CHAPTER 3

Primitive features for relative motion dependent feature extraction

A sequence of images taken in natural environment contains a plethora of different types of visual information like color, form, texture, depth relations, motion, etc., all of which contribute to the vivid perception of the concrete dynamic scene revealed with time. All these visual modalities seem to be processed in parallel by our visual system, each separate modality influencing and contributing to the perception of the others. However, when it comes to the processing of the visual information contained in the same sequence of images by an artificial device like the present-day electronic computer of the von Neumann type, with its intrinsically sequential manner of information processing, and also keeping in mind that the problem of extracting and evaluating the above-mentioned visual modalities by a computer usually amounts to implementing an algorithm involving time-consuming numerical calculations (Marr, 1982; Horn, 1986; Hirai, 1995, p.134), it becomes obvious that for real time implementations it would be

advantageous and much more efficient to discard any information which might be considered inessential for the practical task to be solved. This strategy has been repeatedly and successfully used by researchers in psychophysical experiment settings (e.g. Johansson, 1973, 1975, 1976, for eliminating inessential visual cues in connection with analyzing the perception of motion stimuli). Whether or not a day will come when it would be affordable or even necessary for our computers to "consciously enjoy" the same vivid picture of the surrounding world as humans do is not a major concern for us here. Rather, we would try to venture as far as possible in the opposite direction, discarding any information which might not be essential for our current aim to obtain a system capable to perform a robust recognition of human motion patterns in real-time. In this regard, this chapter will be concerned with trying to find more or less satisfactory answers to the following two major questions:

(1) how much of the visual information contained in a sequence of time-varying images is indispensable for the task of human motion recognition, and how much could be efficiently filtered out as unnecessary (or even impeding), thus contributing to the real-time solution of the problem;

(2) what kind of features could be used to characterize naturally, efficiently and in a robust way the dynamic scenes obtained when various different types of human motions are being performed in front of the camera.

The rest of this chapter is organized in the following order. Section 3.1 will examine some of the evidences available from psycho-physical experiments in an attempt to find an answer to question (1) stated above, and also in support of our choice of primitive features proposed for the needs of the current task. In section 3.2, the primitive features for relative motion dependent feature extraction will be introduced, and section 3.3 will conclude this chapter with a discussion on the proposed features and describing some possible further extensions of our approach.

3.1. Psychophysical evidence in support of the current approach

Motion perception has been studied extensively in a series of psycho-physical experiments known as Moving Light Displays (MLDs). The basic idea of the MLDs (Johansson, 1973, 1975) is to attach a number of small flashlights bulbs to the shoulders, elbows, wrists, hips, knees and ankles of a subject(s) dressed in black and to let him move in a darkened room, or in front of a dark background (Fig. 3-1). A motion-picture film is made and shown to naïve observers, requesting them to identify the type of motions being performed by the subject(s). While the subject is staying motionless or sitting in a chair, what the observers see is only a meaningless random constellation of lights.

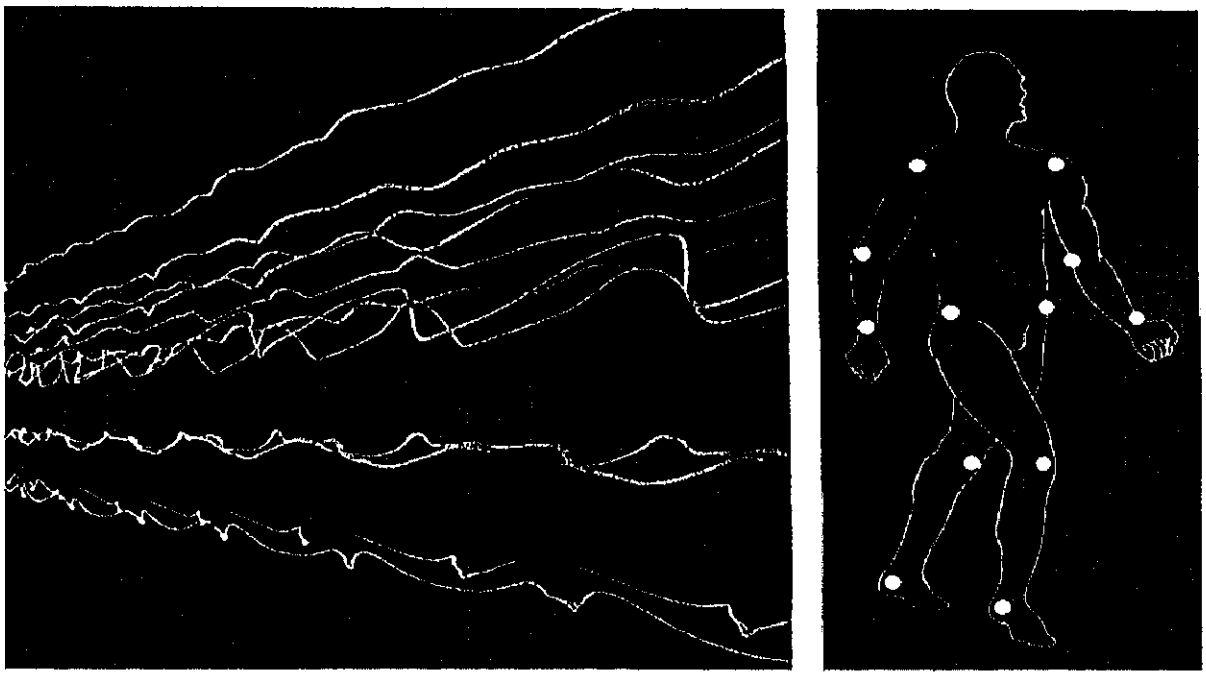


Figure 3-1. Moving Light Displays (MLDs) Experiment conducted by Johansson. On the left are shown the continuous streaks generated by the twelve lights attached to the joints of a subject (shown on the right) as he moves in a dark room, and which are difficult to interpret as a meaningful motion pattern. The same patterns recorded on motion-picture film, however, are easily recognized by naïve observers.

However, once the subject starts moving, the observers instantly manage to recognize the different types of motion patterns performed in the movie, unmistakably discriminating between jogging, normal walking, simulated limp walking, etc., even the gender of the performer or the gait of a friend are easily recognized depending solely on the relative movements of a dozen moving lights.

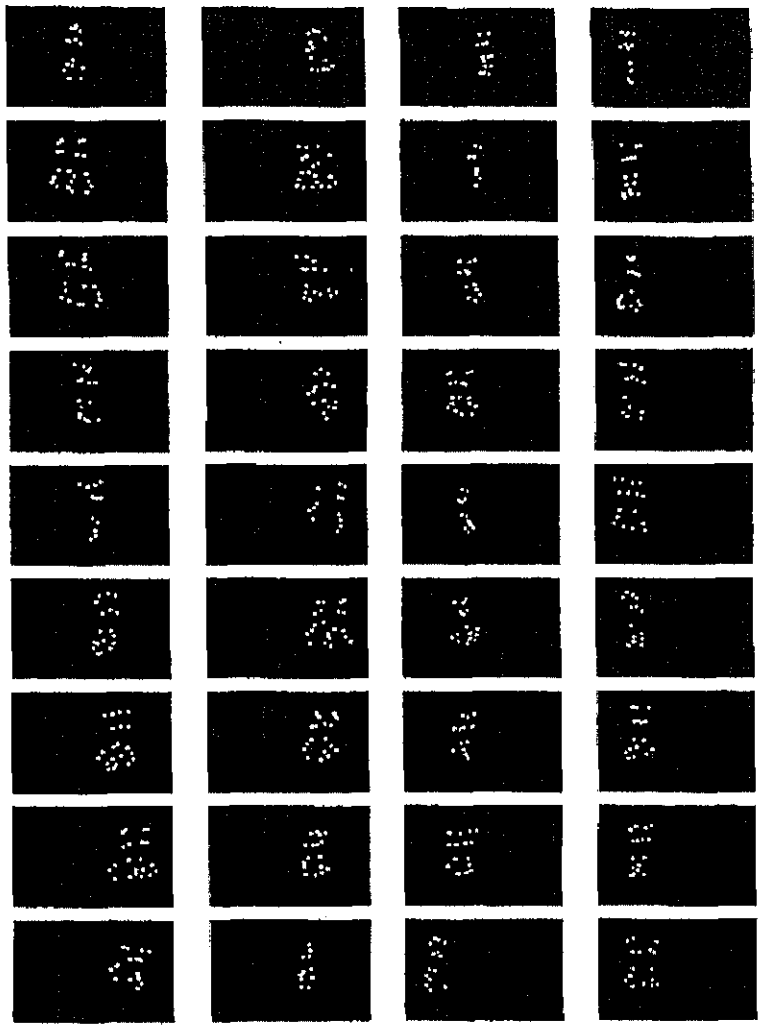


Figure 3-2. Moving Light Displays (MLDs) of two people performing a folk dance in the dark. The sequence proceeds in vertical columns starting at upper left.

Several motion-pictures frames from another experiment conducted by Johansson, consisting of two subjects with twelve bright spots attached in a similar manner and performing a folk dance, are shown in Fig. 3-2. Again, it has been reported that when the film is projected, anyone of the naïve observers is able to correctly decode the nature of the movements.

What this experiments might suggest is that even if the visual input is as radically impoverished as explained above, discarding information about color, brightness (gray-scale values), form, texture, stereo, etc., but retaining only binary motion information (the *moving* bright spots vs. the other black areas), this still contains enough information for the visual system to be able to successfully and rapidly recognize the motion patterns being performed.

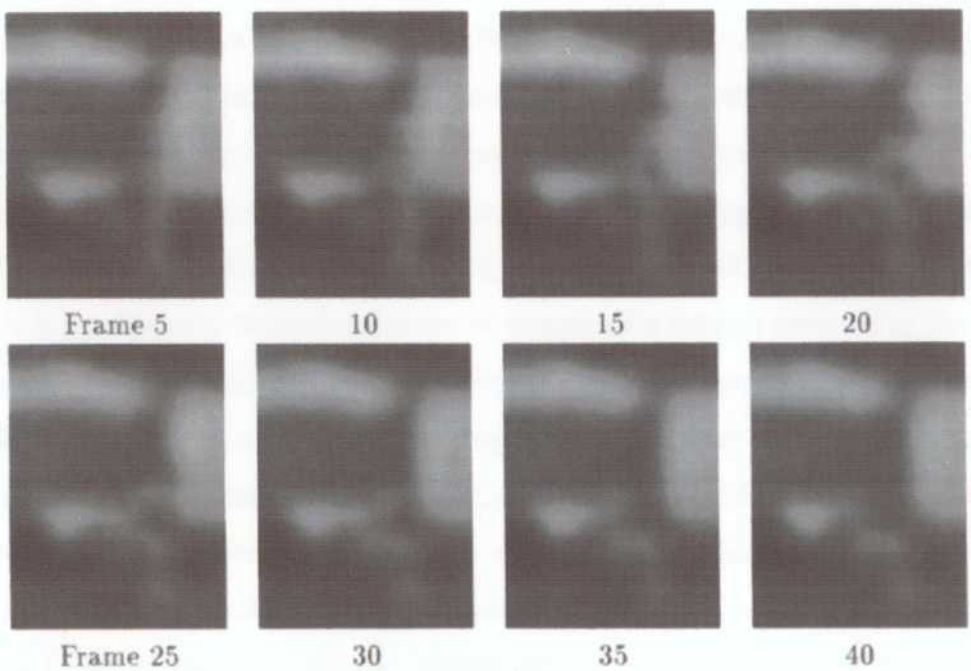


Figure 3-3. Significantly blurred sequence of images where almost no structure is present. Yet observers can easily recognize the action as someone sitting.

However, the MLDs experiments still leave room for speculations that *motion information* might be used first to recover *2D* or *3D structure* (structure from motion), so

that after that *structure* can be used for recognition. The determination of structure from motion (SFM) has been extensively studied in computer vision (Ullman, 1981; Hoffman and Flinchbaugh, 1982; Zhunag and Haralick, 1985; Kenner and Pong, 1990; Broida and Chellapa, 1991). The SFM problem is usually formulated in terms of systems of linear or nonlinear equations, and given the 2D positions of moving points among a few frames, the solution is the 3D coordinates of the points on the moving objects and their 3D motion recovered from the 2D image sequence. Drawbacks to this approach include the facts that the reconstruction is sensitive to noise, and that generally multiple cues like motion, texture, specularities, etc. are needed for robust and accurate recognition. Also, since the SFM methods basically compute intrinsic surface properties like depth or other 2.5D maps (Marr, 1982), their output still needs to be segmented and interpreted/recognized somehow, which might not be a trivial problem.

The approach which we have chosen to follow in this thesis is to directly use motion information for recognition, assuming that the intermediate structure recovery step is unnecessary. Apart from the experiments mentioned above, concerning the easy recognition of MLDs, another line of evidence in support of this approach has been pointed out by Davis and Bobick (Davis and Bobick, 1997; Davis 1998) and is demonstrated in Fig.3-3. A significantly blurred sequence of images of a subject performing the action "sitting" is shown to observers, who are able to recognize it correctly in less than a second, despite the lack of any discernible image detail due to the severe blurring (neither the human being performing the movement, nor any details of the environment are visible). No good features exist in this image sequence upon which to base a SFM algorithm, or which to be used for tracking. Thus, it seems logical to assume that the recognition is achieved directly from the motion pattern relations hidden in the image sequence, without a prior structure recovery or object recognition being necessary. (It might be interesting to notice here a similarity with another visual function - that of stereo perception, - for which it has been shown by B. Julesz, using random dot stereograms (Yulesz, 1971), that object recognition is not necessary.)

The considerations above address and supposedly provide a satisfactory (at least for the level of our current knowledge of that problem) answer to question (1) stated in the beginning of this chapter. Question (2), regarding the design of efficient features for human motion recognition will be the focus of attention in the next section. However, before proceeding to the description of the *relative motion dependent features* for motion extraction, proposed in this thesis, it would be helpful to examine some of the available experimental evidence motivating our choice for the features, and also to introduce some of the relevant terminology from the psycho-physical literature, which might be unfamiliar to researchers working outside this field.

In studying the perception of motion of complex objects consisting of more than one elements (the human body performing various actions can be viewed as such one), many researchers have emphasized the importance of a distinction between three different types of motion: absolute, common and relative motion. The *absolute motion* (AM) is the exact trajectory of each separate element of a moving object, determined in an observer-relative frame of reference, i.e. it is the *absolute* movement of an element without regard to the movement of any other element. *Common motion* (CM) is the perceived movement of the whole object relative to the observer, and *relative motion* (RM) - the movement of a certain element of the object, relative to the movement of other elements of the same object. Two propositions regarding these motions are generally agreed on:

a) Absolute motions are often not seen as such - only RM and CM are usually perceived. This can be illustrated by the following experiment shown on Fig.3-4 (Cutting and Proffitt, 1982) . Three bright spots A, B and C are mounted on an unseen rolling wheel. The absolute motions of each bright spot are as follows: point A describes a cycloid, point B a curtate cycloid, and point C a line (Fig. 3-4a). The typical perception of this configuration, however, is: points A and B have relative circular motion about point C, while the figure as a whole has common motion of linear translation to the right.

b) The motion of any element of the moving object can be represented by the following relation:

$$CM + RM = AM. \tag{3-1}$$

If CM and RM are known, it is a trivial problem to calculate the absolute motion. To the contrary, the problem of determining CM and RM from AM is not as easy, since the number of different combinations of CM and RM which can sum up to exactly the same AM is indefinite.

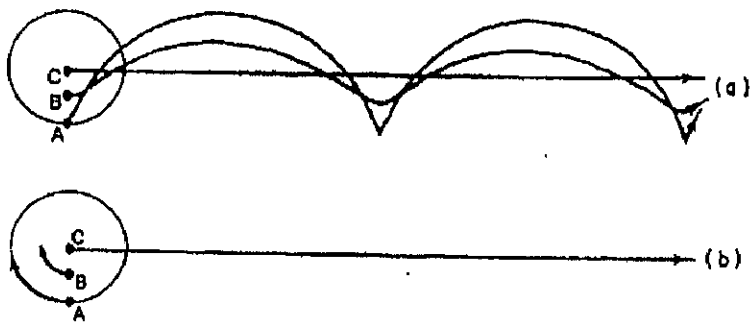


Figure 3-4. Absolute motion often is not perceived as such: (a) absolute motion trajectories for points A, B and C mounted on an unseen wheel rolling to the right; (b) the typical perception of this configuration (see text for details).

How *common* and *relative* motions are extracted from the dynamic displays, the exact order of extraction (CM first, or RM first?), and the importance of each one of those for the resultant motion perception have been the object of extensive debates in the psychophysical literature (Hochberg, 1957; Johansson, 1950/1973; Borjesson and von Hofsten, 1975; Rock, 1975; Restle, 1979; Cutting and Proffitt, 1982). Two opposing theories are generally put forward: one is that CMs are extracted first from the display, leaving RMs as the residual; the other is that RMs are extracted first from the display, leaving CMs as the residual. A third view has been proposed by Cutting and Proffitt (1982) assuming that two processes, one for RM and one for CM are started simultaneously by the visual system, and the process which reaches its solution first (or

is "minimized" first, as they put it), dictates the final percept. Nevertheless, examining numerous experiments, they conclude that in everyday perception *relative motion* is more frequently extracted first, thus determining the overall motion percept in most of the cases. One of the experiments they have considered is demonstrated in Fig. 3-5.

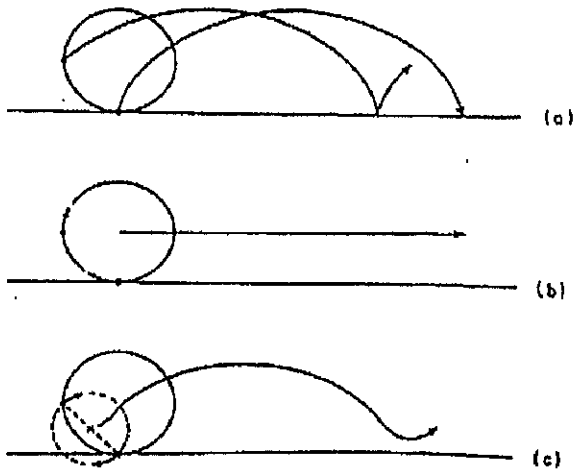


Fig. 3-5. A demonstration of relative motion being extracted prior to common motion: (a) the absolute motion paths of two lights mounted 90° apart on the perimeter of an unseen rolling wheel; (b, c) two possible perceptions of this stimulus, depending on whether CM is extracted prior to RM (version b), or vice versa (version c). Reportedly only version (c) is seen by naïve observers.

Two lights are mounted 90° apart on the perimeter of an unseen wheel rolling to the right. The absolute motion paths of the two lights yield cycloids 90° out of phase, shown in Fig.3-5 (a). Depending on whether CM or RM is extracted first (and thus determining the overall percept), two different motion patterns can be expected. If CM (linear translation to the right) is extracted first, RM would be left as residual motion and determined as a rotation about the center of the wheel (the percept described in Fig. 3-5 (b)). If RM (rotation about the point midway between the two lights) is extracted first, CM would be determined residually as the motion of the point between the two lights. The reported result favors the latter scenario, assigning a leading role to the *relative motion* extraction process. Some other types of motion (again examined in Cutting and Proffitt, 1982), more similar to the motion patterns which we would like to deal

with in this thesis, and for which it would be easy to assume priority of the RM extraction process, also include: the optical flow fields generated as one moves through the environment; mimicking the aging of a human face using cardioidal transformation of its profile (Pittenger and Shaw, 1975; Cutting, 1978); and the human walker in the Johansson's MLDs mentioned above. The importance of the *relative motion extraction* concept for human motion recognition can be perceived most easily when considered in relation to the last case above (whose similarity to any other type of human gestures or actions generated by the human body is obvious): the figural coherence and the nature of the motions performed by the human walker in the MLDs is revealed by the relative pendular motions of the (lights at the) elbows around the shoulder points, the wrists around the elbows and so on, forming a complex nested component structure of relative motions.

Bearing in mind the importance attached to *relative motion* in the psychophysical literature, and its potential priority position for forming and determining the perception of motion patterns in the visual system (and especially in the more relevant to our present interests case of human motion recognition), it is surprising that the possible contribution of relative motion to motion recognition has not yet been explored adequately in the computer vision and pattern recognition field. Investigating the applicability of relative motion dependent feature extraction to gesture recognition in particular (and to motion recognition in the more general case) will be one of the primary concerns in this thesis. A more detailed account of the design of the features we propose will be given in the following section, while possible further extensions of that approach and some alternative methods will be discussed in the last section of this chapter.

3.2. Primitive features for relative motion dependent feature extraction

Feature selection is one of the most important (although comparatively less developed) aspects of pattern recognition. Its importance lies in the following facts:

- a feature extraction process utilizing properly selected and efficient features provides a large reduction of the dimension of the corresponding space in which the patterns are treated, effectively filtering out that aspect of the input patterns which is not related to the task to be solved;
- properly selected features cluster better in the vastly reduced feature space than in the original pattern space, allowing simpler decision surfaces to be used for the recognition process;
- the use of improper or ineffective features necessarily will lead to larger classification errors and less efficient recognition, even if good classification algorithms are used.

Unfortunately, "good features" are most often problem-dependent, i.e. no general all-purpose features are known to exist, and the feature design process very often has to be guided only by intuition and trial-and-error procedures. How to define primitive features with which to represent naturally and efficiently motion in dynamic scenes (especially in the case of human motion) is one of the main problems which we have tried to address in this thesis.

3.2.1. Higher-Order Local Autocorrelation Features

Our first choice in the search for suitable primitive features was to use Higher-Order Local Autocorrelation (HLAC) functions. The use of these functions in pattern recognition was first suggested by Horwitz and Shelton (1961) and McLaughlin and

Raviv (1968), while a practical simplification has been proposed by Otsu (Otsu et al., 1978; Otsu, 1981) and successfully applied to many pattern recognition problems like character recognition (Otsu et al., 1981), face recognition (Otsu and Kurita, 1988; Kurita et al., 1992), texture recognition (Kurita and Otsu, 1993), etc.

The N-th order autocorrelation function $R(\boldsymbol{B})$ of an image $f(\boldsymbol{r}) = f(i, j)$ defined in two dimensional Euclidean space E^2 is given by

$$R(B) = R(b_1,\ldots,b_k) = \int_{E^2} f(r)f(r+b_1)\cdots f(r+b_k)dr \tag{3-2}$$

where $B = (b_1,\ldots,b_k)$ is a $2\times N$ matrix of directional displacement values. A well-known property of the functions (3-2) is that they are shift-invariant (or translation invariant) in the sense that $f(\boldsymbol{r})$ and $f(\boldsymbol{r}+\tau)$ have the same N-th order autocorrelation function, i.e. the results of the calculation are irrelevant to *where* the objects are located in the image. Since the number of the autocorrelation functions (3-2) obtained for different combinations of the displacements B would be enormous, for practical applications it has been proposed (Otsu, 1981) to restrict the order N up to the second order (N = 0, 1, 2) and the range of displacements to within a local 3×3 window, the center of which is the reference point. After eliminating the displacements which are rendered equivalent by the shift, the number of the patterns of the displacements can be reduced to 25, as shown in Fig. 3-6.

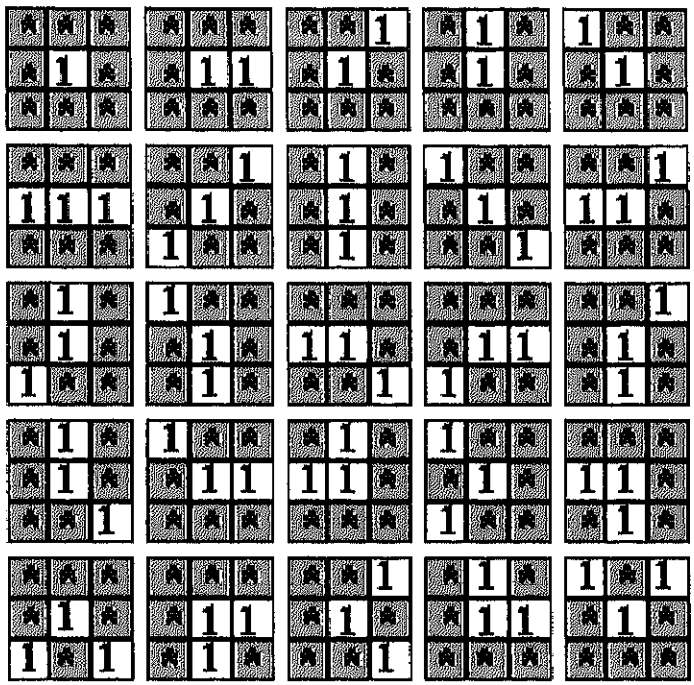
In the discrete case, the primitive features corresponding to the j-th mask pattern in Fig. 3-6 are calculated as

$$R_j = R(B_j) = \sum_r f(r)f(r+b_1)f(r+b_2), \tag{3-3}$$

$$j: \quad 1, \ldots, 25;$$

where $B_J = (b_1, b_2)$ determines the positions of the pixels with value '1' in the mask patterns of Fig. 3-6, relative to the pixel at the center of the mask which is always considered to be '1'.

Local mask patterns for the HLAC features



$N = 0, 1, 2$; * : don't care

Figure 3-6. Local mask patterns for the HLAC feature extraction. The image is scanned with the 25 local masks and the features computed as the sum of the products of the gray values of the pixels corresponding to the "1"s.

An early version of the gesture recognition system proposed in this thesis was built using the features (3-3) and showed some success when tested with the multimodal database of gestures (MMDB) described in chapter 6 (where the gestures are performed in front of an uniformly black background and all subjects wear uniform color clothes; see Fig. 6-1 for some examples), and also with some other types of simple motion data

(puppets of different form and color, e.g. a koala, bear, penguin, etc., being moved in front of an uniformly black background). However, when challenged with real-world data performed in the conditions of normal office environment (i.e. different backgrounds and illumination conditions, different colors and textures of the objects or clothes of the performers, different speed of performance, etc.) the performance degraded significantly (from nearly 100% to less than 50%) disclosing the following problems which render the features (3-3) unsuitable for the task pursued here:

(a) the grayscale (or color) multiplication in (3-3) leads to dependence of the feature values on changes in texture, background and the illumination conditions, which is unacceptable if the system has to operate under real-world conditions;

(b) if instead of grayscale (or color) images, binary motion images are used (created by using image differencing followed by thresholding, as will be explained below), lack of motion (i.e. '0' values) at certain locations *relative to* other locations (with '1's) might contain valuable information about the motion patterns (as will be argued below), but will be lost in the multiplications by '0' in (3-3);

(c) the local masks described in Fig. 3-6 might be suitable for certain problems like character recognition, face recognition or texture recognition, in which local features seem to be relatively more important, but in the case of gesture recognition, especially when relatively broad gestures are concerned, features at a more global level might be more suitable;

(d) the pattern masks in Fig. 3-6 extract structural information from static objects, but recognition of motion is intrinsically different from that (although attempts to represent human motion as a sequence of static postures also exist). Since motion is closely connected with changes over time, an important problem is how to incorporate time-related information into the feature extraction process.

Notwithstanding the above-mentioned problems which are encountered when the features (3-3) are applied to gesture recognition, this approach has the following very useful properties which we will try to retain in the new features proposed below:

(a) HLAC are shift-invariant, which is an important requirement if the gestures are to be recognized correctly independently of the exact location at which they are performed;

(b) the feature extraction does not necessitate the images to be pre-segmented into different elements, which might be difficult and time consuming to achieve;

(c) the feature extraction can be performed in a single pass over the whole image, which is computationally inexpensive and permits operation of the system in real-time;

(d) it is easy to combine the primitive features above (using some form of learning, as will be explained in the next chapter) into new features which are more effective for the concrete task to be solved.

3.2.2. Relative motion dependent features

After experimenting with different algorithms, the basic approach we have found to be most useful for the current task, is to form a set of pattern primitives, which are statistically integrated to obtain information about the quantity and type of *relative motion dependent changes* in consecutive image frames. This method will be described in the rest of this section.

The first step in the algorithm we propose here is to form *binary motion images* from the grayscale (or color) image sequence input from the camera. This step, motivated by the considerations mentioned in the beginning of this chapter, is illustrated in Fig.3-7. Let $I(i, j, t\text{-}1)$, $I(i, j, t)$ and $I(i, j, t\text{+}1)$ be three image frames, obtained correspondingly at time t-1, t and t+1.

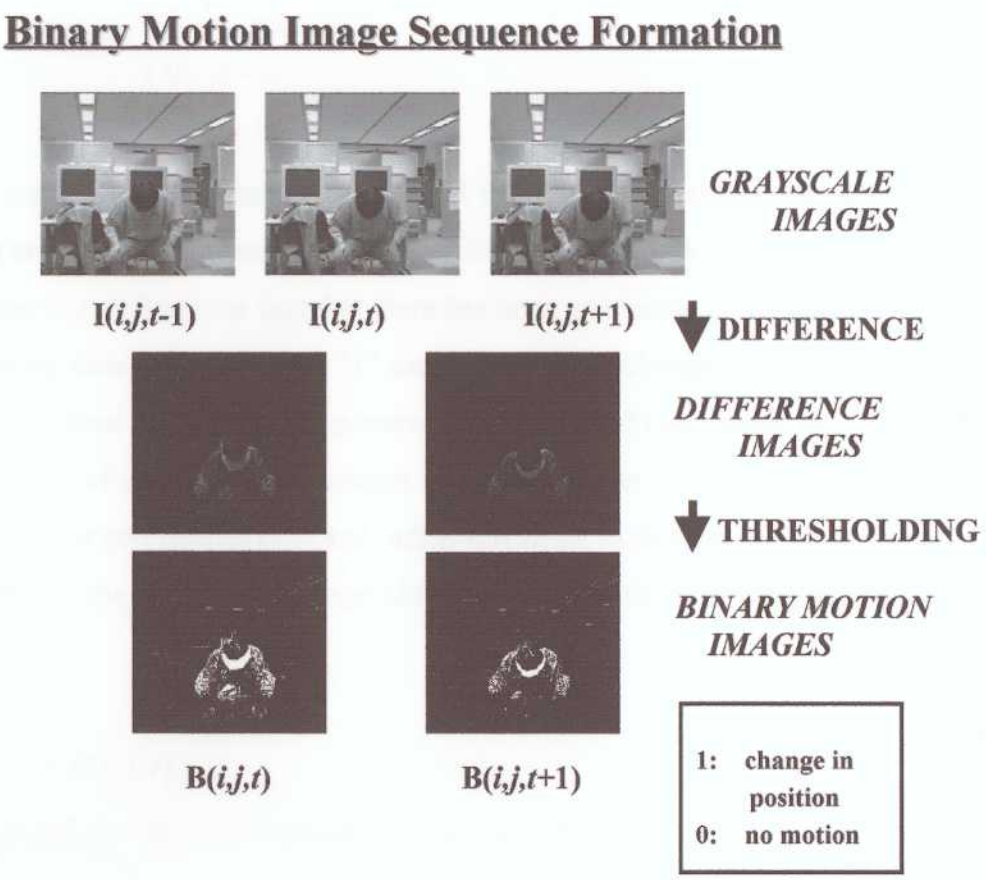


Figure 3-7. *Binary motion* image sequence formation. From the grayscale images (with 256 gray-scale values being used) at the top, *difference images* are obtained by differencing, from which *binary motion images* are formed by thresholding. Several frames from the gesture "bow" are shown in this example. In the binary motion images, a '1' value indicates the fact that there has been a change with time at the corresponding location, while a '0' value implies that the element at that position has remained static during the corresponding time interval.

From these images, after differencing followed by thresholding, the following two binary images are obtained:

$$B(i, j, t) = T(\,|\, I(i, j, t\text{-}1) - I(i, j, t)\,|\,); \qquad (3\text{-}4)$$
$$B(i, j, t\text{+}1) = T(\,|\, I(i, j, t) - I(i, j, t\text{+}1)\,|\,),$$

where the threshold function $T(x)$ is defined as:

$$T(x) = \begin{cases} 1: & x \geq a; \\ 0: & x < a. \end{cases} \tag{3-5}$$

and a is a threshold parameter (a constant value of a = 20.0 has been used throughout the experiments mentioned in this thesis). In $B(i, j, t)$ and $B(i, j, t+1)$, a "0" value at coordinates (i, j) reflects the fact that there has been no change at that location for the corresponding time interval, while "1" means that some change has occurred. The image processing steps described by expressions (3-4) and (3-5) are shown in Fig.3-7.

In each of the two binary images in (3-4), consider an area represented by a circle with center at coordinates (i, j) and radius l. If $B(i, j, t)$ denotes the binary pixel value at the center of the circle in the image obtained at time t, the following pixel functions can be formed:

$$\begin{aligned} & p_1(t) = B(i, j, t); \\ & p_2(l, n\theta, t) = B(i + int(l \cos n\theta), j + int(l \sin n\theta), t); \\ & \qquad\qquad n: 0, 1, 2, \ldots, \frac{360^o}{\theta} - 1; \end{aligned} \tag{3-6}$$

where $int(x)$ returns the integer value of x, l is a scale parameter (also the radius of the circle area), and θ is a directional (angular) resolution parameter (a factor of 360).

By shifting the circle's center (i, j) simultaneously across the two images, and considering the relations between the pixels at the center (the *reference pixels* $p_1(t)$ and $p_1(t+1)$), and the pixels $p_2(t)$ and $p_2(t+1)$ lying on the circumference of the circle, the following features F are extracted:

$$F(u_1,u_2,v_1,v_2;l,n\theta,t) = \frac{1}{Z}\sum_{i,j}\Gamma\{[p_1(t),p_2(l,n\theta,t),p_1(t+1),p_2(l,n\theta,t+1)]\mathbf{NXOR}[u_1,u_2,v_1,v_2]\}; \quad (3\text{-}7)$$

$$u_1,u_2,v_1,v_2 : 0,1;$$

where $[u_1,u_2,v_1,v_2]$ represents the different types of relative binary changes,

$$\Gamma\{\alpha_1,\alpha_2,...,\alpha_N\} = \begin{cases} 1 : \alpha_1 = \alpha_2 = ... = \alpha_N = 1; \\ 0 : otherwise, \end{cases} \quad (3\text{-}8)$$

and $\{(x_1,x_2,x_3,x_4)$ ***NXOR*** $(y_1,y_2,y_3,y_4)\}$ means that a bitwise logical **NXOR** is being performed between x_i and y_i (***NXOR*** is a Boolean operator, such that when the inputs are equal, e.g. both '0' or both '1', the result is '1'; when the inputs differ, the result is '0'). The normalization factor Z in (3-7) is taken to be the average value of the number of '1's in the frames $B(i,j,t)$ and $B(i,j,t+1)$. This normalization is necessary, to compensate for possible differences in size and texture of the moving objects. The mechanics of the extraction process are demonstrated graphically in Fig.3-8.

For different values of the parameters l and $n\theta$, and for all possible combinations of u_1, u_2, v_1, and v_2, different types of primitive features are obtained, each individual feature forming a separate dimension in primitive feature space ***F***. The calculated feature values along each dimension represent the frequency of occurrence of the corresponding relative change patterns. Thereby, during the feature extraction process, for each training sample from each gesture class, the instantaneous "energy" dependent on the relative-motion changes for the time period between t-1 and t+1 is calculated and represented as one value of the feature vector $\Phi(r, s, t)$, where r (r =1,..., number of gesture classes), s (s = 1,..., number of training samples for each class), and t (t = 1,..., number of frames for each training sample) successively take all possible values as the

feature extraction is carried out. As time changes, $\Phi(r; s, t)$ depicts a trajectory in primitive feature space.

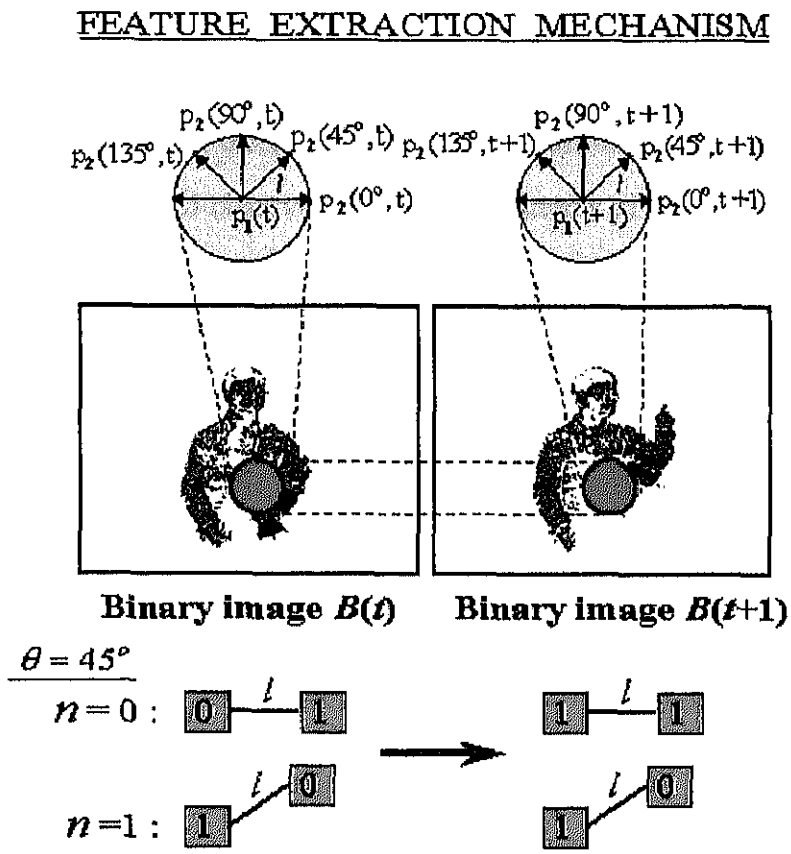


Figure 3-8. Feature extraction mechanism for the relative motion dependent features (shown for the gesture "point to the left" for $\theta = 45^\circ$, $l = 20$, K = 2, X = 2; see text for details). All possible binary transition patterns $[p_1(t),\ p_2(l,n\theta,t);\ p_1(t+1),\ p_2(l,n\theta,t+1)]$ for different values of $n\theta$ and l are extracted and integrated, as the circular areas are shifted across the sequential images $B(t)$ and $B(t+1)$. At the bottom are shown examples for two binary transition patterns: [0, 1; 1, 1] for $n = 0$, and [1, 0; 1, 0] for $n = 1$.

The reasoning behind the feature extraction defined by (1)-(5) is that it might be more informative to assess how different local changes in motion are *related* to each other, i.e. it is not local motion itself that is considered important, but the presence or absence of local motion in one area of the scene *relative* to the presence or absence of

local motion in other areas. Since trying to depict exhaustively all possible relations between different local motion changes would be computationally implausible, in the implementation proposed here a limited set of directionally oriented spatio-temporal relations are used as features, to extract some kind of relative-motion-dependent instantaneous energy spectrum, characterizing the dynamic scene revealed with time. If oriented spatio-temporal relations are to be depicted, the simplest choice is to use only two points (the ends of an oriented line segment with length l), corresponding to $p_1(t)$ and $p_2(l,n\theta,t)$ above, and observe the different patterns of change from B(t) to B(t+1), i.e. how much of the pattern $\{p_1(t) = 0,\ p_2(l,n\theta,t)=1\}$ from B(t), for example, changes to the pattern $\{p_1(t+1)=1,\ p_2(l,n\theta,t+1)= 0\}$, or to $\{p_1(t+1)= 1,\ p_2(l,n\theta,t+1)= 1\}$, and so on in B($t$+1), considering exhaustively all possible binary pattern transitions. It should be noted that while in the feature extraction process based on the higher order local autocorrelation functions (3-3) half of the spatial orientations can be eliminated as equivalent (because of the symmetry) under translation, in the relative motion dependent features all direction have to be retained, because they are not symmetrical under translation in the same direction when the reference point and the related point(s) have different values.

The total number N of the features (3-7), corresponding to the dimension of primitive feature space (if the patterns consisting of only zeroes, and half of the patterns consisting of only '1's, i.e. being equivalent under translation in the same direction, are ignored), is given by

$$N = L\frac{360}{\theta}(2^{KX} - 1.5) \tag{3-9}$$

where L is the number of different scales (different values for the parameter l) at which the features are extracted; K is the number of pixel points used to form the binary relations (e.g. K = 2 for the features given in (3-7), but could be a larger value if more

points are used); and X is the number of consecutive binary images used. Although only two consecutive frames B(i, j, t) and B(i, j, t+1) have been used to form the set of primitives for the feature extraction process, if necessary, equation (3-7) can be easily extended to incorporate information from more frames, as shown below. For example, if (3-7) is calculated at one scale (e.g. for $l = 30$), and at directional resolution $\theta = 45^o$, from (3-9) there will be 120 different features. If three points, p_1, p_2 and p_3, are used to form the binary relations (3-7) in three consecutive binary images B(t), B(t+1), B(t+2), then (3-7) will become:

$$F(u_1,\ldots,r_3;l,n\theta,t) =$$
$$\frac{1}{Z}\sum_{t,j}\Gamma\{[p_1(t),p_2(l,n\theta,t),p_3(l,n\theta,t);p_1(t+1),p_2(l,n\theta,t+1),p_3(l,n\theta,t+1);$$
$$p_1(t+2),p_2(l,n\theta,t+2),p_3(l,n\theta,t+2)]\mathbf{NXOR}[u_1,u_2,u_3;v_1,v_2,v_3;r_1,r_2,r_3]\};$$
$$u_k,v_k,r_k : 0, 1; \quad k: 1,\ldots,3; \tag{3-10}$$

where

$$B(i, j, t+2) = T(\,|\,I(i, j, t+1) - I(i, j, t+2)\,|\,) \tag{3-11}$$

and $p_3(l,n\theta,t)$ might be formed, for example, like:

$$p_3(l,n\theta,t) = B(i + int(\frac{l}{2}\cos n\theta), j + int(\frac{l}{2}\sin n\theta), t) \tag{3-12}$$

for the case when p_1, p_2 and p_3 lie on a line. It is a straightforward matter to extend (3-7) and (3-10) to an even more general case, for arbitrary values of K and X in (3-9). However, if more than two points are used to form the spatio-temporal binary relations, numerous combinations of possible spatial patterns will exist, making it necessary to

justify one's choice of a certain pattern over the others. We have experimented with several different spatially oriented patterns (using different number of points p_2, p_3, etc. for different values of K in (3-9), forming different angles δ with the reference point p_1 as shown in Fig. 3-9), but have found most useful the features defined by (3-6) and (3-12), i.e. the case when all points lie on an oriented line segment. The requirement for real-time performance has determined our choice to use equation (3-7) for the feature extraction performed on the experimental data introduced in the section describing the experimental results, although slightly better result have been observed at off-line speed, if more points p_i, and more consecutive binary images are used to form the binary relations, i.e. if higher values are used for K and X.

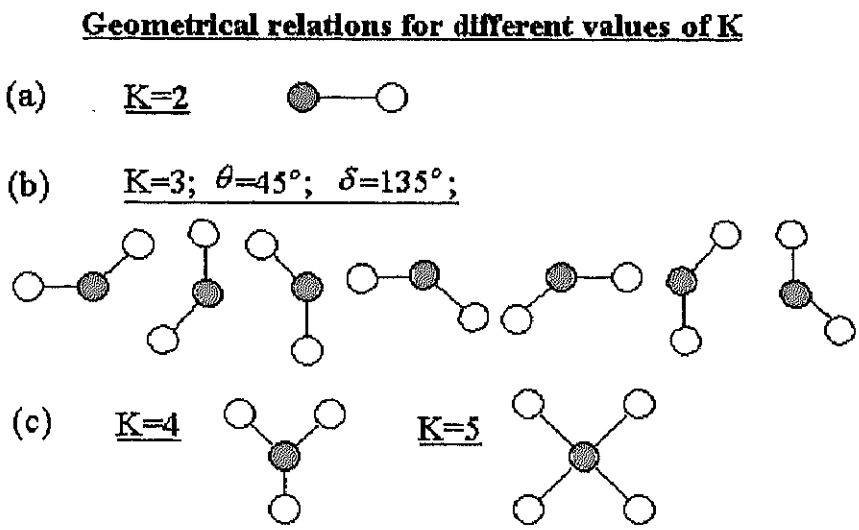


Figure 3-9. Geometrical relations between the *reference point* p_1 (shown in darker color) and the pixels $p_i(t)$ (i : 2, ... , K) defining different types of possible relations for relative motion dependent feature extraction: (a) only two points are used as in the case described by expression (3-7); (b) three points patterns are shown for several values of $n\,\theta$ (n : 1,...,7) and $\delta = 135^\circ$ (the angle formed between the three points); (c) patterns using more than three points.

We will conclude this section with some notes regarding the choice of l and θ, the two main parameters used in (3-4)-(3-12). The directional resolution parameter θ determines the sensitivity of the system to motions at different orientations relative to the reference points. For the needs of the gesture recognition experiments described in chapter 6, it was found that $\theta = 45°$ suffices, although a smaller value for θ might be used to improve directional resolution, if that is necessary (e.g. for recognition of gestures containing more detail) and if incurred increase in computational cost is not a problem.

Regarding the scale parameter l, we found that a relatively large value (e.g. 50 pixels for the 256x240 gesture images used in the experiments in chapter 6) is needed to achieve good performance, i.e. integration of relative motion changes at a more global level seems to be important for the task at hand. The dependence of the recognition rates on the value of the parameter l is demonstrated on Fig. 3-10.

The performance of the proposed algorithm improves if the features are calculated and integrated simultaneously at several different scales (i.e. for several different values of l) and all resulting features are taken to form separate dimensions in primitive-feature space. In this way, both information depending on relative motion at a more global scale (i.e. describing relatively large gesture elements) and information about motion at a more local scale (i.e. pertinent to detail) is simultaneously extracted and supplied to the learning stage of the system (described in the following subsection), where the features extracted at different scales are appropriately weighted and combined to form new features, more suitable for the task to be solved. In order to achieve a real-time performance, in all experiments mentioned in chapter 6 the features (3-7) were calculated at only two scales, i.e. the value of l has been fixed to 20 and 65 pixels.

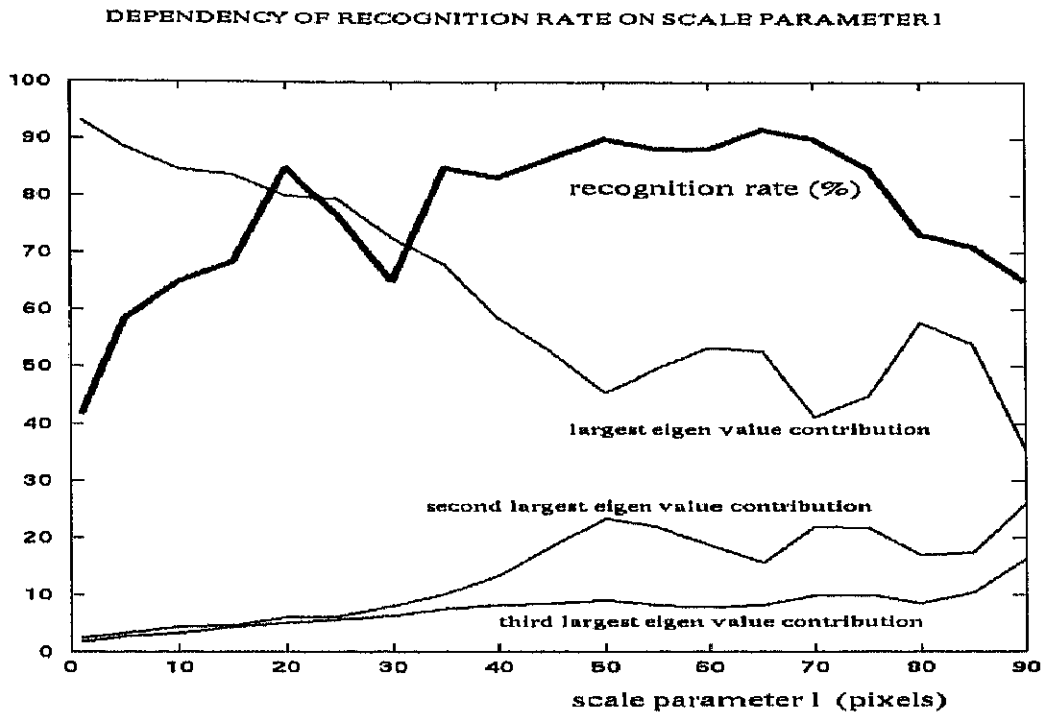


Figure 3-10. Dependency of recognition rate on the scale parameter *l*. The recognition rate for the real-world gesture database (for 6 different subjects) is shown (bold line) for different values of the scale parameter *l*. Two peaks of the recognition rate graph are observed at $l = 20$ and $l = 50...65$, showing that relative motions at the corresponding scale (for the gesture experiments introduced in chapter 6) contain the most useful for the discrimination information. Also shown are the contributions (in percents) of the three largest eigen values determined from the discriminant criterion (4-9) explained in the next chapter.

3.3. Discussion and further extensions of the approach

One obvious advantage of using our method for motion-relevant feature extraction, compared with algorithms which establish motion-correspondences in time, or computing optical flow, is that the feature extraction algorithm proposed here does not necessitate the use of complimentary constraints (e.g. assuming smoothness of motion, proximal uniformity, etc.; these assumptions might work under certain conditions, but could fail in others, e.g. abrupt motions occur frequently in gestures and they shouldn't be smoothed if we want to recognize them properly; smoothing leads to erroneous estimation of motion at occlusion boundaries, etc.) or multiple iterations to converge - all

the necessary information is calculated in only one pass, allowing real-time performance without the use of special image processing hardware. .

It should be noted that the features (3-7) do not extract relative motion in the strict sense defined in section 3-1, i.e. as "the movement of a certain element of the object, relative to the movement of other elements of the same object". To be able to compute this, first one has to identify the different elements of the object (i.e. a model of the object and its elements has to be obtained and maintained throughout the image sequence, involving additional image segmentation and object recognition steps), and after that the dynamic relations between these elements have to be formed. Obviously, this would necessitate a model-based approach, while here we propose bottom-up feature extraction (the features being dependent on the relative change patterns at different locations in the images, without identifying the individual elements of the object), thus avoiding the use of structural information, which can be difficult to obtain and manipulate robustly and in real time.

Because the features (3-7) are calculated *relative to* reference points and integrated over the whole frame (as the coordinates of the reference points consecutively take all possible values), they are *shift-invariant*, i.e. the movements to be recognized may be performed at any location inside the frame and still will be recognized correctly. This is a very important requirement if the system has to operate in real-world conditions, where it is not possible or desirable to constrain the movements of the users to a fixed spatial location.

Although the features proposed in the previous section are *shift-invariant*, in the form defined by (3-7) they are not *size-invariant*, i.e. the same gestures performed by the same subject at several very different distances from the camera will not automatically be recognized as equivalent if a constant value is used for the scale parameter l (although relatively small changes in the size of the objects and the distance from the camera are tolerable, which could be enough for applications in which the position of the subjects in depth does not change significantly). If the gestures have to be recognized

correctly independently of the size and location in depth of the subjects, one possible way to obtain size-invariance of the features (3-7), is to modify the value of l adaptively thus compensating for the changes in size of the motion patterns. There are many possible ways how to do this, but here we will describe only two algorithms which are very simple and easy to implement online.

The first algorithm evaluates the size of the subject (assuming only a single user at a time) and normalizes the scale parameter l accordingly. The size in horizontal and vertical direction can be estimated using the zeroth and first order statistical moments of the image subtracted from the background. The (p, q)th moment m_{pq} of an image function $g(x, y)$ is defined by

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q g(x,y)\,dx\,dy \qquad p, q = 0, 1, 2, \ldots . \tag{3-13}$$

and l is calculated through the following steps:

(a) subtract the low-pass filtered versions (in order to remove high-frequency noise) of an image with the subject in front of the background from an image of the background only and threshold the result to obtain a binary image of the 2D shape of the subject;

(b) calculate the centroid of the subject with coordinates $(m_{10}/m_{00},\ m_{01}/m_{00})$;

(c) calculate from the binary image obtained in (a) the horizontal histogram function $H(x)$ and the vertical histogram function $V(y)$ which give the number of '1's for each value of the coordinates x and y in the respective direction;

(d) calculate the horizontal size dimension l_x and the vertical size dimension l_y as the length of the line starting at the corresponding centroid coordinate and continuing in the opposite directions until more than 90% (or suitable threshold value) of the energy of the corresponding histogram function is contained, i.e.

$$\sum H(m_{10}/m_{00} - \frac{l_x}{2} < x < m_{10}/m_{00} + \frac{l_x}{2}) \geq 0.9 \sum_{all\ x} H(x) \tag{3-14}$$

$$\sum V(m_{01}/m_{00} - \frac{l_y}{2} < y < m_{01}/m_{00} + \frac{l_y}{2}) \geq 0.9 \sum_{all\ y} V(y) \tag{3-15}$$

(e) calculate l as a function of l_x and l_y. We have found that for the gestures experiments conducted here even only the information available from l_y suffices (i.e. the height of the subject seems more important than the width, which is reasonable having in mind that the scope of all meaningful gestures is usually within a circle with a radius of approximately half the height of the human's body) and we have used the following value for l

$$l \approx \frac{7}{10} l_y \tag{3-16}$$

which corresponds to the second peak in the recognition rates graph in Fig. 3-10 for the available experimental data .

The second algorithm can be applied if the distance to the subject is available at any time, e.g. through a sensor, using stereo information, or any other similar method. It is assumed in this case that the size of the motion patterns changes proportionally to the distance between the camera and the objects. The parameter l can be obtained through the following three steps:

(a) set the value l_1 of the parameter l at distance s_1 from the camera, e.g. from experimentally obtained data similar to that shown in Fig. 3-10, so that maximal recognition rates can be obtained if the gestures are performed at distance s_1 from the camera;

(b) calculate the scale parameter k which projects l_1 to $l_2 = kl_1$ at distance s_2 from the camera (the motion pattern itself will be scaled by k^2);

(c) it is easy to show that the value l_x of the parameter l at any distance s_x between the subject and the camera can be calculated as

$$l_x = \frac{l_1}{k} + \frac{(s_2 - s_x)(k-1)l_1}{s_2 - s_1} \quad . \tag{3-17}$$

An alternative, purely hardware solution is also possible, if a camera with zoom control is used so that the size of the subject is always kept constant.

In order to form the binary motion images introduced in the previous section, differencing of sequential images, as defined by expression (3-4), rather than differencing between the image $B(i, j, t)$ at time t and an image of the static background was used. The use of the latter approach (used very often, e.g. in Yamato et al., 1992; also other more elaborate and precise algorithms especially designed for silhouette extraction exist, like Davis and Bobick, 1998; Davis, 1998; Davis and Bradski, 1999) would have led to an extraction of the silhouette of the gesture performer, which is independent to the texture of the clothes of the subjects, but effectively "blind" to any motion performed inside the area of the extracted silhouette. For example, gestures like "clapping hands", "cross hands" or "no motion" used in the present system would have been indistinguishable from each other, leading to significant decrease in the recognition rates. Also, for the latter algorithm to work properly it's necessary that the background reference images are constantly updated over time to compensate for possible changes in the background and in the illumination conditions. On the other hand, since in the algorithm we have used, only the locations which change from frame to frame are extracted (i.e. have a value of '1'), the above problems are circumvented, and although it might seem that different types of textures would be a problem (generating different motion patterns), in the tests we have conducted we found that this is not the case, may be

partly due to the normalization in (3-7) and partly due to the effectiveness of the proposed features for which the *relative* binary transition patterns are more important than the absolute distributions of the concrete binary motion patterns.

Although only binary motion information obtained from difference images has been used as an input to the algorithm proposed here, in principle there would be no problem to extend the algorithm for use with optical flow data. The reason optical flow data has not been used for the present implementation is that still optical flow methods tend to be too unreliable for real-world imagery of human gestures due to sensitivity to noise, textures, occlusions, deformations in non-rigid motions, etc., and also computationally too expensive to allow real-time performance. If, however, these impediments are overcome in the future, the algorithm proposed in the previous section can be applied to optical flow data using some of the methods described below.

The most straightforward way would be to quantize the optical flow data into a set of integer values and after that use multivalue logic instead of binary logic for the calculations of the features. However this would increase enormously the dimension of feature space (in (3-10) 2 has to be substituted by 25) which is impractical. The special case when the optical flow data is quantized into only two values (using a suitable threshold value) can be used directly with our method and possibly would yield better input data than the differencing method used in the previous section, if of course the optical flow data itself is reliable. Another, more practical approach would be to form the relative motion relations using the dot products between the optical flow vectors at the reference point and the related points, in a manner similar to the one introduced in the previous section but using only one frame at a time. The relative motion dependent features then can be defined as:

$$F(l,n\theta)=\int_{E^2}\vec{v}_R\,\vec{v}(r+b_1)+\cdots+\vec{v}_R\,\vec{v}(r+b_N)\,dr\,; \tag{3-18}$$

$$\vec{v}_R = \begin{cases} \vec{v}(r) : \textit{if the direction of } \vec{v}(r) \textit{ quantizes to } n\theta; \\ 0 : \textit{otherwise}; \end{cases} \tag{3-19}$$

$$n : 0,\ldots, \frac{360^o}{\theta} - 1;$$

where $\vec{v}(r)$ is the optical flow's vector value at location r in the image, b is a displacement vector defined as $b_l = [int(l\cos n_l\theta), int(l\sin n_l\theta)]$, l is a scale parameter, θ is a directional (angular) resolution parameter (a factor of 360) and the integration is carried out over the two-dimensional Euclidean space of the image. In this case, instead of keeping track of the frequency of the different binary transition patterns, as was the case with the binary relative motion algorithm from the previous section, the features calculate a measure of the similarity between the optical flow values of the reference point at location r and the related points at locations $r+b_l$. Although some preliminary experiments using the features (3-18) showed promising results (having in mind that the optical flow data which was used wasn't very reliable), more extensive testing with reliable optical flow data is necessary to evaluate the usefulness of these features.