

A Biologically Inspired System for Human Posture Recognition

Shoushun Chen¹, Polina Akselrod² and Eugenio Culurciello²

¹ School of EEE, Nanyang Technological University

² Electrical Engineering Department, Yale University

Abstract—We present a biologically-motivated system to recognize human postures in realtime video sequences. The system employs event-based temporal difference image between video sequences as input and builds a network of bio-inspired Gabor-like filters to detect contours of the active object. The detected contours are organized into vectorial line segments. After feature extraction, a classifier based on simplified line segment Hausdorff distance combined with projection histograms is implemented to achieve size and position invariant recognition. 86% average recognition rate is achieved in the experiment. Compared to state-of-the art bio-inspired categorization methods shows great computational savings, and is an ideal candidate for hardware implementation with event-based circuits.

I. INTRODUCTION

Understanding the humans' actions from their postures is of great interest to several applications like personal health care, environmental awareness, intelligent visual human machine interface, video game systems, and human-robot interaction, just to name a few. Based on commercially available image sensors and powerful personal computers, an impressive series of research work has been reported for human posture categorization. In general, those approaches first detect moving objects by the analysis of video stream, then extract human silhouettes using background subtraction technique [1], [2]. Blob metrics are represented into multiple appearance models [3] and finally posture profiling is conducted based on frame-by-frame posture classification algorithms. Due to the complexity, these algorithms need to be implemented on powerful computers (1GHz processors or better), even when recognizing only a small subset of human body postures [4]. These requirements limit the use of these algorithms in real life applications with low-cost and lightweight wireless platforms, such as embedded computers, sensor networks or smart cellular phones. In addition to the complexity of the algorithms, the conventional frame based image sensors employed in these systems also contribute to lower energy efficiency. In fact, the output of conventional image sensors, as a matrix of pixel color values, contain a very high level of redundancy. Large amounts of unimportant data have to be read and processed before obtaining the features of interest. As a matter of fact, the first step of many computer vision algorithms is to remove the background and extract object edges or motion contours [5].

In this paper, we present an energy-efficient system which combines (1) a custom event-based temporal difference image sensors and (2) an efficient categorization algorithm based on

models of the human visual system. The sensor is able to reduce image redundancy at the sensor level [6], [7] and only report active motion in a series of events. The algorithm then filters individual motion events to extract a very limited number of line features. A modified Hausdorff distance classifier is then employed to measure the similarity of the features with those extracted from a small set of library objects. The goal of our research is to allow mobile platforms to perform sophisticated object discrimination tasks, and to extend this capability to cellular phones and embedded platforms for robotics. The proposed approach is innovative due to its high data-encoding efficiency, large saving in computation complexity as well as an efficient way to achieve robustness to translations and scale while recognizing objects. This is also the first address-event categorization algorithm that provides size and position invariance.

The paper is organized as follows: Section II introduces the system. Section III describes the proposed edge feature extraction algorithm, and Section IV describes the size and position invariant recognition classifier. Section V discusses the implementation and reports the experimental results. Section VI concludes the paper.

II. SYSTEM OVERVIEW

The block diagram of our recognition system is shown in Fig.1. We use a temporal difference image sensor named MotoTrigger [6], a bio-inspired feature extraction unit and a classifier with a set of library postures.

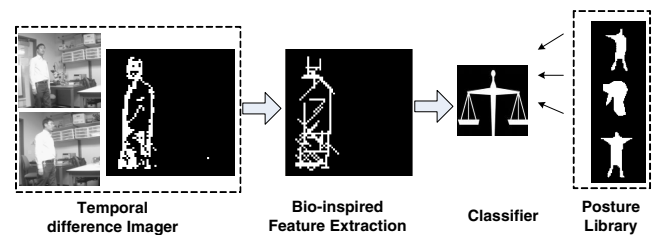


Fig. 1. Block diagram of the system. It is composed of an address-event temporal difference image sensor, a bio-inspired contour-based feature extraction algorithm, a classifier and a reference posture library.

The temporal difference image sensor compares two continuous image frames and only outputs addresses of those pixels whose illumination changes by an amount larger than a predefined threshold. If the scene illumination and object reflectance are constant, the changes in scene reflectance only

result from object movements or camera translation. The background information is thus filtered by the Mototrigger camera. The image sensor encodes the addresses of the active pixels into a stream of events and communicates through a protocol called Address Event Representation (AER) [8]. An 'address-event' refers to the image coordinates of a 'motion' pixel. The address events are sent in parallel to a battery of Gabor-like filters and convolution operation is performed on the fly. The responses of the filters are analogous to neurons in biological networks, where individual synapses deliver charge pulses to targeted neurons. These filters extract zero-crossing or line information from the image. After that, a MAX-like operation is conducted to find the maximal response among the 'neurons'. Only those who reach the maximal response can survive during the competition and each 'neuron' represents a vectorial contour segment in the image. The extracted line segments are fed to the classifier to measure the similarity of the input line segments with those of a set of library objects. The classifier is based on a modified Hausdorff distance scheme and is able to achieve size and position invariance.

III. BIO-INSPIRED FILTERS AND FEATURE EXTRACTION

Our feature extraction algorithm follows a recent model of object recognition in primate visual cortex [9]. As shown in Fig.2, an image is first processed by a network of simple filters 'S1' (after nomenclature in [9]). Each filter models a neuron cell with certain size of receptive field and responses best to basic feature at certain orientation. In the second stage, layer 'C1' combines all the outputs from 'S1' cells that have the same orientation and finds the maximal response (MAX) among them. A neuron cell which reaches the peak response stands for a feature (line or edge) at the same size and orientation as that neuron cell. This extraction procedure is also summarized as Algorithm 1.

Algorithm 1 Procedure for edge extraction

- (1) **S1:** for a giving image, apply filters at 4 orientations ($\theta = 0^\circ, 45^\circ, 90^\circ$ and 135°), and 6 scales ($s = 3, 5, 7, 9, 11, 13$) for a total of 24 neuron maps.
- (2) **Max cross neighborhood:** Within each map, each neuron will compare its own response to all other neurons that fall into its receptive field. If it does not peak, this neuron will be disabled. After this step, only the neurons that locate at the center of an edge can survive.
- (3) **Feature extraction cross scales:** Within each orientation, all the neurons that locate at the same position will be compared, only the neurons that best match the size of the feature will reach the maximum and can thus survive.

As a first step, simple cells are used to build object-selectivity. The temporal difference image is convoluted with a multidimensional array of simplified Gabor filters. We arrange the filters to have six different sizes, ranging from 3 to 13, and four orientations, i.e., $0^\circ, 45^\circ, 90^\circ$ and 135° . Therefore, the network of filters is able to detect features (transitions from black to white or vice versa) as short as 3 and as long as 13, at 4 orientations. The convolution result of each filter will be one

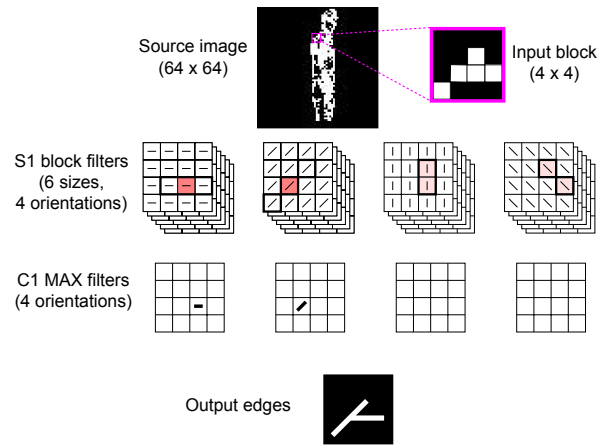


Fig. 2. Hierarchical organization of the feature extraction unit. The magenta square contains a zoomed-in part of the original image below. For the sake of clarity, the feature extraction is exemplified on this 4×4 subimage. It is first processed by a network of simple filters 'S1'. Each filter models a neuron cell with certain size of receptive field that responds best to the basic feature at certain orientation. The neurons of the same size and orientation are organized into 4×4 squares. The latter are shown as 4 piles (by orientation), each pile containing 6 different sizes. The neurons with maximal response among its neighbors are colored with red or pink. In the second stage, layer 'C1' combines the outputs from same orientation 'S1' cells whose response is maximal (red or pink) and sufficiently high (red). For example, the 3-pixel horizontal line gives one high peak, while the 2-pixel vertical line gives two low peaks. In the 'C1' layer above only the surviving neurons are shown. Thus the image is represented by two edges of size 3: one horizontal and one of 45° angle. The edges are visualized as thick white lines on the output image at the bottom.

matrix of neuron cells. Fig.3 illustrates a few maps of neurons for a test image. One can note that, if the size of the feature is larger than the filter size, i.e. the neuron receptive field size, a trapezoid-shaped response is obtained along the direction of the feature. In this case there is no single maximum of the function. When the size of the feature matches the neuron size, a triangle-shaped response is obtained, resulting also in a high peak response. If the size of the feature is smaller than the neuron size, either a low peak is observed or not a local maximum at all. Finding a single peak is thus indicative of what size filter best describes the feature detected.

The next step is to find out the location and the size of the feature. To find out the center of the line, a neighborhood MAX operation is performed [9], and each neuron compares its response to the one of the other neurons that fall within its receptive field. In our implementation, each neuron is built with a flag bit, to indicate whether or not this neuron can survive during competition with other neurons. A neuron will de-select itself from local competition by turning the flag bit to '0' if at least one of its neighbors has a higher response. The principle behind this choice is the following. A neuron has a higher response than its neighbor of the same size and orientation due to a better position. The MAX operation is performed only among neurons that have the same size and orientation. After the above procedure, another round of MAX operation will be performed to find out the size of the feature. This is done by comparing all the neurons at the same position. Only the one reporting the maximum response will survive as best descriptor of the size of the feature.

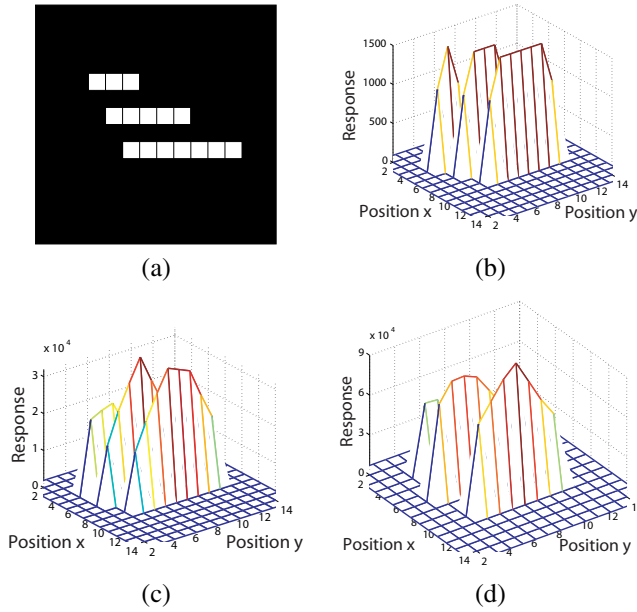


Fig. 3. Map of S1 neuron cells for a test image. This represents the output of the edge Gabor filters. (a) source image which consists of 3 horizontal lines, with lengths of 3, 5 and 7, respectively. (b-d) are neuron cells responses, implemented as convolution of the image with horizontally oriented Gabor filters of sizes 3, 5 and 7, respectively.

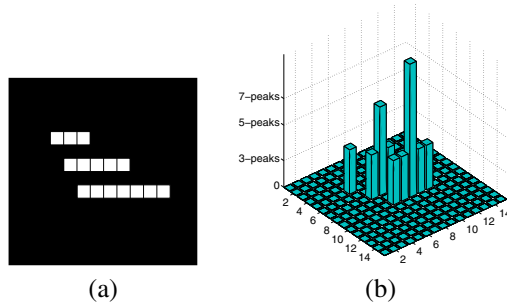


Fig. 4. Selection of most appropriate filter length based on the MAX operation. (a) The processed image, the same as on Fig.3(a). (b) Map of surviving neurons of Fig.3(b)-(d) after neighborhood MAX operation. The surviving neurons corresponding to the sizes 3, 5 and 7 are shown as low (3-peaks), medium (5-peaks) and high (7-peaks) bars, respectively. Not to scale.

Fig.4 shows the map of surviving neurons of Fig.3(b)-(d) after neighborhood MAX operation. Compared to the original map, one can note that only the sufficiently high peak neurons are left. When the size of the neuron exactly matches the size of the feature, only one neuron located at the center of feature will eventually survive, and it will mark the location of the feature.

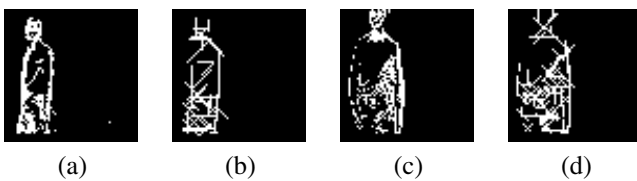


Fig. 5. Feature detection examples with real data. (a) and (c) are two source temporal difference images. (b) and (d) are the corresponding extracted edges.

Fig.5 shows the extraction result of two temporal difference images. In the source image, the outline of the human is composed by scattered pixels. While in the reconstructed image, the outline is replaced by a straight line that best estimates the feature.

IV. SIZE AND POSITION INVARIANT CLASSIFIER

The extracted line segments are fed to a classifier to measure the similarity of the input line segments with those of a set of library objects. Our classifier is based on Line Segment Hausdorff Distance [10] and is made capable of size and position invariance.

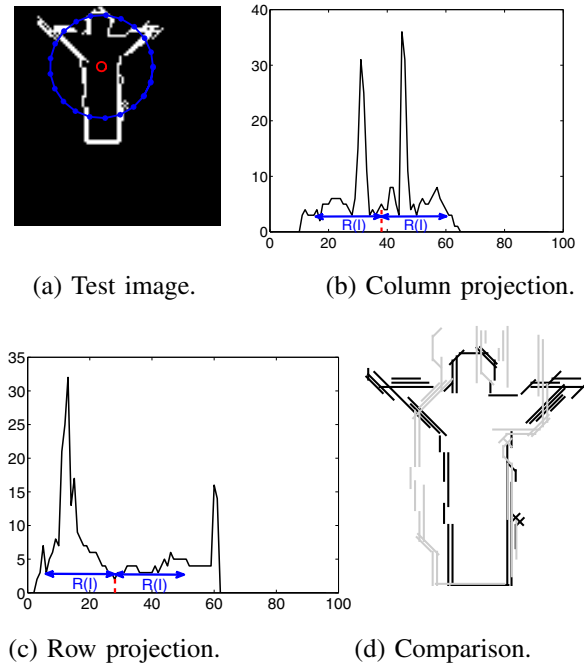


Fig. 6. Alignment and resizing. (a) shows the extracted edges of the test image. The small red circle is the center and the large blue dotted circle is the size. (b) and (c) show the column and row projection histogram, respectively. The dashed red line denotes the mean, and the blue arrows depict the image size. (d) is the graphical representation of the aligned and resized test image (black) with a library image (gray) during the classifying stage.

First, we propose to align two objects using their center position which is easily achievable in address-event because the centroid of a shape is the running average of the y-addresses and x-addresses. Secondly, to achieve size invariance, we propose to normalize the size of the test object and library object. Size information can be obtained from building Projection Histogram. The ideas are exemplified on Fig.6.

We note that our approach demonstrates a great implementation efficiency of the image scaling. For instance, to resize the centered image by a certain ratio α , we simply multiply by α with the coordinates of the edges. This is a built-in advantage of the vectorial feature representation. In conventional approaches scaling an image involves complex operations such as nearest-neighbor interpolations, super-sampling and resolution synthesis.

V. IMPLEMENTATION AND EXPERIMENTAL RESULTS

In this paper we reports on a C++ coded implementation (<http://www.eng.yale.edu/elab/research/svision/svision.html>). The algorithm was designed with the intention of being implemented into event based or address-event hardware. At present, there is no general-purpose hardware that can directly operate on address-events, as a micro-processor operates on digital data. Several groups have proposed some version of general hardware, such as IFAT [11] and CAVIAR project [12]. The work presented here can be thus implemented on both IFAT and CAVIAR hardware, with the appropriate extensions and modifications. Our long term goal is to make these platform converge with address-event algorithms. The proposed algorithm is able to achieve great computational saving, resulting from several novel techniques. First, the object of interest is directly obtained from the output of temporal difference image sensor without any image pre-processing. Only the active pixels are permitted to send address events ($\sim 25\%$ of overall pixels). Secondly, during the MAX operation of convolutional filtering, we found empirically only $\sim 11\%$ of neurons can survive after competition which implies less operands in the stage computation. Thirdly, the contour of the object is decomposed into a very limited number of line segments (~ 60 lines per image). Finally, size and position invariance is an integral part of our approach and no additional preprocessing is needed.







	Bend	Hand1	Hand2
			
No. Images	317	262	236
Success Rate	91%	68%	87%
	Stand	Squat	Swing Hand
			
No. Images	283	327	288
Success Rate	95%	88%	73%

TABLE I

EXPERIMENTAL RESULTS FOR IMAGES TAKEN BY A WEB CAMERA.

To evaluate the system classification performance, we have run the recognition algorithm with a set of human postures. We first built libraries by choosing a number of representative images for each human posture. We extracted the edges of every such image and stored them as library of features. Next, we compared each image in the data base to each image in the library. We have used both a standard web camera and our custom event-based temporal difference image sensor as input video device. Table I and Table II report the successful matches that the algorithm yielded in the two tests, respectively.

VI. CONCLUSION

This paper reports a size and position invariant human posture recognition algorithm. The image is first acquired






	Bend	Hand1	Hand2
			
No. Images	61	61	61
Success Rate	97%	90%	85%
	Stand	Swing Hand	
			
No. Images	61	61	
Success Rate	98%	79%	

TABLE II

EXPERIMENTAL RESULTS FOR IMAGES TAKEN BY OUR TEMPORAL DIFFERENCE IMAGE SENSOR.

using an address event temporal difference image sensor and followed by a bio-inspired hierarchical edge extraction unit. A simplified Line Segment Hausdorff distance scheme is employed for similarity measurement while size and position invariance is achieved by deriving size and position information from projection histograms. The proposed algorithm achieves up to 86% average recognition rate.

VII. ACKNOWLEDGEMENTS

This project was funded in part by NSF award 0622133.

REFERENCES

- [1] K. Takahashi, T. Sakaguchi, and J. Ohya, "Remarks on real-time human posture estimation from silhouette image using neural network," in *IEEE International Conference on Systems Man and Cybernetics*, Oct. 2004, pp. 370–375.
- [2] E. H-Jaraha, C. O-Urunuela, and J. Senar, "Detected motion classification with a doublebackground and a neighborhood-based difference," *Pattern Recognition Letters*, vol. 24, pp. 2079–2092, 2003.
- [3] L. H. W. Aloysius, G. Dong, H. Zhiyong, and T. Tan, "Human posture recognition in video sequence using pseudo 2-d hidden markov models," in *8th Control, Automation, Robotics and Vision Conference*, Dec. 2004, pp. 712–716.
- [4] P. Spagnolo, M. Leo, A. Leone, G. Attolico, and A. Distanti, "Posture estimation in visual surveillance of archaeological sites," in *IEEE Conference on Advanced Video and Signal Based Surveillance*, July 2003, pp. 277–283.
- [5] J. Triesch and C. von der Malsburg, "A system for person-independent hand posture recognition against complex backgrounds," in *IEEE TPAMI*, vol. 23, Dec. 2001, pp. 1449–1453.
- [6] Z. Fu and E. Culurciello, "Fall detection using an address-event temporal contrast vision sensor," in *ISCAS*, May 2008.
- [7] P. Lichtsteiner and T. Delbruck, "A 128×128 120db $15\mu s$ latency asynchronous temporal contrast vision sensor," *IEEE JSSC*, pp. 566–576, Feb. 2008.
- [8] E. Culurciello, R. Etienne-Cummings, and K. Boahen, "Arbitrated address event representation digital image sensor," in *ISSCC. 2001.* IEEE, 2001, pp. 92 – 93.
- [9] T. Serre, "Learning a dictionary of shape-components in visual cortex: Comparison with neurons, humans and machines," Ph.D. dissertation, MIT, Apr. 2006.
- [10] S. Chen, B. Martini, and E. Culurciello, "A bio-inspired event-based size and position invariant human posture recognition algorithm," *ISCAS*, vol. 0, no. 0, May 2009.
- [11] R. Vogelstein, U. Mallik, E. Culurciello, R. Etienne-Cummings, and G. Cauwenberghs, "Saliency-driven image acuity modulation on a reconfigurable silicon array of spiking neurons," in *Adv. Neural Information Processing Systems NIPS '04*, vol. 17. MIT Press, December 2004.
- [12] R. Serrano-Gotarredona, T. Serrano-Gotarredona, A. Acosta-Jimenez, and B. Linares-Barranco, "A neuromorphic cortical-layer microchip for spike-based event processing vision systems," *IEEE TCAS-I*, vol. 53, pp. 2548–2566, Dec. 2006.