INVARIANT PATTERN RECOGNITION SYSTEM USING RT AND GMDH

*Ján TURÁN, *Ľuboš OVSENÍK, **Ján TURÁN, Jr.
*Department of Electronics and Multimedia Communications, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 042 00 Košice, Slovak Republic, tel. 055/602 2943, E-mail: jan.turan@tuke.sk
**3D People gmbh, Kaiser Passage 6, D-72766 Reutlingen, Germany

SUMMARY

The proposed paper gives the results of the development work related to design pattern recognition system based on Application of fast translation invariant Rapid Transform (RT) and GMDH. The system was implemented as a software package on PC and tested with identification of classes of real objects. Experimental results are given for applying the proposed invariant pattern recognition system to recognition of Nativity Symbols, Informative Symbols and Cuneiform Writings corrupted by noise.

Keywords: *GMDH, Rapid Transform (RT), Modified Rapid Transform (MRT), pattern recognition, invariant feature extraction, information symbol classification*

1. INDRODUCTION

Transformation methods can be used to obtain alternative descriptions of signals. These alternative descriptions have many uses such as classification, redundancy reduction, coding, etc., because some of these tasks can be better performed in the transform domain [1,2].

Various transformations have been suggested as a solution of the problem of high dimensionality of the feature vector and long computation time. Such transforms are RT and modified RT (MRT), which are fast translation invariant transforms from the class CT [1-4]. We apply the RT in feature extraction stage of the recognition process.

Whereas conventional empirical modelling techniques require an assumed model structure, new procedures have been developed which generate the model structure as well as the model coefficients from a database [1,5-7]. One of these procedures is he GMDH (Group Method of Data Handling) algorithm usually used for creating polynomial networks with active units. GMDH is a useful data analysis technique for the modelling of non-linear complex systems [5,6,7]. We apply the GMDH algorithm as intelligent network classifier in the proposed new Invariant Pattern Recognition System.

2. RAPID TRANSFORM

In the field of pattern recognition and also scene analysis is well known the class of fast translation invariant transforms - Certain Transforms (CT) [1,3,4] based on the original rapid transform (RT) [3] but with choosing of other pairs of simple commutative operators. The RT results from a minor modification of the Walsh-Hadamard transform (WHT). The signal flow graph for the RT is identical to that of the WHT, except that the absolute value of the output of each stage of the iteration is taken before feeding it to the next stage. The signal flow graph for one-dimensional RT is showed on the

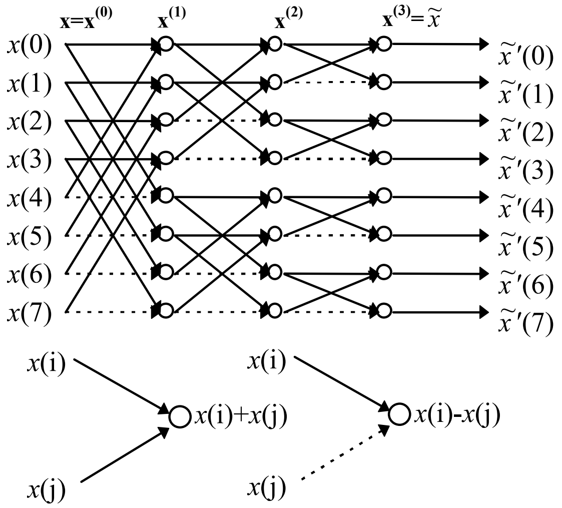

Fig. 1 Signal flow graph for 1D RT

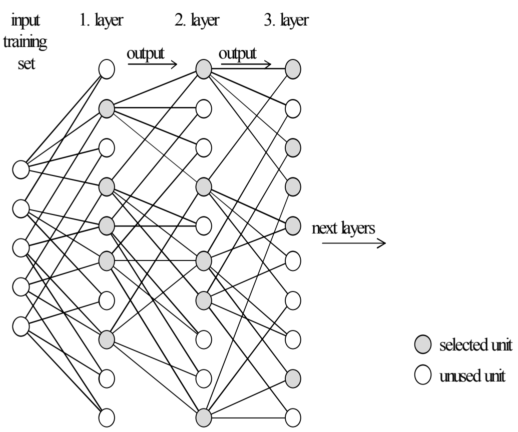

Fig. 2 Functional flow graph of multilayer GMDH algorithm

Fig. 1. RT is not an orthogonal transform, as no direct inverse exists. With the help of additional data, however, the signal can be recovered from the transform sequence, i.e. invertible rapid transform (IRT) can be defined [1,8]. RT has some interesting properties such as invariance to cyclic shift, reflection of the data sequence, and the slight rotation of a two-dimensional pattern. It is applicable to both binary and analogue inputs and it can be extended to multiple dimensions [1].

3. GMDH ALGORITHM DESCRIPTION

The idea of GMDH (Group Method of Data Handling) is the following: we are trying to build an analytical function (called "model") which would behave itself in such a way that the estimated value of the output would be as close as possible to its actual value [5]. For many applications such an analytical model is much more convenient than the "distributed knowledge" representation that is typical for neural network approach [6,7,11].

The most common way to deal with such a problem is to use linear regression approach. In this approach, first of all we must introduce a set of basis functions. The answer will then be sought as a linear combination of the basis functions [5]. For example, powers of input variables along with their double and triple cross products may be chosen as basis functions. To obtain the best solution, we should try all possible combinations of terms, and choose those that give best prediction. The decision about quality of each model must be made using some numeric criterion. To reduce computational expenses, one should reduce the number of basis functions (and the number of input variables), which are used to build the tested models. To do that, one must change from a one-stage procedure of model selection to a multistage procedure.

GMDH is based on sorting out procedure, that is successive testing of models selected out of a set of candidate models according to a specified criterion [6]. Most of GMDH algorithms use polynomial support functions. General connection between input and output variables can be found in form of a functional Volterra series, whose discrete analogue is known as the Kolmogorov-Gabor polynomial [5,7]:

$$y = a_0 + \sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} x_i x_j + \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} a_{ijk} x_i x_j x_k + \dots \qquad (1)$$

where $\boldsymbol{X}=(x_1, x_2, \dots , x_n)$ is the vector of input variables and $\boldsymbol{A}=(a_i, \dots ,a_{ij}, \dots , a_{ijk,\dots})$ is the vector of the summand coefficients. Components of the input vector $\boldsymbol{X}$ can be independent variables, functional forms or finite difference terms [5]. The method allows finding simultaneously the structure of the model and the dependence of modelled system output on the values of most significant inputs of the system.

The multilayer GMDH algorithm enables to construct Kolmogorov-Gabor polynomial by a composition of lower-order polynomials (partial function) of the form [5,11]:

$$y = a_0 + a_1 x_i + a_2 x_j + a_3 x_i x_j + a_4 x_i^2 + a_5 x_j^2 \qquad (2)$$

where *i,j = 1, 2, ... , m; i≠j.*

To find these polynomials (the coefficients) it is sufficient to have only six data points at our disposal. Repeated solution of this quadratic polynomial (2) enables to construct the complete polynomial (1) of any complexity.

The input data of m input variables x are fed randomly; for example, if they are fed in pairs at each unit (node or processing element PE), then a total of $C_m^2 = \frac{m(m-1)}{2}$ partial functions (PEs) of the form below are generated at the first layer (Fig. 2):

$$y = f(x) \qquad (3)$$

where $f(x)$ is partial function as in (2) and y is its estimated output.

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N_A} \\ \vdots \\ y_{N_A+N_B} \\ \vdots \\ y_N \end{bmatrix}, \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N_A,1} & x_{N_A,2} & \cdots & x_{N_A,m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N_A+N_B,1} & x_{N_A+N_B,2} & \cdots & x_{N_A+N_B,m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,m} \end{bmatrix} \begin{array}{l} \left.\vphantom{\begin{matrix} a \\ a \\ a \end{matrix}}\right\} \textbf{A} \text{ contain } N_A \text{ data of } N \\ \left.\vphantom{\begin{matrix} a \\ a \end{matrix}}\right\} \textbf{B} \text{ contain } N_B \text{ data of } N \\ \left.\vphantom{\begin{matrix} a \\ a \\ a \end{matrix}}\right\} \textbf{C} \text{ contain } N_C \text{ data of } N \end{array}$$

Fig. 3 Data splitting

Then outputs of $F_1 \le C_m^2$ functions (PEs) are selected (freedom of choice) as per the threshold measure to pass on to the second layer as input in pairs (Fig. 2). To avoid over fitting, available input-output data are divided into two sets: one set is used for training (training data set) and the other is the selection data set (Fig. 3). Parameters of the polynomial are calculated using only the training data set. In the next layer the partial functions of the same form (2) are generated. The number of such functions (nodes) is $C_{F_1}^2$. The process continues until the stopping criterion is satisfied: typically, the mean squared error (MSE) of the best performing PE (node) of every layer will decrease until a minimum is reached at layer k; if further layers are added, the MSE will actually rise [5]. The best performing PE on layer k (or first p best PEs) is selected as the output node for entire network.

4. IMPLEMENTATION OF THE GMDH ALGORITHM

Data can be previously normalized by:

$$\mathbf{x}_i = \frac{\mathbf{x}_i}{\mathbf{x}_{i,\max}}, \mathbf{y} = \frac{\mathbf{y}}{\mathbf{y}_{\max}}, \mathbf{x}_i, \mathbf{y} \in [0,1] \tag{4}$$

Most of the selection criteria require the division of the data into two or more sets. Suppose we have a sample set of N data points $(\mathbf{x}_1, y_1)$, $(\mathbf{x}_2, y_2)$, ..., $(\mathbf{x}_N, y_N)$. First thing to do is splitting the data set into three sets: the training data set **A**, the selection data set **B** (**W** = **A** ∪ **B**) the test data set **C** (Fig. 3). The first two sets are used to construct the network, and the test data set is used to obtain a measure of its performance (to find the optimal model or models).

The data splitting can be performed in several ways, which is depending on the application. In general, the data can be ordered (according to their variance, time, etc.) or unordered and proportions of splitting can be 40%, 25% and 35%, or 50%, 25% and 25% or other for **A**, **B** and **C** set correspondingly (Fig. 3). In the case, when data are arranged according to their variance, data with higher variance belong to the training set.

In our experiments each processing element receives three input variables x_i, x_j, x_k $i \neq j$, $i \neq k$, $j \neq k$ and generates output using linear and polynomial activation function respectively:

$$\hat{y}_l = a_0 + a_1 x_{l,i} + a_2 x_{l,j} + a_3 x_{l,k}\ ,\ l = 1,\ldots,N_A \tag{5}$$

$$\hat{y}_l = a_0 + a_1 x_{l,i} + a_2 x_{l,j} + a_3 x_{l,k} + a_4 x_{l,i}^2 + a_5 x_{l,j}^2 + a_6 x_{l,k}^2\ ,\ l = 1,\ldots,N_A \tag{6}$$

The weights $\mathbf{a} = [a_0, a_1, \ldots a_3]$ and $\mathbf{a} = [a_0, a_1, \ldots a_6]$ are computed by least squares technique:

$$\hat{\mathbf{a}} = (\mathbf{X}_{A,i,j}^T \mathbf{X}_{A,i,j})^{-1} \mathbf{X}_{A,i,j}^T \mathbf{y}_A \tag{7}$$

where

$$\mathbf{X}_{A,i,j} = \begin{bmatrix} 1 & x_{1,i} & x_{1,j} & x_{1,k} \\ 1 & x_{2,i} & x_{2,j} & x_{2,k} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{N_A,i} & x_{N_A,j} & x_{N_A,k} \end{bmatrix}, \quad \mathbf{y}_A = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N_A} \end{bmatrix}$$

and $\hat{\mathbf{y}} = \mathbf{X}_{A,i,j}\hat{\mathbf{a}}$.

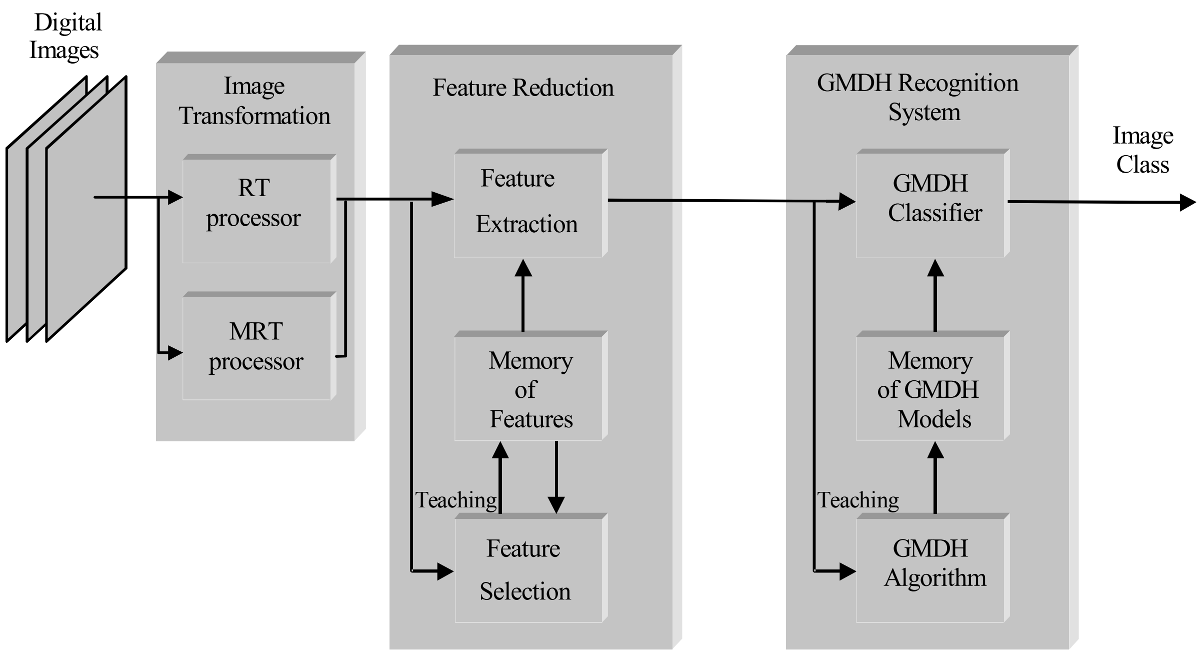

Fig. 4 Block scheme of pattern recognition system based on RT and MRT transforms and GMDH algorithm

All partial functions are evaluated by follow external criterion:

$$\Delta^2 \hat{=} \frac{\sum_{p \in C} (\hat{y}_p^c - y_p)^2}{\sum_{p \in W} y_p^2} \qquad (8)$$

where W=A∪B and C is test data set.

The algorithm will stop when:

- a maximum number of layers has been reached ($k = k_{max}$)
- the performance of the best-fitted node on each layer has reached a minimum.

5. INVARIANT PATTERN RECOGNITION SYSTEM

Block scheme of the invariant pattern recognition system based on RT (or MRT) transform and GMDH algorithm is on Fig. 4. Digital pattern enter to "Image Transformation " module, where is transformed using RT or MRT. The amount of data is reduced in "Feature Reduction" module. Features that will be in feature vector are selected during the teaching process and stored in “Memory of Features” module.

In GMDH classification system (Fig. 4) each independent category of patterns (images) has it’s own model computed in teaching process. These models are stored in "Memory of GMDH Models" module. Output of each model is value 1, if the input pattern corresponds with the class of that model and output value is 0 otherwise.

6. EXPERIMENTAL RESULTS

The proposed new invariant pattern (image) recognition system was tested in the recognition of a set of 120 independent classes of Nativity Symbols (Fig. 5), of Informative Symbols (Fig. 6) and of Cuneiform Writings (Fig. 7). We implemented feature extraction with RT. As teaching sets we use sets containing 60 to 252 symbols for each class of symbols. As a recognition sets we use eight sets of 120 noised symbols with noise rate 1%, 2%, ..., 8%. The results of experiments with RT using simple Euclid classifier and polynomial (linear and non-linear) GMDH classifier are on Tab. 1 for Nativity Symbols, on Tab. 2 for Informative Symbols and on Tab. 3 for Cuneiform Writings. As can be seen, the recognition system with GMDH classifier gives better performance. The recognition efficiency is increasing if we use teaching sets with noised patterns. The best performance is for the system based on combination of RT and GMDH algorithm.

Tab. 1 The efficiency of the recognition process for nativity symbols

	Teaching set				
	Without noise and 1% of noise	Without noise, 1% and 2% of noise	Without noise, 1%, 2% and 3% of noise	Without noise, 1%, 2%, 3% and 4% of noise	Without noise, 1%, 2%, 3%, 4% and 5% of noise
RT	80,274%	84,167%	88,333%	88,981%	90,833%
RT + linear GMDH	77,592%	82,407%	87,315%	95,092%	96,111%
RT + non-linear GMDH	69,722%	68,333%	84,537%	89,722%	94,074%

Tab. 2 The efficiency of the recognition process for informative symbols

	Teaching set				
	Without noise and 1% of noise	Without noise, 1% and 2% of noise	Without noise, 1%, 2% and 3% of noise	Without noise, 1%, 2%, 3% and 4% of noise	Without noise, 1%, 2%, 3%, 4% and 5% of noise
RT	89,091%	90,101%	91,111%	92,828%	94,545%
RT + linear GMDH	85,859%	91,313%	94,646%	95,253%	95,960%
RT + non-linear GMDH	93,131%	93,838%	94,141%	97,576%	97,576%

Tab. 3 The efficiency of the recognition process for cuneiform writings

	Teaching set				
	Without noise and 1% of noise	Without noise, 1% and 2% of noise	Without noise, 1%, 2% and 3% of noise	Without noise, 1%, 2%, 3% and 4% of noise	Without noise, 1%, 2%, 3%, 4% and 5% of noise
RT	78,549%	80,452%	82,778%	84,028%	86,354%
RT + linear GMDH	86,736%	90,973%	94,271%	95,209%	96,320%
RT + non-linear GMDH	77,222%	84,167%	87,361%	91,875%	95,174%

7. CONCLUSION

The proposed paper gives the results of the development work related to design a new invariant pattern recognition system based on the combination of the RT and the GMDH algorithm. The proposed system was realised as a software tool on the PC and tested in experiments with recognition of noised Nativity Symbols, Informative Symbols and Cuneiform Writings. The obtained experimental results are satisfied and recognition efficiency, which was obtained, are up to 69% - 97% for Nativity Symbols, up to 85% - 98% for Informative Symbols and up to 77% - 97% for Cuneiform Writings. The obtained results are satisfied.

ACKNOWLEDGEMENTS

The authors are thanking for the financial support from the COST 276 and COST 292 grant and VEGA grant No. 1/0381/03.

REFERENCES

[1] Turan, J.: Fast Translation Invariant Transforms and Their Applications. Elfa, Košice, 1999.

[2] Chmurny, J. - Turan, J.: Two-dimensional Fast Translation Invariant Transforms and Their Use in Robotics. Electronic Horizon, Vol.15, No.5, 1984, pp. 211-220.

[3] Reitboeck, H. - Brody, T.P.: A Transformation with Invariance Under Cyclic Permutation for Application in Pattern Recognition. Inf. and Control, Vol.15, 1969, pp. 130-154.

[4] Wagh, M.D. - Kanetkar, S.V.: A Class of Translation Invariant Transforms. IEEE Trans. on Acoustic, Speech and Signal Proc., Vol. ASSP-25, No.3, 1977, pp. 203-205.

[5] Madala, H.R. - Ivakhnenko, A.G.: Inductive Learning Algorithms for Complex System Modeling. CRC Press, London 1994.

[6] Ivakhnenko, A.G. - Ivakhnenko, G.A.: The Review of Problems Solvable by Algorithms of the Group Method of Data Handling (GMDH). Pattern Recognition and Image Analysis, Vol. 5, No. 4, 1995, pp. 527-535.

[7] Farlow, J.S.: Self-organizing Methods in Modelling - GMDH Type Algorithms, Statistics: Textbooks and Monographs. MARCEL DEKKER, INC., New York 10016, USA, 1984.

[8] Turan, J. - Chmurny, J.: Two-dimensional Inverse Rapid Transform. Computers and Art. Intelligence, Vol.2, No.5, 1983, pp. 473-477.

[9] Fang, M. - Hausler, G.: Modified Rapid Transform. Applied Optics, Vol.28, No.6, 1989, pp. 1257-1262.

[10] Turan, J. - Kovesi, L. - Kovesi, M.: CAD System for Pattern Recognition and DSP with Use of Fast Transformation Invariant Transforms. Journal on Communications, Vol. XLV, 1994, pp. 85-89.

[11] Ivakhnenko, A.G. – Ivakhnenko, G.A. - Müller, J.A.: Selforganization of Neurones with Active Neurons. Pattern Recognition and Image Analysis, Vol. 4, No. 2, 1994, pp. 185-196.

BIOGRAPHY

Ján Turán (Prof, Ing, RNDr, DrSc) was born in Šahy, Slovakia. He received Ing (MSc) degree in physical engineering with honours from the Czech Technical University, Prague, Czech Republic, in 1974, and RNDr (MSc) degree in experimental physics with honours from Charles University, Prague, Czech Republic, in 1980. He received a CSc (PhD) and DrSc degrees in radioelectronics from University of Technology, Košice, Slovakia, in 1983, and 1992, respectively. Since March 1979, he has been at the University of Technology, Košice as Professor for electronics and information technology. His research interests include digital signal processing and fiber optics, communication and sensing.

Ľuboš Ovseník (Ing, PhD.) was born in Považská Bystrica, Slovakia, in 1965. He received his Ing. (MSc.) degree in 1990 from the Faculty of Electrical Engineering and Informatics of University of Technology in Košice. He received PhD. degree in electronics from University of Technology, Košice, Slovakia, in 2002. Since February 1997, he has been at the University of Technology, Košice as Assistant professor for electronics and information technology. His general research interests include optoelectronic, digital signal processing, photonics, fiber optic communications and fiber optic sensors.

Ján Turán Jr. (Ing) was born in Košice, Slovakia. He received his Ing. (MSc.) degree in computer engineering in 1999 from the Faculty of Electrical Engineering and Informatics of University of Technology in Košice. He works in 3D People gmbh as research manager. His research interests include digital signal and image processing and computer games design.

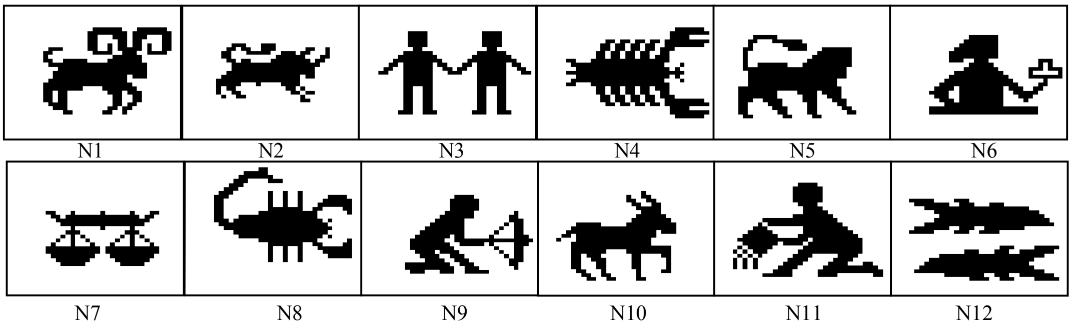

Fig. 5 Nativity Symbols used in experiments

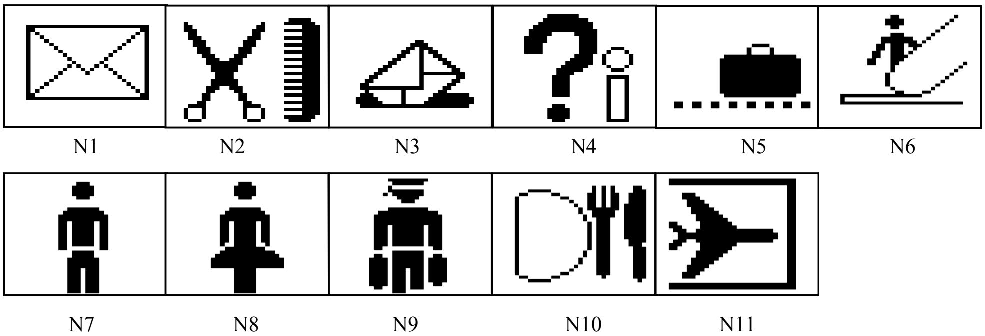

Fig. 6 Informative Symbols used in experiments

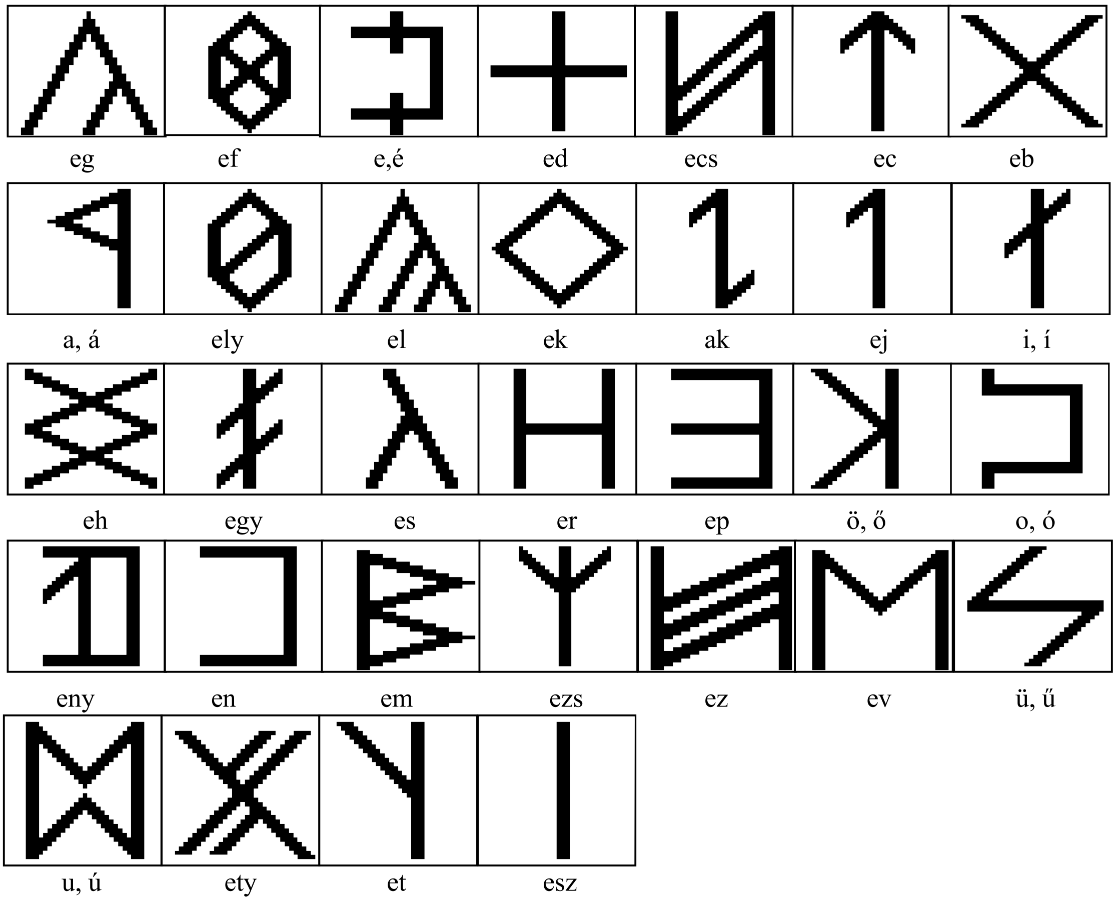

Fig. 7 Cuneiform Writings used in experiments