Arm Gesture Detection in a Classroom Environment

Jie Yao and Jeremy R. Cooperstock
Centre for Intelligent Machines, McGill University
3480 University Street, Montreal, QC H3A 2A7
{yaojie|jer}@cim.mcgill.ca

Abstract

Detecting human arm motion in a typical classroom environment is a challenging task due to the noisy and highly dynamic background, varying light conditions, as well as the small size and multiple number of possible matched objects. A robust vision system that can detect events of students' hands being raised for asking questions is described. This system is intended to support the collaborative demands of distributed classroom lecturing and further serve as a test case for real-time gesture recognition vision systems. Various techniques including temporal and spatial segmentation, skin color identification, as well as shape and feature analysis are investigated and discussed. Limitations and problems are also analyzed and testing results are illustrated.

1 Introduction

An architectural overview of our hand-raising recognition system is presented below in Figure 1. The system is designed to detect events based on the assumptions of no camera motion and a known subject with relatively well defined motion patterns, i.e. upward arm movements. These assumptions allow us to exploit prior knowledge about the attributes of subjects, such as typical arm motion speed and color information.

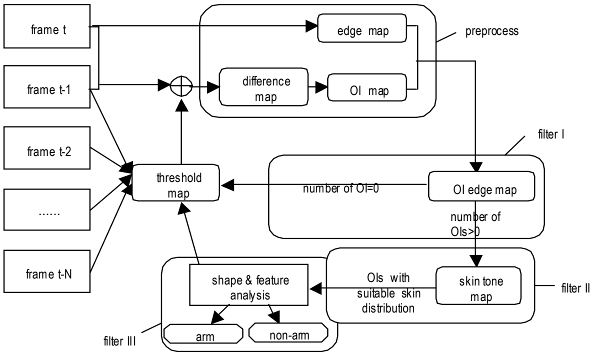


Figure 1. Architectural Overview

Motion information is extracted by temporal differencing. The difference map at time *t* is calculated by subtracting the reference frame *t-1* from the current frame *t* and then thresholding the subtraction result using a threshold map, which is updated using the previous *N* frames in the sequence. Interesting objects are located based on this inter-frame data in an object of interest (OI) map. The current OI map and the edge map of frame *t* are logically combined and stored into an OI edge map, which is then segmented in order to identify likely candidates. Provided a non-zero number of candidates remain from the OI edge map, the system further refines the set through analysis of skin distribution, geometrical extent ratio, horizontal span deviation, shape similarity energy and other features. Finally, the threshold map is updated again using the information obtained from the earlier analysis steps.

Figure 2 illustrates a typical classroom scene in which the rows of seats are aligned on a slight slope from the horizontal plane.

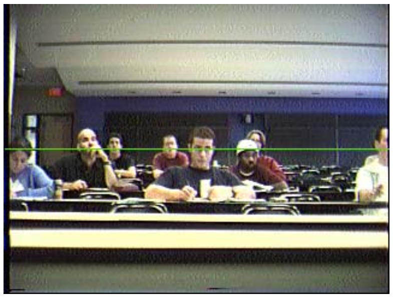


Figure 2. Typical classroom scenario with horizontal reference line drawn to indicate approximate expected position of students' heads.

It can be assumed that the students' heads are randomly distributed and lie approximately on the same horizontal plane, indicated by the reference line in the figure. By placing the camera at the front of the classroom, with a center of focus aligned parallel to this plane, we can confine the analysis, in general, to those parts of the image above the students' heads.

Motion information is extracted by temporal differencing. The difference map at time *t* is calculated by subtracting the reference frame *t-1* from the current frame *t* and then thresholding the subtraction result using a threshold map, which is updated using the previous *N* frames in the sequence. Interesting objects are located based on this inter-frame data in an object of interest (OI) map. The current OI map and the edge map of frame *t* are logically combined and stored into an OI edge map, which

is then segmented in order to identify likely candidates. Provided a non-zero number of candidates remain from the OI edge map, the system further refines the set through analysis of skin distribution, geometrical extent ratio, horizontal span deviation, shape similarity energy and other features.

Finally, the threshold map is updated again using the information obtained from the earlier analysis steps.

The remainder of this paper describes the image processing approach of our system in greater detail. Section 2 discusses temporal and spatial techniques used to extract motion information. Section 3 investigates various edge detection methods and introduces the OI edge map construction algorithm. Section 4 deals with color analysis for skin tone distribution. Algorithms for shape and feature analysis are reviewed and discussed in Section 5 and in Section 6 experimental results and conclusions are presented.

2 Segmentation

To extract motion data from the video sequence and identify objects of interest (OI), both temporal and spatial segmentation operations are applied. Temporal differencing is used to measure change between a current and reference image. The reference image, *R(x,y),* is subtracted from the current frame, *F(x,y)*, and the result is thresholded by *T* to obtain a binary difference map, *D(x,y)*, as follows:

$$D(x,y) = \begin{cases} 1 & if\left|R(x,y) - F(x,y)\right| > T \\ 0 & otherwise \end{cases}$$

The difference between pixels is typically measured as a Euclidean distance in RGB space.

The reference image can be defined in several ways. A common approach uses a static background without any subjects in the scene [10, 11, 12]. However, given our exclusive interest in observing students' arms, we find it more appropriate to utilize the inter-frame difference directly, i.e., each frame uses its previous frame as a reference. As an example of this approach, Figure 3 includes the results of two frame differences, in which the raised hands of students A and B can be segmented clearly. Note that in the difference map generated by frames *t-2* and *t-1*, a blob is formed as student A lowered his hand. However, this will be discarded by our recognition algorithm in a later stage.

A key step in temporal differencing is the selection of an appropriate threshold, which may vary from pixel to pixel, and should be updated dynamically, in response to lighting variations over time. It may be assumed that intensity changes continuously and statistically, suggesting that we compute the threshold map, T_t from a sequence of *N* previous frames $\{F_{t-N}, F_{t-N+1}, ..., F_{t-1}\}$. Between these frames, the differences of RGB components are calculated to find the largest span:

$$X_l = \max_{t-N \le i,j \le t-1} | X_i(x,y) - X_j(x,y) |$$

where X takes on the respective values of *R*, *G*, and *B* components in turn. The threshold at pixel (x,y) is then determined by:

$$1.4826K\sqrt{R_l(x,y)^2 + G_l(x,y)^2 + B_l(x,y)^2}$$

as suggested by Levine [1], where 1.4826 is a normalizing factor of Gaussian distribution, and K is a balance coefficient, which we choose to be 1.2 based on experimental results. Figure **3** illustrates the difference image obtained using an adaptive threshold map.

Since the difference image is generally noisy with highly irregular edges, morphological filtering, including erosion and dilation, is typically employed to reduce small, noisy blobs, fill in holes within OIs, and smooth boundaries [2]. Figure 3(d) illustrates the result of a single erosion followed by two dilations to the difference image.

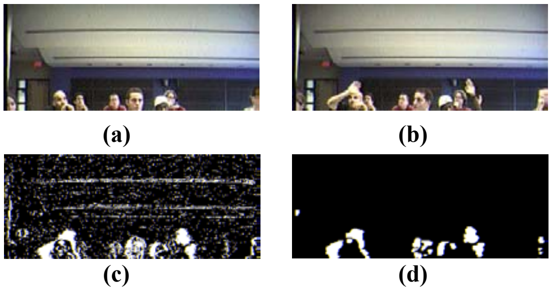

(a) **(b)**
(c) **(d)**

Figure 3. Temporal differencing (a) reference frame; (b) current frame; (c) difference frame with adaptive threshold; (d) output image after morphological operations.

With the difference image now significantly cleaner, a region labeling algorithm is used to assign a unique label to each OI, using a connected component operator [3] in the building block. An example is provided in Figure 4, in which the resulting candidate OIs are enclosed by bounding boxes.

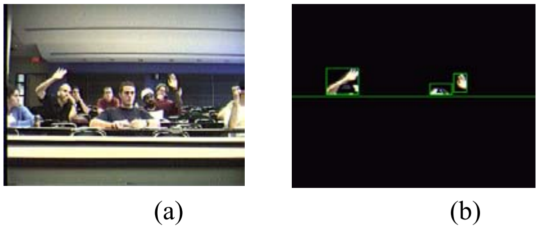

(a) (b)

Figure 4. OI labeling (a) original frame; (b) candidate OIs.

3 OI edge construction

With the location of the OIs now defined, we turn to the problem of shape classification by edge extraction. The raw edges are obtained by application of a Canny edge detector, based on derivatives of gray level intensity *I*, obtained from (R, G, B) tuples by

$$I = 0.299 * R + 0.587 * G + 0.114B .$$

To obtain a cleaner map of the desired outlines, we take advantage of the observation that edges due to a noisy surface are usually *weaker* than object boundaries. Thus, we downscale the grayscale image before edge detection in order to remove these weak edges, as shown in Figure 5.

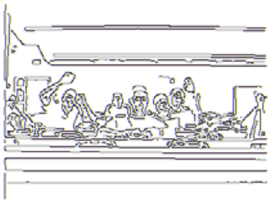
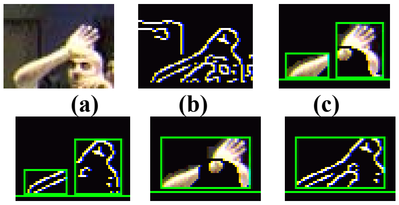


Figure 5. Canny edge detection after downscaling the original image by 0.7, with T_h = 150 , T_l = 1 and σ = 1.2.

In order to focus our attention on those edges indicative of change in the scene, we generate an OI edge map, P_t by applying a logical AND operation on the Canny edge map, E_t and the binary difference map, D_l:

$$P_t(x, y) = \begin{cases} D_l(x,y) & \text{if } D_l(x,y) > 0 \text{ and } E_t(x,y) > 0 \\ 0 & \text{otherwise} \end{cases}$$

In the event that an arm movement between two successive frames is relatively small, or the individual's clothing color is similar to the background, the edges of P_t are likely to be broken into multiple, small fragments. To combine these, we must apply more dilation operations, taking care not to introduce unintended edges. As one such precaution, we do not dilate edges along the horizontal, since a raised arm generally has a non-horizontal slope. The process of integrating fragments of OIs and combining their associated bounding boxes is illustrated below in Figure 6.



(a) (b) (c)

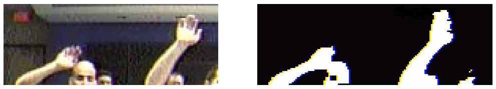
(d) (e) (f)

Figure 6. OI edge recovery (a) original object; (b) primary edges from Canny operator (c) fragmented OIs from temporal change detection; (d) fragmented edges of OIs; (e) integrated OI after recovery operation, in which dilation is applied to the edges of 8d following the outlines of 8b; (f) recovered edges.

4 Skin tone identification

Assuming that the background wall color of a typical classroom is relatively different from that of flesh tones [4], the use of skin distribution information within the edges of the OIs could be helpful for human arm recognition. We use a predefined-range method [4, 9] over a normalized RGB space to define the range of skin color distribution. Sample results are illustrated below in Figure 7.

Since the algorithm has already isolated the edges corresponding to OIs, we need only analyze the color of pixels within their defined boundaries. In this example, we were fortunate to have several students wearing short sleeves, thus presenting good skin color data. In general, however, we can only rely on students' hands being exposed.



(a) (b)

Figure 7. Skin detection results; (a) original image, (b) matching pixels in normalized RGB space.

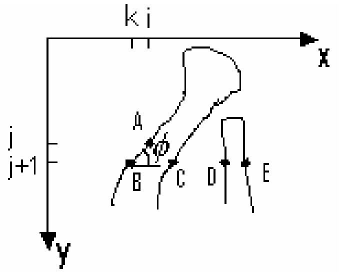
5 Shape and feature analysis

From the skin map, a skin edge map is extracted indicating the outlines of the skin blobs. Next, the thinning algorithm described by Stentiford, is applied, to compute the medial axis of the skin outlines using a 3×3 template [5]. Taking into account the specific shape of our targets, we note that knowledge of the object boundary is sufficient to produce a skeleton. The outline is smooth, relatively noise-free, and tends to have a monotonically increasing or decreasing spatial distribution; furthermore, the width varies gradually and smoothly. Thus, the skeleton can be approximated reasonably by a straight line.

Our algorithm raster scans the image, matching all boundary points with Stentiford's templates, selecting target points according to the following criteria:

1. The *span*, or column separation between two target points on the same row is within an appropriate range, for which the average is defined by a dynamic parameter, w_0, adjusted according to spans already accepted.
2. Boundary points tend to maintain a *stable* orientation.

Figure 8. Orientation criteria for selecting target points.



For example, a left boundary point (A in Figure 8) at position (*i*, *j*) constitutes a member of a span with tangent $\phi < 90^0$. Thus the left boundary point on the next scan line should be near B at position (*k*, *j+1*) (*k*<*i*) in order to keep

ϕ relatively unvaried. Further, the right boundary point C on this row is also selected according to its relative position with B, while candidates D and E are rejected, according to our selection criteria. This can be seen in Figure 9, in which the rows where the student's hand and head are connected in the skin map are excluded from the medial axis computation. This helps avoid an erroneous computation of the center.

Similar to Baruch [6], we extract skeleton candidates as the *middle* pixels of these boundary points, taken on the same row, as illustrated in Figure 9.

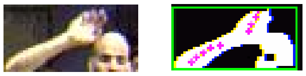


Figure 9. Voted medial axis candidates (cross-hatches in figure on right).

Although these candidates do not fully represent the object's skeleton, they are sufficient to construct an approximate medial axis using line regression. Most importantly, unlike other skeletonization methods, this algorithm needs to perform only a single raster scan, which is critical for efficient operation.

Once the medial axis is constructed, we estimate the slope of the arm using the linear function $y = ax + b$. As the simple least squares regression methods are not robust to outliers, we consider the use of LTS or LMS [7], but favor the less costly Hough transform [8].

The orientation and four extent ratios are calculated for our OIs and used as constraints against valid arm poses. Orientation is represented by the slope of the extracted arm skeleton, which, for a completely raised arm, is typically in the range of [45^0, 135^0]. The extent ratios (expressed as height/width) are measured for the bounding box, the primary difference blob, the skin tone region, and the geometric outline of the OI.

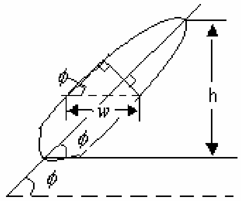


Figure 10. Geometric ratios for skin tone and outline.

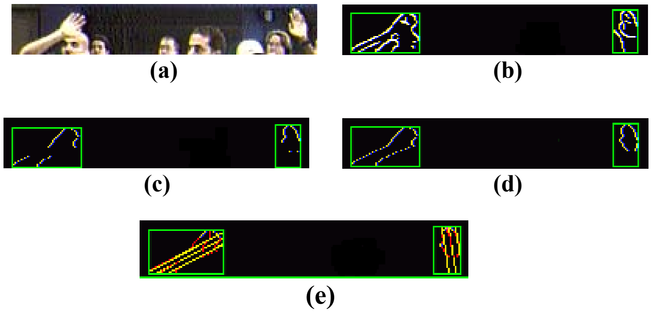
The geometric ratios are illustrated in Figure 10, in which w is the median horizontal width and ø is the orientation angle. The actual width and length of the object can be approximated as $w$sin ø and h/sin ø, respectively, assuming non-horizonal orientation of the arm. Thus the length/width ratio is: $\Lambda = \dfrac{h}{w\sin^2\phi}$

Based on our assumption of smoothness, the variance of horizontal span of an arm should be fairly small. We use the statistical standard deviation to measure the smoothness:

$$\sigma^2 = \sum_i \frac{(x_i - \mu)^2}{N}$$

where μ is the mean of $\{x_i\}$.

Since the line segments obtained by the methods of Section 3 include some unwanted interior edges, we now sweep the constructed arm edges to remove invalid outline candidates, whose separation exceeds the standard deviation by some empirically determined threshold. Any gaps along the orientation of the retained candidates are filled to preserve continuous segments, which may then be approximated by parallel lines, symmetric about the medial axis. This process is illustrated in Figure 11.



(a) (b)

(c) (d)

(e)

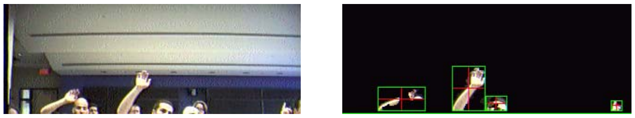
Figure 11. Detection of outline candidates (a) original image; (b) OI's edges; (c) candidates after removal of pairs exceeding threshold of standard deviation; (d) reconstructed outlines; (e) simplified model representation with parallel lines on sides of medial axis.

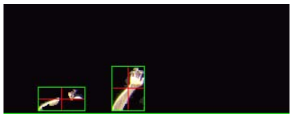
Similar to the snake method [13] we can measure the quality of fit of our simplified model to the actual outlines using their mutual attraction energy, defined as the average of the squared Euclidian distance between the outline pixels and the parallel lines of the model.

In order to evaluate the likelihood that our parallel line model represents an actual arm outline, we consider two factors. First, the energy terms must be reasonably small and the percentage of outlier pixels among the overall candidate set should be low. Second, based on our assumption of symmetry about the media axis, the energy terms of the left and right outlines should also be symmetric.

6 Results and conclusions

Two examples of our algorithm in operation are provided in the figures below, illustrating detection of multiple targets (Figure 12) and successful rejection of a distracter (Figure 13).

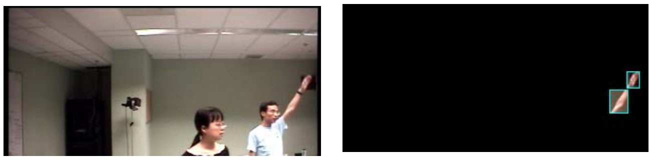




(c)

Figure 12. Test of multiple targets (a) original scene; (b) OIs; (c) detected targets.

Under conditions in which the arm is largely covered by clothing of similar color to the background, it is difficult for the skin-tone detector to isolate regions of movement, thus leading to small difference blobs. However, in combination with outline analysis, it remains possible to detect a raised arm, provided a sufficient area of skin from the hand is observed.



(a) (b)

Figure 13. Test of distracter rejection: (a) the distracter (the student in the center of the image) has just appeared in the scene (b) detected targets.

Our experimental results yield a correct recognition rate of approximately 80% and a false positive rate of 20%. The latter are generally due to unexpected actions, such as a student standing up or walking, and environmental factors such as the occasional similarity of a background color to skin tone. To reduce the occurrence of false positives, we could enforce a minimum on the size of subjects in the frame, as smaller subjects are more difficult to detect reliably. A second improvement would be the construction of a more complex skin color model in order to minimize the influence of environmental effects.

At present, the system performs adequately in detecting most hand raising events under various test scenarios, although, there is obvious room for improvement before it can be used reliably as a communication assistant in remote lecturing applications. An additional area for future work would be to automate the determination of algorithm-specific thresholds that are sensitive to focal length and view angle. Finally, our ongoing efforts aim to incorporate tracking and motion analysis for camera control, so that the camera can zoom automatically to a student with a question.

Acknowledgements

The completion of this research was made possible thanks to Bell Canada's support through its Bell University Laboratories R & D program and to a CANARIE Inc. ANAST research contract. This support is gratefully acknowledged.

References

[1] Y. Yang, M.D. Levine. 1992. The Background Primal Sketch: an Approach for Tracking Moving Objects. Machine Vision and Applications, Vol. 5, 17-34.

[2] R.M. Haralick, S.R. Sternberg, X. Zhuang. 1987. Image Analysis Using Mathematical Morphology. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 9, No. 4, 532-550.

[3] A. Bovik. 2000. Handbook of Image and Video Processing. Academic Press.

[4] M.M. Fleck, D.A. Forsyth, C. Bregler. 1996. Finding Naked People. European Conference on Computer Vision, Volume II, 590-602.

[5] F. W. M. Stentiford and R.G. Mortimer. 1983. Some New Heuristics for Thinning Binary Hand Printed Characters for OCR. IEEE Transactions on Systems, Man, and Cybernetics. Vol. 13, No.1: 81-84.

[6] O.Baruch. 1988. Line Thinning by Line Following. Pattern Recognition Letters, Vol. 8: 271-276.

[7] P. J. Rousseeuw. 1984. Least Median of Squares Regression. Journal of the American Statistical Association, 79: 871-880.

[8] Z. Zhang. 1997. Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting**.** Image and Vision Computing Journal, Vol. 15, No. 1: 59-76.

[9] K. Sobottka, I. Pitas. 1996. Segmentation and Tracking of Faces in Color Images. Second International Conference of Automatic Face and Gesture Recognition: 236-241.

[10] C.R. Wren, A. Azarbayejani, T. Darrell, A. P. Pentland. 1997. Pfinder: Real-time Tracking of the Human Body. IEEE Transactions on Pattern Analysis and Machine Intelligence, 780-785.

[11] M. Bichsel. 1994. Segmenting Simply Connected Moving Objects in a Static Scene Trans. Pattern Analysis and Machine Intelligence, Vol. 16, no. 11: 1,138–1,142.

[12] I. Haritaoglu, D. Harwood, and L.S. Davis. 1998. W4: Who? When? Where? What? A Real System for Detecting and Tracking People. International Conference on Automatic Face and Gesture Recognition, 222-227.

[13] M. Kass, A. Witkin, and D. Terzopoulos. 1987. Snakes: Active Contour Models. First International Conference on Computer Vision: 259-268.