# Hierarchical Motion History Images for Recognizing Human Motion [*]

James W. Davis
Dept. of Computer and Information Science,
Center for Cognitive Science
583 Dreese Lab, 2015 Neil Avenue
Ohio State University
Columbus, OH 43210
jwdavis@cis.ohio-state.edu

## Abstract

*There has been a recent and increasing interest in computer analysis and recognition of human motion. Previously we presented an efficient real-time approach for representing human motion using a compact "Motion History Image" (MHI). Recognition was achieved by statistically matching moment-based features. To address previous problems related to global analysis and limited recognition, we present a hierarchical extension to the original MHI framework to compute dense (local) motion flow directly from the MHI. A hierarchical partitioning of motions by speed in an MHI pyramid enables efficient calculation of image motions using fixed-size gradient operators. To characterize the resulting motion field, a polar histogram of motion orientations is described. The hierarchical MHI approach remains a computationally inexpensive method for analysis of human motions.*

## 1. Introduction

The tracking and recognition of human motion, action, and events using computer vision has recently gained widespread interest in both academic and industrial research, with much emphasis on real-time systems [7, 21, 20, 14, 1, 24]. The applied significance for machines capable of human motion recognition can be found in automatic surveillance and monitoring systems, video content analysis, and Perceptual User Interfaces [26]. With the emergence of increasingly faster computers enabling real-time video analysis, there will be a growing role for computer recognition of human motion and activity.

Previously, we developed a real-time approach for representing human movement that recursively integrates the motion throughout a video action sequence into a single "Motion History Image" (MHI) [9, 3]. For recognition, higher-order moment features computed from the template were statistically matched to trained models. The approach was used within several interactive virtual environments requiring computer interpretation of the participant's motions (e.g., [4, 10, 5]). Though the approach has proven a useful real-time method in constrained situations, it still has a number of limitations related to the *global* image feature calculations and *specific* label-based recognition.

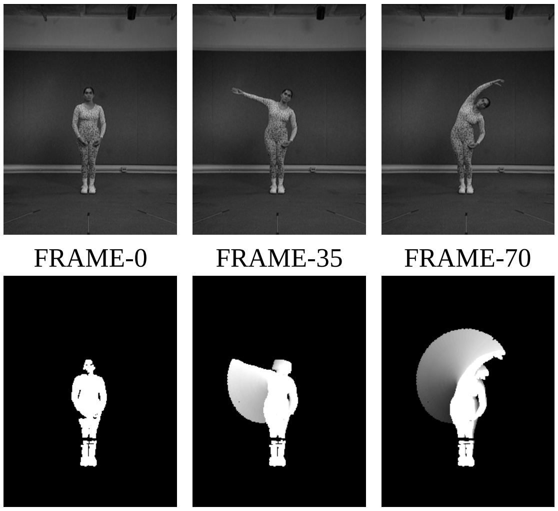
In this paper, we address these limitations by extending the approach with a mechanism to compute a dense (local) motion vector field directly from the MHI for describing the movement. Raw motion information is not globally affected by minor occlusions and may be favored in recognition situations when a precisely labeled action is not required or even possible. For example, a system may be designed to respond to leftward motion, but may not care if a person, hand, or car created the motion. The main contribution of this new approach is the use of a hierarchical MHI representation that compensates for the varying speeds that are common to articulated human motion.

We begin with an overview of the general MHI framework (Sect. 2). Then we describe a gradient-based mechanism for extracting motion flow (Sect. 3) within a hierarchical MHI pyramid representation (Sect. 4). From the motion gradients, motion orientation histograms are constructed and evaluated as a means of movement characterization (Sect. 5). Lastly, we discuss additional capabilities and the system implementation (Sect. 6), followed by a summary of the research (Sect. 7).

## 2. Motion History Images

In prior work we developed the basic MHI framework for representing and recognizing human motions [9, 3]. The

[*] Support for this research was provided by Intel Corporation.


FRAME-0 FRAME-35 FRAME-70

MHI-0 MHI-35 MHI-70

**Figure 1. Top row: Keyframes of an arm stretching exercise movement. Bottom row: MHIs corresponding to keyframes in the top row.**

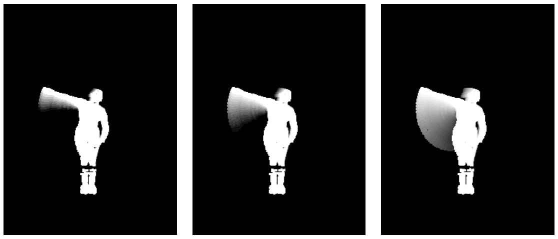

$\delta = 0.25$ $\delta = 0.5$ $\delta = 1.0$

**Figure 2. Effect of altering the decay parameter $\delta$ (in seconds) in Eqn. 1.**

method focuses on accumulating and recognizing holistic "patterns of motion" rather than trajectories of structural features. Similar use of templates for characterizing motion include work by [22, 19, 12], but are constrained to very particular domains (e.g., periodicity, facial motion). Our general template method is targeted at representing *arbitrary* human (and other) movements. The strength of the approach is the use of a compact, yet descriptive, real-time representation capturing a sequence of motions in a single static image (similar to [18]). The MHI is constructed by successively layering selected image regions over time using a simple update rule:

$$MHI_\delta(x,y) = \begin{cases} \tau & \text{if } \Psi(I(x,y)) \neq 0 \\ 0 & \text{else if } MHI_\delta(x,y) < \tau - \delta \end{cases} \tag{1}$$

where each pixel (x,y) in the MHI is marked with a current timestamp $\tau$ if the function $\Psi$ signals object presence (or motion) in the current video image $I(x,y)$; the remaining timestamps in the MHI are removed if they are older than the decay value $\tau - \delta$. This update function is called for every new video frame analyzed in the sequence.

The function $\Psi$ that selects a pixel location in the input image for inclusion into the MHI can be arbitrarily specified. Since the template representation captures both the position and temporal history of a moving object, many possibilities for selecting regions of interest are applicable. Detectors may include background subtraction, image differencing, optical flow, edges, stereo-depth silhouettes, flesh-colored regions, etc. With an *object* selection process for $\Psi$ (e.g., background subtraction), the representation can accommodate slowly moving regions ($<$ 1 pixel/frame) that would otherwise be missed by image differencing or standard optical flow. For the results presented here, we used a threshold-difference background subtraction method.

To illustrate the construction of an MHI, keyframes from a sequence of a person performing an "arm stretch" movement and the corresponding (cumulative) MHIs are presented in Fig. 1 ($\delta = 2.33$ sec.). For display purposes the timestamp pixel values in the templates are linearly mapped to graylevel values 0–255. Here the brightness of a pixel corresponds to its recency in time (i.e., brighter pixels are the most current timestamps). Depending on the value chosen for the decay parameter $\delta$, an MHI can encode a wide history of movement (See Fig. 2).

Our initial approach to recognition with MHIs [9] was to extract several higher-order scale and translation invariant moment features [16] (also from a binarized version) and statistically match them to stored model examples using the Mahalanobis distance [25]. Though successful in constrained situations with single and multiple cameras, a limitation with that recognition method was the holistic generation (and matching) of the moment features computed from the *entire* template. Any occlusions of the body or errors from the implementation of $\Psi$ resulted in serious recognition failures. Also the recognition method was limited to only label-based (token) recognition, where it could not yield any information other than specific identity matches (e.g., it could not report that "upward" motion was occurring at a particular image location). This led us to consider a more *localized* approach to motion analysis of the MHI.

## 3. Motion Gradients

From Eqn. 1, the MHI layers the $\Psi$ regions over time in such a way that the visual appearance of the layered regions gives the impression of motion directly from the intensity gradients in the template. It is quite apparent from MHI-70 in Fig. 1 that the upward progression of movement is captured in the dark-to-light intensity gradients.

Since motion can be perceived from the displayed timestamp gradients in the template, one could theoretically convolve gradient masks with the timestamp values in the MHI to extract a motion vector at each pixel (this is similar in concept to computing normal flow along brightness contours [15]). We demonstrated this basic concept in [8, 11].

The use of a discrete fixed-sized gradient mask to calculate the template's motion vectors, though, has an inherent bias to particular motion speeds, where the motion displacement (gradient) must appear within the spatial resolution of the gradient mask. For example, the standard 3×3 Sobel gradient masks

$$F_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} /8 \qquad F_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} /8$$

limit the range of recoverable motion to a 2-pixel maximum object displacement in each dimension. The convolution of 3×3 gradient masks (or masks of similar dimension) will therefore not produce motion vectors at faster moving regions because no overlapping or layered object regions are visible within the window. If we were to simply employ larger spatial gradient masks, this would alternatively reduce the granularity of measurements attainable in slower moving regions.

We need an adaptable mechanism to appropriately extract the MHI motions of different velocities. One approach could employ multiple, competing gradient masks of different spatial coverage. Our approach instead is to exploit the gradient information with a hierarchical representation using fixed-size gradient operators.

## 4. Hierarchical MHIs

Much like the work on hierarchical motion estimation, stereo matching, and image coding using image pyramids [2, 23, 6], we extend the original MHI representation into a hierarchical pyramid format to provide us with a means of addressing the gradient calculation of multiple image speeds. An image pyramid is constructed by recursively lowpass filtering and sub-sampling an image (i.e., power-of-2 reduction with anti-aliasing) until reaching a desired size of spatial reduction. With a pyramid representation, two images having large motion displacements between them will have smaller displacements when compared at increasingly higher (reduced) pyramid levels. This permits us to use fixed-size gradient masks in each pyramid level (along with some constraints) to calculate motions of different speeds. The result is a hierarchy of motion fields where the resulting motion computed in each level is tuned to a particular speed (with faster speeds residing at higher levels). In addition, the spatial reductions in the pyramid generation help to fill in any small gaps that may have been missed by $\Psi$.

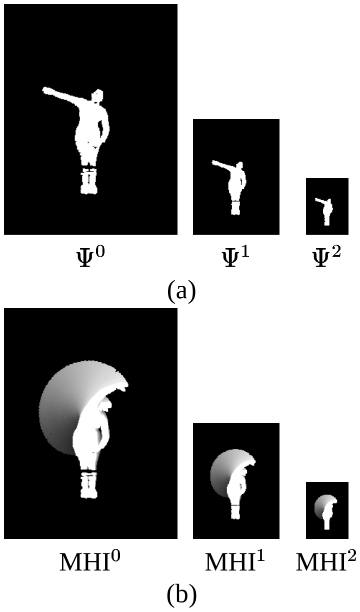


**Figure 3. Image pyramids. (a) $\Psi$ pyramid of a person silhouette. (b) MHI pyramid created from (a).**

Unfortunately, we cannot simply convert the MHI directly into a pyramid using the anti-aliasing reduction techniques described. Since the MHI is composed of timestamps rather than graylevels, anti-aliasing the border values of the motion regions with the null background (or regions missed by $\Psi$) under a lowpass operation will corrupt the timestamp values. Alternatively, the $\Psi$ silhouette image itself has no such temporal sequencing and it can be transformed directly into a pyramid and used to create the targeted MHI pyramid. Since $\Psi$ is required to produce a binary image for Eqn. 1, we post-convert each grayscale $\Psi$ pyramid level into a binary image by thresholding the anti-aliased images above zero (equivalent to dilating each image by half the size of the blur window). A binary $\Psi$ image pyramid for frame 35 of the arm stretching movement is shown in Fig. 3(a). To create the corresponding MHI pyramid, each level from the $\Psi$ pyramid is used to update an MHI of that particular resolution. This indirect method of generating the MHI pyramid remains computationally effective and can be done in parallel, resulting in the pyramid shown in Fig. 3(b).

### 4.1. Motion orientation and speed

Before calculating the motion orientations and speeds in each level of the MHI pyramid using the gradient masks, a few constraints are initially required. First, the boundaries of the motion regions should not be examined by the gradient operators because of their adjacency to the null background values. We therefore impose an 8-connectedness test for each timestamp pixel to verify that it is "interior" to the motion region. Similarly, we do not examine the current $\Psi$ silhouette timestamps ($\tau$). We additionally constrain the $F_x$ or $F_y$ gradient to have an absolute value above some minimum to ensure that $>2$ object regions are being layered as a ramp (rather than a step of uniform region) within the gradient mask. For the implementation a threshold of 1/(2·FPS) is used. An upper bound on the gradient can also be enforced to discount those large temporal changes beyond some reasonable delay (e.g., 10/FPS).

Unlike the standard reintegration component (warp, expand, re-estimate) in the motion estimation of [2], we do not require an iterative propagation of the course-to-fine motion measures back to the size of the original MHI. Instead, for each pixel we 1) choose the pyramid level that passes the gradient constraints, 2) compute the motion from that level, and 3) scale the result to the size of the original image. If multiple pyramid levels pass the constraints in Step 1), then we choose the level $L$ with the minimum acceptable temporal disparity (finest temporal resolution):

$$L(x,y) = argmin_i(F_x^i(x,y)^2 + F_y^i(x,y)^2) \qquad (2)$$

The resulting motion orientation $\phi$ and speed $S$ computed from level $L$ for pixel $(x,y)$ and scaled to level-0 follows with

$$\phi(x,y) = \arctan \frac{F_y^L(x',y')}{F_x^L(x',y')} \qquad (3)$$

$$S(x,y) = \sqrt{V_x(x,y)^2 + V_y(x,y)^2} \qquad (4)$$

where $x' = x/2^L$, $y' = y/2^L$, and velocities $V$ are

$$V_x(x,y) = \frac{2^{L+1}}{2 \cdot F_x^L(x',y')} \qquad (5)$$

$$V_y(x,y) = \frac{2^{L+1}}{2 \cdot F_y^L(x',y')} \qquad (6)$$

The numerator in the velocity calculation corresponds to the 2-pixel spatial extent of a 3×3 gradient window in level-L scaled to the size of the original image. The denominator represents the change in time visible in the gradient window. For example, an object seen in level-2 moving a horizontal distance of 1 pixel every frame (at 30 Hz) corresponds to $V_x(x,y) = \frac{2^3}{2 \cdot 0.0333} = 120$ pixels/sec $\rightarrow$ 4 pixels/frame in the original image.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **1** | 1 | **1** | 2 | **2** | 2 | **3** | 3 | **3** |

(a) Level-0

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 2 | 3 | 3 |

(b) Level-1

| | | | | |
|---|---|---|---|---|
| - | $\frac{4}{3}$ | $\frac{6}{3}$ | $\frac{8}{3}$ | - |

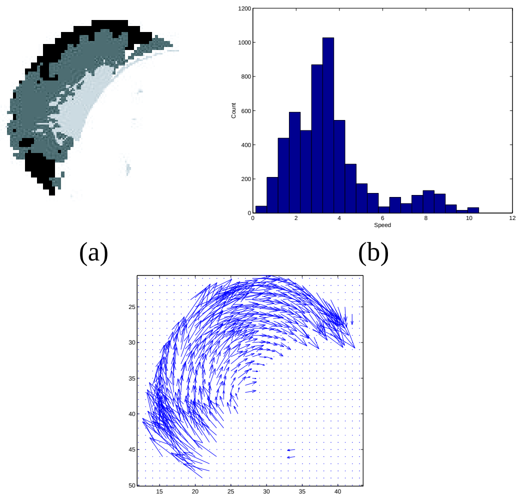
(c) Level-1 (blurred)

**Figure 4. Sub-sampling process. (a) Original level-0 image. (b) Sub-sampling level-0 produces level-1. (c) Smoothed version of (b).**

Since we are in effect using sub-sampled dilated $\Psi$ images to create the MHI pyramid, we need to first smooth "inside" the aliased motion regions in each level of the MHI pyramid to accommodate those timestamp values deleted between the even numbered locations during sub-sampling in the $\Psi$ pyramid. We illustrate this with a 1-D example in Fig. 4. In the top row we have a collection of timestamps at level-0 in an MHI pyramid. These values correspond to a 3-pixel wide object moving rightward at 3 pixels/frame. After sub-sampling, only the timestamps in bold from level-0 survive to level-1. None of the locations in level-0 pass the min-value gradient constraint. In level-1, the valid gradient at the middle location (using Eqn. 5) produces a velocity of 2 pixels/frame $(4/(2 \cdot 1))$ when in fact the motion is 3 pixels/frame. This is a result of no anti-aliasing while constructing the pyramid. But if we apply a 3×1 average mask to level-1 (post reduction) to extend the gradient information (See Fig. 4(c)), the timestamps can be used to recover the correct velocity $(4/(2 \cdot \frac{2}{3}) = 3)$.

In Fig. 5(a) we show the selected pyramid levels (using Eqn. 2 with the validation constraints) for each pixel in the arm raising MHI from Fig. 1. As expected with arc motion, the radially distant regions have a faster speed and are thus calculated at a higher pyramid level. In Fig. 5(b) we plot a histogram of the speeds computed using Eqn. 4. The resulting motion field is displayed in Fig. 5(c) and captures the overall expected pattern and organization of motions, with the larger motion vectors most radially distant from the point of rotation.

When body motions become extremely fast between frames, the approach requires multiple pyramid levels to reduce the displacements for calculating the proper gradients. A problem that can arise is illustrated with the MHI for a jumping-jack exercise movement as shown in Fig. 6(a). The level-0 MHI shows the quickly moving arms appearing strobed and unlayered. The actual pixel distance between the small hand region in consecutive frames is quite large — approximately 12 pixels. The pyramid approach cannot accommodate these small, fast arm motions for the size of the regions becomes too small in the higher pyramid levels



(a) (b)

(c)

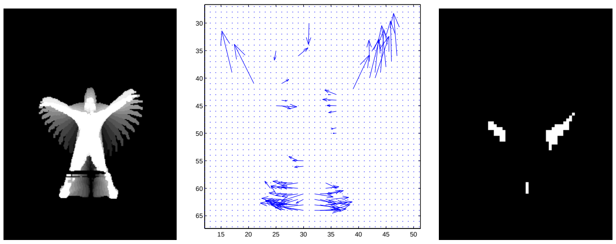
**Figure 5. (a) Pyramid levels assigned to pixels in the MHI (three-level pyramid with level-0=light-gray, level-1=medium-gray, and level-2=black). (b) Histogram of calculated pixel speeds. (c) Resulting motion field.**

to be analyzed[1]. In Fig. 6(b) we show the motion components that *are* recoverable with the approach using three pyramid levels, capturing mostly the final arm motions and the legs kicking out. The approach can however signal the non-recoverable fast motion that does not pass the gradient constraints in the upper-most level of the pyramid (See Fig. 6(c)). This way, the approach can announce that additional faster motion is present, but not recoverable.

## 5. Motion Orientation Histograms

Once the motion field for the MHI has been computed (e.g., Fig. 5(c)), several recognition methods could be applied to characterize particular movements. In the gesture recognition work of [13], a single histogram of image *edge orientations* of a user's hand was used to recognize various static gestures, with dynamic gestures formed by concatenating histograms of individual poses. We follow this approach and develop a *motion orientation* histogram by accumulating the orientations $\phi$ of the motion flow computed from the MHI pyramid (this method could be extended to incorporate speed as well).

In the top row of Fig. 7 we present cumulative MHIs for the arm stretching movement at frames 25, 45, and 70. The corresponding cumulative motion orientation histograms plotted in polar form are shown in the bottom row of Fig. 7. The histograms are quantized by 5 degrees and smoothed with a 5-tap Gaussian filter. The motion from frames 0–25 appears as movement at 125°, from 0–45 the motion extends clockwise to 60°, and from 0–70 includes the remaining rightward motion.



(a) (b) (c)

**Figure 6. Jumping-jack movement. (a) Level-0 MHI. (b) Resulting motion field using three pyramid levels. (c) Regions identified with speeds beyond pyramid range.**

We examined several different movements from the database of exercise activities used in [9]. In Fig. 8, we present the final keyframe, MHI, and polar motion orientation histogram for the movements. There are many similarities in the actions, where moves (a) and (b) have general upward motion (move (b) also has downward motion), moves (c) and (d) have upward arcing motion, and moves (e) and (f) possess diagonal downward motion. It is clear from the polar motion orientation histograms that these movements are significantly different from one another, which may not be entirely expected given the coarse nature of the MHI representation and histogram features.

For recognition, an unknown polar histogram can be normalized and compared to a model histogram using the Euclidean $L_2$ norm distance metric as a measure of similarity. The comparative $L_2$ distances for the movements in Fig. 8 are shown in Table 1. To handle linear changes in performance speed during recognition, the approach outlined for the original MHI framework using a backward-looking time window remains applicable [9].

Since histograms ignore the spatial configuration, movements comprised of the same motion vectors, no matter where they appear in the template, appear similar in a polar motion orientation histogram. As shown in Fig. 9, sideways leaning and sitting from a particular viewpoint have very similar polar motion orientation histograms but quite different body configurations (actions).

To test the sensitivity of the representation to changes in view angle, we examined two different movements each

[1] Inside blurring followed by the 8-connected check effectively erodes regions at each level by approximately 2 pixels.

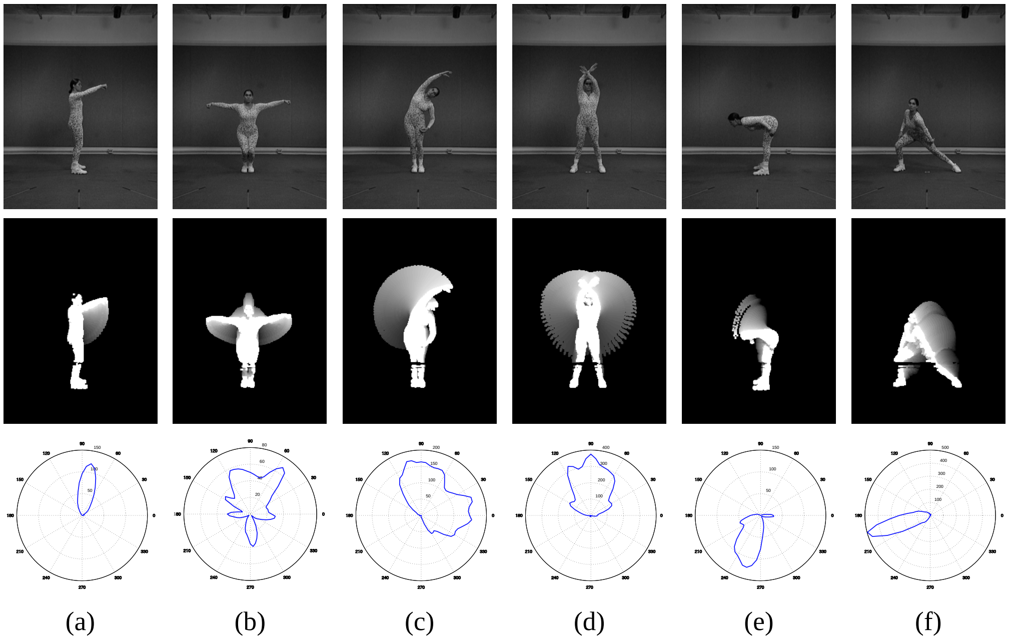


**Figure 8. Top row: Key-frames from six aerobic exercise movements. Middle row: MHIs for the movements. Bottom row: Polar plots of motion orientation histograms computed using an MHI pyramid.**

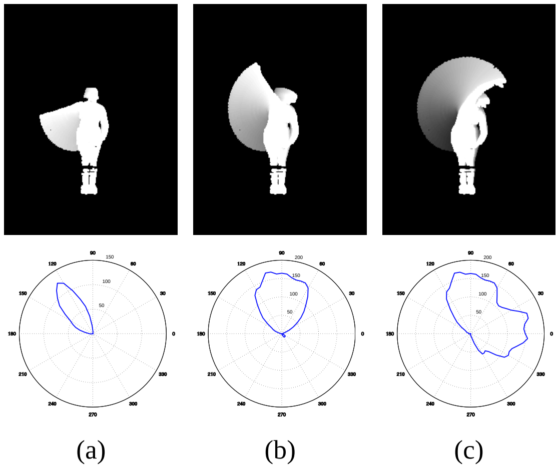


**Figure 7. Cumulative MHIs and polar motion orientation histograms. (a) Frames 0-25. (b) Frames 0-45. (c) Frames 0-70.**

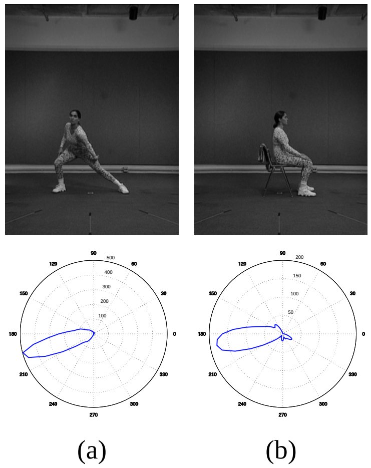
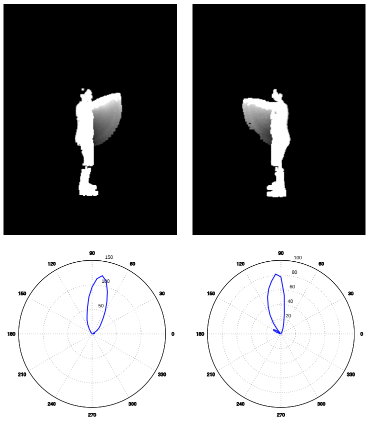


**Figure 9. Movements having similar polar motion orientation histograms but different action. (a) Sideways lean. (b) Sitting.**

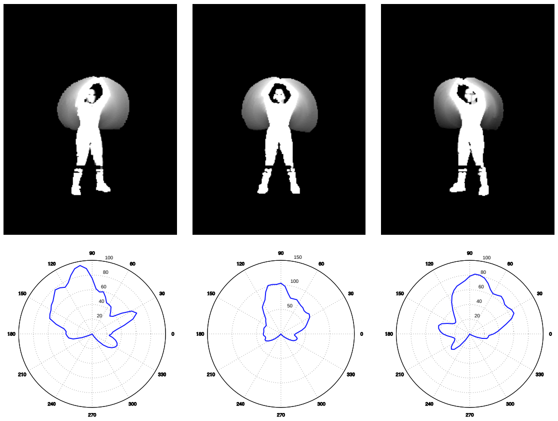


(a) (b)

**Figure 10. Reflection discrimination in the polar motion orientation histograms.**

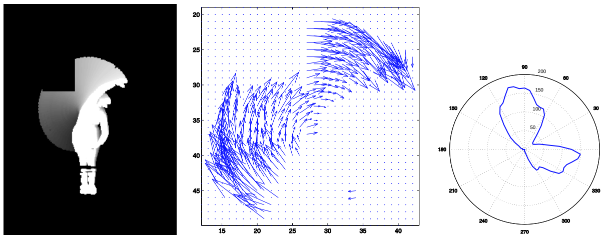
| | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ |
|---|---|---|---|---|---|---|
| $P_1$ | 0.000 | 0.205 | 0.205 | 0.155 | 0.338 | 0.370 |
| $P_2$ | – | 0.000 | 0.085 | 0.089 | 0.207 | 0.262 |
| $P_3$ | – | – | 0.000 | 0.105 | 0.243 | 0.294 |
| $P_4$ | – | – | – | 0.000 | 0.269 | 0.301 |
| $P_5$ | – | – | – | – | 0.000 | 0.260 |
| $P_6$ | – | – | – | – | – | 0.000 |

**Table 1. Euclidean $L_2$ norm $||P_i - P_j||$ measures for normalized polar plots in Fig. 8.**



(a) (b) (c)

**Figure 11. View change discrimination with $\{-30°, 0°, +30°\}$ views of the same movement.**



(a) (b) (c)

**Figure 12. Occlusion example. (a) MHI with occlusion region. (b) Motion field computed from occluded MHI. (c) Polar motion orientation histogram of occluded MHI.**

performed from different viewpoints. In Fig. 10, we show MHIs for ±90° views of a forward arm-lift movement. The motion calculation and polar motion orientation histograms capture the subtle differences between these motions caused by reflection, as shown in the bottom row in Fig. 10. In Fig. 11, we show the MHIs for $\{-30°, 0°, +30°\}$ views of a side arm-lift movement. The polar motion orientation histogram representation quite distinctly captures the differences resulting from the view-based changes.

To report the effect of occlusion on the representation, we blocked out a region in the MHI containing motion (prior to pyramid generation), computed the polar motion orientation histogram, and compared the results with the un-occluded version. In Fig. 12(a) we show the MHI for the arm stretching movement with a 44×44 pixel motion region removed. The calculated motion field for the occluded MHI is shown in Fig. 12(b). From the motion field, the polar motion orientation histogram was computed and is displayed in Fig. 12(c). For comparison, refer to the un-occluded histogram in Fig. 7(c). The occlusion does not preclude the calculation of the remaining visible motions in the MHI, and thus the two versions are similar except for the missing occlusion data. The normalized $L_2$ distance between the histograms is 0.056 and is much smaller than the distances of the un-occluded motion to the other exercise movements in Fig. 8. A shape-based characterization will have a quite different response.

## 6. Discussion

In addition to employing the motion orientation, the motion speed could be utilized for the task of motion segmentation. Earlier work [5] presented a segmentation of regions in an MHI using a downward-stepping floodfill algorithm over the motion regions to identify areas of motion directly attached to body parts. With the pyramid representation, each level in the MHI hierarchy has associated with it motion of the same energy, thus providing a possible means of segmentation by clustering motion regions in the same pyramid level (similar to the layers work of [27]). The *relative* differences of motion between levels may be applicable for differentiating multiple objects or motions.

We are currently implementing the hierarchical MHI approach in C++ using the highly optimized routines provided in the Intel Image Processing Library (IPL) and Open Source Computer Vision (OpenCV) library [17]. Many of the MHI functions and other necessary operators have already been incorporated into these packages. The main advantage to using the Intel libraries in terms of hardware is that faster processing is now accessible on standard PC-based platforms rather than on specialty systems or costly workstations. With this infrastructure, we are seeking stable real-time performance for a system that can be easily ported and made available to other researchers or developers.

## 7. Summary

We presented a useful hierarchical extension for computing a local motion field from the original MHI representation. To alleviate sensitivities to minor occlusion and limitations from global analysis, we developed a gradient-based motion pyramid method to extract local motion flow directly from the MHI. The MHI was transformed into an image pyramid to permit efficient fixed-size gradient masks to be convolved at all levels of the pyramid to extract motion information at a wide range of speeds. Polar histograms of the motion orientations were used to characterize several human movements. The hierarchical MHI approach remains a computationally inexpensive algorithm to represent, characterize, and recognize human motions in video.

## References

[1] J. Aggarwal and Q. Cai. Human motion analysis: a review. In *Nonrigid and Articulated Motion Workshop*, pages 90–102. IEEE, 1997.

[2] J. Bergen, P. Anadan, K. Hanna, and R. Hingorami. Hierarchical model-based motion estimation. In *Proc. Euro. Conf. Comp. Vis.*, pages 237–252, 1992.

[3] A. Bobick and J. Davis. The recognition of human movement using temporal templates. *IEEE Trans. Patt. Analy. and Mach. Intell.*, 23(3), 2001.

[4] A. Bobick, S. Intille, J. Davis, F. Baird, L. Campbell, Y. Ivanov, C. Pinhanez, A. Schutte, and A. Wilson. The KidsRoom: action recognition in an interactive story environment. *Presence: Teleoperators and Virtual Environments*, 8(4):367–391, 1999.

[5] G. Bradski and J. Davis. Motion segmentation and pose recognition with motion history gradients. In *Workshop on Applications of Computer Vision*. IEEE, Dec. 2000.

[6] P. Burt and E. Adelson. The laplacian pyramid as a compact image code. *IEEE transactions on communications*, com-31(4):532–540, April 1983.

[7] C. Cedras and M. Shah. Motion-based recognition: a survey. *Image and Vision Comp.*, 13(2):129–155, 1995.

[8] J. Davis. Recognizing movement using motion histograms. Technical Report 487, MIT Media Laboratory, Cambridge, MA, 1999.

[9] J. Davis and A. Bobick. The representation and recognition of action using temporal templates. In *Proc. Comp. Vis. and Pattern Rec.*, pages 928–934. IEEE, 1997.

[10] J. Davis and A. Bobick. Virtual PAT: a virtual personal aerobics trainer. In *Proc. Workshop on Perceptual User Interfaces*, pages 13–18, 1998.

[11] J. Davis and G. Bradski. Real-time motion template gradients using Intel CVLib. In *Proc. ICCV Workshop on Frame-rate Vision*. IEEE, 1999.

[12] I. Essa and A. Pentland. Facial expression recognition using a dynamic model and motion energy. In *Proc. Int. Conf. Comp. Vis.*, pages 360–367, June 1995.

[13] B. Freeman and M. Roth. Orientation histograms for hand gesture recognition. In *Proc. Int. Conf. Auto. Face and Gesture Recog.*, 1995.

[14] D. Gavrila. The visual analysis of human movement: a survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.

[15] B. Horn. *Robot Vision*. MIT Press, Cambridge, 1986.

[16] M. Hu. Visual pattern recognition by moment invariants. *IRE Trans. Information Theory*, IT-8(2):179–187, 1962.

[17] Intel Open Source Computer Vision Library, www.intel.com/research/mrl/research/opencv.

[18] R. Jain and H. Nagel. On the analysis of accumulative difference pictures from image sequences of real world scenes. *IEEE Trans. Patt. Analy. and Mach. Intell.*, 1(2):206–214, April 1979.

[19] F. Liu and R. Picard. Finding periodicity in space and time. In *Proc. Int. Conf. Comp. Vis.*, pages 376–383. IEEE, 1998.

[20] T. Moeslund. Computer vision-based human motion capture - a survey. Technical Report LIA 99-02, University of Aalborg, 1999.

[21] T. Moeslund. Summaries of 107 computer vision-based human motion capture papers. Technical Report LIA 99-01, University of Aalborg, 1999.

[22] R. Polana and R. Nelson. Detection and recognition of periodic, nonrigid motion. *Int. J. Comp. Vis.*, 23(3):261–282, 1997.

[23] L. Quam. Hierarchical warp stereo. *Proc. Image Understanding Workshop*, pages 137–148, 1984.

[24] M. Shah and R. Jain, editors. *Motion-Based Recognition*. Kluwer Academic, 1997.

[25] C. Therrien. *Decision Estimation and Classification*. John Wiley and Sons, Inc., New York, 1989.

[26] M. Turk and G. Robertson. Perceptual user interfaces. *C-ACM*, 43(3):33–34, 2000.

[27] J. Wang and E. Adelson. Representing moving images with layers. *IEEE Trans. Image Proc.*, 3(5):625–638, 1994.