Hierarchical Interpretation of Human Activities Using Competitive Learning

Harry Wechsler, Zoran Duric, Fayin Li
Department of Computer Science
George Mason University
Fairfax, VA 22030
{wechsler,zduric,fli}@cs.gmu.edu

Abstract

In this paper we describe a method of learning hierarchical representations for describing and recognizing gestures expressed as one and two arm movements using competitive learning methods. At the low end of the hierarchy, the atomic motions ("letters") corresponding to flow fields computed from successive color image frames are derived using Learning Vector Quantization (LVQ). At the next intermediate level, the atomic motions are clustered into actions ("words") using homogeneity criteria. The highest level combines actions into activities ("sentences") using proximity driven clustering. We demonstrate the feasibility and the robustness of our approach on real color-image sequences, each consisting of several hundred frames corresponding to dynamic one and two arm movements.

1. Introduction

Humans are non-rigid, articulated objects with many degrees of freedom. We are interested in describing and recognizing gestures expressed using human motions over various periods of time [3]. Such motions can be understood with the aid of a three-level descriptive hierarchy based on concepts of *atomic motions*, *actions*, and *activities* [2]. At the low end of the hierarchy, the atomic motions ("letters") corresponding to flow fields computed from successive color image frames are derived using Learning Vector Quantization (LVQ). At the next intermediate level, the atomic motions are clustered into actions ("words") using homogeneity criteria. The highest level combines actions into activities ("sentences") using proximity driven clustering. We refer to one arm activities as "simple" and to simultaneous two arm activities as "compound".

Learning has long been a central issue in the understanding of intelligence as it plays a fundamental role in regulating the balance between internal representations and external regularities. As "versatility, generalization, and scalability are desirable attributes in most vision systems, the only solution is to incorporate learning capabilities within the vision system" [7]. Gesture interpretation involves both motion analysis and pattern recognition, often referred to as the *where* and *what* problems. The motion analysis component, described in detail elsewhere, consists of moving arm(s) detection, motion estimation, and tracking. Normal flow is used to detect a moving arm automatically. Expectation Maximization (EM), uniform sampling, and a shortest path algorithm are used to estimate the boundary of the arm. An affine motion model is fit to the arm region using residual analysis and outlier rejection for robust parameter estimation. The estimated parameters are used for both predicting the location of the moving arm and encoding its motion. The novelty of our approach resides in the recognition component. It comes from learning gestural representations at different abstraction levels using competitive learning methods; this leads to a linguistic approach based upon signal to symbol mappings. In the remainder of this paper we describe our method and present experimental results.

2. Competitive Learning

It is widely accepted that automatic detection of moving objects, their accurate tracking, and interpretation and recognition of long image sequences remains very challenging. The challenge comes from the need to process raw image streams and convert them to *summative representations*. Note that "Signal to symbol integration and transformation is an old but difficult problem. It comes about because the world surrounding us is a mixture of continuous space time functions with discontinuities. Symbols not only provide nice abstractions for low-level strategies, but also allow us to move one level up the modeling hierarchy

1051-4651/02 $17.00 © 2002 IEEE

and observe the properties of the systems and their interactions between each other and their environment at a more macroscopic level. Symbolic representation mediates reasoning about the sequential and repetitive nature of various tasks" [1]. Our approach to this problem is based on competitive learning methods driven by clustering.

Learning Vector Quantization (LVQ) [4] first abstracts the affine motion parameters of the flow fields corresponding to atomic motions as morphological primitive units, i.e., *letters*. Clustering driven by homogeneity defines then the dictionary *word* entries corresponding to actions. Sequences of actions are recognized as *sentences* corresponding to activities. One arm activities are recognized as simple activities; synchronized two arm activities are recognized as compound activities. Similar to a phonotopic map [5], one can then parse ("read") and interpret ("recognize") arm movements along the "linguistic" trajectories they trace. The learning component implements a linguistic approach; it provides for computational efficiency, conceptual abstraction and hierarchical modeling, and robustness. The first few cycles of each moving arm color image sequence are used for training, while the remaining cycles are used for testing.

2.1. Parsing and Interpretation

Simple arm activities include *striking*, *grind*, *swing*, *stirring* (augmented by tool using), (*slow* or *fast*) *pounding* (see examples in Fig. 1); compound two arm activities include *in-phase striking*, and *opposite phase striking* (see Fig. 2). Video color-image sequences are represented using three hierarchical layers of abstraction in order to facilitate parsing and interpretation of arm movements. LVQ self-organizes and compresses the original input space spanned by the affine parameters of the flow fields using 13 (thirteen) symbols, i.e., "letters", each of them corresponding to an individual tile in the Voronoi tesselation. It can be shown using the inner product of the prototype vectors that the letters are quite dissimilar. Homogeneity driven clustering derives the next layer, which consists of 13 (thirteen) "words", corresponding to specific actions; the transitions ("edges") as segmentation ("punctuation") points between actions are grouped into the fourteenth "word". Examples of words include ("up"- "AAAAAAAAA"), ("down" - "BBBBBBBBB"), while examples of transitions include (BPO) and (FGL). Please note that some actions can map to more than one cluster. One can again show that the dictionary "word" entries are quite dissimilar.

The last representational layer encodes simple arm activities in terms of simple word sequences using proximity driven clustering, e.g., *pounding* consists of repeating cycles of *up* and *down* segments, *swirling* consists of repeating *circle* segments, *swing* consists of repeating *left* and *right* segments and so on. Examples of compound activities, *in-phase striking* and *opposite phase striking*, are encoded as repeating cycles of pairs (*up*, *up*) and (*down*, *down*), and (*up*, *down*) and (*down*, *up*), respectively.

Arm activities can now be recognized as follows : (a) derive the letter sequence using the prototypes found using LVQ; (b) derive the word sequence via flexible and robust matching between the sequence of letters and the action/transition clusters using as distance a properly weighted largest common letter subsequence and the nearest neighbor as the classification rule; (c) label the activity by matching the sequence of words against sentence prototypes learned earlier. We find that the sentences corresponding to different arm activities are quite dissimilar.

3. Experimental Results

We show here the feasibility of our method using the eight kinds of arm activities referred to in the previous section. Each sequence consists of several hundred (100 - 500) frames. These images were collected using a Sony DFW-VL500 progressive color scan camera; the frame rate was thirty frames per second and the resolution was 320×240 pixels per frame. The moving arm was detected and tracked automatically as was briefly described in Sect.1: Fig. 1 and 2 show 4 frames, randomly chosen, and the corresponding tracked contours for each kind of arm movement.

-0.00067	-0.0014	0.28	-0.0012	0.0026	-0.92	A
0.00064	0.0013	-0.17	0.00001	-0.0033	0.95	B
-0.016	-0.0052	-1.20	-0.0031	-0.0069	-0.070	E
-0.00023	-0.0023	-0.99	-0.0034	-0.0061	-0.25	F
0.0071	0.0016	-1.09	-0.0018	0.0084	0.37	G
0.0020	-0.00064	-0.97	0.0016	0.0010	-0.24	H
0.0016	-0.00082	-0.82	0.0032	0.00074	-0.58	I
-0.0049	-0.011	0.95	0.026	-0.0012	-0.46	K
-0.0081	-0.0062	1.02	-0.017	-0.017	0.050	L
-0.0016	0.00017	0.96	-0.00010	0.00007	0.25	M
0.0019	-0.00009	0.52	-0.0043	0.021	0.89	N
0.0022	-0.0040	-0.67	0.0013	0.00019	-0.74	O
0.00009	0.0070	0.54	-0.0095	0.00082	0.86	P

Table 1. The numerical values of affine parameters corresponding to the derived letters.

The input to LVQ consists of the affine motion parameters derived for one cycle (approx. 60 frames) for each of the eight kinds of arm movements. The output of LVQ yields thirteen prototypes/letters; the corresponding affine flows are shown in Fig. 3; the corresponding affine parameters are shown in Table 1. After the letters are derived

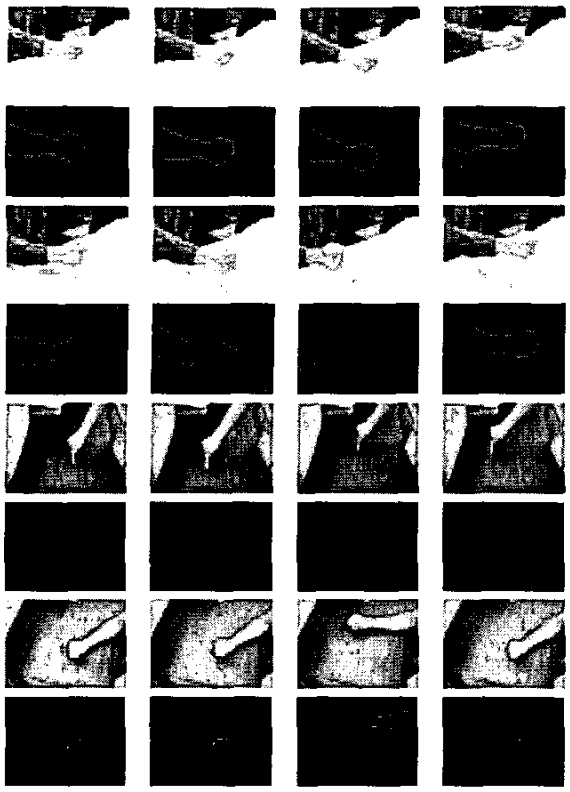

Figure 1. Frames from the "striking", "grind", "stirring", and "swing" arm movement sequences. Odd rows: selected color frames. Even rows: tracked contours.

by LVQ, each video frame is represented by a letter corresponding to the nearest motion prototype. Clustering then compresses the letter sequences into word sequences. The thirteen words (marked by numbers 0–12) correspond to letter sequences. These sequences can be of variable length. The words are $0 : A^*$, $1 : N^*$, $2 : B^*$, $3 : K^*$, $4 : L^*$, $5 : E^*$, $6 : G^*$, $7 : H^*$, $8 : P^*$, $9 : F^*$, $10 : I^*$, $11 : O^*$, and $12 : M^*$, where A^* stands for a sequence of A's possibly interspersed with a (very) few other letters.

Using this code, each color video image sequence is compressed by an order of magnitude. The results are shown in Fig. 4. The feasibility and the robustness of the proposed method is shown using long and inherently noisy image sequences corresponding to different arm activities, some of them possibly recorded at different speeds. Testing performed on the activity cycles not used during training yields perfect recognition accuracy - 100% – if one allows that slow and fast pounding are recognized as one activity.

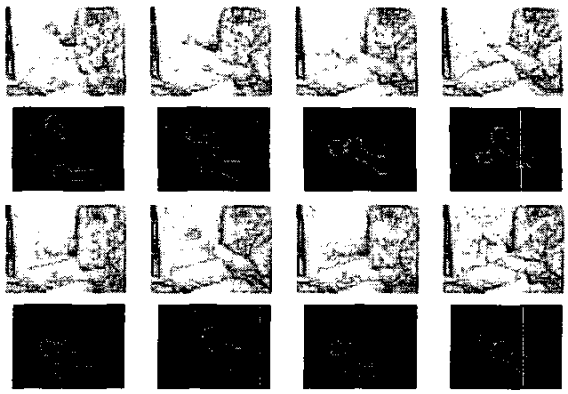

Figure 2. Frames from the compound activities sequences. Top rows: in-phase striking. Bottom rows: opposite phase striking. Odd rows correspond to selected color frames. Even rows correspond to tracked contours.

4. Conclusions

This paper describes a new method for the hierarchical interpretation of lower arm and hand movements from color video image sequences using a linguistic approach driven by competitive learning. The feasibility and the robustness of the proposed method has been shown using long and inherently noisy image sequences corresponding to different one and two arm activities recorded at different speeds.

The linguistic approach described in this paper can be easily expanded in terms of both alphabets and dictionary entries. A possible extension would include coupling arm and wielded tools movements. The stirring activity considered earlier as a part of our experimental data involved the use of one such tool. Our preliminary experiments suggest recognition of arm activities can be enhanced by coupling the parsing and interpretation processes using the motions traced by both the arm and the wielded tool.

References

[1] R. Bajcsy, R. and J. Kosecka The problem of signal and symbol integration: A study of cooperative mobile autonomous agent behavior *DAGM Symposium*, Bielefield, Germany, 1995

[2] A.F. Bobick. Movement, activity, and action: The role of knowledge in the perception of motion. *In Proc. Royal Society Workshop in Knowledge-based Vision in Man and Machine, London, England*, 1997.

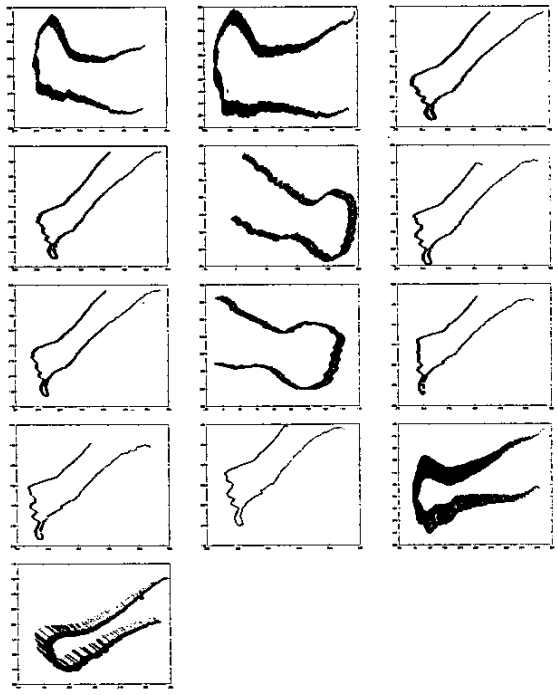

Figure 3. Affine flows corresponding to the thirteen letters/prototypes: A, B, E, F, G, H, I, K, L, M, N, O, and P.

[3] D.M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73:82–98, 1999.

[4] S. Haykin. *Neural Networks* (2nd ed). Prentice Hall, 1999.

[5] T. Kohonen. The "neural" phonetic typewriter. *Computer*, 21:11–24, 1988.

[6] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, 1995.

[7] S. Nayar and T. Poggio. Early visual learning. In *S. Nayar and T. Poggio (Eds.), Early Visual Learning*, Oxford University Press, 1996.

striking:
0[14]1[2]2[9]1[5]2[2]1[3]0[7]11[4]0[14]2[10]
1[11]0[8]11[4]0[10]1[2]2[11]1[6]2[6]0[13]11[2]
0[11]1[3]2[20]11[4]0[2]11[2]0[6]11[4]0[13]11[2]
2[16]1[5]2[4]1[4]0[2]11[2]0[8]11[4]0[14]2[13]
1[2]8[3]2[7]0[29]1[4]2[11]1[2]2[4]1[2]0[12]11[3]
0[17]1[2]2[13]1[3]2[3]1[2]0[2]

grinding:
3[4]4[3]5[3]6[11]7[2]3[15]11[3]3[5]4[2]6[9]7[6]
6[6]7[2]11[2]0[4]3[2]0[3]3[10]0[8]3[3]2[2]6[7]
5[3]6[12]7[2]5[2]

swing:
10[1]10[2]11[15]1[3]8[3]1[3]8[4]1[2]11[16]10[2]
1[3]8[9]11[17]8[2]1[4]8[7]1[2]11[15]0[4]1[2]8[7]
1[2]1[2]

stirring:
2[6]6[6]5[3]7[3]7[4]10[7]11[13]0[11]3[15]4[30]
8[2]1[7]2[11]6[3]2[2]6[10]7[6]10[10]11[9]0[11]
3[14]4[8]4[15]8[13]1[3]2[19]6[11]7[7]10[8]11[10]
0[13]3[7]4[13]4[10]8[3]1[8]2[8]6[12]5[2]7[5]7[3]
10[6]11[11]0[13]3[5]

slow pounding:
0[28]2[4]0[6]3[2]0[4]2[12]6[3]2[30]6[6]2[3]
0[56]2[4]6[2]2[58]6[3]0[42]3[2]0[48]2[57]

fast pounding:
0[42]2[27]6[3]8[2]0[50]6[4]2[29]6[3]0[21]3[6]
0[9]2[38]0[37]6[2]2[37]0[15]3[10]0[6]2[3]6[4]
2[24]0[18]3[4]

in-phase striking—left hand:
0[23]3[4]2[25]0[32]3[3]2[32]0[17]11[3]0[24]2[33]
11[5]0[11]

in-phase striking—right hand:
0[28]2[29]0[33]2[29]1[3]0[3]11[3]0[41]2[30]1[2]
6[2]2[4]0[11]

opposite phase striking—left hand:
2[11]1[3]11[6]0[30]2[35]1[2]1[3]11[5]0[3]11[5]
0[22]2[39]11[13]0[27]2[28]1[3]2[9]11[8]

opposite phase striking—right hand:
0[16]2[35]1[3]0[36]3[2]2[30]1[4]1[2]0[36]3[2]
2[31]1[9]0[38]2[9]

Figure 4. Words sequences corresponding to the arm activities. The numbers in square brackets correspond to the word length.