# Low Level Recognition of Human Motion
## (Or How to Get Your Man Without Finding his Body Parts)

Ramprasad Polana and Randal Nelson

Department of Computer Science
University of Rochester
Rochester, New York 14627
Email: polana@cs.rochester.edu and nelson@cs.rochester.edu

**Abstract**

*The recognition of human movements such as walking, running or climbing has been approached previously by tracking a number of feature points and either classifying the trajectories directly or matching them with a high-level model of the movement. A major difficulty with these methods is acquiring and tracking the requisite feature points, which are generally specific joints such as knees or ankles. This requires previous recognition and/or part segmentation of the actor. In this paper, we show that the recognition of walking or any repetitive motion activity can be accomplished on the basis of bottom up processing, which does not require the prior identification of specific parts, or classification of the actor. In particular, we demonstrate that repetitive motion is such a strong cue, that the moving actor can be segmented, normalized spatially and temporally, and recognized by matching against a spatiotemporal template of motion features. We have implemented a real-time system that can recognize and classify repetitive motion activities in normal gray-scale image sequences.*

## 1 Introduction

The emphasis on visual motion as a means of quantitative reconstruction of world geometry has tended to obscure the fact that motion can also be used for recognition. In fact, in biological systems, the use of motion information for recognition is often more evident than its use in reconstruction. Humans have a remarkable ability to recognize different kinds of motion, both of discrete objects, such as animals or people, and in distributed patterns as in windblown leaves, or waves on a pond. The classic demonstration of pure motion recognition by humans is provided by Moving Light Display experiments [Johansson, 1973], where human subjects were able to distinguish activities such as walking, running or stair climbing, from lights attached to the joints of an actor. More subtle movement characteristics can be distinguished as well. For example, human observers can identify the actor's gender, and even identify the actor if known to them, by his or her gait. Such abilities suggest that, in the case of machine vision, it might be possible to use motion as a means of recognition directly, rather than indirectly through a geometric reconstruction.

Human motion, specifically walking, has been studied extensively using model-based approaches (for example [Hoffman and Flinchbuagh, 1982]). Rashid [1980] addresses the problem of correspondence of the points in an MLD sequence between successive frames and obtaining trajectories of those points. Goddard recognizes MLDs involving single actors moving parallel to the image plane using a connectionist approach utilizing the lower-level features of trajectories. Gould et al. [1992] build a trajectory primal sketch that represents significant changes in motion with the purpose of identifying objects using trajectories of a few representative points. The curvature features of trajectories have been used to detect cyclic motion by Allman and Dyer [1990] and by Tsai et al. [1993].

Very few researchers attempted motion recognition directly using purely low-level features of image motion information. Anderson et al. [1985] use spatiotemporal energy measures to characterize different sources of motion. Overall, there is little literature concerning the question of whether motion recognition can be achieved using purely low-level features of motion and if so how it can be achieved and what features to use. Our research answers this question affirmatively and yields specific demonstrations of motion recognition using low-level statistical features of motion. Further, the focus of earlier research was mainly on human gait recognition, whereas our techniques apply equally well to any source of motion.

In this paper, we describe a robust method for recognizing activities, including ones, such as walking, that involve simultaneous translation of the actor. The recognition scheme is based on low-level features of motion, and does not require the recognition or tracking of specific parts of the actor. We make use of the motion field computed between successive frames to segment and track the actor, detect scale changes and compensate for translation and scaling. The resulting gray-level image sequence consists of the activity at a constant distance from camera while the actor remains stationary in the image frame. We com-

bine this with earlier reported methods [Polana and Nelson, 1994a], [Polana and Nelson, 1994b] for detecting stationary activities and classifying them to obtain a real-time implementable system of activity recognition. Such techniques for the recognition of activities have potential applications in areas such as automated surveillance.

## 2 Activities

Activities involve a regularly repeating sequence of motion events. If we consider an image sequence as a spatiotemporal solid with two spatial dimensions $x, y$ and one time dimension $t$, then repeated activity tends to give rise to periodic or semi-periodic gray-level as well as motion signals along smooth curves in the image solid. We refer to these curves as *reference curves*. If these curves could be identified and samples extracted along them over several cycles, then frequency domain techniques could be used in order to judge the degree of periodicity and thus detect periodic activities.

Consider the case of human walking. This is an example of a non-stationary activity; that is, if we attach a reference point to the person walking, that point does not remain at one location in the image. If the person is walking with constant velocity, however, the reference point moves across the image in a path composed of a constant velocity component modulated by whatever periodic motion the reference point undergoes. Thus, if we know the average velocity of the person over several cycles, we can compute the spatiotemporal curves of motion along which the periodicity can be observed.

For the current research we assume that the object producing the periodic activity is undergoing locally linear translatory motion (in 3D), so that we can estimate the local velocity of the object and compensate for the translation and looming so as to make the object stationary. We recompute the motion field between successive frames of the resulting gray-level image sequence and use the recomputed motion to detect and classify the activity.

## 3 Activity Recognition

We use a spatiotemporal motion magnitude template as a basis for the recognition of activities. In order for this to work, the motion to be identified must be normalized with respect to spatial scale, spatial translation and temporal scale and translation. Template matching is a well studied and frequently effective method of recognition. It fails when sufficiently rigid normalization cannot be carried out. It turns out that periodicity inherent in motion such as walking or running is a sufficiently strong cue to allow strong normalization to be performed.

If there are multiple actors in the scene, it is important to initially detect each actor and spatially isolate them. Fortunately, independent motion provides an exceptionally strong segmentation cue. Nelson [Nelson, 1991] has demonstrated a real-time method of detecting independently moving objects even in the case that the observer is itself moving. Using such a method, we can detect the pixels in an image sequence that exhibit motion independent of that of the background and segment the image frames into distinct regions corresponding to different moving objects. Other common methods of segmenting multiple moving objects are: using color cues, distance from camera obtained from a range sensor, or selecting objects moving in a certain velocity range.

Given a gray valued image sequence, we first detect pixels corresponding to independently moving objects. These pixels are grouped using spatial clustering methods and an object of interest (the actor performing the activity) is selected and tracked. The subsequent activity detection and recognition can be applied to each independently moving object. For each selected object, a spatiotemporal template of motion features is obtained from the motion of the corresponding pixels, and it is used to match the test sample with the reference motion templates of known activities.

Spatial scale invariance is achieved by measuring the spatial size of the object through successive frames, estimating the spatial scale parameters and compensating for the changes in scale. Spatial translation invariance is achieved by tracking the object of interest through successive frames, estimating the spatial translation parameters and compensating for the translation of the object. The spatial translation parameters are estimated using a least squares technique assuming the object is moving along a locally linear trajectory.

Temporal scale invariance is achieved by detecting the frequency of the activity and obtaining one cycle of activity by averaging motion information of multiple cycles. A more complete discussion of the periodicity detection and frequency estimation can be found in [Polana and Nelson, 1994a]. Temporal translation has turned out to be hard to estimate from the motion information, but it was handled in the matching stage by matching the test template with reference template at all possible temporal translations.

In the following subsections we focus on the normalization procedures with respect to spatial translation and scale changes and in a later subsection we describe the steps involved in feature vector computation and matching.

### 3.1 Tracking in the Presence of Other Moving Objects

If there is a single moving object in the scene, the object can be effectively tracked by following the centroid of the moving pixels corresponding to that object. Such a simple method of tracking does not work if there is more than one moving object in the field of view. Instead, we make use of an estimate of shape of the object and its predicted position in the image frame to restrict the centroid computation to the area that is most likely corresponds to the object. Suppose $S(t)$ is the set of pixels that corresponds to the estimated object, and $(x_t, y_t)$ is the position of the object in flow frame $t$. From the position estimates of the past few (say $K$) flow frames, we obtain an estimate of the velocity of the object (assuming local linear trans-

latory motion as before). Let $(u_t, v_t)$ be the velocity estimate at flow frame $t$. Then the predicted position of the object in flow frame $t+1$ is

$$p(t+1) = (x_t + u_t, y_t + v_t).$$

And, an estimate for the set of pixels corresponding to the object in flow frame $t+1$ is

$$S'(t+1) = \{(x + u_t, y + v_t) : (x, y) \in S(t)\}.$$

We measure the centroid of motion

$$c(t+1) = \sum_{(i,j,t) \in S'(t+1)} (i, j)/\|S(t)\|$$

in frame $(t+1)$, and then updated estimates of position of the object and its corresponding set of pixels respectively in flow frame $t+1$ as

$$(x_{t+1}, y_{t+1}) = w * p(t+1) + (1-w) * c(t+1),$$

$$S(t+1) = \{(x + x_{t+1} - x_t, y + y_{t+1} - y_t) : (x, y) \in S(t)\}.$$

This is continued for every frame estimating the position of the object using its velocity estimated from past $K$ frames and centroid of motions in the current frame.

A demonstration of the tracking algorithm in the presence of multiple moving objects and occlusions by other objects is shown in figure 1. The illustration shows an image sequence consisting of two persons walking towards and crossing each other. The object of interest in this case is the person walking from right to left. The first eight frames of the 64 frame image sequence given here consist of only the first person walking. The second person temporarily occludes the first person. In the first eight frames, we thresholded the motion to highlight significant motion. Using those locations we estimated the shape of the object of interest in the form of a rectangle surrounding the object, which is illustrated in the figure. The estimated positions are shown with a plus (+) sign. It can be seen that the tracking algorithm smoothly tracks the first person even when there is occlusion. The sequence on the right shows the tracking in detail through successive frames during the occlusion.

### 3.2 Changing Scale

In this section, we show how the changes in spatial scale of the activity can be detected and compensated for. We make a key assumption here: that the height of the object of interest does not change over time. This is certainly true for the activities of human walking, running etc., and it is a reasonable assumption for a host of other activities. (Even when the height is changing, the periodic repetition of the activity requires that the same height recur through successive cycles of the activity, and hence fitting the model described below over many periods in this case will give good estimates of scale changes).

Let $H$ be the actual height of the object in three-dimensional world. (It is assumed that this $H$ is unchanging over time). According to the projective imaging model with image plane at unit distance from the origin, the image coordinates $(x_t, y_t)$ in image frame $t$ are related to the three-dimensional world coordinates $(X_t, Y_t, Z_t)$ as $(x_t, y_t) = (X_t/Z_t, Y_t/Z_t)$, where $Z_t$ is the distance of the object from the camera at image frame $t$. From this it can be derived that $h_t = H/Z_t$ where $h_t$ is the projected image height of the object at image frame $t$. Now, if we assume the object is approaching or moving away from camera at a locally constant velocity, say $W$, then $Z_t = Z_0 + W * t$ is the distance of the object from camera. Using the relation $h_0 = H/Z_0$, we find the image height of the object over time to be

$$h_t = h_0/(1 + w * t)$$

where $w = W/Z_0$ is a constant scale factor. (Notice that $w$ is negative if the object is approaching the camera, positive if the object is moving away, and it is exactly equal to zero if the object's distance from the camera does not change).

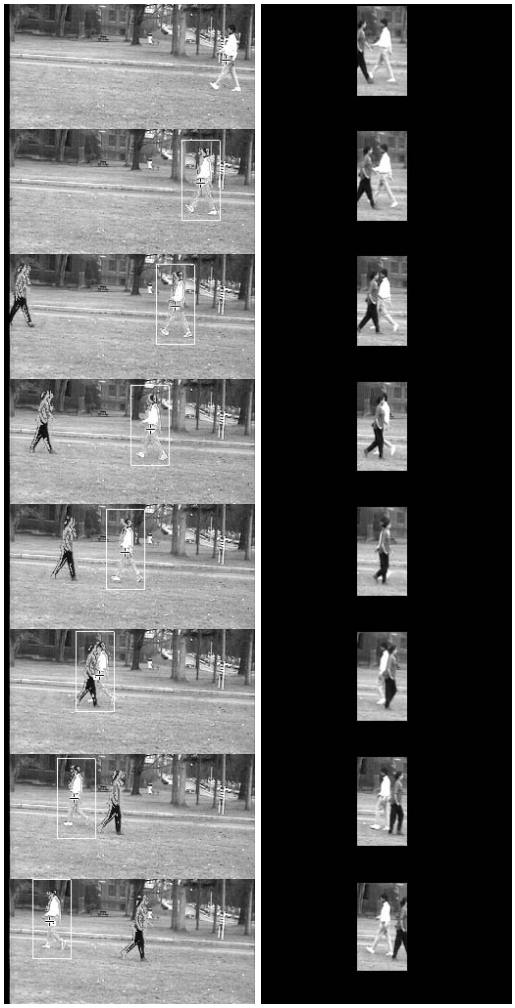

Figure 1: Left: Tracking applied to a walking person (every eighth frame shown); Right: image frames surrounding the interfering motion (consecutive frames)

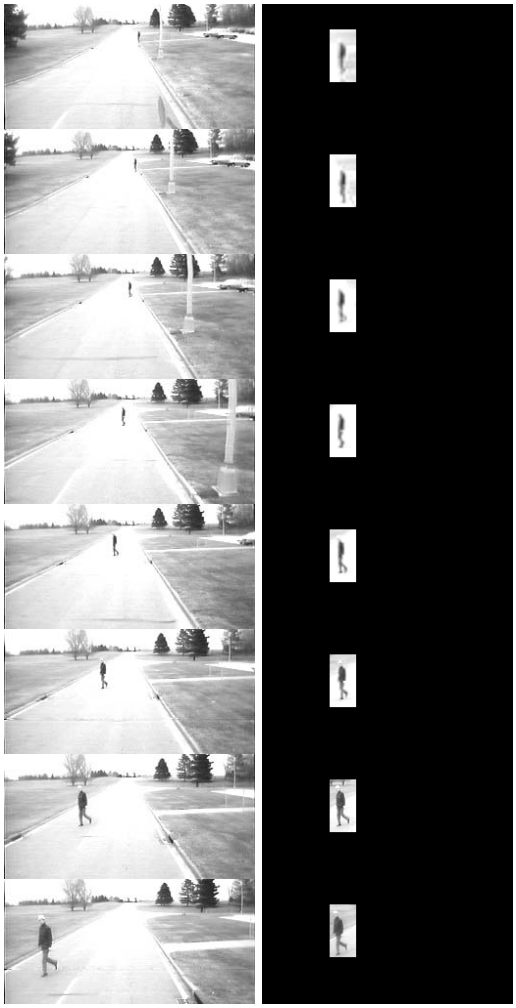

Figure 2: Left: Image sequence of a person walking across the street taken by a camera mounted on a van moving at about 30 mph (every eighth frame); Right: sequence after tracking (every eighth frame)

By estimating height of the object in each flow frame and using the model described above, we obtain an estimate for the locally constant scale factor $w$ and then compensate for the scale changes by scaling the image frame $t$ so as to match the scale of the activity in the reference database (which is fixed before hand). Unfortunately, the relation $h_t = h_0/(1 + w * t)$ is not linear in $w$ and so we can not directly use the least squares technique to estimate $w$. Instead, we use an approximation to the model $h_t = h_0(1 - w * t)$ which is a good approximation if the term $w * t$ is small. We keep $w * t$ small by using the model in small temporal neighborhoods, so that $t$ is very small (also $w = W/Z_0$ is small if the distance of the object from camera is large compared to the speed with which it is approaching or moving afar, which is true in most circumstances). Note that the relation $1/h_t = (1 + w * t)/h_0$ is linear in $w$, but using least squares to minimize the error between observed and model heights is not same as minimizing the error between observed and model inverted heights, and hence the estimate of $w$ is not same, even though it may produce a reasonable estimate.

Thus the steps involved in detecting and compensating for changes in scale are: measure the image height of the object in each flow frame, use the heights in the last $K$ frames to estimate the scale factor $w$ and scale the image frame $t$ to match the fixed scale corresponding to the activities in the reference database.

To demonstrate the above technique, we have digitized an image sequence from the video recorded by NIST (National Institute of Standards & Technology) using a video camera mounted on a van looking straight ahead while the van is being driven on the road at about 30mph around the NIST grounds in Gaithersberg, Maryland. The image sequence is shown in figure 2 which consists of a person walking across the street as the van is approaching. The image heights of the person for this sequence are hand-measured and plotted (dotted-line) in figure 3. As can be seen, a linear fit over the entire image sequence is inappropriate for this data. We estimated the scale factor over the entire sequence using the least squares technique for inverted heights and plotted (solid-line) the resulting fit to the data in the same figure. It gives a reasonably good approximation in the beginning where the distance from camera is large and at the right end a slight deviation from actual heights is seen where the distance of the person from camera is smaller compared to the speed of the vehicle. By using local linear models a better approximation is obtained and when the image frames are scaled and tracked as before, we obtain the stationary walking activity shown in figure 2. The motion magnitude feature vector is computed for this image sequence and classification algorithm applied and it was correctly classified as walking (there being six other choices and unknown).

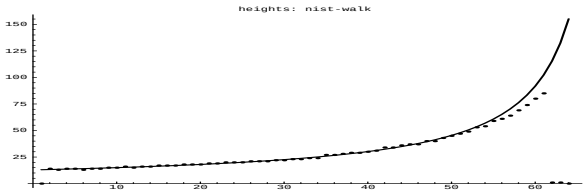


Figure 3: Actual image heights of the person (dotted) and the fitted model (solid)

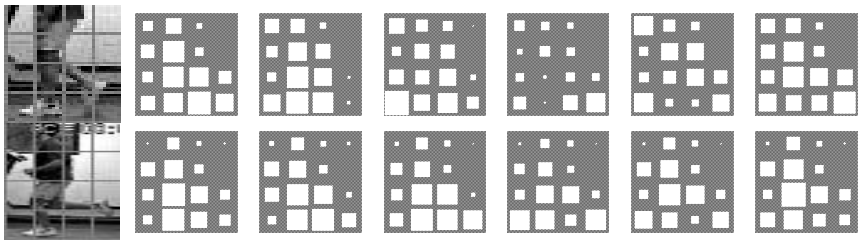

Figure 4: Sample total motion magnitude feature vector for a sample of walk (top) and a sample of run (bottom), one cycle of activity is divided into six time divisions shown horizontally, each frame shows spatial distribution of motion in a 4x4 spatial grid (size of each square is proportional to the amount of motion in the neighborhood).

### 3.3 Feature Vector Computation and Matching

Given a gray-valued image sequence, the actor is detected, tracked and spatial scale changes are estimated. The image sequence is transformed so as to compensate for the spatial translation and scale changes of the actor. The resulting image sequence consists of the actor at the center of the image frame and at the same distance from the camera throughout the image sequence. The image frame is reduced to the size of the object and the motion is computed between successive image frames.

Compensating for translation and scale changes so that the object remains in the center of the image frames causes the previously stationary background to appear to be moving. In the transformed image frames, the background will be moving with the same velocity magnitude as the object velocity estimated but in the opposite direction. After computing the flow fields between successive frames of the transformed image sequence, we eliminate any motion that is consistent with the background velocity by making the estimate at that point unknown. (Alternatively, instead of recomputing motion after transforming and then eliminating background motion, we could update the previously computed flow field by subtracting the estimated trajectory motion of the object. Such a subtraction, however leads to large inaccuracies in the measured flow, primarily because the differential techniques we use for speed have low accuracy, and subtraction can cause the values to lose all significance.)

Each flow frame $t$ is divided into a spatial grid of $X$x$Y$ dimension and the motion magnitudes in each spatial cell are summed. Let $M(x, y, t)$ be the motion magnitude in flow frame t corresponding to spatial cell $(x, y)$. According to the definition of a periodic activity, for each fixed $(x, y)$, the signal $M(x, y, t)$ over time should be periodic. For each $(x, y)$, we compute the periodicity index as described in [Polana and Nelson, 1994a] and combine the individual periodicity indices to get a periodicity measure for the whole image frame. By thresholding the resulting periodicity measure it is possible to determine if the motion produced by the object is periodic. If it is found that there is sufficient periodicity in the motion, we proceed to compute the feature vector of motion magnitudes.

The frequency of the activity is found along with the periodicity measure and it is used to divide the entire image sequence into a number of cycles of the activity. The flow frames are folded over temporally, so as to obtain a single cycle of the activity and the motion in different cycles is averaged to obtain motion in a single combined cycle of activity. The length of the cycle is divided into $T$ temporal divisions and motion is summed in each temporal division corresponding to each spatial cell $(x, y)$ separately. The resulting spatiotemporal motion template is used as the feature vector to match against reference motion templates of known activities.

The classification method we have used is the nearest centroid algorithm, which is simple to implement and effectively shows the discriminating power of the feature vector. To recognize an activity as unknown, we need to fix thresholds for the distance between a test sample and the reference classes. These thresholds can be taken as the average distance of reference samples from the centroid. This way, we would be recognizing a test vector as belonging to class $k$ if the test vector falls within a circular region of radius threshold and center as the centroid. To achieve greater accuracy we first find the principal components of the reference vectors in each class and weigh the test vector elements inversely proportional to the corresponding coefficients in the first principal component. Feature vector elements which are more consistent within the class are given higher priority in matching by the above process and the elements whose variability is large are weighed down. The net result of this procedure is to make the recognition regions around each class centroid ellipsoidal instead of circular.

### 3.4 Real-Time Implementation

We have implemented the above algorithm on SGI architecture with multiple processors. The complexity is proportional to the number of pixels involved in the activity. The majority of the work involved is the normal flow computation at the original resolution of the image sequence. With four processors, the flow computation at 128x64 resolution takes 40 to 50 milliseconds between two successive image frames. The remaining processing involves frames of much reduced resolution and it takes from 20-30 milliseconds. The implementation includes displaying the original image frame, gradient, flow, and the $X$x$Y$ grid motion magnitudes and the classification result at every frame of

the image sequence. The total computation for each frame takes 60-80 milliseconds. Of course, more processors can be used for faster running times.

## 4 Results

The algorithm was used to classify seven different types of activities which included: walking, running, jumping jacks, exercises on a machine, swinging, skiing and swimming activity of a toy frog. The image sequences consist of 128 frames of 128x128 8-bit graylevel pixels (except walking and running whose frames are of 128x64 pixels). The image sequences contained a minimum of four cycles of activity to reliably detect the fundamental frequency given that there is a considerable amount of non-repetitive structure from the background in the case of translating actors. We used four samples of each activity to create a reference database. The test samples differed from the reference database samples in frequency, speed of motion, spatial scale, different lighting conditions and different background and foreground gradients. The test database contained four separate samples of each activity and in addition contained ten samples of walking by a different persons six of which involve simultaneous translation and scale changes (these samples had frames of 128x256 pixels so that the actor remains in the frame through the entire sequence).

We have tested classification using a feature vector of motion magnitudes in a spatiotemporal grid of size 4x4x6. The recognition algorithm could successfully classify all samples achieving a 100% correct classification. To test the degree of robustness of the activity recognition algorithm, we have attempted classifying degraded the samples of walking by adding motion clutter of leaves blowing in the wind at increasing motion magnitudes. The motion of leaves was chosen in stead of random noise for the degradation because it is a realistic example of structured motion clutter that is commonly present in the background. The results showed that the classification scheme can tolerate structured motion clutter whose magnitude is equal to one half that of the activity, and it displayed degraded, but still useful performance for even higher clutter magnitudes.

## 5 Conclusions

We have described a general technique for activity recognition and applied it to the case of walking recognition. This technique uses a periodicity measure to detect the activity and a feature vector based on motion information to classify the activity into one of several known classes. We have illustrated the technique using real-world examples, and shown that it robustly recognizes the activity under various complications. It is robust to varying image illumination and contrast because the method uses only motion information which is invariant to these. It is also invariant to spatial and temporal translation and scale due to the normalization of the feature vectors, and the multiple temporal matching. It is also fairly robust with respect to small changes in viewing angle (i.e on the order of 20 degrees). The swing and exercise sequences were taken outdoors where there is a small amount of background motion. This comprises not only moving trees and plants, but also moving people and an occasional crossing of a car. That the activities can be detected even in these cases demonstrates that the technique is tolerant of the usual background clutter and an occasional disturbance.


## Acknowledgements

This work is supported by contracts NSF IRI-9010692 and AFOSR 91-0288.


## References


[Allmen and Dyer, 1990] M. Allmen and C.R. Dyer. Cyclic motion detection using spatiotemporal surface and curves. In *Proc. Int. Conf. on Pattern Recognition*, pages 365–370, 1990.

[Anderson *et al.*, 1985] C. H. Anderson, P. J. Burt, and G. S. van der Wal. Change detection and tracking using pyramid transform techniques. In *Proc. SPIE Conference on Intelligent Robots and Computer Vision*, pages 300–305, 1985.

[Gould *et al.*, 1992] K. Gould, K. Rangarajan, and M.A. Shah. Detection and representation of events in motion trajectories. In Gonzalez and Mahdavieh, editors, *Advances in Image Processing and Analysis*. SPIE Optical Engineering Press, 1992.

[Hoffman and Flinchbuagh, 1982] D.D. Hoffman and B.E. Flinchbuagh. The interpretation of biological motion. *Biological Cybernatics*, pages 195–204, 1982.

[Johansson, 1973] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics*, 14:201–211, 1973.

[Nelson, 1991] R.C. Nelson. Qualitative detection of motion by a moving observer. In *Proc. of IEEE CVPR*, pages 173–178, 1991.

[Polana and Nelson, 1994a] R. Polana and R.C. Nelson. Detecting activities. *Journal of Visual Communication and Image Representation*, 5(2):172–180, 1994.

[Polana and Nelson, 1994b] R. Polana and R.C. Nelson. Recognizing activities. In *Proceedings of ICPR*, 1994.

[Rashid, 1980] R.F. Rashid. *LIGHTS: A System for Interpretation of Moving Light Displays.* PhD thesis, Computer Science Dept, University of Rochester, 1980.

[Tsai *et al.*, 1993] P.-S. Tsai, M. Shah, K. Keiter, and T. Kasparis. Cyclic motion detection. Technical Report CS-TR-93-08, Computer Science Dept, University of Central Florida, 1993.