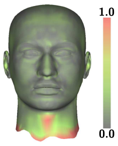
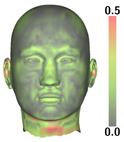
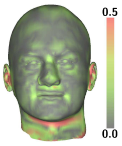
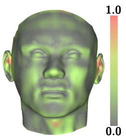
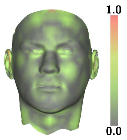
					
Min	1.1×10^{-5}	4.5×10^{-5}	0.3×10^{-5}	6.6×10^{-5}	0.4×10^{-5}
Max	3.4282	1.2055	1.5712	1.2969	2.0151
Mean	0.1650	0.1163	0.1681	0.2285	0.2033
Avg.	0.3783	0.2621	0.3083	0.2894	0.2989
Std. dev.	0.3175	0.1283	0.1762	0.1710	0.1959
Hausdorff	3.4282	1.6747	3.4446	1.9193	3.6731

Table 1: Distance computations between our results and the approximated input head models. The last row of the table shows the *two-sided Hausdorff distances*, while the values in the other rows are computed by the *one-sided Hausdorff distance*. In the case of the last two reconstructed model, we cut the back of the head model to get accurate measurements. For the sake of better comparison, we also measured the height of the models, which are 24.5233, 23.1186, 24.3174, 25.3531, and 23.5528, from left to right, respectively.

To validate our resulting models, we generated FAPs files using 3D reconstructed human facial expressions from the BU-3DFE (Binghamton University 3D Facial Expression) Database [27]. Then, these FAPs files were played on our resulting face models to get the same expressions. The original reconstructed model and our deformed model with different facial expressions can be seen in Figure 10.

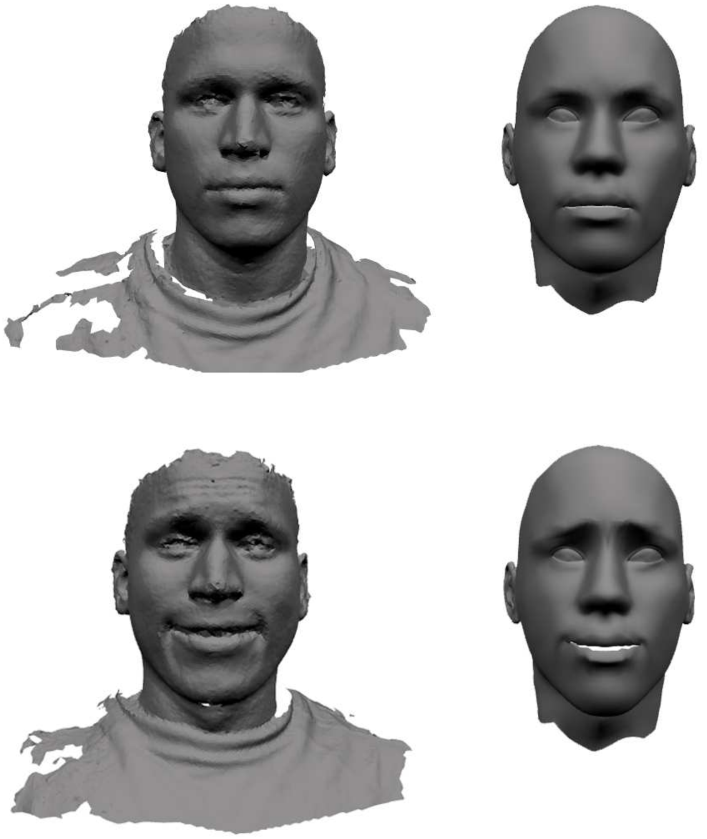


Figure 10: Left: 3D reconstructed human facial expressions from [27]. Top right: Our resulting model with neutral expression. Top left: Our resulting model using the generated FAPs file.

6 Conclusions and future works

In this paper, we have proposed a robust and semi-automatic method for creating an MPEG-4 compliant face model. Only the definition of the feature points needs user interaction, but it takes just a few minutes. In the case of a high-resolution face model (≈3500 vertices, ≈7000 faces), the process takes 8 minutes approximately. After applying our method, the resulting talking heads can be used in any MPEG-4 compatible facial animation player immediately.

Our solution uses a cage based deformation method called harmonic coordinates to approximate the input model with the generic one. The main difference from earlier systems is that we do not have to define the MPEG-4 parameters. Therefore, calibration errors do not influence the quality of the animation of the model. Based

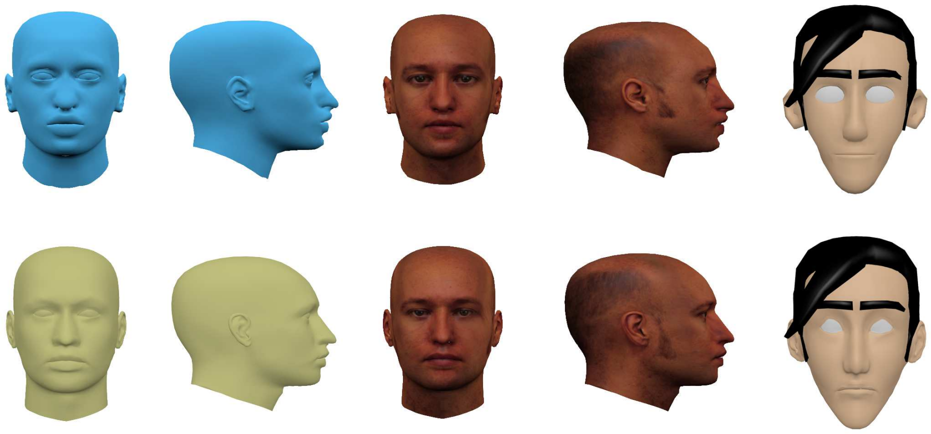


Figure 11: Top: Input models. Bottom: The resulting models after the calibration. The used textures are from [14]. In the last case, the hair is a separate model that has been attached to the result by hand.

on our measurements (see Section 5.2), we can say, that the resulting face models (shown in Figure 11) are similar to the original input, but there may be small differences in some parts of the faces, especially around the eyes, the ears, and the necks. We think that these errors stem from the applied distance reduction technique. Therefore, we would like to try different techniques for minimizing the distance between the cage and the model. A self-organizing neural network may be able to solve this problem. Additionally, we would like to attempt to minimize the number of user interactions by using a face tracking method which can mark the necessary facial feature positions for us. We assume that our method can work with different parameterization (e.g., FACS), but we used the MPEG-4 standard in this paper to create an operating prototype.

References

[1] Autodesk, Inc. Autodesk® 123D Catch®. `http://www.123dapp.com/catch/`. Accessed: 28 May 2015.

[2] Balci, K. Xface: MPEG-4 Based Open Source Toolkit for 3D Facial Animation. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 399–402, 2004.

[3] Blender Online Community. Blender - a 3D modelling and rendering package. `http://www.blender.org/`. Accessed: 7 March 2015.

[4] Cao, C., Weng, Y., Lin, S., and Zhou, K. 3D Shape Regression for Real-time Facial Animation. *ACM Transactions on Graphics*, 32(4):41:1–41:10, 2013.

[5] CNR, Visual Computing Lab ISTI. MeshLab. `http://meshlab.sourceforge.net/`. Accessed: 1 February 2015.

[6] Cosi, P., Fusaro, A., and Tisato, G. LUCIA a New Italian Talking-Head Based on a Modified Cohen-Massaro's Labial Coarticulation Model. In *INTERSPEECH*, pages 2269–2272, 2003.

[7] de Rosis, F., Pelachaud, C., Poggi, I., Carofiglio, V., and De Carolis, B. From Greta's mind to her face: modelling the dynamics of affective states in a conversational embodied agent. *International Journal of Human Computer Studies*, 59(1–2):81–118, 2003.

[8] Escher, M., Pandzic, I., and Thalmann, N. M. Facial deformations for MPEG-4. In *Proceedings Computer Animation '98 (Cat. No.98EX169)*, pages 56–62, June 1998.

[9] Feng, A., Rosenberg, E. S., and Shapiro, A. Just-in-time, viable, 3-D avatars from scans. *Computer Animation and Virtual Worlds*, 28(3-4):e1769, 2017.

[10] Fergus, P., El Rhalibi, A., Carter, C., and Cooper, S. Towards an avatar mentor framework to support physical and psychosocial treatments. *Health and Technology*, 2(1):17–31, 2012.

[11] Jia, J., Zhang, S., Meng, F., Wang, Y., and Cai, L. Emotional Audio-Visual Speech Synthesis Based on PAD. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(3):570–582, 2011.

[12] Joshi, P., Meyer, M., DeRose, T., Green, B., and Sanocki, T. Harmonic Coordinates for Character Articulation. *ACM Transactions on Graphics*, 26(3):71, 2007.

[13] Lavagetto, F. and Pockaj, R. The facial animation engine: toward a high-level interface for the design of MPEG-4 compliant animated faces. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(2):277–289, 1999.

[14] Lundqvist, D., Flykt, A., and Öhman, A. The Karolinska Directed Emotional Faces (KDEF). CD ROM from Department of Clinical Neuroscience. Psychology section, Karolinska Institutet, ISBN 91-630-7164-9. 1998.

[15] Nieto, J. R. and Susín, A. Cage based deformations: A survey. In *Deformation Models: Tracking, Animation and Applications*, pages 75–99. Springer Netherlands, Dordrecht, 2013.

[16] Pandzic, I. S. and Forchheimer, R., editors. *MPEG-4 Facial Animation: The Standard, Implementation and Applications.* John Wiley & Sons, Inc., New York, NY, USA, 2003.

[17] Parke, F. I. and Waters, K. *Computer Facial Animation.* CRC Press, 2008.

[18] Pyun, H., Kim, Y., Chae, W., Kang, H. W., and Shin, S. Y. An Example-based Approach for Facial Expression Cloning. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 167–176, 2003.

[19] Rácz, R., Tóth, Á., Papp, I., and Kunkli, R. Full-body animations and new faces for a WebGL based MPEG-4 avatar. In *CogInfoCom 2015: 6th IEEE International Conference on Cognitive Infocommunications*, pages 419–420, 2015.

[20] Rhalibi, A. E., Carter, C., Cooper, S., Merabti, M., and Price, M. Charisma: High-performance Web-based MPEG-compliant Animation Framework. *Computers in Entertainment*, 8(2):8:1–8:15, 2010.

[21] Sheng, Y., Willis, P., Gonzalez Castro, G., and Ugail, H. PDE-Based Facial Animation: Making the Complex Simple. In *Advances in Visual Computing*, pages 723–732. Springer Berlin Heidelberg, 2008.

[22] Singh, K. and Kokkevis, E. Skinning Characters using Surface Oriented Free-Form Deformations. In *Proceedings of the Graphics Interface 2000 Conference*, pages 35–42, 2000.

[23] Technologies, Visage. Visage Technologies - Face Tracking and Analysis. `http://visagetechnologies.com/`. Accessed: 15 March 2017.

[24] The MakeHuman team. MakeHuman. `http://www.makehuman.org/`. Accessed: 26 September 2014.

[25] Tóth, Á. and Kunkli, R. *Standardize Me.* `https://arato.inf.unideb.hu/kunkli.roland/software/standardize-me-toolkit/` and `https://github.com/dragostej/standardizeme`.
Accessed: 2 October 2017.

[26] Weise, T., Li, H., Van Gool, L., and Pauly, M. Face / Off : Live Facial Puppetry. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 7–16, 2009.

[27] Yin, L., Wei, X., Sun, Y., Wang, J., and Rosato, M. J. A 3D Facial Expression Database for Facial Behavior Research. In *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pages 211–216, 2006.

Acta Cybernetica **23** (2018) 1071–1087.

Gaze-based Cursor Control Impairs Performance in Divided Attention*

Róbert Adrian Rill[ab] and Kinga Bettina Faragó[a]

Abstract

In this work we investigate the effects of switching from mouse cursor control to gaze-based control in a computerized divided attention game. We conducted experiments with nine participants performing a task that requires continuous focused concentration and frequent shifts of attention. Despite carefully controlling experimental and design aspects, the performance of subjects was considerably impaired when using gaze-based control. The participants were experienced users of the mouse control version of the task, we adjusted the difficulty to the more demanding conditions and selected the parameters of gaze input based on previous research findings. In contrast to our assumptions, experienced users could not get used to gaze-based control in the amount of experiments we performed. Additionally we consider the strategies of users, i.e. their method of problem solving, and found that it is possible to make progress in our task even during a short amount of practice. The results of this study provide evidence that the adoption of interfaces controlled by human eye-gaze in cognitively demanding environments require careful design, proper testing and sufficient user training.

Keywords: gaze-based control, eye tracking, divided attention, human performance, cognitive load, Midas Touch, dwell time

1 Introduction

The seminal work of Yarbus [30] had an enormous impact on consequent researches on eye movements. He showed that it is possible to infer the task an observer is performing from their fixation patterns. Not only do eye movements indicate the search target during visual exploration, but they also reveal emotions, intentions

*This research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013).

[a]ELTE Eötvös Loránd University, Budapest, Hungary. Faculty of Informatics, 3in Research Group, Martonvásár, Hungary.
E-mail: `rillroberto88@yahoo.com (ORCID iD: https://orcid.org/0000-0002-3004-7294)`, `kinga.farago@gmail.com (ORCID iD: https://orcid.org/0000-0002-2402-7123)`

[b]Faculty of Mathematics and Computer Science, Babeş-Bolyai University, Cluj-Napoca, Romania.

DOI: 10.14232/actacyb.23.4.2018.6

and cognitive processes [2]. Eye contact and gaze direction represent a powerful means of communication in regulating interaction and establishing socio-emotional connection [17].

In addition to having a long history in studying human visual behavior in medical and psychological research, eye tracking has the potential to revolutionize interface design in the field of human-computer interaction. Due to the technological advancements, non-intrusive and accurate hardware solutions are readily available (see, e.g., [8, 17]). Moreover, explosion in computer vision in recent years lead to the development of state-of-the-art appearance-based gaze estimation methods that use only images and videos from off-the-shelf monocular RGB cameras, and have an acceptable range of errors in prediction (see, e.g. [22] and the references therein).

Eye movements can be divided into several different categories [5, 12, 16, 17]. Saccades are the most common way of moving the eyes in a sudden, ballistic way of 2 degrees or larger taking about 30-120 ms each. They are typically followed by fixations of at least 100 ms, generally between 200-600 ms periods of relative stability. However, during fixations the eyes still make small, jittery motions, covering usually less than one degree. Blinks of up to 200 ms may occur during a fixation without terminating it. Smooth pursuit movements are less sudden than saccades and occur only in response to a moving target in the field of view. There are also other eye movements which, however, are not significant in human-computer interaction. For more information about eye movements, see [17] and the references therein.

Modern technological solutions facilitate the design of complex human-computer interaction interfaces. These often require the divided attention of users. Compared to the slow and deliberate way of operating a mouse or other input device, eye movements usually scan the screen involuntarily, for example the user is not aware of the jittery motions during a fixation. Moreover, eyes are used primarily for perception [18] and they typically precede actions [1, 14, 17, 27]. Thus, the additional use for control requires careful design of human-computer interaction interfaces [3, 13, 20] in order to provide adequate feedback and to avoid false activation.

Controlling computers by gaze could also improve performance in multitasking situations by taking advantage of the nature of eye movements, reducing this way cognitive load or increasing safety. For instance, gaze-based control of secondary displays in automotive or aviation environments can reduce pointing and selection times while also performing the primary task of driving and flying, respectively [23].

1.1 Related work

Despite the problems of using the same modality for both perception and control, gaze estimation and eye tracking have important application areas ranging from medical diagnosis and psychological research to the design of interfaces and usability studies of gaze-controlled applications from the field of human-computer interaction [17]. Probably the most common example is eye-typing [15, 16, 18, 19, 25, 28]. Other applications include object selection on interfaces [12] and in real world [26], target search and selection [28], computer game control (ranging from simple puzzles [1] and classical games [6] to role-playing and first-person shooter video games [11, 27]), facilitating attention switching between multiple visual displays [13], robotic device control [7], web-browsing [5, 14], interacting with geographic information systems [3], developing interactive graphical user interface elements [20], projected display control in automotive and military aviation environments [23].

Using gaze as an input method may not be comparable to the universal mouse and keyboard [14] because of the nature of the human eye movements and the physiology of the eye [17], yet it can still have several advantages. For people with physical disabilities gaze-based interaction provides a means to communicate and interact with technology and other people [5, 6, 17, 18, 20]. The performance of children with disabilities can be enhanced considerably through gaze controlled computers [10]. In the case of older adults, it may be able to compensate for the declined motor functions when using mouse input [21].

Another relevant factor to consider is the engagement of users. Eye movements are extremely fast and require little effort [1, 3, 7, 14, 25, 27]. Thus, gaze response may be well suited for visual search tasks and novice users may find captivating the natural quality of selection by looking [28]. Gaze can represent a superior input modality in simple computer games in terms of achievements, engagement or gameplay experience [1, 27], even for users without any previous training [6]. Furthermore, gaze-based interaction facilitates attention switching when using multiple displays [13], and eye movements represent a means of communication in collaborative virtual environments too [6, 27].

In contrast, controlling by gaze also faces considerable challenges [17]. Probably the most common issue is the *Midas Touch* problem [1, 3, 4, 5, 7, 12, 14, 17, 18, 20, 23, 26, 27], i.e. activating a command involuntarily only by looking at a specific target. Eye movements are largely automatic and unconscious. Normal visual perception requires that the eyes of the user scan the scene, gather information about the environment before making an action [1, 14, 17, 27]. Ideally the system should distinguish casual viewing from intentional control.

When using gaze as input, there is no natural counterpart of a mouse click [6, 12, 20]. For systems using solely gaze-based control for selection the most obvious and common alternative is *dwell time* [1, 4, 5, 11, 12, 14, 15, 16, 17, 18, 19, 28], i.e. gazing at a specific object for a sufficiently long time to trigger an activation command. However, this also raises further questions. Dwell time must be short enough to be comfortable for the subject, but this brings up the Midas Touch problem. On the other hand, a long dwell time might ensure that unintentional selections are not made, but it limits exploration time, diminishes the advantage of fast and natural eye movements and reduces the responsiveness of an interface.

Other alternatives for dwell-based selections to avoid unintentional commands are gaze-gestures [3, 11, 17, 20, 26], predefined gaze patterns as pre-programmed strategies for control [7] or combining gaze-based search and pointing with other modalities such as additional hardware button (cf. [14]) or feet [3].

Investigating feedback modalities is also in the focus of researchers. When using dwell-time, the user only initiates the action and it is the responsibility of the system to provide a clear indication of the status [17]. Majaranta et al. [19] compared auditory and visual feedback during eye-typing. Their results show that the feedback method influences both text entry speed and error rate, and suggest that a simple auditory confirmation of selection is more effective than visual feedback alone. In their follow-up works the authors also compare long versus short [16] as well as adjustable [15] dwell time duration. In a more recent work Majaranta et al. [18] also found haptic feedback to produce results close to those of auditory feedback. Other feedback methods to consider are using animation to indicate the progression of dwell time [15, 18], zooming into the area of focus [13].

The state-of-the-art, low-cost and easily accessible eye tracking technology makes gaze input a useful, fast and convenient way of communication. Although a considerable amount of work has already been done on interaction techniques, there is still no general procedure on how to incorporate eye-movements into human-computer interaction interfaces in a natural and unobtrusive way. Furthermore, little research has been conducted to examine the effects of different input modalities on users' problem solving strategies. Present-day computer interfaces require users to employ a range of complex strategies, including planning, goal searching, handling interruptions and information coordination. Bednarik et al. [1] compared dwell time, gaze-augmented interaction and mouse input using a simple puzzle game and found that the interaction methods affected performance, problem solving strategies and user experience. Dorr et al. [6] showed that gaze is superior to mouse input in a classic computer game and found that expert and novice players differ in their employed eye movement strategies.

Investigating people's problem solving strategies, using gaze as input and comparing it to more classical response methods also represents a powerful tool in psychological research [7, 21, 25, 27, 28] (e.g., in visual search tasks), because it may reveal new aspects of cognitive processes and may have implications on the design process of interfaces employing gaze tracking. Therefore we investigate gaze-based control in a special dynamic divided attention task. Particularly, we designed and implemented a simplified version of the popular Train of Thought game from the Lumosity[1] online platform. Lumosity is comprised of a set of computerized games designed by scientists, each aiming to train one of five core cognitive abilities: attention, processing speed, memory, flexibility and problem solving [9]. In the following we will refer to our version of Lumosity's Train of Thought game as the *Divided Attention (DA) Game*.

The contributions of this paper consist in investigating the effects of switching from the traditional mouse to gaze-based input in a divided attention task not examined by previous works to the best of our knowledge. Despite carefully considering several experimental and design aspects, the performance of participants was considerably impaired by gaze-based control in this cognitively demanding task requiring the divided attention of players. The subjects in our experiments were

[1] `http://www.lumosity.com/`

experienced in the conventional mouse control version of the game, we adjusted the difficulty of the task to the more demanding conditions and chose dwell-time and other parameters based on previous research findings.

The paper is organized as follows. In Section 2 we describe briefly the DA Game, detail our design choices, present the experiments with gaze-based control and define the performance measures. Section 3 presents the results of the experimental and statistical analysis. This is followed by a discussion in Section 4, which highlights future directions as well. Finally, Section 5 concludes the paper.

2 Methods

2.1 Design of task

In our previous work [24] we conducted a longitudinal study with mouse control and presented in detail the design process of the DA Game used in our experiments. Thus, here we only describe briefly the purpose of the task and present the design elements of the gaze-based control version.

The DA Game tests the divided attention and working memory of the players by requiring them to continuously focus on multiple simultaneous targets, to switch frequently between them keeping track of each one. The task of the user is to direct continuously oncoming objects to their color-matching destination through selecting and flipping switches at forks and changing this way the direction of the tracks and the path of the moving objects. For a snapshot of the game see Figure 1.

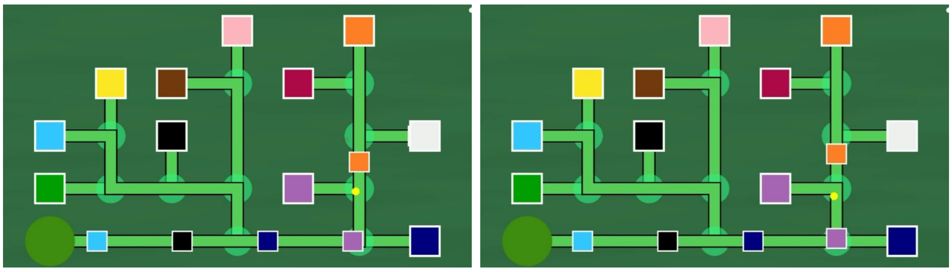


Figure 1: Snapshots of two frames from the Divided Attention Game used in our experiments: the small squares are moving continuously and have to be directed to their color-matching destinations by flipping switch nodes represented by transparent green circles. The yellow dot corresponds to the screen coordinates of the user's gaze direction. The player started to fixate on the switch next to the purple destination on the left image; the predefined dwell time of 500 ms has just elapsed and the switch was flipped as seen on the image on the right.

The traditional mouse cursor was replaced by a yellow dot displayed at the screen coordinates of the gaze direction. Although in some cases it might distract users' attention, due to the nature of eye movements, we decided to show the cursor at all times in order to provide continuous feedback for the players in a task with time constraints and to allow the possibility to compensate for potential drifts of the eye tracker during one gameplay. The noise of the device and the jittery movements of the eyes during fixations may also distract the concentration of users [17] when performing the task. It is easier to keep a steady cursor in one place until the target is selected. Accordingly, we applied a smoothing to the cursor movement using a moving average window on each of 5 consecutive samples. This does not slow down the responsiveness in the DA Game, which requires fast-paced user actions.

The most essential part of gaze-based control interfaces is generating a selection, i.e. flipping a switch node in the DA Game in our case. We chose the most common method, namely dwell time. We selected the length of the interval based on previous works enumerated next. We note that after the dwell time elapsed, we did not give any additional feedback to the user about the fact that the switch was flipped, since this is clearly noticeable as seen on Figure 1. Also, in order to be consistent with mouse-based selection, if the user continued to fixate on the target, it was selected repeatedly when the dwell time had elapsed again.

2.1.1 Dwell time duration

Jacob [12] found that a short dwell time of 150-250 ms gave excellent results, while duration over 750 ms was not useful at all in object selection tasks. In [28] the authors state that fixations longer than 500 ms are often seen during cognitive integration phases of difficult tasks. Their pilot studies indicated that 700 ms or less works well for simple tasks. They also found that a dwell time of 1000 ms makes false selections unlikely in a target selection task and that 750 ms is subjectively slow in their eye-typing task.

Majaranta et al. compared short and long dwell time duration, i.e. 450 ms vs 900 ms [16, 19]. Experienced participants achieved faster typing speed but higher overall error rates. The authors concluded that with short dwell time sharp and clear feedback is essential. In a later work, Majaranta et al. [15] also investigated adjustable dwell time in a longitudinal study and found that dwell time decreased from an average of 876 ms to 282 ms, and error rates also decreased. It is important to note that the learning rate was rapid during the first few sessions and decelerated prominently. More recently Majaranta et al. [18] found a dwell time of 860 ms in a practice session too long, and 500 ms seemed to work for them. They also mention that 500 ms might be too fast for novices in eye-typing. Expert typists may even use dwell times that correspond to their normal fixation times (for more details see [18] and the references therein).

Kern et al. [13] used a delay of 600 ms for marking gaze positions to reduce attention switching costs between multiple computer screens. Hyrskykari et al. [11] used a dwell time of 700 ms in a multi-user role-playing game where the user's gaze has to be maintained in the center of the screen for most of the time. Fedorova et al. [7] employed fixations of 500 and 300 ms long for robot control. They note that this resulted in slow but reliable communication, in situations where distractors are common and false alarms can have high costs. Lutteroth et al. [14]

used only a 200 ms activation dwell threshold in a web browsing task and achieved a fairly close performance to the mouse click alternative. Chen and Shi [5] investigated variable dwell time in a web-browsing task using probabilistic models and their best model reduced error rate by 50% and response time by 60% while maintaining the other performance measure constant when compared to a uniform dwell time of 100 and 300 ms, respectively. They also used in their practice experiments a fixed 500 ms dwell time.

Based on the above studies and also taking into consideration the fact that visual reaction time is considerably less than 500 ms [29], we selected a dwell time duration of 500 ms.

2.2 Participants and experiments

In our previous work [24], we have performed a longitudinal study with 10 participants, who were asked to play with the regular mouse control version of the DA Game. The volunteers were aged between 25 and 30 (mean age was 27 years, SD=1.76), had normal or corrected-to-normal vision and reported no attention disorders nor color vision deficiency. The experiments lasted several days, with multiple trials played each day. We manipulated the difficulty of the game, i.e. the moving speed of the squares, according to the score of the players from the previous trial. Based on this, the experiments were separated into three phases: beginner, intermediate and advanced.

For the experiments with the gaze-based control version, 9 out of the 10 participants were invited back for ten additional trials. The participants were instructed that data about their gameplays will be logged for further analysis and they were asked to sign a consent form before the experiments. We also allowed rest periods after each trial if the subject requested so.

For gaze tracking the Tobii EyeX Controller[2] [8] device was used, which is attached to the bottom of the display, has a sampling rate of 60 Hz and requires personal calibration before each data collecting session. Although the manufacturers claim that no continuous recalibration is required [8], drifts may occur over time [28] due to illumination or head position changes. Accordingly, we repeated the calibration procedure between trials when necessary.

2.3 Performance measures

The details of the experiments with the mouse control version of the DA Game are presented in our previous work [24]. For the purposes of this study we selected 10 consecutive trials from the intermediate phase to compare them in all of our analyses with the 10 gaze control trials. The difficulty of the game was set to a default value, meaning a decrease of 15% on average compared to the last trial selected for comparison.

The performance of the participants is determined by the user errors, which can be separated into two categories:

[2] `https://tobiigaming.com/product/tobii-eyex/`

(i) *errors of omission* are the cases when the player misses an action; these are the more common ones and can have several causes such as the place of the action is outside the visual field or too little time to handle multiple parallel tasks;

(ii) *errors of commission* occur when the player performs a wrong action, and does not correct it. These mistakes are the more rare ones in the DA Game and can happen when the player confuses two colors, performs an action too early or acts recklessly because of pressure.

In our analysis we computed the number of each type of user error and compared the means between the mouse control and the gaze-based control versions using the repeated measures analysis of variance (ANOVA) statistical model. Furthermore, we fit linear regression lines on the number of user errors to analyze the change over the trials from our experiments.

We calculate the length of the time intervals passed from the moment of a proper switch flip until the square actually passes the switch node. This latter event corresponds to the last moment when the switch could have been still flipped. We compared the distributions of these remaining time intervals between the mouse and the gaze-based control versions, to see whether there are considerable differences, i.e. whether the dwell time is limiting performance.

We also analyze the strategies of the participants. Particularly, we define two measures that characterize their decision making. The first one, called *double switch*, refers to flipping the same switch twice in a row, where after the first proper switch the player fails to look away and the dwell time elapses again resulting in another erroneous switch flip. This action corresponds to performing a double click with the mouse. We fit linear regression lines on the number of double switches to check whether they show an increasing or decreasing pattern during the trials of our experiments.

The second strategic measure is called *planning* or *planning ahead* and is defined in detail in our previous work [24], where it was found the most important predictor of performance in a regression analysis. It involves thinking in advance, executing an action before the situation would become critical and has the effect of reducing future timing constraints and/or cognitive load. We compare the planning strategic measure between the mouse and gaze-based control experiments using repeated measures ANOVA.

In our analysis we test the following experimental hypotheses.

H1 Despite carefully controlling experimental and design aspects, the number of user errors is considerably increased compared to the mouse control version.

However, we expect to observe a slow decrease in the number of errors and in the number of double switches over time in the amount of experiments we performed, i.e. players would start to get used to the gaze-based control version of the DA Game.

H2 The 500 ms dwell time does not influence considerably the distribution of the remaining times, i.e. it does not impair performance by limiting the available times to perform switch flips.

H3 The planning strategic measure is decreased when using gaze-based control, most likely because of the higher cognitive load.

3 Results

We calculated the total number of user errors for each trial in both conditions (mouse control and gaze control). There was a statistically significant difference in means of user errors between the two control types, as determined by the repeated measures ANOVA, $F(1, 8) = 61.19$, $p < 0.001$. Figure 2 shows the mean of the user error numbers across trials in each of the two control versions of the DA Game, separately for every subject. Clearly, gaze control yielded lower scores on average.

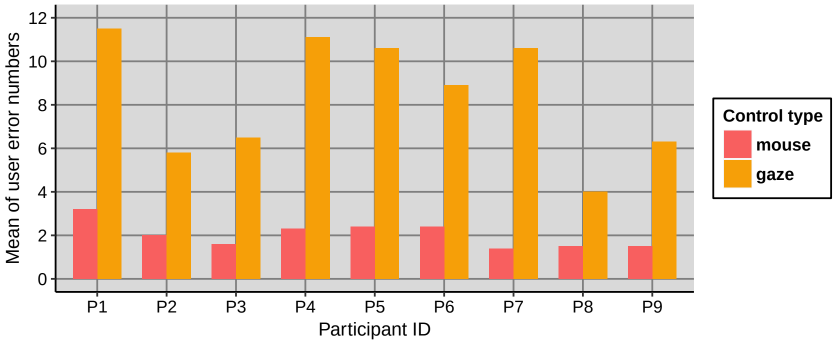


Figure 2: Comparison of overall mean of user error numbers, separately for participants.

The proportion of the commission type of errors to the total number of errors was also calculated in both conditions. Figure 3 compares these percentages computed over all 10 trials for each participant. We can see that generally the proportion of commission errors is considerably higher in the gaze version. Also, the repeated measures ANOVA for proportion of commission errors (computed for each trial separately) showed significant main effects of control type (mouse vs. gaze), $F(1, 8) = 20.28$, $p = 0.002$.

Figure 4 shows regression lines fitted on the number of user errors across trials. The errors of omission are decreasing in case of six subjects (P1, P3, P5, P6, P7, P9) and increasing in case of two subjects (P2 and P8). The errors of commission are decreasing in case of five (P3, P5, P7, P8, P9) and increasing in case of two subjects (P4 and P6). For some players we can see a reasonable learning rate when considering the sum of errors (P3, P5, P7, P9). However, the average of the

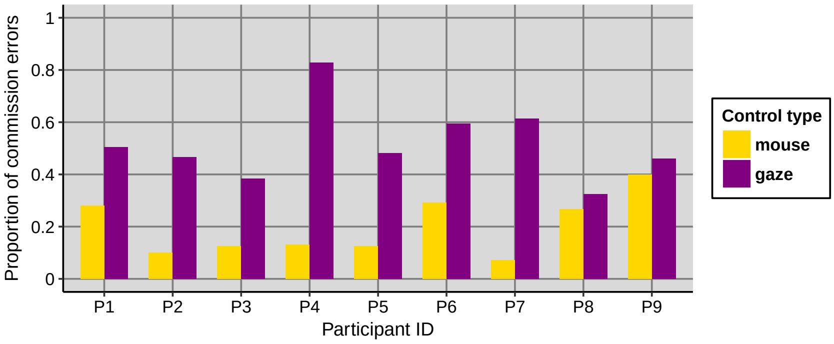


Figure 3: Comparison of overall proportion of commission type user errors, separately for participants.

user error numbers still remained considerably higher when compared to the mouse control version, as seen on Figure 2.

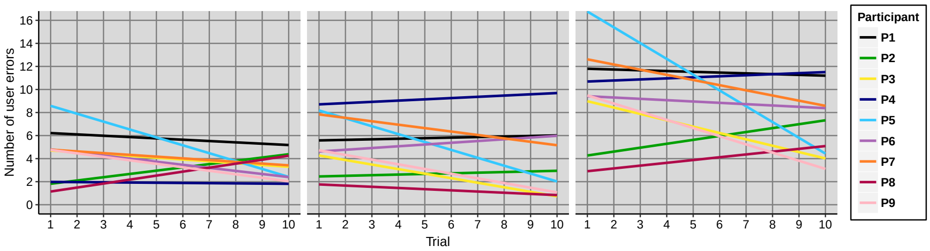


Figure 4: Linear regression lines fitted on the number of errors separately for participants, from left to right: errors of omission, errors of commission, sum of the two error types.

Figure 5 visualizes the smoothed distributions of the remaining time interval lengths, for both the mouse and gaze versions. We used the two-sample one-sided Mann-Whitney-Wilcoxon test to assess whether there are statistically significant differences between the distributions of the data for each of the 9 participants. The tests indicated that the remaining times were greater for the mouse control version than for the gaze-based control version in case of only 3 participants (P1, P8 and P9), $p < 0.001$. Inspecting the graphs on Figure 5 we can see that the distributions are surprisingly similar in case of each participant, the peaks for mouse control are higher and the curves are not shifted consistently to the left when switching to gaze-based control.

Figure 6 shows regression lines fitted on the number of double switches across trials. The lines show a considerable decrease for P5 and P7, and a substantial increase for P3.

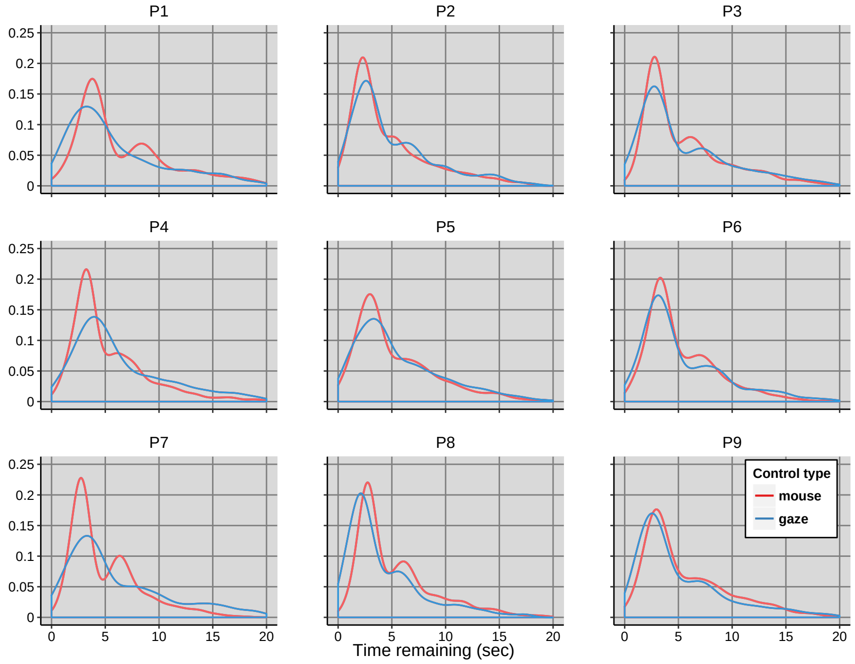


Figure 5: Comparison of distributions of remaining time intervals from the moment of switch flip until last possibility of performing the action, separately for participants. The title of each subplot is the identifier of the participant.

Figure 7 shows the comparison of the planning strategic measure between the mouse and gaze control versions for every participant. The results are mixed, confirmed also by repeated measures ANOVA, which revealed no significant main effect of control type, $F(1, 8) = 1.35$, $p = 0.28$. Particularly, planning was increased in case of four subjects (P4, P5, P6, P7), while it decreased for the others when switching to gaze-based control.

4 Discussion

We investigated the effect of switching from mouse control to gaze-based control in a complex divided attention task. Three hypotheses were tested and we elaborate the findings below.

H1 This was confirmed, since the number of user errors was increased in the gaze control version as shown on Figure 2, and this change was also statistically significant, as determined by the repeated measures ANOVA.

Regarding the second part of this hypothesis, it was confirmed only partially. Not all participants started to get used to the gaze-based control version of the

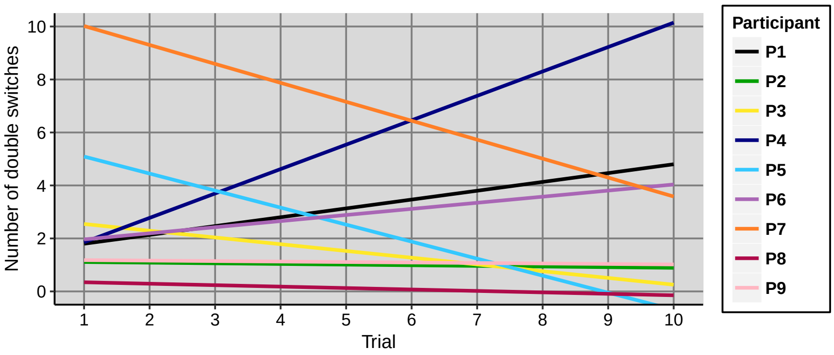


Figure 6: Linear regression lines fitted on the number double switches separately for participants.

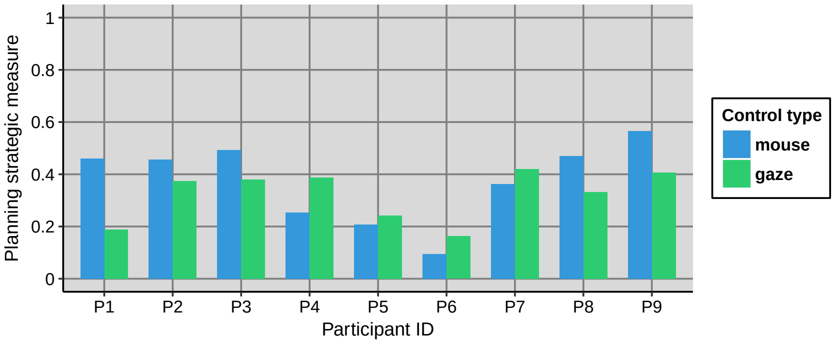


Figure 7: Comparison of planning strategic measure separately for participants.

DA Game, because the total number of user errors and the number of double switches decreased during our experiments only in the case of P3, P5, P7 and P9, as demonstrated by Figures 4 and 6.

H2 This was confirmed, i.e. the 500 ms dwell time did not limit considerably the available times to perform switch flips. Although the Mann-Whitney-Wilcoxon test indicated statistically significant differences between the distributions of the remaining times for the mouse and gaze control versions for three subjects, the distributions were similar for each participant, as shown on Figure 5, and the differences were not consistently reflecting the dwell time.

H3 This was confirmed only partially, because the planning strategic measure was actually increased in case of four participants, as seen on Figure 7.

Several notes have to be made regarding our experiments and results. The sum of the user errors was considerably increased in the gaze-based control version of the DA Game. But what was really unanticipated is the significant increase of the proportion of errors of commission, as seen on Figure 3. This demonstrates that performance decrements can be attributed in large part to reckless actions from the increased cognitive load and not time constraints. Our subjects had difficulty in restraining themselves to not double-check the switches nodes after they made the proper action. Also they fail to avoid double switches, although the dwell time of 500 ms should be long enough to react and look away after the switch has been flipped once [29].

One might argue that our results are biased because of the selection of the trials for comparison from the mouse control version. However, the intermediate phase was the part where subjects were familiar with the DA Game and could play comfortably after the beginner phase [24]. In the advanced phase difficulty was high in order to test the effects of time pressure, so this would not provide a proper comparison. In addition, the difficulty of the DA Game was decreased in the experiments presented here.

The sample size in this study is small, which restricts the strength of generalizability of our findings and the statistical power of our analysis. It is plausible that the performance differences are due to the lack of practice with gaze-based control. A balanced study, where subjects would get experienced in gaze-based control first, is almost impossible to perform. Nevertheless, analyzing the strategies of participants shows that it is possible to achieve a fast learning rate in the gaze version of the DA Game. Specifically, the number of both user error types and the number of double switches were decreasing during our experiments and also the planning strategic measure was increased compared to mouse control in case of subjects P5 and P7 (see Figures 4, 6 and 7). One important lesson to learn from our experiments is that since using gaze as an input method in dynamic environments requires conscious effort from the user to carefully avoid looking at prohibited targets, the implementation of such interfaces requires careful design and experimentation.

4.1 Future work possibilities

It can be argued that choosing dwell time as the selection method in gaze-based control can limit performance. Indeed this latency contributes to cognitive load because it limits the exploration time in the DA Game. The choice of dwell time duration represents a trade-off between speed and accuracy. To achieve better performance adaptive dwell time [5, 15, 16] might represent a plausible alternative. This may require machine learning techniques in order to find a good model for predicting dwell times in our dynamic task.

One can implement an animation to indicate the progression of dwell time [15, 18]. Also the item in focus might be highlighted to increase the responsiveness of the interface. However, these could distract attention in spatial tasks with timing constraints. Also zooming into the focus area might be counterproductive as it means losing context information too [13].

An alternative for dwell time based selection is using a blink for the signal. But this would disrupt the natural interaction by requiring the user to think about it before blinking. Another option for performing a click is to combine gaze-pointing with a hardware button. This may be faster than simple dwell time, but less accurate since users may tend to click before gaze has fully settled on the target [14]. One could combine gaze with other input modalities [17], for example speech, head movements or even feet [3]. However, these would not work for people with disabilities who could potentially use only gaze as an input method.

Gaze gestures [11, 17, 26] might provide a robust alternative to dwell-based interaction to avoid unintentional commands. Some researchers also studied predefined gaze patterns as pre-programmed strategies for control [7, 14]. However, participants with disabilities would have considerable difficulty in performing such gaze patterns or gestures.

Gaze input requires concentration to control the eyes consciously. Implicit use of gaze for control can release users from this burden [3], while explicit gaze input should be applied carefully, since it may have cognitive drawbacks. Combining gaze-based control with EEG signal analysis can help to gain further insight into cognitive processes [25].

All these alternative options for employing gaze-based control require further studies, possibly using a larger sample size.

5 Conclusion

In this paper, we conducted a small scale experimental study to analyze the impact of switching from mouse to gaze-based control in a special divided attention task, requiring continuous focused concentration and frequent shifts of attention. We conducted experiments with 9 participants and carefully controlled design and experimental aspects: the mouse control version of the task was well practiced, the difficulty was adjusted to the more demanding conditions and the parameters of gaze-based control were selected based on previous research findings. Despite all these circumstances, gaze control had a significant negative impact on the performance of participants.

In contrast to our assumptions, experienced users could not get used to gaze-based control in the amount of experiments we performed. On the other hand, by investigating the problem solving strategies of users, we showed that some subjects could make considerable progress in our task even in a short amount of practice. Our efforts suggest that with careful design, proper testing and sufficient user training, gaze controlled computer interfaces can become helpful in environments requiring divided attention.

Acknowledgements

The authors would like to thank their supervisor, András Lőrincz, for his expertise during the project, Zoltán Tősér for supervising the implementation process of the divided attention game used in this study, and Richárd Bellon and Dávid Hornyák for implementing the software. The authors also express their gratitude to the volunteers who kindly participated in the experiments.

References

[1] Bednarik, Roman, Gowases, Tersia, and Tukiainen, Markku. Gaze interaction enhances problem solving: Effects of dwell-time based, gaze-augmented, and mouse interaction on problem-solving strategies and user experience. *Journal of Eye Movement Research*, 3(1), 2009.

[2] Borji, Ali, Lennartz, Andreas, and Pomplun, Marc. What do eyes reveal about the mind? algorithmic inference of search targets from fixations. *Neurocomputing*, 149:788–799, 2015.

[3] Çöltekin, Arzu, Hempel, J., Brychtova, A., Giannopoulos, Ioannis, Stellmach, Sophie, and Dachselt, Raimund. Gaze and feet as additional input modalities for interacting with geospatial interfaces. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume III-2. ETH-Zrich, 2016.

[4] Chen, Chun-Ching and Huang, Yen-Yi. Exploring the effect of color on the gaze input interface. In *2018 IEEE International Conference on Applied System Invention (ICASI)*, pages 620–623, April 2018.

[5] Chen, Zhaokang and Shi, Bertram E. Using variable dwell time to accelerate gaze-based web browsing with two-step selection. *International Journal of Human-Computer Interaction*, 2018.

[6] Dorr, Michael, Pomarjanschi, Laura, and Barth, Erhardt. Gaze beats mouse: A case study on a gaze-controlled breakout. *PsychNology*, 7(2):197–211, 2009.

[7] Fedorova, Anastasia A., Shishkin, Sergei L., Nuzhdin, Yu O., and Velichkovsky, Boris M. Gaze based robot control: The communicative approach. In *2015 7th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 751–754, April 2015.

[8] Gibaldi, Agostino, Vanegas, Mauricio, Bex, Peter J., and Maiello, Guido. Evaluation of the Tobii EyeX Eye tracking controller and Matlab toolkit for research. *Behavior Research Methods*, 49(3):923–946, 2017.

[9] Hardy, Joseph L., Farzin, Faraz, and Scanlon, Michael. The science behind Lumosity, Version 2, 2013. Lumos Labs, Inc.

[10] Holmqvist, Eva, Derbring, Sandra, and Wallin, Sofia. Participation through gaze controlled computer for children with severe multiple disabilities. *Studies in Health Technology and Informatics*, 242:1103–1108, 2017.

[11] Hyrskykari, Aulikki, Istance, Howell, and Vickers, Stephen. Gaze gestures or dwell-based interaction? In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, pages 229–232, New York, NY, USA, 2012. ACM.

[12] Jacob, Robert J. K. What you look at is what you get: Eye movement-based interaction techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, pages 11–18, New York, NY, USA, 1990. ACM.

[13] Kern, Dagmar, Marshall, Paul, and Schmidt, Albrecht. Gazemarks: Gaze-based visual placeholders to ease attention switching. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 2093–2102, New York, NY, USA, 2010. ACM.

[14] Lutteroth, Christof, Penkar, Moiz, and Weber, Gerald. Gaze vs. mouse: A fast and accurate gaze-only click alternative. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, pages 385–394, New York, NY, USA, 2015. ACM.

[15] Majaranta, Päivi, Ahola, Ulla-Kaija, and Špakov, Oleg. Fast gaze typing with an adjustable dwell time. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 357–360, New York, NY, USA, 2009. ACM.

[16] Majaranta, Päivi, Aula, Anne, and Räihä, Kari-Jouko. Effects of feedback on eye typing with a short dwell time. In *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications*, ETRA '04, pages 139–146, New York, NY, USA, 2004. ACM.

[17] Majaranta, Päivi and Bulling, Andreas. *Eye Tracking and Eye-Based Human-Computer Interaction*, pages 39–65. Springer, London, 2014.

[18] Majaranta, Päivi, Isokoski, Poika, Rantala, Jussi, Špakov, Oleg, Akkil, Deepak, Kangas, Jari, and Raisamo, Roope. Haptic feedback in eye typing. *Journal of Eye Movement Research*, 9(1), 2016.

[19] Majaranta, Päivi, MacKenzie, I. Scott, Aula, Anne, and Räihä, Kari-Jouko. Auditory and visual feedback during eye typing. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, pages 766–767, New York, NY, USA, 2003. ACM.

[20] Menges, Raphael, Kumar, Chandan, Sengupta, Korok, and Staab, Steffen. eyegui: A novel framework for eye-controlled user interfaces. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction*, NordiCHI '16, pages 121:1–121:6, New York, NY, USA, 2016. ACM.

[21] Murata, Atsuo. Eye-gaze input versus mouse: Cursor control as a function of age. *International Journal of Human-Computer Interaction*, 21(1):1–14, 2006.

[22] Park, Seonwook, Spurr, Adrian, and Hilliges, Otmar. Deep pictorial gaze estimation. In *The European Conference on Computer Vision (ECCV)*, pages 721–738, September 2018.

[23] Prabhakar, Gowdham and Biswas, Pradipta. Eye gaze controlled projected display in automotive and military aviation environments. *Multimodal Technologies and Interaction*, 2(1), 2018.

[24] Rill, Róbert Adrian, Faragó, Kinga Bettina, and Lőrincz, András. Strategic predictors of performance in a divided attention task. *PLOS ONE*, 13(4):1–27, 2018.

[25] Sengupta, Korok, Sun, Jun, Menges, Raphael, Kumar, Chandan, and Staab, Steffen. Analyzing the impact of cognitive load in evaluating gaze-based typing. In *2017 IEEE 30th International Symposium on Computer-Based Medical Systems (CBMS)*, pages 787–792, June 2017.

[26] Shao, Yuan-Fu, Wang, Chiuan, and Fuh, Chiou-Shann. Eyelasso: Real-world object selection using gaze-based gestures. In *28th IPPR Conference on Computer Vision, Graphics, and Image Processing*, 2015.

[27] Smith, J. David and Graham, T. C. Nicholas. Use of eye movements for video game control. In *Proceedings of the 2006 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, ACE '06, New York, NY, USA, 2006. ACM.

[28] Stampe, Dave M. and Reingold, Eyal M. Selection by looking: A novel computer interface and its application to psychological research. In Findlay, John M., Walker, Robin, and Kentridge, Robert W., editors, *Eye Movement Research*, volume 6 of *Studies in Visual Information Processing*, pages 467–478. North-Holland, 1995.

[29] Woods, David L., Wyma, John M., Yund, E. William, Herron, Timothy J., and Reed, Bruce. Factors influencing the latency of simple reaction time. *Frontiers in Human Neuroscience*, 9:131, 2015.

[30] Yarbus, Alfred L. *Eye movements and vision.* Plenum Press, 1967.

CONTENTS

ISSN 0324—721 X

Felelős szerkesztő és kiadó: Csendes Tibor