Published in the Lecture Notes in Artificial Intelligence - Vol. 3171 - XVII Brazilian Symposium on Artificial Intelligence - SBIA'04, São Luis, Sept. 29, 2004

An Experiment on Handshape Sign Recognition using Adaptive Technology: Preliminary Results

Hemerson Pistori[1] and João José Neto[2]

[1] Universidade Católica Dom Bosco, Dept. Engenharia de Computação, 79117-900 Campo Grande, Brasil
pistori@ec.ucdb.br
[2] Universidade de São Paulo, Escola Politécnica, 05508-900 São Paulo, Brasil
joao.jose@poli.usp.br

Abstract. This paper presents an overview of current work on the recognition of sign language and a prototype of a simple editor for a small subset of the Brazilian Sign Language, LIBRAS. Handshape based alphabetical signs, are captured by a single digital camera, processed on-line by using computational vision techniques and converted to the corresponding Latin letter. The development of such prototype employed a machine-learning technique, based on automata theory and adaptive devices. This technique represents a new approach to be used in the far more complex problem of full LIBRAS recognition. As it happens with spoken languages, sign languages are not universal. They vary a lot from country to country, and in spite of the existence of many works in American Sign Language (ASL), the automatic recognition of Brazilian Sign Language has not been extensively studied. ...

1 Introduction

Brazilian census, conducted by IBGE [3] in 2000, reveals an absolute number of 166.000 deaf persons. The most used sign language in Brazil is LIBRAS, which has been officially recognized as a legal communication mean just in 2002 [4]. This same law imposes that teachers, special educators and speech therapists be trained for using LIBRAS. The delay in the recognition of brazilian deaf sign language reflects intense and long dated quarrels between oralists and gestualists. Oralists claim that deaf persons should learn the national spoken language, while gestualists defend a bi-linguistic approach. The dominance of the oralist approach until recently (80's) led to extreme measures, as the complete prohibition of sign language in children education [1]. However, two main factors are gradually changing this scenario toward a bi-linguistic approach: (1) the growing acceptance of the richness and importance of the deaf culture, which includes

[3] Instituto Brasileiro de Geografia e Estatística (Brazilian Institute of Geography and Statistics)
[4] Brazilian Federal Law number 10.436 - April 24, 2002

their natural sign language, and (2) scientific discoveries in psychology, indicating that sign communication is essential for early language acquisition and complete development of deaf children [2].

Natural sign languages are not mere transcription of spoken languages. Instead, they are evolving systems, as expressive as any other language, encompassing complex lexical, syntactical and semantical substructures, with specificities absent in oral languages, like spatiality and iconicity [1]. However, as it happens with spoken languages, natural sign languages are much more comfortable, flexible and expressive than written languages. Deaf persons communicate by signs much faster than by writing or typing [3]. Hence, all justifications for researches in computer speech technologies also applies to sign languages, including text-to-sign and sign-to-text translators and human-computer interaction based on sign languages. Besides, as sign languages are not universal - they vary from country to country, sometimes comprising many different dialects inside the same country - sign languages translators are also important research goals to be pursued: an estimate of the number of sign languages around the world ranges from 4000 to 20000 [4].

The specific features of gestural languages turns it impossible to fully reuse algorithms and methods developed for the speech recognition domain. That is seldom recognized by computer scientists and engineers who neglect the cooperation of deaf communities and specialists in this kind of work [1]. The present work is just a first step toward the construction of a LIBRAS-to-Portuguese text converter, which will be used, in the future, as a front-end for applications such as a LIBRAS-to-Speech translator, or to LIBRAS-controlled interfaces. It comprises a simple prototype editor which may be trained to understand alphabetical hand signs corresponding to Latin letters, captured by a single digital camera, and a software framework allowing the integration of different machine learning and computer-vision techniques. Such editor is intended to be used both as a testbed for different techniques and as a dataset capturing tool, where body signs and corresponding portuguese text will be collected to form a huge database of samples, similar to Purdue's RVL-SLLL ASL database for automatic recognition of American Sign Language [5].

Our editor is already being used experimentally on a novel machine learning approach, based on adaptive decision trees (AdapTree [6]), which has as one of its main advantage the fact of being incremental, allowing the dynamic correction of errors even after the training phase (by the user himself), unlike many classic approaches, for instance, traditional feed-forward artificial neural networks with backpropagation.

The next section presents some previous work on the automatic recognition of sign language, followed by a brief introduction to the brazilian sign language. Section 4 introduces the adaptive technology and the machine learning approach used in our work. The prototype and some related experiments are shown and commented in sections 5 and 6. Results analysis, conclusions and future works are discussed in the last section.

2 Related Work

Despite some recent work on the recognition of head movements, like nods and shakes [7], which are very important in sign communication, most of works concentrates on hand movement and shape tracking. Besides being an important feature for sign language recognizers, hand tracking has important applications in human-computer interaction, virtual reality and sign compression. Applications include hand-controlled computer games [8], TV-set control by hand movements [8], video-conference compression techniques for deaf communication [3] and pointing devices that replace the traditional mouse.

The two major classes of hand tracking systems are: (1) data-glove-based, which rely on electromechanical gloves for virtual reality, comprising several sensors for detecting the position of wrists, hands, fingertips and articulations [9] and (2) vision-based, that works on continuous frames of digital images containing the hands [10–13]. The first group usually provides faster and more accurate systems, but the non-intrusive vision-based approach experiments growing interest with recent advances and larger availability of digital cameras, powerful micro-processors and computational vision techniques. Vision-based techniques may also benefit from the use of hand marks (such as colored gloves) [14] and infra-red cameras, which facilitate the identification of human-body parts with noisy backgrounds, but in some degree, at the expense of lower non-obstructiveness and hardware availability.

Works on hand-tracking which are focused on sign language recognition are also frequent, and are available for several different languages such as Australian, Chinese, German, Arabic and American Sign Language [9, 12]. Most of current sign language recognition systems typically include four modules: (1) hands segmentation, (2) parameter extraction, (3) posture recognition and (4) dynamic gesture recognition.

At the segmentation phase, hands are extracted from the background. This task may be very hard if bare hands are required with complex background, but techniques based on luminescence-invariant skin color detection [15], pre-stored background images and subsequent image differencing are showing promising results [11]. Parameter extraction reduces the size of the searching model by detecting interesting features in the segmented hand, such as position and relative angle of fingers and fingertips, hand center and direction, vectors from the hand center to the board, image moments and orientation histograms [16, 12, 14].

Posture and gesture recognition involves searching for the sign model which best fits the extracted parameters. Gesture models must deal with temporal information and much work has been done in the adaptation of speech recognition techniques, based on Hidden Markov Chains, to such problems [9, 15]. Machine learning techniques are also being used both in posture- and gesture-recognition, including artificial neural-networks and neuro-fuzzy networks [12]. Some other recognition techniques include principal component analysis [17], elastic graph matching and Kalman filters [13].

3 Brazilian Sign Language

(*a*) (*b*) (*c*)

Fig. 1. Examples of Iconic Signs used in LIBRAS
(a) Home (b) Small (c) Keep Silent

The Brazilian Sign Language, LIBRAS, is a complex, structured and natural Brazilian language, whose origin is found in the French Sign Language [18]. As it happens with spoken languages, sign languages naturally evolve to take advantage from its basic communication mean, the relation between space and body gestures in the case of sign languages, in order to make communication efficient. For instance, different positions in the space in front of a LIBRAS user may be assigned to different subjects of discourse, which may be referenced afterward, simply by pointing operations. Another feature of LIBRAS, important for its visual nature, is the use of symbols with iconic meaning: figure 1 shows LIBRAS gestures used to represent *home*, *small* and *keep silent*, respectively.

Though full LIBRAS communication includes head, main body, arms and movements of other body parties; hand shapes, position (with respect to the body of the signer) and movement are essential to LIBRAS, as they are used as basic elements from which more complex sentences may be constructed. A set of 46 basic hands shapes, or configurations, are found in LIBRAS. This set includes the 19 alphabetic "static" symbols presented in figure 2 (signs for letters *h,j,k,x,y* and *z* involve hand movements and are not shown). In LIBRAS, fingerspelling, the use of alphabetic sequences to form words is restricted to special cases, such as the communication of acronyms and proper nouns. However, alphabetic symbols appear as components in a variety of other signs. The sentence *what is your name*, for instance, is expressed by doing the sign for the *q* letter (referring to the interrogation *what* - "qual", in portuguese) near the mouth followed by an horizontal hand movement, shaping the letter *n* (referring to *name* - "nome"), from left to right, near the chest. The later example also illustrates the importance of the position and movement of the hands. More information on LIBRAS, including a LIBRAS-English dictionary, may be found in [18].

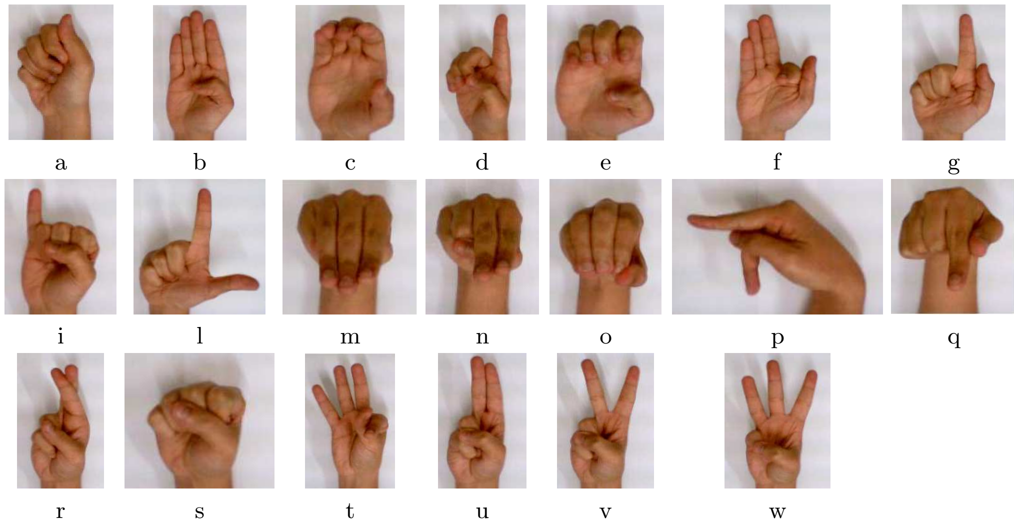

Fig. 2. Some LIBRAS alphabetic symbols

4 Adaptive Devices

The prototype editor developed in this work includes a module that learns to recognize certain hand signs by analyzing a set of examples previously interpreted by a human. Even during execution, eventual system recognition errors may be corrected by the user, simple by typing the letter and showing the corresponding hand sign. The ability of the system to evolve its recognition capability by dynamically changing its internal structure when new examples arrive is modeled, in the current work, using adaptive device theory. It is out of the scope of this paper to detail the full complexity of this theory but the next paragraphs give a brief introduction to the topic, referencing some works where more information may be found.

A rule-driven adaptive device [19] is a formal tool obtained by empowering some device whose operation relies on static rules, such as finite state automata, with an adaptive layer which transforms the former device's rules into dynamic ones. Such adaptive layer, which preserves much of the syntax and semantics of the subjacent mechanism, is based on simple deletion and insertion actions that operate on the subjacent mechanism's set of rules and is activated while the subjacent mechanism operates. In this way, the initially static structure of the subjacent device becomes dynamic, although the whole device is essentially expressed in terms of the subjacent formalism.

Figure 3 shows an adaptive automaton that recognizes the context-dependent language $a^n b^n c^n$. The subjacent mechanism, figure 3.(a), follows the standard representation for finite-state automata, with the exception of the $[.F]$ label attached to the transition consuming the symbol a. This label indicates that an adaptive function, called F (figure 3.(b)), is expected to be executed just after performing the transition to which it is attached. Adaptive functions are described using a notation inherited from semantic nets, with variables being denoted by a question mark prefix ($?x$, $?y$ and $?z$). The asterisk prefix indicates generators: new, unique, symbols that should be reinstantiated each time the adaptive function is executed ($*n_1$ and $*n_2$). Vertical parallel arrows give the function execution direction, from the pattern that should be matched and extracted to the pattern that should be included into the subjacent structure. The illustrating example works as follows: each time some symbol a is consumed, the adaptive function F finds the single empty transition in the automaton, removes it and inserts two new transitions for consuming the substring bc, with an empty transition strategically placed between them.

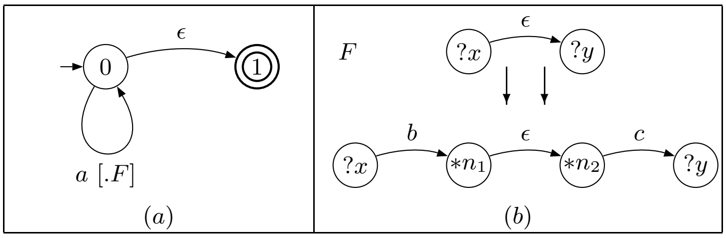

Fig. 3. Adaptive Automaton for $a^n b^n c^n$
(a) Subjacent Mechanism (b) Adaptive Layer

The adaptive technique is already being applied in the solution of problems in areas as diverse as grammatical inference, automatic music composition, natural language processing, robotics and computer vision [20, 19] with subjacent mechanisms that includes context-free grammars, structured pushdown automata, Markov chains, decision tables, Petri-nets and decision trees [5]. In the work reported in the present paper, adaptive decision trees, which are basically an adaptive automata that grows on a tree-like structure, have been used in the implementation of our machine learning strategy. The main features of adaptive decision trees, which elect it as a good alternative are: (1) efficient learning phase, even with large training sets and (2) incremental learning capabilities [6].

5 Development

We have implemented a prototype fingerspelling LIBRAS editor in Java by integrating to our algorithms three auxiliary software packages: an image processing

[5] Extensive information on adaptive technology may be found at http://www.pcs.usp.br/~lta. A free graphical IDE for experimenting adaptive automata may be downloaded from the same site.

environment, the ImageJ [6]; SUN library for time-based devices, the Java Media Framework (JMF); and a toolkit that support the development and confrontation of different machine learning and data-mining algorithms, WEKA [7]. The choice of reusing existing packages, all of them freely available from Internet, besides facilitating future maintenance and integration to other systems, allowed a very short and cheap development schedule for the whole environment.

Our prototype works as follows: initially, there is a training phase, when users must hand-sign letters to a camera, with one hand, and use the other-hand to type the same letter. Training image capture keeps running until a second click is issued. This procedure is then repeated, several times, for all desired letters: usually, the more examples are input, the higher the achieved accuracy. After a special reserved key is pressed, the system enters the next phase, where an adaptive decision tree is induced (learned), and the user may start the finger-spelling section, in which no typing is needed. The learning module may be recalled whenever a wrong guess is detected by the user (the system types a wrong symbol - different from that which is being signed), just by typing (two times - one to start and other to stop collecting image frames) and simultaneously signing the correct symbol. The ability of the system to incorporate new training instances and improve its performance during its operation phase is a differentiated feature of our incremental learning approach.

Besides the simple editor and the learning module, the prototype comprises a further digital image processing unit that pre-processes the image and extracts some attributes. The first image processing step involves the application of a Sobel edge detector. More sophisticated edge detectors, like Canny's, has also being tried, but limitations, due to real-time processing requirements, lead us to choose Sobel. Next, the image is binarized by using the iterative threshold technique, proposed by Riddler and Calvard [21]. An homogeneous, white, background (as in figure 2) and a fixed hand position were provided so that handtracking was not the issue in the current work.

Parameters extraction is based on image moments [16] with a novel approach that consists in dividing the image in equal-sized rectangular regions, and iteratively calculating the image moments for each region. This strategy increases the parameter set with some local information, which may be important for discriminating some handshapes. Figure 4 illustrate 12 parameters that our method extracts from three different hand signs. The grid, in red, is placed in the center of mass of the image and a new center of mass (red rectangle), calculated on the points inside the grid, delivers the first two parameters (x and y coordinates of the center). The next parameters capture the direction of the hand (illustrated as green arrows) and may easily be calculated as the standard deviation of the points with respect to the center of mass. The blue rectangles complete the parameter set, and correspond to the center of mass of each grid region.

[6] ImageJ is available at http://rsb.info.nih.gov/ij/
[7] WEKA is available at http://www.cs.waikato.ac.nz/~ml/WEKA/

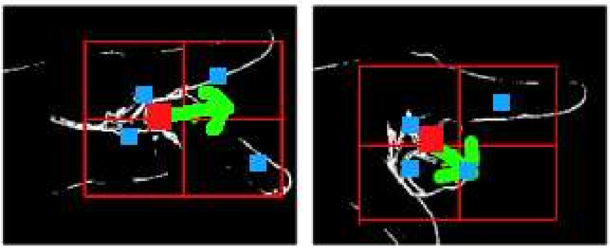

Fig. 4. Parameters extracted from two different handshapes

6 Experiments

For a first test of the learning algorithm performance, we collected a few images from 9 different signs and created some datasets in WEKA's file format (arff). Using the WEKA benchmark utilities, our machine learning algorithm, AdapTree, has been compared to Quinlan's C4.5, an state-of-art decision tree learning algorithm. Table **??** presents the ratio of correctly classified instances (RCC) for both algorithms, using WEKA's random split test, with a 66% cut, for a dataset containing 270 instances (30 for each sign). This test randomly chooses 66% of the dataset instances for training the learning scheme and applies the learned model to the remaining 34% instances. Table **??** also shows the average training time. These experiments indicates that the performance of AdapTree and C4.5 is not significantly different, in a non-incremental environment.

Table 1. Comparative Results

	AdapTree	C4.5
Ratio of correctly classified instances	95.02%	95.23%
Learning Time	0.25s	0.29s

The graphic in figure 5 shows the cumulative training time when a set of 10 new training instances are presented to each of the compared algorithms. In contrast to C4.5, AdapTree does not need to rebuild the tree, from scratch, each time new instances are obtained. The superiority of AdapTree over C4.5, regarding incremental learning, becomes evident in the graphic.

7 Conclusion

In this work we presented a prototype for a fingerspelling LIBRAS editor, an innovative way for parameter extraction based on the iterative calculation of image moments and a new machine learning approach. Such learning schema was chosen because this work is part of a larger project that aims to develop incremental

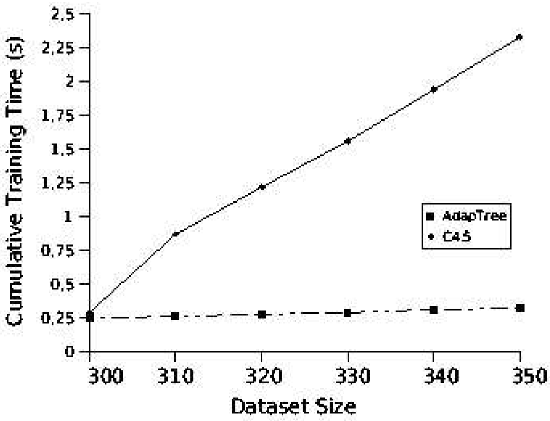

Fig. 5. Cumulative training time for an increment of 10 instances

learning approaches, mixing symbolic and sub-symbolic learning strategies. The current solution still lacks some robustness in face of different context training instances, although, it has the advantage of being incremental allowing continuous training and adjustments. An indirect result of this project was the integration of different free-software packages into a single framework for the application of machine learning techniques into machine vision tasks, involving time-based devices (like webcams). Changing the learning algorithm is very easy, due to the integration with WEKA. A prototype based on artificial neural networks, for instance, may be easily achieved, using WEKA's implementation.

The prototype should now be enhanced with a handtracking phase, possibly based on skin-color and image differencing [11]. The next version of the prototype will be developed with the help of deaf specialists in order to determine a representative LIBRAS dataset of training instances to be captured from different signalers, and in several environment conditions, in order to form a huge database for the brazilian sign language, similar to the one already available for the american sign language [5].

References

1. Braffort, A.: Research on computer science and sign language: Ethical aspects. In: Gesture Workshop. (2001) 1–8
2. Schirmer, B.: Language and literacy development in children who are deaf. Macmillan Publishing Co, New York, NY (1994)
3. Schumeyer, R., Heredia, E., Barner, K.: Region of interest priority coding for sign language videoconferencing. In: IEEE First Workshop on Multimedia Signal Processing, Princeton (1997) 531–536

4. Woll, B., Sutton-Spence, R., Elton, F.: Multilingualism - the global approach to sign languages. In: The Sociolinguistics of Sign Languages. Cambridge University Press, Cambridge, UK (2001)
5. Martnez, A., Wilbur, R., Shay, R., Kak, A.: Purdue RVL-SLLL ASL database for automatic recognition of american sign language. In: Proc. IEEE International Conference on Multimodal Interfaces. (2002)
6. Pistori, H., Neto, J.J.: Decision tree induction using adaptive FSA (to appear). CLEI Electronic Journal (2004)
7. Erdem, U.M., Sclaroff, S.: Automatic detection of relevant head gestures in american sign language communication. In: Proceedings of the International Conference on Pattern Recognition - ICPR 2002, Qubec, Canada (2002)
8. Freeman, W.T., Tanaka, K., Ohta, J., Kyuma, K.: Computer vision for computer games. In: 2nd International Conference on Automatic Face and Gesture Recognition, Killington, VT, USA (1996) 100–105
9. Fang, G., et al: Signer-independent continuous sign language recognition based on SRN/HMM. In: Gesture Workshop. (2001) 76–85
10. L.Bretzner, I.Laptev, T.Lindeberg: Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. In: Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition (FGR'02), Washington, USA (2002) 423–428
11. Martin, J., Devin, V., Crowley, J.: Active hand tracking. In: In IEEE Third International Conference on Automatic Face and Gesture Recognition, FG '98, Nara, Japan (1998)
12. Al-Jarrah, O., Halawani, A.: Recognition of gestures in arabic sign language using neuro-fuzzy systems. Artificial Intelligence **133** (2001) 117–138
13. Stenger, B., Mendona, P.S., Cipolla, R.: Model-based hand tracking using an unscented kalman filter. In: Proc. British Machine Vision Conference. Volume I., Manchester, UK (2001) 63–72
14. Davis, J., Shah, M.: Recognizing hand gestures. In: Proceedgins of the Third European Conference on Computer Vision (ECCV), Stockholm, Sweden (1994) 331–340
15. Kapuscinski, T., Wysocki: Hand gesture recognition for man-machine interaction. In: Proceedings of the Second International Workshop on Robot Motion and Control, Bukowy Dworek, Poland (2001) 91–96
16. Freeman, W.T., Anderson, D.B., Beardsley, P.A., Dodge, C.N., Roth, M., Weissman, C.D., Yerazunis, W.S., Kage, H., Kyuma, K., Miyake, Y., ichi Tanaka, K.: Computer vision for interactive computer graphics. IEEE Computer Graphics and Applications **18** (1998) 42–53
17. Martin, J., Crowley, J.L.: An appearance-based approach to gesture recognition. In: Proc. of 9th Int. Conf. on Image Analysis and Processing, Florence, Italy (1997)
18. Capovilla, F.C., Raphael, W.D.: Dicionário Enciclopédico Ilustrado Trilíngue da Língua de Sinais Brasileira. Editora da Universidade de São Paulo, São Paulo, Brasil (2001)
19. Neto, J.J.: Adaptative rule-driven devices - general formulation anda case study. In: CIAA'2001 Sixth International Conference on Implementation and Application of Automata, Pretoria, South Africa (2001) 234–250
20. Costa, E.R., Hirakawa, A.R., Neto, J.J.: An adaptive alternative for syntactic pattern recognition. In: Proceeding of 3rd International Symposium on Robotics and Automation, ISRA, Toluca, Mexico (2002) 409–413
21. Ridler, T.W., Calvard, S.: Picture thresholding using an iterative selection method. IEEE transactions on Systems, Man and Cybernetics (1978)