Motion-Based Recognition: A Survey*

Claudette Cédras and Mubarak Shah
Department of Computer Science
University of Central Florida
Orlando, Florida 32816-2362
email: shah@cs.ucf.edu

Abstract

Motion perception and interpretation plays an important role in the human visual system. It helps us recognize different objects and their motion in a scene, infer their relative depth, their rigidity, etc. In psychology, this process has been studied extensively by Johansson using moving light displays (MLDs). MLDs consist of bright spots attached to the joints of an actor dressed in black, and moving in front of a dark background. The collection of spots carry only 2D information and no structural information, since they are not connected. A set of static spots remained meaningless to observers, while their relative movement created a vivid impression of a person walking, running, dancing, etc. The gender of a person, and even the gait of a friend can be recognized based solely on the motion of those spots.

There are two theories about the interpretation of MLD type stimuli, from a psychology point of view. In the first, people use motion information in the MLD to recover the 3D structure and subsequently use the structure for recognition (structure from motion problem). The second theory of motion analysis deals with the direct use of motion information for recognition. In motion-based recognition approach, the emphasis is not on the static structure, and motion information is not extracted one frame at a time. Instead, a sequence containing a large number of frames is used to extract motion information in its continuum. The advantage here is that a longer sequence leads to recognition of higher level movements, like walking or running.

This paper provides a review of recent developments in the computer vision aspect of motion-based recognition. We will identify two main steps in motion-based recognition. The first step is the extraction of motion information and its organization into motion models. The second step consists of the matching of some unknown input with a model. Several methods for the recognition of objects and motions will then be reported. They include methods such as cyclic motion detection and recognition, lipreading, hand gestures interpretation, motion verb recognition and temporal textures classification. Tracking and recognition of human motion, like walking, skipping, and running, will also be discussed. Finally, we will conclude the paper with some thoughts about future directions for motion-based recognition.

*The research reported in this paper was supported by NSF grants CDA-9122006 and IRI-9220768.

Contents

1 Introduction **1**

2 Extraction of Motion Information and Matching **3**
2.1 Optical Flow 4
2.2 Motion Correspondence 5
2.3 Trajectory Parametrization 7
2.4 Relative Motion 8
2.5 Motion Events 9
2.6 Region-Based Features 10
2.7 Matching and Classification 12
2.8 Summary 13

3 Motion Recognition **13**
3.1 Cyclic Motion Detection 14
3.2 Lipreading 16
3.3 Gesture Interpretation 18
3.4 Motion Verb Recognition 20
3.5 Temporal Textures Classification 21
3.6 Summary 22

4 Human Motion Tracking and Recognition **22**
4.1 Modeling of the Human Body 22
4.2 Modeling of Human Motion 24
4.3 Recognizing Body Parts 26
4.4 Three-Dimensional Tracking 28
4.4.1 Tracking with Stick-Figure Models 28
4.4.2 Tracking with Volumetric Models 29
4.5 Human Motion Recognition 32
4.5.1 Recognition of Human Movements 32
4.5.2 Discrimination Between Humans From Their Motion 34
4.6 Summary 35

5 Conclusion and Future Directions **35**

1 Introduction

Motion perception and interpretation plays an important role in the human visual system. We have the ability to recognize a distant walking person by his/her gait, particular hand gestures, dance steps, flying birds, all of which are made up of a complex sequence of movements. Motion perception helps us recognize different objects and their motion in a scene, infer their relative depth, their rigidity, etc. Our visual system is very sensitive to motion, and we tend to focus our attention on moving objects. Motionless objects, in a scene, are not as easily detectable, and several camouflage strategies of the animal kingdom rely on that fact. Our ease of perception and interpretation of motion suggests that our visual system is very well adapted to process temporal information.

In psychology, motion perception has been studied extensively using Johansson's moving light displays (MLDs) [31, 32]. MLDs consist of bright spots attached to the joints of an actor dressed in black, and moving in front of a dark background. The collection of spots carry only two dimensional information and no structural information, since they are not connected. A set of static spots remained meaningless to observers, while their relative movement created a vivid impression of a person walking, running, dancing, etc. The gender of a person, and even the gait of a friend can be recognized based solely on the motion of those spots [8]. It has been shown that inverted (upside down) MLDs are usually not recognized, even for some simple movements [69]. This would suggest that the familiarity of an observer with a particular motion plays an important role in the ease with which one can recognize it: an inverted movement is not natural nor familiar, thus it is more difficult to recognize. Nevertheless, our easy recognition of MLDs would indicate that we can directly use motion as a means for recognition.

There are two theories about the interpretation of MLD type stimuli. In the first, people use motion information in the MLD to recover the three dimensional structure, and subsequently use the structure for recognition (structure from motion). In this case the moving object would be identified first, then the motion it performs in the image sequence would be sought. According to the second theory, motion information is directly used to recognize a motion, without any structure recovery.

There has been significant interest over the last decade, in the computer vision community, in the determination of structure from motion (SFM) (e.g. [27, 34, 66, 68]). In SFM, the three dimensional coordinates of points on the moving objects and their three dimensional motion is recovered from a sequence of frames. This problem is formulated in terms of systems of nonlinear or linear equations given two dimensional positions of moving points among a few frames. Interesting theoretical work (e.g. [10, 66, 74, 76]) related to the number of points required for a solution, the uniqueness of such a solution, and the effect of noise on the solution has been studied. In these approaches, it is assumed that the recovered three dimensional structure will subsequently be used for recognition. However, three dimensional structure is not sufficient alone for robust and accurate recognition, and the reconstruction is sensitive to noise. Multiple cues like motion, specularities, textures, etc., are needed. The structure from motion methods compute intrinsic surface properties, such as depth. But depth maps and other maps of 2.5D sketch are still basically images, which still need to be segmented and interpreted before they can be used for more sophisticated tasks.

Another approach for motion analysis deals with the direct use of motion information for recognition, as our easy recognition of MLDs would suggest. In this approach, emphasis is not necessarily on the static structure, and motion information is not extracted one frame at a time. Instead, a sequence containing a large number of frames is used to extract motion information in its continuum. The use of a longer sequence leads to recognition of higher level movements, like walking or

running. Those movements consist of a complex and coordinated series of events that cannot be understood by looking at only a few frames. Therefore, more complex movements can be examined at a more appropriate level.

Motion-based recognition consists of the recognition of objects or motions directly from motion information extracted from the sequence of images. Knowledge about the object or motion is used to construct models that will ultimately serve in the recognition process. There exists a distinction between *motion-based recognition* and *motion recognition*: motion-based recognition is a general approach that favors the use of motion information for the purpose of recognition, while motion recognition is one goal that can be attained with that approach.

There are two main steps in motion-based recognition. The first step consists of finding an appropriate representation for the objects or motions we want to model, from the motion cues of the image sequence. Those representations can be relatively low-level, for instance the trajectory of a particular point on a moving object, a speed or direction throughout the sequence, and can, if necessary, be organized into very high-level representations, for example the scenarios in Goddard's work [24, 23], or motion verbs as described by Koller et al. [37] and Tsotsos [72, 73]. The low-level processing of the images consists of the extraction of features, which are "manipulated" and organized into those representations. The models are then created and extended as necessary. The second step consists of the matching of some unknown input with a model. The methods here are more standard, and are often common pattern classification techniques. The term recognition is sometimes used as an equivalent to classification, although a distinction must be made. Feature vectors are classified, i.e. associated with a cluster representative, according to a distance measure to that cluster. Clusters can also be grouped or split, depending on a predefined distance measure, or according to some parameter introduced by the user. Recognition, on the other hand, is an association to only one possible model. If the input's representation does not "fit" a model's representation, then it is not recognized as such, however close it might be. In this paper, recognition is used in the usual broader sense that includes classification or association with a particular class of object or motion.

Another way to use motion for recognition is to explicitly use shape and motion models to predict and recover the motion performed by an object. A motion here is defined by a sequence of the specification of parameters defining the shape of an object in time. Instead of using the motion information from an image sequence to create representations that will be used to recognize an object or motion from models, shape and motion models of an object are used to recover the shape parameters of the object, throughout the sequence. This approach is often designated as tracking. Most of the work done in this area pertains to the walking motion of a person. A precise body model and motion model of walking are first defined and are then used, along with information from the image sequence, to determine the correct sequence of body configurations for the walking motion.

Applications for motion-based recognition are wide. It has already been used in the medical field for the study of left ventricular motion, to reveal damages, impairments or abnormalities [72, 73]. The framework developed can furthermore be adapted to other types of diagnostic methods. In clinical gait analysis, the location of various joints of a patient's body are tracked and analyzed for abnormalities using a computer. The study of the relative joint angles throughout the cycle, along with some other type of tools like electromyographical activity of involved muscles, provides a basis for the comparison of normal and abnormal gaits and can pinpoint the location of problems. For example, the assessment of the locomotion of patients suffering from cerebral palsy can help in the determination of appropriate surgical or orthotic interventions; progression of neuromuscular disorders can be examined; the evaluation of the effectiveness of a prosthetic joint replacement, the

improvements of orthotic design, the changes in prosthetic design, all help in the physical therapy of joint diseases, biomechanical simulations, kinesiological analysis and ergonomic designs [51, 29, 11].

Similarly in sports and athletic training, some initial systems already exist, for example, to digitize the path of a javelin and compute its velocity and angle of attack, and to compare values against a mathematical model of the perfect javelin throw. Biomechanists use underwater cameras and computer systems to produce three dimensional coordinates of the swimmer's arm, and calculate the curve from the velocity of the hand, the angle of pitch, and the speed of the swimmer [51]. In the future, systems could be built for analyzing golf swings or tennis strokes. Motion-based recognition systems could also be used in the analysis of dance movements and as an instructional tool for dancers.

Systems for the automatic interpretation of spoken words (lipreading) and of sign language are under way. One application is in the development of better human-machine interfaces. Another use would allow hearing impaired people to communicate more easily by using visual phones. Telephone networks cannot support real-time graylevel image transmission, however the requirements in precision and resolution for American Sign Language are relatively low as compared to grayscale images [67]. Methods that create some image compression could be used for real-time sign language communication over the phone.

Surveillance systems, for which simple motion detection might not be sophisticated enough, could be another application. Motion recognition techniques could help disambiguate between possible or allowable types of motion from a non-desirable type of motion in a particular scene. In automatic monitoring systems, those techniques could help locate problems leading to possible failures in an automated line; given the constraints of the allowed machine motions, an unexpected motion, or lack of motion, could be isolated faster, and therefore be attended to without waste of time. Other uses for motion-based recognition systems include obstacle avoidance of moving objects for robots, and satellite monitoring of weather disturbances.

This paper will survey different methods used for motion-based recognition. Section 2 will tackle the problem of motion representation i.e., how to extract the spatial and temporal features from a sequence, and how to organize them into a coordinated set. Once a representation is defined, a set of models can be built up using the encoded features, then unknown sequences, following the same process of encoding and organization, can be compared for matching and recognition. The matching step is also described in section 2. Section 3 will be concerned with the description of several methods for motion recognition. Section 4 will describe methods for the tracking and recognition of human motion. We will finally conclude in section 5 with a brief summary, and future directions for motion-based recognition in computer vision applications and research.

2 Extraction of Motion Information and Matching

The first important step in motion-based recognition is the extraction of motion information from a sequence of images. There are generally two methods for extracting two dimensional motion: motion correspondence and optical flow. Motion correspondence is concerned with the matching of characteristic tokens through time, while optical flow consists of the computation of the displacement of each pixel between frames.

Motion correspondence deals with extracting interesting points, characteristic features in an image, that can be tracked in time. This correspondence for multiple frames results in what is called a motion trajectory, i.e. a sequence of locations (x_i, y_i), for $i = 1 \ldots n$, where n is the number of frames in the sequence. A motion trajectory can thus be considered a vector valued

function, that is, at each time we have two values x and y. However, a single valued function is better suited for computations, and therefore parametrization of trajectories is necessary. The trajectories can be parametrized in several ways, for instance speed and direction, velocities v_x and v_y, and spatiotemporal curvature. Parametrized representations can be analyzed to identify important motion events. Motion events can be seen as particular occurrences happening in the motion, for example a change in the direction, a stop, an acceleration. Gould, Rangarajan and Shah [25] and Gould and Shah [26] identify motion events by detecting zero-crossings in v_x and v_y. Engel and Rubin [19] use a polar velocity representation to determine five types of motion events: smooth starts, smooth stops, pauses, impulse starts and impulse stops. Parametrized representations can also be used in matching and recognition. Rangarajan, Allen and Shah [59, 58] used the scale-space of speed and direction to match two trajectories, while Allmen and Dyer [3] use the scale-space of the curvature of a spatiotemporal curve to detect cyclic motion.

Optical flow can be computed from a sequence by considering two consecutive frames at a time. Several researchers used optical flow to extract two dimensional motion, and from which more elaborate motion representations are later built [2, 43, 44, 47, 54, 56].

Extraction of motion information over a region or over a whole image can also be used, as opposed to motion trajectories that carry information about a single point on an object. Features derived from the use of an extended region or from a whole image are here called region-based features. For instance, Polana and Nelson [55, 47] compute several features from the normal flow (component parallel to the gradient) of the whole image. One of them is the average flow magnitude divided by its standard deviation. Martin and Shah [43], using dense optical flow fields over a region, perform correlation between different sequences for matching. Storing different views of a non-rigid object, and expressing motion of that object as a sequence or combination of those views is another approach that has been used with both binarized and graylevel images [52, 16, 17]. Yamato et al. [75] computed a mesh feature from each binarized image of a sequence, which consists of an ordered set of ratios of black to white pixels, each from an element of an image divided into a grid. Another way to use grayscale images would be to compute a set of eigenvectors (eigen images) forming an orthonormal basis, and expressing an image as a function of those, as used by Kirby et al. [35].

Figure 1 presents an overview of extraction of motion information from a sequence of frames. The different features extracted and their organization will be further discussed in this section.

Once the motion information is extracted and organized, matching between a model and an input needs to be performed for recognition or classification. Several methods use clustering techniques since their models and inputs are encoded by feature vectors, but other approaches have also been described, for example a probabilistic method, and a connectionist method.

In this section, we will examine the kind of information that can be extracted from a sequence of images, and how recognition or classification takes place. Section 2.1 will briefly describe methods for the computation of optical flow, while section 2.2 discusses methods for generating trajectories from an image sequence. In section 2.3, procedures for extracting information from those trajectories will be reported. Relative motion will be discussed in section 2.4, motion events and region-based motion features will be the topics of sections 2.5 and 2.6, respectively. Matching techniques will be covered in section 2.7, and a summary of the section will follow.

2.1 Optical Flow

Optical flow methods are very common for assessing motion from a set of images. Optical flow is an approximation of the two-dimensional flow field from image intensities. Several methods have been developed (see Barron et al. [9] for a recent review), however, accurate and dense measurements are

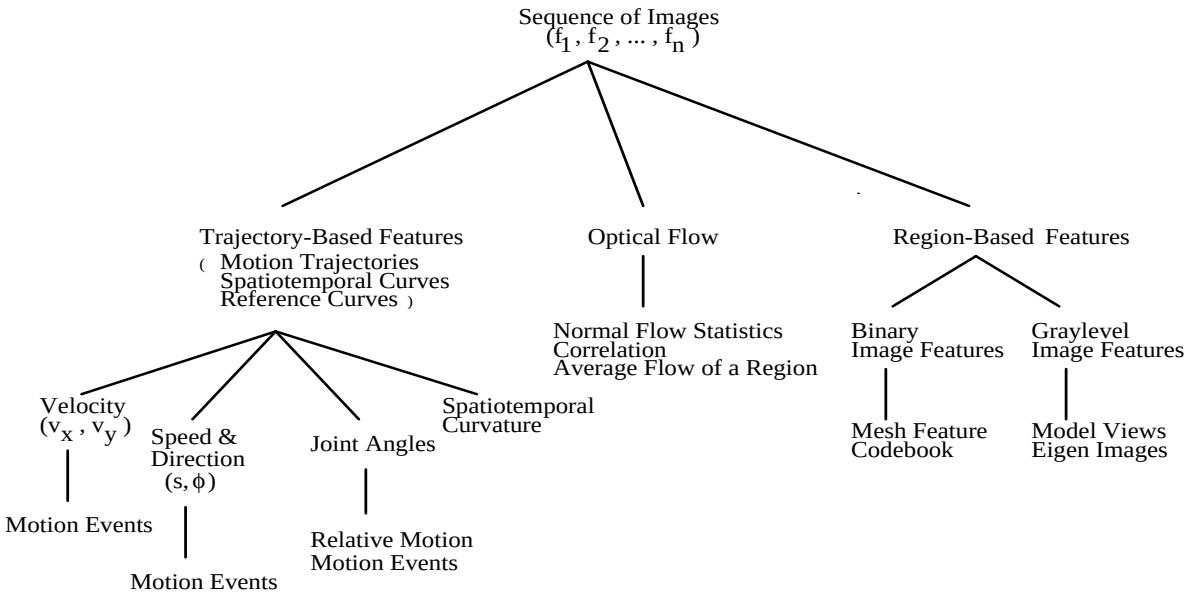

Figure 1: Extraction of motion information from a sequence of frames.

difficult to achieve. The methods are divided into four classes: differential methods, region-based matching, energy-based techniques and phase-based techniques. Differential methods compute the velocity from spatiotemporal derivatives of image intensity. Methods for the computation of first order and second order derivatives were devised, although estimates from second order approaches are usually poor and sparse. In region-based matching, the velocity is defined as the shift yielding the best fit between image regions, according to some similarity or distance measure. Energy-based (or frequency-based) methods compute optical flow using the output from the energy of velocity-tuned filters in the Fourier domain, while phase-based methods define velocity in terms of the phase behavior of band-pass filter outputs, for example the zero-crossing techniques.

One problem with optical flow in general, is that it is susceptible to the aperture problem, which, in some conditions, only allows the precise computation of the normal flow, i.e. the component parallel to the gradient. It is also prone to boundary oversmoothing, i.e. background pixels around object boundaries might have a non-zero flow value. Problems also arise with multiple moving objects, where segmentation can be difficult to achieve, and by the multiple legitimate velocities in a small neighborhood that might occur as a result of transparency. Despite all, it has successfully been used as a source of information; methods based on optical flow are described in section 2.6.

2.2 Motion Correspondence

The trajectories derived from the location of particular points on an object, in time, are very popular because they are relatively simple to extract, and their interpretation is obvious. The generation of motion trajectories from a sequence of images typically involves the detection of tokens in each

frame, and the correspondence of such tokens from one frame to another. The tokens need to be distinctive enough for easy detection, and be stable through time so that they can be tracked. Tokens include edges, corners, interest points, regions and limbs. Corners are points where the gradient direction changes rapidly, and correspond to physical corners of objects. They are useful for scenes containing polyhedral objects, automobiles, etc., but might be rare in other types of scenes. Moravec's interest operator [45] is another feature detector that has proved useful in many applications.

The correspondence problem can be defined as: given n frames taken at different time instants, and given m points in each frame, map a point in one frame to another point in the next frame such that no two points map onto the same point. This problem is combinatorially explosive, and the occurrence of occlusion and disocclusion also adds to the difficulty of the problem. One needs to introduce constraints to limit the search space. Constraints include *proximal uniformity* [60], *maximum velocity, small velocity change* or *smoothness of motion* [13, 65], *common motion, consistent match, rigidity*, etc. [7]. An important issue in the correspondence problem is to convert the above *qualitative* heuristics (constraints) into quantitative expressions, which become cost functions. Jenkin [30] proposed an algorithm for tracking the three dimensional location of points from a stereo view of the two dimensional location in the images, which is based on a general smoothness assumption, stating that the location, scalar velocity (speed) and direction of motion of a given point are relatively unchanged from one frame to the next. Sethi and Jain [65] used the principle of path coherence and of smoothness of motion as the basis for their algorithm for monocular image sequences, called the *Greedy Exchange*. They proposed an iterative optimization algorithm to find optimal trajectories, in order to maximize the smoothness of each trajectory and of the set of trajectories. Rangarajan and Shah [60] proposed a method based on a proximal uniformity constraint, which says that most objects in the real world follow smooth paths and cover a small distance in a small amount of time. The resulting trajectories are smooth and uniform and do not show abrupt changes in the velocity vector over time. Their algorithm minimizes a proximal uniformity cost function and establishes correspondence over n frames. Cheng and Aggarwal [13] devised a method for the correspondence problem using a two stage algorithm. The first stage is the sequential forward searching algorithm, which extends trajectories up to the current frame, and the second stage is a batch-type rule-based backward correcting algorithm, whose purpose is to correct the wrong correspondences among the last few frames. The only assumptions made are that of the smoothness of motion, for the first stage, and simple error (no chain error) for the second stage. In all the methods above, the smoothness of motion constraint was defined in a very similar manner, and so far this assumption has been very useful. However, it might not necessarily be true in all conditions, and sometimes abrupt motion might need to be interpreted as such. In those cases, the smoothness assumption will not be sufficient and the algorithms will probably fail.

In the papers surveyed, several methods were developed for token extraction and trajectory determination. For instance, a trajectory can be extracted from a spatiotemporal cube (ST-cube), which is the representation created by stacking a sequence of image frames. In Allmen [2], the instantaneous motion of each point in an ST-cube is first computed to produce what is called the spatiotemporal surface flow (ST-surface flow) $F(\mathbf{x}) = (\Delta x, \Delta y, \Delta t)$, where $\mathbf{x} = (x, y, t)$. A spatiotemporal curve (ST-curve), defined as a three dimensional curve $\alpha(t) = (x(t), y(t), t)$, is a trajectory through the ST-cube such that time is strictly increasing, and such that the tangent vector at a point on the curve is equal to the ST-surface flow. Another example for determining trajectories is to track the centroid of some "blobs" approximately indicating the presence of motion or of some object. Polana and Nelson [56] compute the normal flow direction and magnitude between successive pairs of frames; pixels with significant motion are marked and their centroid

determined and tracked. An approximate trajectory is then computed by fitting a line through the sequence of centroids. Davis and Shah [18], in their hand gesture recognition algorithm, use the direction of motion and displacement of each finger. The fingertips were marked and tracked through the sequence by locating the centroids of the markers after a binarization step, using the algorithm proposed by Rangarajan and Shah [60] as described above. Koller et al. [37] used methods originally developed by Kories and Zimmermann [38] and Sung [70] to generate the trajectories subsequently used for the recognition of vehicle trajectories using motion verbs. The trajectories of the vehicles are determined by first classifying each pixel into one of eight categories, according to local gray-level information. The centroid of connected pixels belonging to the same class (blobs) is computed, and then tracked along a number of frames. Their displacement is mapped to a vector chain, and the starting position of that vector along with the total displacement define an image displacement cluster.

As can be seen, the tracking of some points throughout the sequence can be performed in a variety of ways. In the case of MLD type of input, the location of tokens is already given, such that the correspondence process can be performed directly. In other cases, interest points, centroids and other tokens have to be extracted first, in a robust and consistent way, before the correspondence process can take place.

2.3 Trajectory Parametrization

Simple trajectories do not usually provide sufficient information for recognition and, as mentioned earlier, they are vector valued functions, which are not as easy to work with as single valued functions. Parametrization of the trajectories is thus very useful. One representation is the trajectory velocity, v_x and v_y, i.e. the velocity in x and in y relative to time. They are simply computed as follows:

$$v_x = \frac{x_i - x_{i-1}}{\Delta t} \quad \text{and} \quad v_y = \frac{y_i - y_{i-1}}{\Delta t},$$

where (x_{i-1}, y_{i-1}) and (x_i, y_i) are the location of a point in frames $i-1$ and i. The velocity is translation invariant, but not rotation nor scale invariant. Speed and direction are another useful parametrization. They are defined as:

$$\begin{aligned} s_i &= \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}, \\ d_i &= \arctan\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right), \end{aligned}$$

where s_i is the speed and d_i is the direction of a point at frame i, x and y are as above. Speed and direction are both translation and rotation invariant. Furthermore, the direction is scale invariant, but not the speed. The direction is however more sensitive to noise, due to the nonlinear operation of *arctan*. Velocity parametrization and speed and direction parametrization are fairly easy to compute from the trajectories, and generate curves which are easy to interpret. However it is not obvious how to combine those trajectories for higher level interpretation, for example to detect cyclic motion.

The spatiotemporal curvature κ of a trajectory is another common representation. It is determined as follows:

$$\kappa = \frac{\sqrt{A^2 + B^2 + C^2}}{((x')^2 + (y')^2 + (t')^2)^{3/2}},$$

where

$$A = \begin{vmatrix} y' & t' \\ y'' & t'' \end{vmatrix}, \quad B = \begin{vmatrix} t' & x' \\ t'' & x'' \end{vmatrix}, \quad \text{and} \quad C = \begin{vmatrix} x' & y' \\ x'' & y'' \end{vmatrix}.$$

The notation $|\cdot|$ denotes the determinant. A discrete approximation is used to compute the derivatives, for example, $x'_i = x_i - x_{i-1}$ and $x''_i = x'_i - x'_{i-1}$. It is assumed that Δt is constant, so $t' = 1$ and $t'' = 0$. It has been shown in differential geometry that any space curve is completely defined by its curvature and torsion, up to a rigid rotation and translation. The curvature has the advantage over other trajectory parametrizations that a single function is able to capture information, as opposed to two functions with v_x and v_y or speed and direction. Curvature of a trajectory was used by Allmen and Dyer [3] and Tsai et al. [71] for cyclic motion detection.

Mase and Pentland [44] used multiple trajectories, although in a very different manner. The average optical flow was computed in four windows around the mouth of a speaker, in successive pairs of frames. A principal component analysis was performed with the flow components, and two functions, $O(t)$ expressing mouth opening in time, and $E(t)$ reflecting the elongation of the mouth as a function of time, were then created:

$$O(t) = v_b + v_l + v_r,$$
$$E(t) = \tfrac{1}{2}v_a - u_l + u_r,$$

where v_a, v_b, v_l, v_r are the y components of the flow vector of the upper, lower, left and right windows, and u_r, u_l are the x component of the flow in the right and left windows, respectively.

Any type of trajectory can be further processed in order to be used for recognition. For example, the scale-space can be computed, by increasingly smoothing a curve using Gaussian masks, and by detecting zero-crossings or level-crossings at each level. Scale-space of trajectories were used in several studies described in section 4.5.

2.4 Relative Motion

In the trajectory parametrizations of the previous section, absolute values of velocity, speed, direction and curvature were used. However, absolute values might sometimes be inadequate. In the case of human motion, the absolute velocity of a body part has less significance than the relative velocity between moving parts, and relative joint angles with respect to time carry important information. Cutting and Proffitt [15] showed that relative motion is an important aspect in human visual perception. The perception of an object's absolute motion can be divided into two component motions, common motion and relative motion. Common motion is the perceived global movement of an object relative to the observer, each element of the object sharing that motion, while relative motion is the movement of an element with respect to other elements. The absolute motion of a point, i.e. its trajectory, is defined as the sum of common and relative motions. Cutting and Proffitt hypothesized that a minimization process is applied to both the common and the relative motions, and that the one that is resolved first explains what is perceived by the observers. The authors found that relative motion between the elements of an object was usually extracted first, i.e. relative motion is more revealing for the understanding of the motion and recognition of an object than common motion.

In his thesis, Allmen tries to define more formally this minimization process [2]. The absolute motion $A(t)$ as a function of time is defined as $A(t) = C(t) + R(t)$, where $C(t)$ and $R(t)$ represent common and relative motions, respectively, as a function of time. $A(t)$ is recovered from the image

sequence for $t \in [t_i, t_j]$, and those equations are used as constraints to be satisfied. The following function is then minimized:

$$f(C, R) = \sum_{t=t_i}^{t_j} [\kappa_t^2(C(t)) + \kappa_t^2(R(t))],$$

where κ_t is the time derivative of the curvature along a path. Allmen proposes two approaches for solving this system, and he believes that the relative and common motions can be computed such that the results explain the observers' perception.

Because relative motion was shown to be involved in the human visual system and the perception of motion, this kind of information should therefore be very helpful in computer vision systems. Multiple trajectories can be used to compute relative motion. For example, relative angles can be computed between pairs of points relative to a given axis, or as the joint angle between three points, in each frame. The difference of angle is then determined between successive frames, which can be considered as angular velocities, if we assume the time between each frame is constant. This method of computing angular velocity using relative angles between pairs of points can be used, for instance for discriminating between two actors performing the same action. One problem with multiple trajectories is that the computation of relative motion between every element is combinatorially explosive. Also, relative motion between two unrelated elements might not be very revealing. Heuristics can be developed for selecting appropriate relative motions among all possible. Knowledge about the shape of the object will obviously help in that determination. Nevertheless, relative motion plays a very important role in human perception, and this avenue should thus be further investigated.

2.5 Motion Events

Motion events are defined as significant changes or discontinuities in motion. A sudden change of direction, a stop, a pause, can provide important clues to the type of object and its motion. A study of temporal subsampling of American Sign Language sequences showed that choosing images where low activity occurs, i.e. between motion events, was better for understandability than to pick images at constant time intervals [50]. Motion events are usually detected by the presence of discontinuities, which can be found, for instance, by computing the scale-space of a speed curve.

Gould and Shah's Trajectory Primal Sketch (TPS) [25, 26] is a representation for the significant changes in motion. Changes are identified at various scales by computing the scale-space of the velocity curves v_x and v_y extracted from a trajectory. This results in a set of TPS contours, each contour corresponding to a change in motion. The strength of the contour, i.e. the number of zero-crossings belonging to that contour, the shape or straightness of the contour i.e. the sum of the distance between each successively linked zero-crossing from the contour divided by strength-1, and the frame number at which the contour originated, reveals a lot about the event. This representation has been shown to distinguish basic motions like translation, rotation, projectile and cycloid. Their results show that the first derivative discontinuities in the rotation and cycloid trajectories have a sine/cosine relationship with respect to each other, so that they can be separated from the projectile and translation type trajectories. Rotation can be further distinguished from cycloid by the presence of a sine/cosine relationship in the first derivative of the velocity, i.e. in the acceleration, which is absent in the cycloid trajectory. As for translation, it can be classified into straight line translation or curved translation, depending on the respective values of the acceleration. On a straight line translation, both slopes of v_x and v_y must be the same, while in a curved translation, the slopes will

be different. By studying the velocity, acceleration and TPS from different primitive trajectories, it was thus possible to discriminate them. Composite TPS or CTPS [25] is an extension of the TPS in which the trajectory of several points on an object can be reconstructed given some initial information like their starting point, the TPS of one of the trajectories, the frequency, in the case of rotation. The reconstruction has been shown for translation and rotation motions.

Based on psychophysical considerations, Engel and Rubin [19] described the significant changes in motion as motion boundaries, of which they found five types: smooth starts, smooth stops, pauses, impulse starts and impulse stops. They are considered as motion events that partition a global motion into its psychological parts. Their method of detecting those boundaries use polar velocity representation (s, ϕ), and the features used for the detection of the perceptual boundaries are first and second derivatives of the speed s' and s'', and the second derivative of angle ϕ''. Force impulses imply a discontinuity, i.e. zero-crossing in consecutive pairs of s'' and ϕ'' estimates. If a zero-crossing exists in either or both, and its slope exceeds some predefined threshold, then force impulses are confirmed. Starts and stops are found when speed is low, but increasing or decreasing sufficiently, again determined using thresholds. The simultaneous presence of a start or stop and a force impulse indicates an impulse start or stop, while a pause occurs when a start (smooth or impulse) follows a stop (smooth or impulse). The authors studied cycloidal motion at two different speeds, fast and slow. At each cusp, a boundary was detected; the slow motion detected a ϕ'' zero-crossing, along with a pause, while the fast motion asserted force impulses for both s'' and ϕ'', but no start, stop or pause. Both boundaries were found to agree with human's perception.

Goddard [24], in his work on human motion, used changes in rotational velocity of body segments along with changes in direction, as motion events. For example, six ranges of angular velocity values were used ($\leq -200; [-200, -100]; [-100, 0]; [0, 100]; [100, 200]; \geq 200$ $degrees/second$), along with four quantizations of the orientation, taken as the four 90^o quadrants. Any change in the orientation or angular velocity constitutes a motion event, which will trigger some action in his connectionist system.

Motion events have been shown for simpler motions as for projectile and cycloid motions, as well as for complex motions like human movements. Motion events are thus of wide applicability, and can be used alone or in conjunction with other types of features.

2.6 Region-Based Features

For certain types of objects or motions, the extraction of precise motion information for each single point is not desirable nor necessary. Instead, the ability to have a more general idea about the content of a frame might be sufficient. Features generated from the use of information over a relatively large region or over whole images are referred to here as region-based features. This approach has been used in several studies. For instance, Polana and Nelson [55, 47] gathered a set of four features based on the computation of the normal flow, i.e. the component of the flow field parallel to the gradient, over regions of interest. The first feature is the *mean flow magnitude divided by its standard deviation.* The normal flow is computed at each pixel of a region, and its mean magnitude calculated, then divided by the standard deviation, ensuring for scaling invariance. The *positive and negative curl and divergence estimates* were also used. Divergence is the dot product of the gradient operator and the flow vector, while the curl is their cross product. They are computed for every pixel of a region, and the positive and negative values separated. The features used are the average values of the positive and negative curl and divergence over the region of interest. The *non-uniformity of flow direction* feature requires the computation of the histogram of the eight discretized directions over the relevant region of the image. The sum of the absolute deviation

from a uniform distribution will give an approximation of the non-uniformity of direction. This approximation is then normalized by using the four-way histogram of gradient directions, in order to reduce the dependency of the flow direction with respect to the intensity texture. Finally, the *directional difference statistics in four directions* were computed. Using the eight discretized directions of motion, along with the first feature above, the authors compute second order statistics of the normal flow direction. Co-occurrence matrices are built for four directions, using pixel pairs at a distance proportional to the average flow magnitude, again ensuring for scaling invariance. In each direction, the ratio of number of pixel pairs differing in direction by at most one over the number of pairs differing by more than one is computed. The logarithm of the ratio is then used as feature. Four new features then emerge from the directional statistics, one from each direction. Once all those values are computed, they are then put into vector form for classification.

In another paper, the same authors describe a representation for periodic motions like walking, running, jumping, and exercising [54]. It is, as above, based on the computation of the normal flow direction and magnitude, from which statistics are gathered over a selected portion of the sequence of images, called the spatiotemporal (ST) cube. The ST-cube consists in x, y and $time$ dimensions, as was described in section 2.2. The authors devised two different motion models. The first one is to compute a set of four statistics over the whole ST-cube, namely the vertical and horizontal stick-like motion, and the vertical and horizontal worm-like motion. Those statistics use the direction of motion at every pixel of the ST-cube. The stick-like motion is characterized by a set of consecutive pixels, arranged vertically or horizontally, showing the same motion direction, horizontal or vertical, respectively. The worm-like motion is characterized by a set of horizontally or vertically arranged pixels showing motion in the horizontal or vertical direction, respectively. The second representation is a three-dimensional feature vector, created by partitioning the x, y and $time$ dimensions of one cycle of the ST-cube (one sequence contains four cycles). In each of the resulting three dimensional cells, a statistics is then computed. Experiments with three different statistics were performed: the summed normal flow magnitude in each cell, the dominant motion direction in each cell, and the summed motion magnitude in the dominant motion direction of each cell. This dominant direction is approximated by computing the histogram of the discretized normal flow direction of motion weighted by its corresponding magnitude, and taking the direction with the highest value.

Eigen images extracted from a set of graylevel images of an object provide enough information to directly represent new images of that object. Kirby et al. [35] used that method with mouth images. The eigen images are the eigenvectors of the ensemble averaged covariance matrix C:

$$C = \frac{1}{P} \sum_{j=1}^{P} u^{(j)} u^{(j)t},$$

where $u^{(j)}$ is vector formed from the concatenation of the columns of the j'th image and u^t is the transpose of u. C is non-negative, and its eigenvectors form an orthonormal basis. They are computed within the Karhunen-Loève framework, since the system would be too large to be otherwise solved efficiently. Once the eigen images are extracted, an image can then be expressed as a linear combination of those eigenvectors, i.e. by a vector of coefficients, each coefficient associated to an eigenvector. An image sequence of a spoken word could then be represented by a vector for each frame of the sequence, thus forming a matrix of coefficients.

A set of model views of an object is the approach taken by Darrell and Pentland with hand gestures [16, 17]. Their method automatically stores the appropriate number of views necessary to represent an object using correlation. If optical flow can be reliably extracted from a sequence of

images, correlation using optical flow might be more appropriate as compared to correlation using plain graylevels. With this idea in mind, Martin and Shah [43] use a sequence of dense optical flow fields around the mouth of a speaker, which are then correlated, for matching with a sequence of optical flow frames. Dense flow fields create redundancy in data that allows for better performance and robustness.

Using binarized images, Petajan et al. [52] created a codebook of mouth images, i.e. a set of images of the different shapes of the mouth while speaking. The motion of a spoken word can be described as a sequence of elements of this codebook. This sequence can be defined using the images directly, or by using an index identifying each image: this process is called vector quantization.

Yamato et al. [75] used a mesh feature extracted from each binarized frame of a sequence. An image is divided into a grid, and the proportion of black to white pixels in each grid element is computed; the ordered set of ratios for an image is called the mesh feature. Motion is detected by variations in the ratios from one image to the other.

There are advantages and disadvantages in each representation. For instance, in Darrell and Pentland's work, the number of views necessary to represent an object might get too large for storage and/or recognition purposes. In this perspective, Kirby's approach seems more elegant and efficient, since once the eigenvectors are found, any image can be expressed as a small set of coefficients. The interpretation of Petajan et al.'s codebook images is straightforward, but their storage (they used at most 255) is costly in space. The mesh feature used by Yamato et al. can represent an image in a very simple form, but this abstract format prevents easy interpretation of the feature.

2.7 Matching and Classification

Once the representation has been defined and the features from both models and unknown images properly encoded, a comparison must be made so that classification or recognition can take place. Of course, this comparative step depends on the type of features that are used to encode both models and inputs. The methods described here usually involve some kind of distance calculation between a model and an unknown input, the model with smallest distance is taken to be the class of motion to which the input belongs.

Scale-space of trajectories have been used in different ways. Rangarajan, Allen and Shah [58] computed the diffused scale-space of speed and direction of different points extracted from their trajectory. They argue that for similar motions, the scale-space will be similar, such that the point by point difference between the scale space of the speed and direction curves from different points undergoing similar motion will be much smaller than with points undergoing very different motions. Allmen and Dyer [2, 3] used the scale-space of curvature values computed from spatiotemporal curves, to detect cycles in the motion of the moving object. If cycles occur, the scale-space will also contain repeating patterns, which can be detected.

If the features can take the form of a vector, then clustering of features is a method which has been often employed, in particular by Polana and Nelson [47, 54, 55]. The approach is based on the assumption that similar motions will generate similar feature vectors, which can be classified using a nearest centroid classifier, or else any kind of classifier based on vectors. A principal component analysis can also be useful in determining which features are more important in the classification. Petajan et al. [52], in their lipreading scheme, matched quantized vectors, which are the representation of word samples. The matching process involved computing the distance between an input and a model vector, and the model with the smallest distance was chosen as the representation of the input. Finn and Montgomery [21] also used feature vectors, and recognition

consisted of choosing the model that generated the smallest root mean square distance. Mase and Pentland [44] used a sampling of their two functions (mouth opening and elongation functions, see section 2.3) to create a vector used for input and model comparisons, and a match was established with the model that gave the smallest weighted squared difference.

Another approach taken was the connectionist or neural network, found in Goddard's work [24]. His representation consists of an ordered sequence of events which are coordinated by temporal and motion events. A hierarchy is used: at low-level, the presence of a low-level feature triggers an event which is sent to the layer above in the hierarchy. Combination of events at that level trigger other events at yet higher levels, and so on, until the coordinated sequence of events of a body in motion can trigger one motion model at the highest level. This is all done in a connectionist framework, where the detection of a feature or an event activates one or more units, which might trigger units at higher levels, up to the output level, representing the global motion of walking, running or skipping.

Yamato et al. [75] took a probabilistic approach to the classification of different motions. They used a sequence of symbols, one per frame, derived from a mesh feature at the image level, sequence determining an intermediate-level representation. Sequences like these are used to train Hidden Markov Models (HMMs), which in this case are symbol generating machines. Matching of an unknown sequence with a model is done through the calculation of the probability that an HMM could generate the particular unknown sequence. The HMM giving the highest probability is the one that most likely generated that sequence.

Clustering techniques are very common and well documented, and since several methods use a vector type of representation, the matching can be done in a fairly efficient manner. Other matching methods will obviously depend on the type of representation used, and will thus vary accordingly.

2.8 Summary

A wide variety of representations are built from basic information extracted from an image sequence. The most common kind of information is related to motion trajectories, as described in section 2.2, which are usually simple to compute and interpret. Region-based features, like the mesh feature, are more abstract and their interpretation is not as obvious. Optical flow was frequently computed, often as part of a region-based feature. Features using graylevel image information are not as common, might be more costly in space or computation, and must be able to cope with the inherent noise involved. However the use of whole images and their intrinsic redundancy might capture some subtle information that a small set of discrete features could not provide.

Some features are more appropriate for the motion of particular kinds of objects. For the motion of rigid objects in a scene, trajectory representations are very useful: speed, direction, velocity, and their time derivatives, curvatures. Region-based features were also reported, for instance codebooks containing images of an object's different shapes, and eigenvectors forming a basis for describing an object's shape. Features based on optical flow have also been used, like normal flow statistics. With articulated objects like the human body, joint angles and angular velocity computations are commonly used.

3 Motion Recognition

Relatively few papers encountered in the literature describe the whole process of recognition, from the image sequence up to the recognized object or motion. Most of them only describe various representations, without taking their system a step further by discussing ways to create a database

of models and to index it for recognition or classification. Sometimes this step is relatively simple. However, if the system is not designed to do so, it might be very difficult and/or costly to "add it". It is thus important to have this goal in mind when designing a recognition system.

The previous section was concerned about the two stages of recognition, namely motion information extraction and representation, and the recognition/classification. This section will describe indvidual recognition methods. Section 3.1 will examine methods whose purpose is to detect and recognize the presence of cyclical motion. Sections 3.2 and 3.3 will discuss approaches that have been designed for lipreading and hand gestures recognition, respectively. Motion verb recognition and temporal textures classification will be discussed in sections 3.4 and 3.5. A summary will then conclude this section.

3.1 Cyclic Motion Detection

The presence of cyclic motion in a sequence of images can reveal a lot about the object showing that type of motion. A rigid object can perform a cyclic movement, for example a ball in a pendulum motion, while an articulated object can perform much more complex motions. Furthermore, different cyclic motions could occur concurrently with the same or with different frequencies and phase relative to each other. The following three studies describe how cyclic motion is detected, and in one case how this information can be used to recognize human motion.

Based on studies of the human visual system, Allmen, along with Dyer [3, 2] argue that cyclic motion detection (1) does not depend on prior recognition of the moving object, i.e. cycles can be detected even if the object is unknown; (2) does not depend on the absolute position of the object; (3) needs long sequences (at least two complete cycles); (4) is sensitive to different scales, i.e. cycles at different levels of a moving object can be detected. To study cyclic motion detection, the authors use curvature as a low level description of motion. Given the three dimensional location of a point over a long sequence, its path can be parametrized by $\alpha(t) = (x, y, z)$, for all frames of the sequence. $\kappa(t)$ is defined as the curvature at $\alpha(t)$. A cyclic motion of period $\Delta t = t_2 - t_1$ is detected if

$$|\kappa(t) - \kappa(t + \Delta t)| < \varepsilon, \ \forall t \in [t_1, t_2].$$

Only two cycle intervals are needed for detection. The two dimensional projection of three dimensional paths are used, and it is assumed that if cycles occur in the projection, they will also occur in three dimensional space. The definition of cyclic motion is further extended to rigid objects and articulated objects (see [2]).

The authors used ST-curves recovered from the ST-cube representing the sequence of images (section 2.2). Allmen [2] shows that if a solid object in a scene undergoes cyclic motion such that the periodic motion is preserved under projection, the curvature of the ST-curves extracted and corresponding to that object will be cyclic. The ST-curves are equivalent to $\alpha(t)$ defined above, and the cyclic motion definition can be applied. To detect cycles, curvature scale-space was proposed as a representation. For the detection of cycles, a modified version of a uniform cost algorithm was used, in which a node is associated for every possible pair of features in the scale space, where the features are defined to be local maxima, along with their left and right contours. A match cost is determined with each node, measuring how different the two features are. The pattern with lowest cost and which repeats over the sequence is the result of the match.

The advantage of curvature scale-space for cycle detection is that cycles are position invariant and the curvature scale-space is position invariant as well. Another advantage is that it should be possible to detect cycles at different scales: coarser cycles would appear at coarser scale, while finer cycles could be detected at finer scale. However, due to the way the curvature is computed, points

of sudden changes in the trajectory will create high and narrow impulses in the curvature function, containing high frequency components that may interfere with lower frequency components. It is not obvious how the two can be separated without getting rid of those impulses in some way [71]. It is not clear how Allmen and Dyer's method is effective in doing so.

Polana and Nelson [56] also used a ST-cube to detect cycles. In their approach, the first part consists of extracting a reference curve, as described in section 2.2. The reference curve is in fact the approximate trajectory of an object, and simply provides the approximate location of the centroid of an object in time. The frames are then aligned with respect to the centroid of the object, such that it remains stationary in time. However, if the object presented some periodic motion, for example a person walking, the motion of the legs and arms remains, which will create some periodic graylevel signals over the image, especially around the centroid. The periodic motion will be extracted from the graylevel signals using a Fourier transform. The periodicity measure p_f of the signal f is defined as the normalized difference of the sum of the power spectrum values at the highest amplitude frequency and its multiples, and the sum of power spectrum values at the frequencies halfway in between:

$$p_f = \frac{\sum_i F_{iw} - \sum_i F_{(iw+\frac{w}{2})}}{\sum_i F_{iw} + \sum_i F_{(iw+\frac{w}{2})}},$$

where F is the energy spectrum of the signal f, and w the frequency corresponding to the highest amplitude of the spectrum. Since the signal along any one curve may be ambiguous, the periodicity measure must be computed for a number of reference curves of the same moving object, all parallel to the original curve, and be combined together. To do so, a form of non-maximum suppression was devised. Given n reference curves, the periodicity of the signal along each curve was calculated. Each frequency w is then assigned a value P_w equal to the sum of the periodicity measured from all signals whose highest amplitude is w. The maximum value of this combined signal is taken as the fundamental frequency, and its periodicity P is defined as the average of the periodicity measures of the contributing signals. The periodicity detection is invariant to the magnitude of motion, speed of activity, and is fairly robust in small changes in viewing angles. In this paper, linear motion of the object (constant velocity, linear path) was assumed, but nonlinearly moving objects can be handled by tracking objects given a coarse estimate of its initial location and velocity, to give a reference curve that is not a straight line.

The authors go even further by describing a method for recognizing different periodic motions, in an upcoming paper [54]. The overall periodicity measure described above is first used to refine the segmentation of the moving object. A normalization procedure then produces a spatio-temporal solid (ST-solid) that is invariant to spatial scale and translation, and invariant to temporal scale, i.e. frequency. The motion models, represented by feature vectors, are then created using information from this ST-solid. They are based on the computation of statistics based on the normal flow direction and magnitude, and were described in section 2.6. The classification is performed using a nearest centroid algorithm.

Tsai et al. [71] used the spatiotemporal curvature for cycle detection. As explained in section 2.3, the trajectory of a point on an object that performs some cyclic motion is used to compute curvature as a function of time. A median filtering is then applied to suppress high and narrow impulses that can interfere with the detection of cycles; the DC component, i.e. the component that doesn't change with time, is also removed to avoid zero frequency impulses. An autocorrelation is performed to emphasize self-similarity within the curvature function. A Fourier transform is finally applied to that signal to detect the presence of cycles and their period: a high impulse indicates the presence of cycles and their fundamental frequency.

Both Allmen and Dyer [3] and Tsai et al. [71] used a curvature function as input, although Tsai performed some preprocessing steps on the curvature function. Polana and Nelson [56] used graylevel signals of the aligned frames as input. Allmen and Dyer use a uniform cost algorithm on the scale-space of the curvature to detect cycles, while Tsai et al. and Polana and Nelson use Fourier transforms to directly detect the frequency of cyclic motion. It seems reasonable and sensible to use Fourier transforms to detect cyclic motion, and the Fast Fourier Transform algorithm makes the process more efficient. Furthermore, it is more robust to uncorrelated noise. Motion-based recognition from cyclic motion has been reported in Tsai et al. (see section 4.5.2) and Polana and Nelson [54]. Their method provides good results on a variety of periodic motions. Furthermore, it possesses several desirable invariances, for instance to spatial and temporal translation and scale, and can handle small variations in viewing angle, varying image illumination and contrast, and even some amount of background motion.

3.2 Lipreading

Humans have the very complex ability of interpreting facial expressions, gestures, even the so called "body language". Hearing impaired people further develop this ability since most of them can perform some lipreading and/or understand sign language. Lipreading is a very difficult task, especially since certain phonemes can appear visually identical (phonemes are minimal meaningful units of sound from which two words can be distinguished). For instance, the phonemes "b", "p", and "m" sound different but look the same when spoken [21]. Acoustically-based automatic speech recognition (ASR) is still not completely speaker independent, limited in vocabulary and sensitive to noise [21]. Combination of acoustic and visual speech recognition is one possibility to better achieve lipreading capability. This section will describe several lipreading methods developed so far; gesture interpretation will be discussed later.

In Petajan et al. [52], the lipreading task is performed by using the mouth opening area of a speaker to create a codebook, as described in section 2.6. Tracking the nostrils in order to recover the precise location of the mouth window, the mouth images were binarized, then thresholded, such that only the mouth opening created a dark area. This large set of mouth images was reduced by clustering to about 255 clusters. A representative of each cluster was stored in a codebook of mouth images. The images in this codebook were ordered by increasing size of dark area, and identified with an index value. Once the mouth images codebook has been created, an inter-cluster distance table is computed, for faster computation during the matching process. The models of spoken words (the spoken letters and digits from zero to nine) are stored and vector quantized. Vector quantization replaces each mouth opening image of a sequence by the index of the closest image in the codebook, thus creating a vector of indices representing the sequence. Recognition is done by computing the distance between vector quantized word samples and every vector quantized word model. The model with smallest distance represents the sample. Recognition without vector quantization was also performed, by directly using the image distance measurements. Results were actually better than with vector quantization. Combination of visual and acoustic recognition was also tested, which increased the recognition rate as compared to the acoustic method. Results were however divided as compared to visual recognition alone. The authors however believe the combination of methods would probably produce optimal recognition performance.

Finn and Montgomery's algorithm [21] uses distances between different points around the mouth, in combination, to distinguish between different spoken sounds. Twelve dots were placed around the mouth of a speaker and tracked during the experiments; a total of fourteen distances were measured, and used as a feature vector. The data were normalized relative to time and overall

amplitude of distance measurements. The recognition consisted of computing a total root mean square value between two utterances: the model with smallest difference was considered the correct model. Some optimization was also performed by using a weighted set of the distances contributing the most to the recognition.

A different scheme was developed by Mase and Pentland [44]. They observed that the most important features that affect mouth shape relate to the elongation of the mouth, and to the mouth opening, affecting upper and lower lips. Using optical flow as described in section 2.3, the authors came up with the specification of two principal types of motions of the mouth expressed as functions with respect to time: mouth opening $O(t)$ and elongation of the mouth $E(t)$. $O(t)$ and $E(t)$ are computed in each frame, then smoothed and normalized to a fixed variance. Word boundaries were taken to be times when $O(t) = 0$, i.e. when the mouth is closed, and can easily be located on the $O(t)$ plots. Templates were used for recognition, and matching was performed, after a resampling step that normalizes the time to speak one word (time warping). A sampling of the templates was used for matching with a similar sampling of the data. A match was established between a template and the data if the weighted squared difference between the sampled templates and data was smallest among all templates. Because the data consisted of several contiguous words, the templates were compared to data between each set of potential word boundaries.

In Martin and Shah [43], the authors use a sequence of dense optical flow fields around the mouth of a speaker, which are then spatially warped, temporally warped, then correlated, for matching with a sequence of optical flow frames. Spatial warping is used to locate the window containing the lips of each model frame with each input frame. Each model optical flow frame is compared to each input optical flow frame using a correlation procedure. The best match is found and the location and match score are stored in a table. Temporal warping addresses the problem of one speaker speaking faster or slower, from one time to the next or from another speaker, and it uses the table created from the spatial warping step. The purpose is to find the best representatives of a longer sequence so that model and input can be directly compared. This is done through Sakoe and Chiba's algorithm, which take the above table to be the adjacency matrix of a graph, and find the best path through this graph [64]. This list of points of the best match are stored for the next step. Correlation matching takes the average of the correlation values of the frames of the input and in the best path, for each model, of the above procedure. The model with the highest average is taken to be the one representing the input.

Kirby et al. [35] chose to express mouth images as a function of a fixed set of feature images. This fixed set is the set of the eigenvectors of the ensemble averaged covariance matrix C (see section 2.6). A spoken word made up of a sequence of P images can then be expressed as a $Q \times P$ matrix of coefficients computed with respect to the set of Q "eigenlips". The recovery of images in this way was performed, with very good results, except for some smoothing due to the elimination of smaller scaled eigenfunctions. Although the main goal of this work is to provide a low-dimensional vocabulary for the analysis and synthesis of lip motion and as a means of compression for transmission, some experiments were done on word classification. Identification of particular words in a sequence using spatial eigenfunctions was performed using a template matching technique in order to find the minimal necessary number of coefficients for correct match, which was determined to be three. With temporal eigenfunctions, correlations were computed word-wise, i.e. data used in constructing the eigenfunctions came from a single word. Each eigenfunction was expressed as a one-dimensional function with respect to time. The values from the second and third eigenfunctions were plotted against each other to create a "signature" graph, which the authors estimate is distinctive enough for discrimination. However no precise method was described for recognition using the signature graphs.

Finn and Montgomery's method [21] is the simplest, but the markers around the mouth are not very practical: lip reading should be performed without any "clues" around the mouth, and ultimately in real-time. Martin and Shah's work [43] use dense optical flow as main feature, and matching using correlation between every frame of the model and input sequences, which is computationally expensive since the input needs to be compared to every model. Mase and Pentland's mouth opening and elongation functions are simple enough and address continuous speech, but the difficult part, according to the authors, is to find the beginning of the first word. Petajan et al.'s method [52] is interesting since when the codebook is complete, a distance table is computed, which makes image comparison somewhat faster. However, the vector quantization did not work as well as direct image comparison, so the number of codebook images might need to be increased. Kirby et al.'s method [35] is also very nice, and has the advantage of reducing tremendously the quantity of data for image transmission, by only transmitting a coefficient matrix for a sequence, as opposed to transmitting whole images. More extensive work on lipreading *per se* is needed, but the authors provide a nice basis for such work. The biggest problem for any of those methods is that of speaker dependence. Integration of both acoustic and visual inputs was attempted by Petajan et al. [52] with better results. Finn and Montgomery [21] didn't actually experiment on combination of visual and acoustic input, but they discuss the possible contribution of visual information to acoustic speech recognition.

3.3 Gesture Interpretation

Humans have the capability or can develop the ability to interpret gestures, and gestural languages have been developed to allow hearing impaired people to communicate more easily. Systems built for gesture interpretation might serve several goals. First, they could be used to develop a human-machine interface for a more natural way to interact with humans; so far, the human had to learn how to communicate with the computer. The inverse so far hasn't been very successful!... Second, efficient methods of data compression for image transmission need to be developed. Although the era of visual phones is right around the corner (see [36]), actual bandwidths on the telephone network cannot support real-time grayscale visual transmission: static images or very low image transmission rates only have been achieved. For hearing impaired people, the possibility of using visual phone to sign would be most useful, since it is the most usual and fastest way for them to communicate. It was shown that the requirements in precision and resolution for American Sign Language (ASL) are relatively low as compared to grayscale images. Sperling et al. [67] compared several effective methods for creating compressed ASL images, for example using binarized images, space and time subsampling of grayscale images, and outline images. A previous study shows that moving light displays can successfully be used in ASL [53]. A temporal subsampling of ASL based on event boundaries was reported in [50].

Methods devised for gesture recognition however are not as common as for compression and transmission. One method uses color markers on a glove, which are tracked during the motion of the hand, and from which three dimensional information is extracted, using a structure from motion approach. This method is performed in real-time and is used to modify an image of a model [14]. Another system, called Finger-Pointer, recognizes pointing actions and simple hand forms, also in real-time, but needs stereoscopic image sequences to do so. It is used in the remote control of computer-aided presentations [22]. The disadvantages are that color imaging is required in the first case, and stereoscopic images necessary in the second. Two studies are described below, one that automatically learns, tracks and recognizes human gestures and that performs in real-time, and the other providing a simple data structure for representing and recognizing hand gestures,

which also performs in real-time.

Darrell and Pentland present a new method for learning, tracking and recognizing human gestures from a sequence of images [16, 17]. The method uses an automatic view-based approach to build the set of view models from which gesture models will be created. The model views of an object are built using normalized correlation. The first view is chosen by the user as one of the images from a sequence. The object in the subsequent input images is tracked, and when the correlation score r_m drops below a predetermined threshold, a new model view is created with the current input image. This process is repeated until no more models are necessary. Once all views of an object have been gathered, gesture models need to be created. A gesture is a set of views over time. A gesture will be correlated with each stored view of the object (the hand), and the score plotted, for each view, with respect to time. Several examples of the same gesture are used, and the mean $\tilde{g}_m(t)$ and variance $\sigma^2(g_m(t))$ of the correlation scores with respect to model view m will be used to represent that particular gesture $\mathbf{g}$. The gesture models need first to be adjusted to the same length, and this is done through dynamic time warping (DTW). To compare a new input gesture, each frame of the new sequence is correlated with a model view and its score determined. The score results for the whole sequence is plotted with respect to time. The same process is repeated for all model views, and the score results for each model stored in a vector $\mathbf{r}(t)$. The input gesture will be compared to all gesture models. However since this new sequence might contain several gestures, the cost of the optimal path of the DTW is defined differently, as a function of $\tilde{g}_m(i)$, $\sigma^2(g_m(i))$, i.e. the mean score and corresponding variance of view model m at time i, and $r_m(j)$, the correlation score for model m at time j. The score of the model gesture is the minimum of any of the partial sums that account for all time samples in the gesture:

$$\delta(\mathbf{g}, \mathbf{r}) = \min_{0 \leq t \leq T'} C_{T,t}$$

where T and T' are the number of time steps in $\mathbf{g}$ and $\mathbf{r}$ respectively, and $C_{T,t}$ is the minimal cost to align $\mathbf{g}[0, T]$ and $\mathbf{r}[0, t]$ for $0 \leq t \leq T'$. The score is defined as $1/\delta(\mathbf{g}, \mathbf{r})$. Such a score is computed at each time frame for the new sequence, and for each model. The scores for each model can then be plotted, and the peaks indicate a match for the model in the new sequence.

Davis and Shah [18] report a simple method for hand gesture recognition by tracking the trajectory followed by each finger and using their motion as a basis for recognition. The direction of motion and displacement of each finger was computed as described in section 2.3. The models are constructed by averaging the direction and displacement information from several samples of the same gesture; they are then stored along with the name of the gesture. A motion code is then derived from this information: it consists of a five-bit number, each bit associated with a finger. A bit will be set if motion of its associated finger is detected, and reset otherwise. This motion code is used for matching in the following way: the model gestures are stored as an array of linked lists of gestures. The motion code serves as an index for all gestures sharing a particular motion code. The direction of motion and displacement, along with the motion code of an unknown gesture is then computed, and the gesture will be compared only to those models with the same motion code, with the use of thresholds. A finite state machine (FSA) is used to model a generic gesture. A gesture is characterized by four phases: (1) static start position, for at least three frames; (2) smooth motion of the hand and fingers until the end of the gesture; (3) static end position, for at least three frames; (4) smooth motion of the hand back to the start position. The user must be restrained however in following these four phases in order for a gesture to be properly modeled or recognized.

Automatic interpretation of hand gestures is in a way more difficult than lipreading because

the motion of the hand can become very complex. Darrell and Pentland's system [16, 17] performs in real-time on a special purpose image processing machine. Their method was tested with four gestures, one of them almost always successfully recognized even when performed by different users. On the other hand, Davis and Shah's algorithm [18] represents very succinctly any gesture, and yet produces very good results on a set of seven real image gesture sequences. Their method also performs in real-time. Their use of a finite state diagram prevents the need for dynamic time warping to align gesture sequences of different lengths, which was demonstrated in the experiments. It is however very important to be able to find the correct bounds of each gestures, otherwise the finite state machine might work asynchronously with the sequence and jeopardize the recognition for the rest of the sequence. So far, the two methods devised were tested on very small set of simple gestures and thus have very limited scope. The problem remains on how to design a system that can work with a large "vocabulary", not so simple gestures, and remain user independent.

3.4 Motion Verb Recognition

Motion verb recognition deals with the association of natural language verbs with the motion performed by a moving object in a sequence of images. Artificial intelligence and machine vision are thus combined to provide us with a natural language description of scene motion. Badler [5] and Neumann and Novak [48] did some initial important work in motion verb recognition.

Koller, Heinze and Nagel [37] describe a method which automatically characterizes the trajectories of moving vehicles using natural language motion verbs. Association of a motion verb to trajectories of those candidates is done using an artificial intelligence type approach. A set of 119 verbs with 67 different definitions which apply to vehicle motion were extracted from a German dictionary. Those verbs were divided into four categories: verbs describing the action of an agent (vehicle) only; verbs which make additional reference to the road; verbs which make additional reference to other objects; verbs which make additional reference to other locations. Another subdivision discriminates between inchoative verbs, durative verbs and resultative verbs, i.e. verbs describing the beginning, middle and end of a motion event. Attributes are computed from the sequence, which help describe more precisely the trajectory segments, the position of a vehicle with respect to the street or other objects, its orientation, velocity, etc. A total of 13 such attributes were defined. In addition, for all verbs there exists a set of 3 predicates, whose truth value is determined at each time instant: (1) a precondition, which is true if some predefined attribute values are within the expected range at the beginning of an interval of validity for a verb; (2) a monotonicity condition, which indicates if the direction and amount of change of associated attributes remain acceptable; (3) a postcondition, indicating if attribute values are within the range expected at the end of an interval of validity for a verb. An interval of validity is the time period for which a particular interpretation is true, and depends on the truth values of the predicates. A finite state automaton is used to determine intervals of validity. The transitions from one state to another depend on the predicate values, i.e. monotonicity, pre and postconditions. The algorithm can successfully determine the motion verb most appropriate for an object in a frame interval.

The goal of Tsotsos' work [72, 73] was to build an artificial intelligence system, called ALVEN, capable of using motion information to recognize normal and abnormal behavior of a heart's left ventricular motion. When the heart is impaired or damaged, abnormalities occur regionally, and different segments of the left ventricle (LV) often behave differently. Furthermore, some evidence indicates that if a segment is damaged, other segments may overperform, thus creating a new behavior on other segments too. The authors used a semantic net representation with frames, type hierarchies, inheritance and exception handling, as the basis of organization of their knowledge-

base. Based on previously released studies, natural language semantic components were developed to describe motion concepts using English motion verbs.

In patients that were receiving corrective heart surgery, nine markers were implanted on the LV wall, roughly in the same plane, and two clips were attached to the aorta, which served as reference points. Cineradiography was performed in a follow-up examination. The assessment of the left ventricular motion is based on the velocity profile of its wall segments, using location changes of points, length changes of axes and perimeter, area and shape variations. The extent, velocity and acceleration of contractions must be measured, and processing is done frame by frame. The wall is divided in three segments: the anterior, posterior and apical segments. The image sequence was divided into two phases, the expansion phase or diastole, and the contraction phase or systole; each phase was further subdivided into subphases. For the normal case, those different phases were characterized by assigning them different time "constraints": minimum start time, maximum end time, minimum/maximum duration, minimum/maximum rates of contraction. Each constraint has exceptions associated with it: too short/long (duration of event), too fast/slow (rate of area change) and too late/early (time slot in which the motion was recognized with respect to the LV cycle). Abnormalities such as asynchrony, hypokinesis and diskinesis were defined, among others. ALVEN provides a summary describing the dynamics of the LV as a whole, and if necessary, at the segment or even at the marker level, with their associated quantities, for verification or for precisely locating reasons for an abnormal summary description.

Both methods describe systems that are very interesting because of the interaction of computer vision an artificial intelligence, for the description of the motion of vehicles at an intersection or the behavior of the left ventricle of the heart.

3.5 Temporal Textures Classification

In their paper, Nelson and Polana describe how the movement of the ripples on water, the wind in the leaves of trees, a cloth waving in the wind, can be classified [47, 55]. Those motions, referred to as temporal textures, show complex and non-rigid motions. The term temporal texture is used to emphasize that the motion patterns are of indeterminate spatial and temporal extent. Different features based on optical flow fields, when combined together, provide enough information for classification. The four features chosen were the mean flow magnitude divided by its standard deviation, the positive and negative curl and divergence estimates, the non-uniformity of flow direction, and the directional difference statistics in four directions. They are based on optical flow fields, and were described in section 2.6. The feature values are arranged into a vector, and the authors used, for classification, a nearest centroid classifier. Three sequences were used to compute the cluster centers, i.e. train the classifier, while using a fourth one as the unknown. None of the single features is sufficient, alone, for correct classification. However using curl and divergence estimates along with the directional difference statistics in four directions provide enough information to correctly classify all motions used in the experiments. A principal component analysis of the features was also performed, which also confirmed the relative importance of those two features.

Because Polana and Nelson's algorithm seems to use all the data from all the frames of a sequence, this method looks computationally expensive, but it might be necessary to provide the stability and invariance needed for a good classification.

3.6 Summary

Recognition schemes for objects and motions were described in this section. Cyclic motion detection and recognition were discussed, along with lipreading, gesture interpretation, motion verb recognition and classification of temporal textures. Among those studies, several are very versatile. Tsotso's framework could be modified for use with different types of images for different diagnostic tools, for instance arrythmia interpretation from electrocardiograms, or foetal aortic valve function evaluation using ultra-sounds [72, 73]. Petajan et al.'s codebook of mouth openings [52] and Kirby et al.'s eigenlips [35] could be easily generalized for other types of objects, perhaps even articulated. Although a framework general enough for all sorts of applications has yet to be found, some of the methods described here have a great potential for a large range of use.

4 Human Motion Tracking and Recognition

This section will be concentrating on methods designed to recognize human motion. There are several ways to view this task. The first one is to recognize the action performed by a person in a scene, among a database of human action models, in a way similar to what was described in the previous section. The second way is to be able to recognize the different body parts like arms, legs, etc. throughout a sequence, using motion. This approach is referred to here as labeling. The third way is to define motion as a sequence of object configurations or shapes through time. The knowledge of the shape and motion of an object, in this case the human body, is used to *guide* the interpretation of an image sequence in order to analyze the motion between frames, to determine the most plausible configuration of the body or to recognize and label different parts of the body. This approach has been used mostly with humans, and is called here tracking of human motion.

The modeling of the human body and of human motion is an important step in body labeling and tracking. A good model should allow the system to easily recognize a human body and any of its postures, i.e. all allowable positions of its parts with respect to each other, yet be simple enough to minimize the number of parameters necessary to represent the body adequately. This kind of balance is seen between the volumetric models and the stick figure models, and they will be described in section 4.1. Section 4.2 will describe how the walking motion has been modeled, i.e. what kind of qualitative and quantitative information is necessary for good tracking. Section 4.3 will describe two pieces of work for body labeling in a sequence, while three dimensional tracking methods will be examined in section 4.4. Section 4.5 will be concerned with the recognition of the different motions that can be performed by a person, and how it is possible to discriminate between different persons performing the same action.

4.1 Modeling of the Human Body

To properly study human motion, good body models must be defined. The models differ somewhat if they are used for body parts labeling or for tracking. Stick figure models and volumetric models are used for three dimensional tracking. In the case of labeling, only the projection of three dimensional models have been used in the methods that will be described below.

The stick figure model consists of segments usually connected at their endpoints and representing the body. This model can thus be seen as a skeleton (taken in a computer vision sense) of a human body and can be as detailed as necessary. Akita's model [1] consists of six segments: two arms, two legs, torso and head. However no joints are explicitly defined. Lee and Chen's model [39, 12] contains fourteen joints and seventeen segments. The joints are left and right shoulders, elbows,

wrists, hips, knees and ankles, plus a pelvis and neck joint. The segments are the left and right lower legs, thighs, forearms, upperarms, plus the segments joining the two shoulder and the two hip joints, the neck with each shoulder joint and with the pelvis joint, and finally the pelvis joint with each hip joint. The torso is formed by the neck, shoulders and pelvis joints, and the hip part is formed by the pelvis and hip joints. Both torso and hip parts are assumed rigid. Six additional points characterize the head, but they are only used in a camera calibration step. The length of each segment is also given. Tracking with such a model consists of finding the location in space of each joint and the three dimensional angle between each pair of contiguous segments. Parameters used are the segments' length and location in space of their extremities.

Stick figure models can be described using only a few parameters. In the case where the trajectory of each joint is given, as in MLD studies, and the connectivity is known in advance, then the segments are implicitly given. If the connectivity is not known, or if the only available information is a set of point locations in each frame of a sequence, then some intermediate steps are needed to determine the connectivity. Rashid tackled this problem with his system, called *Lights* [61]. *Lights* was provided with the location of a set of points on one or more object in each frame of a sequence. It could track and cluster points belonging to independently moving objects. Within a cluster, the relative motion of object points was also analyzed, and groups representing independently moving subparts were segmented.

A problem with stick figures, in general, is that depth is difficult to judge from an image sequence. Occlusion and disocclusion helps in depth evaluation; since line segments cannot occlude one another, depth cannot be determined [6]. If the three dimensional location of the endpoints of the segments are known, then it can be determined which segment is closer to the observer.

Volumetric models are intended to better represent the complexity of the human body. Generalized cones are the most commonly used models. A generalized cone is the surface swept out by moving a cross-section of constant shape but smoothly varying size along an axis [42]. However the volumetric models described below all use generalized cones restricted to having a cross-section of constant shape and size, and are called generalized cylinders. The cost for better representation is an increase in the number of parameters in order to describe the cylinders and their degrees of freedom. Usual parameters are thus the length of the long axis along with the radius of both major and minor axes, the location of the origin of the local coordinate system, and the transformation matrix relating the local coordinate system to the coordinate system of contiguous (connected) segments and/or to the origin of the model. The model proposed by Marr and Nishihara [42] consists of a hierarchy of cylinders, starting with a unique cylinder describing coarsely the size and orientation of the body. This overall model can be refined using a collection of component cylinders representing the different body segments, giving more detailed information about the spatial organization of the human shape. Each component cylinder is attached to another cylinder and its location in space defined relative to the principal axes of the model, in this case of the torso, by predefined relations. Marr and Nishihara specify those relations using cylindrical and spherical coordinates, which they called *adjunct relations*. One axis is determined relative to another by specifying the location of one of its endpoints in cylindrical coordinates with respect to the other axis, then specifying its orientation in spherical coordinates. The model has the advantage of being as refined as needed.

In O'Rourke and Badler's work [49], the model has a well-defined structure, consisting of segments and joints. Segments are defined as abstract rigid bodies with an associated embedded coordinate system. Each segment may have a number of joints at fixed locations in its coordinate system. A joint is a unique point joining two segments. The model is made up of 24 segments and 25 joints. The "flesh" is modeled by a collection (about six hundreds) of spheres located at fixed positions within a segment's coordinate system. The model includes constraints on joint angles,

and a collision detection method for non-adjacent segments.

Hogg's body model [28] follows Marr and Nishihara's. The prior information relates to the three dimensional shape and structure of the body, which is modeled by a set of fourteen elliptical cylinders representing the feet, legs, thighs, hands, arms, upperarms, head and torso. Each part is defined by its length and the size of the major and minor axes of the cross-section. The origin of the coordinate system of a part is the center of its corresponding cylinder, and its principal axis X, Y and Z are running to the left, top, and forward. Relative position of body parts is determined using geometric transformations that carries the coordinate axes of one part onto the other. The connection between parts is done explicitly; the joints between parts are specified in terms of the parts it connects, and a geometric transformation that defines the position of one part relative to the other. The transformations are composed of a translation from the origin of the first part to the joint, followed by a rotation to align the axes of both parts, then a second translation to overlay the axes.

Rohr [62, 63] also follows Marr and Nishihara's model, and his model also consists of a set of fourteen elliptical cylinders. Each cylinder is described by its length, and the size of the major and minor axes of the cross section. The coordinate system is aligned with the natural axes; the origin of the whole body is the center of the torso. Transformations between different coordinate systems are done through transformation matrices using homogeneous coordinates. The absolute size of the body parts is used and includes clothing. The author argues that the most usual way of seeing people is when they are clothed, so the parameters of the body should be representative of that fact.

One way to represent the projection of a human body in an image plane is to model it as a collection of ribbons or of antiparallel lines, also called *apars* (a special case of a ribbon). The body is actually modeled by the regions enclosed in the ribbons. In Leung and Yang's work [40], the body is represented as a collection of six apars (two legs, two arms torso and head). The relative location of each segment is of little importance; more weight is given to the width and length ratios between each segment, and the length/width ratio of the head. No precise description of each body part is necessary.

There is a reasonable equilibrium to be attained between the ability of the model to truly represent the body, and its simplicity, i.e. the number of parameters necessary to represent it. Obviously, good models are nicer to the eye, while simpler models might seem crude, but the overhead generated by the "nice" model might not be worth the trouble. Generalized cylinders are a good middle ground and are often used. Stick figures are simpler and could lead to faster implementations, but are not as interesting.

4.2 Modeling of Human Motion

Human motion can be modeled using joint angles. Joint angles have been extensively studied in physical medicine; reference [46] provides an extensive study of gait parameters for a group of normal and some abnormal subjects, as well as a bibliography on human gait. Joint angles are more formally expressed as flexion/extension, abduction/adduction and rotation angles. Flexion occurs when two body segments change their relative position such that the angle formed between them decreases; for example, bringing the thigh up toward the abdomen is a hip flexion. Extension is the return from flexion, i.e. when the angle between the two segments increases. Abduction (adduction) is the movement of a body segment away (towards) the midline of the body or of the body part to which it is attached. An example of abduction at the shoulder is the movement of the arm up from its original position; bringing it back would be the adduction of the shoulder

joint. Rotation occurs when a segment rotates about its longitudinal axis, for instance rotation of the head left or right with respect to the spine. Joint angles are usually expressed relative to one walking cycle, which is defined as the time interval between successive instants of contact of one foot to the floor, for the same foot. The forward motion has been shown to be almost constant within a cycle, while the vertical displacement of the head is relatively small considering the global motion. From the study above, normal locomotion is characterized by a smooth forward translation of the trunk and rythmicity in the length of successive steps, as well as in the duration of successive temporal components of the walking cycle. In patients impaired in some way in their walking, the smoothness and rythmicity are not necessarily present.

References [29, 33, 57] describe some systems aiming at the computation of the characteristic parameters of normal and abnormal walking motion, and which can be used to generate motion models for vision systems. The common systems use a set of cameras to track the displacement of markers placed on the limbs of a person, including at the joints, and from which the three dimensional location of the markers can be recovered. Others use more direct measurement devices like electrogoniometers, which directly assess joint angles, but require a more elaborate equipment attached to the subject that might slightly modify the motion and thus produce artifacts. Electromyographic data, to assess muscle activity, is sometimes used in addition to the above data.

In the computer vision field, the joint angles plotted in time (joint curves) for one walking cycle are used as a walking motion model for humans. They provide sufficient information for the determination of the posture of a person, i.e. the relative position of each body segment, throughout the cycle. Not all of the possible angle curves are used: the most common are the hip flexion/extension and the knee flexion/extension curves, along with the shoulder and elbow flexion/extension curves. Those angles curves can be used in two ways. First, given a particular time with respect to the walking cycle, the angles of all the joints above can be determined; conversely, given the angles at several joints, the time within the cycle can be estimated. Those angle curves are a very useful tool for modeling human walking motion. Other type of information can also be extracted to provide additional knowledge, for example constraints on possible angles for each joint, along with constraints on angular velocities. This kind of information could reduce the search space during tracking by constraining the possible angles and angular velocity between frames.

Hogg uses flexion/extension curves for the left hip, knee, shoulder and elbow joints in his walking model [28]. Those curves were generated by hand using data from one walking subject. Each curve is defined as a periodic cubic B-spline specified by nine control parameters, and is seen as an idealized joint curve. Each curve gives an angle, in degrees, as a function of the walking cycle, represented by a parameter called $PSTR$ varying from 0 to 1. The beginning of a cycle ($PSTR = 0$) corresponds to the left leg stretched forward and right leg stretched backward, and a $PSTR$ value of 0.5 means the body is halfway through the cycle. The walking motion on the right side is assumed symmetrical with respect to the left side, and the angle value for the right knee at $PSTR = 0$ hence corresponds to the angle value of the left knee at $PSTR = 0.5$. Given a value for $PSTR$, the eight joint angles can be extracted and a body posture can be determined. In Hogg's method, the world coordinate system has its origin on the ground plane. The $X - Z$ plane is parallel to the ground plane, and the Y axis is vertical. A geometric transformation connects the torso to the world. A parameter, TRS_B, determines the direction of motion of the body relative to the ground plane, and is actually a rotation about the vertical axis Y. A parameter called SPD represents the speed of the body in that direction. Parameters TRS_X and TRS_Z determine the position of the torso with respect to the ground, and their time derivative is a function of both TRS_B and SPD. Constraints on the values of $PSTR$, TRS_B and SPD and their time derivatives can

be used to restrain the search space in the next frame. For example, knowing the current $PSTR$ parameter and its time derivative range, the $PSTR$ value for the next frame can be evaluated. Constraints on SPD and the derivatives of $PSTR$ or TRS_B are given in advance. Assignment of values to the joint angles, the speed and direction of the body model, in a frame, defines its static representation, or posture. A set of such assignments, one for each frame of a sequence, defines a motion. A particular motion, like walking, is specified by placing constraints on the parameters of the model and their time derivatives. This set of constraints associated with a particular model and motion is called the model constraints.

Rohr [62, 63] also uses the flexion/extension curves of the hip, knee, shoulder and elbow. However, the curves originate from the data of a study on the gait of sixty men [46]. Each curve is the result of taking the joint angle at ten time instances within a walking cycle, and smoothing them by using periodic cubic splines. Rohr is the first to use kinesiological data as a basis for a walking motion model. He also uses a parameter called *pose*, similar to Hogg's PSTR, i.e. it represents a time instance within the cycle such that given *pose*, all joint angles can be found, and vice-versa.

More specific knowledge about the walking motion has been used by Chen and Lee [12]. They defined five rules pertaining to walking. The first two general rules are that (1) the two arms (legs) cannot be both in front of behind the torso simultaneously, and (2) the arm/leg pair on the same side of the body cannot swing forward or backward at the same time. Three other more stringent rules are also defined: (3) when the shoulder and elbow joints on either arm swing, they must do so in a cooperative manner (a similar rule applies for the hip/knee joints); (4) the trajectory plane on which the arm or leg swings is generally parallel to the moving direction; (5) at any time, there exists at most one knee having a flexion angle. Moreover, when there is such a flexion in one leg, the other leg stands nearly vertically on the ground.

A different approach for the modeling of motion was taken by Akita [1], who used a sequence of stick figures, called key frame sequence, to model rough movements of the body. Key frames were used traditionally in animation, where they provided the essential frames of a sequence. They were created by "master" animators, while the frames in between those key frames were filled-in by other animators [6]. In the case of Akita's work, the key frame sequence consists of an ordered sequence of stick figures, each differing from its predecessor and successor such that a body segment has crossed or uncrossed another body segment. The key frame sequence is determined in advance and referred to in the prediction process. Both Rohr and Hogg used keyframes in early versions of their work, but discarded this motion model in favor of the model based on joint angles from real humans, as described above.

4.3 Recognizing Body Parts

The main goal of the methods described here is to track and label each region of a body performing some action. The tracking consists of determining the location and shape of body parts, from frame to frame, while labeling involves identifying them. The movements performed are arbitrary.

Akita [1] used a key frame sequence of stick figures to approximately predict the location of a body part with respect to the other part, in the current image. The actual body is modeled using generalized cones. However, as far as the paper is concerned, only their projection on the image plane is used. The recognition of the parts is done is the following order: legs, head, arms and trunk. The author feels that the legs are the most "stable", i.e. more prominent and easier to recognize, while the trunk is the least prominent, because of its frequent change of shape due to occlusion. The legs are determined by finding the position of the inner and outer boundary pixels of each leg, using the original gray-level image and a binary edge map image. The head is

then detected using the predicted region estimated from the previous image along with the head diameter. The arms are then detected and labeled: all pixels outside what is estimated as the trunk's side lines and that don't belong to the legs or the head are arm pixel candidates. Their boundaries are determined at a later stage. The outline pixels from the estimated trunk region are labeled as trunk. When occlusion occurs, the authors devised a method in which difference images are used in order to find the precise outline of the body parts. The determination of legs, head, arms and trunk parameters is performed at every frame. Correspondence between frames is established using one of two methods. When the position change of a segment is small enough, its position can be predicted from the previous frame, using window code distances, which are defined in the paper. If window codes cannot find a correspondence, then the key frame sequence is used to find the current posture of the body. The position of the body parts is then recomputed.

In this method, the body structure must be known in advance, along with the key frame sequence of stick figures. The problem with this model is that the key frame sequence only gives us information about the relative position of the body segments, thus no temporal information is available from that representation; it cannot be predicted when a segment will cross another in the image sequence, nor the time (number of frames) between those events. Only the order in which the events will occur is given. The authors never mention explicitly how constraints on the body parts are computed and used. The authors do not show how the feature points or characteristic points are chosen for correspondence with window codes. Also, the computation of velocity and acceleration is not defined.

Leung and Yang [40] also tackle the problem of body labeling in a sequence of images. In a concurrently released paper [41], the authors described a segmentation method, using difference images and past history. The advantage of difference images is that motion can be localized to a particular portion of the image, therefore focusing attention. History referencing handles the case where no current motion is detected for a region, but where motion was detected, for that same region, in the previous frame. These regions where motion previously occurred are added to the current segmented picture.

The process is made up of two steps: region description and body part identification. The region description process abstracts the segmented image from the segmentation algorithm, to extract the antiparallel lines (apars) that will be used for labeling. An oriented line estimation of the region boundaries is first performed. An apar detection is performed, followed by a growing procedure which recursively finds new pairs of antiparallel lines and concatenates them to the current apar. More processing is done on the detected apars in order to delete unwanted ones or concatenate more apars, according to some heuristics defined by the authors. Finally, apars are selected according to the following: (1) if they have been concatenated; (2) if the ratio length/width > 1; (3) if they are moving apars closed at either or both ends.

The labeling is performed in three steps. The first will, according to some constraints, map potential apars to the model. The constraints are that (1) the width ratio of arms, legs, head and trunk is approximately 1:2:2:4; (2) the length/width ratio of the head smaller than 1.2; (3) if a pair of apars are labeled as arms or legs, they must possess similar intensity distribution; and finally (4) the width of the trunk is a bit larger than twice leg width. These heuristics have been determined experimentally. After labeling the apars, the choice for the best representatives is done by assigning weights that reflect the likelihood of an apar to be one of the six body apars. The choice of the 6 body apars is done by: (1) interval partitioning, which partitions the apars in 3 groups with width ratios 1:2:4; (2) pairing (for legs and arms), to associate apar pairs in a same interval according to the constraints mentioned above; (3) selection process, which determines the most appropriate apars to represent the model. The arms are the two pairs with highest total weight of the first

interval. The legs and head, both from the second partitioned interval, are chosen next, again with highest total weight. The trunk is the apar with highest weight of its category, but its width must be twice that of the legs.

Both Akita's and Leung and Yang's works are to label body parts in a sequence of frames. Both use a similar body model consisting of six body parts; Leung and Yang however use a collection of apars bounding a region, while Akita uses generalized cones (more precisely their two dimensional projection) to describe the regions. In both cases, length and width of each region needs to be computed, as well as the axis in Akita's case. As mentioned previously, Leung and Yang's approach is more dependent on predefined thresholds. Akita's method seems more sound: the key frame sequence is a good idea for rough representation of motion, although no temporal information can be extracted from it. Both works involve labeling of body parts, and Akita's also include correspondence in successive frames. Another advantage of tracking is that computation from frame to frame can use current information, which is done in Akita's work through window codes. In Leung and Yang's work, the labeling process seems to be repeated at every frame; use of the current information to predict even a rough position of the body parts might be more cost effective than total recomputation. Akita's work explicitly addresses the problem of occlusion of body parts: its key frame sequence is based on occlusion events, and a method is described for occluding/occluded parts. Leung and Yang's implicitly address the problem in their segmentation algorithm. Both use difference images for that purpose. Akita's work, to our opinion, is simpler yet more complete than Leung and Yang's, which depends too much on thresholds that might vary eventually with every person. However Akita's work need more a priori information; the motion must be known and be explicitly described with the key frame sequence.

4.4 Three-Dimensional Tracking

Three-dimensional tracking consists of determining the sequence of three dimensional positions of a body, along with its posture, i.e. the relative position of its parts, from the frame by frame analysis of a sequence. The object, in this case a human body, is modeled using stick figures or generalized cones. The analysis of a frame provides the information necessary to update the posture and position of the body model in space. The validity of the updated model is verified by comparing its projection on the image plane with the object in the sequence. An initial estimate of the body model's position in space needs to be given or determined in order for the frame by frame analysis to be used subsequently. This step is performed in different ways for the systems that will be described. Section 4.4.1 will describe a system using a stick figure model, while section 4.4.2 will describe systems where volumetric models were used.

4.4.1 Tracking with Stick-Figure Models

The goal of the work done by Chen and Lee [12, 39] is to find the sequence of three dimensional body configurations (postures) of stick figures which, when projected on an image plane, would give rise to the image sequence, given the set of end point positions in each frame. The first part consists of finding all possible three dimensional configurations for each frame. After this is done, the task is then to find the best sequence of configurations.

The structure of the model consists, as mentioned in section 4.1, of fourteen joints and seventeen segments, plus six more feature points on the head. The three dimensional length of all segments is given, along with the relative three dimensional location of the 6 head feature points. Using those six feature points, the transformation matrix from body coordinate system into a camera

coordinate system is determined. After the location of the head features are determined in camera coordinate system, the location of the joints of the body model can be recovered from joint to joint, using the given segment lengths. It was shown in [39] that there are generally 2 possible solutions for the three dimensional coordinates of a feature point serving as the end point of a segment, given the three dimensional location of a start point and the length of the segment. The set of joints are represented as a tree with the known neck joint serving as root. The nodes at each level represent the possible three dimensional locations of a joint. Each node has two successors corresponding to the two possible solutions, except for degenerate cases where there is only one solution. A path from the root to a leaf determines a body configuration or body posture. However, among those, quite a few are not allowed because of body constraints. Knowledge of physical constraints and motion constraints are used to prune this tree.

Physical constraints comprise angle constraints, distance constraints and collision-free constraints. Four categories of angles are associated with the body joints: flexion/extension, abduction/adduction, rotation and bending (lateral flexion). The ranges allowed for each joint are precisely defined in [39], for each of the categories above. Distance constraints are used in particular to recover the coordinates of the joints. Collision-free constraints imply checking if arm segments penetrate the torso, and whether arm and leg segments collide with each other. Angle constraints are not sufficient to limit the body configuration that are valid for the walking motion. A set of rules has been defined for this purpose, and they can be as general or as stringent as one needs. These were defined in section 4.2. If a configuration does not satisfy those rules, it will be rejected.

The normal movement during walking is a smooth and continuous motion. In this study, this movement is considered as a collection of smooth and continuous angular motions of all body segments, which is expressed as a nearly constant angular velocity, or equivalently a close to null angular acceleration. For a configuration $\{X_i\}$ in frame i, the position of each joint is used to compute the relative translational velocity V of segments between two consecutive frames. The relative angular velocity $\omega_{\overline{AB}}$ and acceleration $\alpha_{\overline{AB}}$ of segment $\overline{AB}$ are also computed. An angular acceleration function associated with body configurations x_i, x_{i+1} and x_{i+2} was defined as:

$$f(x_i, x_{i+1}, x_{i+2}) = \sum_{\overline{AB}} |\alpha_{\overline{AB}}(x_i, x_{i+1}, x_{i+2})| \ ,$$

i.e. the sum, over all segments, of their angular acceleration. The overall angular acceleration function over N frames is defined as:

$$f(x_1, x_2, \ldots, x_N) = f_1(x_1, x_2, x_3) + f_2(x_2, x_3, x_4) + \ldots + f_{N-2}(x_{N-2}, x_{N-1}, x_N).$$

So finding a smooth motion is done by minimizing the function $f(x_1, \ldots, x_N)$, which is solved as a graph search problem.

In this method, the relative three dimensional location of the head points are necessary at every frame in order to compute the transformation matrix for the camera coordinate system and then determine the rest of the joint locations.

4.4.2 Tracking with Volumetric Models

Tracking with volumetric models is more complex because of the increase of the number of parameters required to represent the model itself. The body models consist of a cylinder for each hand, arm, forearm, foot, leg, thigh, trunk and head. The three dimensional position of the body model and the relative position of its segments (its posture) must be determined in each frame.

The posture is parametrized by a parameter, $PSTR$ in Hogg's work, *pose* in Rohr's ($PSTR$ and *pose* are the same), ranging from 0 to 1. In both, this parameter is determined and plotted as a function of the frame number, along with its position with respect to the world coordinate.

Hogg

In Hogg's work, the three dimensional body model described in section 4.1 is used to estimate the three dimensional motion of a person seen in an image sequence. A frame analysis will provide an estimate of the person's three dimensional position and posture. As mentioned in sections 4.1 and 4.2, the body model uses a set of cylinders to represent body parts. Each cylinder is determined by the length of its axes and the center of its coordinate system. The relative position between the parts are defined explicitly by geometric transformations. Important parameters are $PSTR$, speed SPD and direction of motion TRS_B. Assigning a value to each parameter of the body model defines its posture, and a sequence of such assignments, one for each frame, specifies a motion of the body. The purpose here is thus to find the set of assignments to the parameters that satisfies the constraints that define the walking motion for that model. The system will track, i.e. will estimate, at each frame, the best parameters and verify that the projection of the body model on the image plane fits with the person seen in the image sequence.

The tracking at each frame is done through a function called TRACK. TRACK uses a function called SEARCH, which searches the optimal parameter assignments for the current image. The parameters chosen for the previous frame, along with the model constraints, define a range of possible postures for the current frame, which is a subset of the model constraints. This subset will be searched in order to find the best set of parameters. Box constraints are used to simplify the process, i.e. constraints are partitioned into sets of closed intervals. An evaluation function will be given representatives of those intervals, and the result used as part of a plausibility function. The evaluation function takes a set of parameters and determines if the model it defines is plausible. All the possible combinations of parameters will be evaluated. The plausibility of the model is computed by projecting the model on the image plane and matching its projection with the actual edge features of the image. If a good match occurs, then the instantaneous description will be highly plausible. The plausibility EVAL of the model is defined as the sum of the plausibility of its parts:

$$EVAL = \sum_i w_i * PEVAL_i$$

where $\sum_i w_i = 1$ and $PEVAL_i$ is the plausibility of the i'th part. The weights are given beforehand, and relate to the confidence we have on each part's assessment. For instance, more weight has been placed on the legs than on the arms because the latter are more often obscured by the torso.

The global search is organized in the following way: all possible positions of the torso are generated. For each torso position, an independent search for each limb is performed, in order to find the optimal limb positions relative to the torso. Combining the evaluation for each limb with of the torso and head provide the optimal plausibility of the entire model relative to the position of the torso. This calculation is repeated for all possible torso positions, and the model with the highest plausibility value is then chosen as the posture or configuration for the current frame. The whole process is then repeated for the next frame, and so on.

When strong constraints already exist, SEARCH is the best method to provide the best configuration. But when the search domain is too large for SEARCH, as in the first frame of the sequence when no previous information restricts the position of the body within the scene, and the $PSTR$ parameter is unknown, two alternate functions are described, HSEARCH and DIFF, which will

provide estimates of the parameters that will be used subsequently by SEARCH.

Rohr

In Rohr's approach, the body model consists of fourteen cylinders with elliptical cross-sections, as described in section 4.1. The motion model uses joint angle curves based on walking motion studies. The method comprises two phases. The first phase, called the initialization phase, provides an estimate for the posture and three dimensional position of the body using a linear regression method; the second phase, starting with the estimate from the first phase, uses a Kalman filter approach to incrementally estimate the model parameters.

The initialization phase analyzes the first ten to fifteen images in order to obtain starting values for the next phase. Image analysis consists of the segmentation into moving and non-moving regions, using a change detection algorithm. The fits are compared from frame to frame, and the image points where change occurred are marked. A rectangle surrounding the marked pixels is determined, along with the outline of the moving object and the velocity field. An estimate of the three dimensional position, using this enclosing rectangle and an assumption about the height of the pedestrian is found by solving a system of linear equations. For a better estimate of the three dimensional position and the determination of the *pose* parameter, model contours are matched with image edges. Edge points in the image are linked using an eigenvector line fitting algorithm. For each projected model contour, i.e. for the projection, on the image plane, of each contour of each body part, an enclosing window is determined, and the line fitted edge points inside this window will be used in a similarity calculation. The similarity involves computing the length l_i of the projection of the fitted edge on the model contour whose length is l_{Mi}, its angle $\Delta\phi_i$ relative to the contour, and the distance d_i between the midpoint of the edge (inside the window) and its corresponding projection on the contour line:

$$s_i = l_i \ \exp\left(-\frac{1}{2}\left(\frac{(l_{Mi}-l_i)^2}{\sigma_{l_i}^2}+\frac{d_i^2}{\sigma_{d_i}^2}+\frac{\Delta\phi_i^2}{\sigma_{\Delta\phi}^2}\right)\right),$$

where $\sigma_{l_i}, \sigma_{d_i}$ depend on l_{Mi} and $\sigma_{\Delta\phi}$ is constant. The overall similarity between the model edges and the graylevel edges is the sum of s_i for all visible model edges, normalized by the sum of corresponding values l_{Mi}:

$$s(p) = \frac{\sum_{i=1}^{n} s_i}{\sum_{i=1}^{n} l_{Mi}}.$$

The model's *pose* and three dimensional location, represented by parameter p, is chosen such that $s(p)$ is maximized. Fixing the three dimensional position and varying the *pose* parameter within the walking cycle, a similarity curve is created; the state of motion with highest similarity is the one chosen as *pose*. Once the image analysis is done and the parameters determined for those images, a linear regression analysis is performed to provide initial values for the Kalman filter. From this point on, the image analysis will be limited to the search for the three dimensional position and parameter values by matching with graylevel edges, to provide a current measurements' vector at each frame for the filter. The knowledge of the *pose* parameter and its time derivative allows for a greatly reduced search space, as opposed to the whole cycle of the initial phase. The size of this search space will depend on the parameters' uncertainties, represented by the covariance matrix of the Kalman Filter. They can vary somewhat from frame to frame, a high uncertainty leading to a larger search space. Typically, the search space will be around ± 0.2 of the estimated *pose* parameter. The state vector for the Kalman filter is $p_k = (X_k, X'_k, Y_k, Y'_k, Z_k, Z'_k, pose_k, pose'_k)^T$,

where (X_k, Y_k, Z_k) is the three dimensional position in frame k, x' represents the first derivative with respect to time. At each frame, the three dimensional position and *pose* parameter is thus computed and fed to the Kalman filter, which will then provide an estimate for the new position and *pose* in the next frame. These two steps are repeated, and the model parameters are thus determined for the whole sequence.

Comparison Between Methods

Rohr's and Hogg's approach might seem very similar to one another from a global point of view. Both use a similar three dimensional model, Hogg's $PSTR$ and Rohr's *pose* parameter carry the same information, they both use joint angle curves for the same joints of the body. However they differ in several ways. Rohr removes hidden contours of his body model, arguing the recognition will be more robust. The joint angle curves in Rohr's approach were taken from kinesiological data, while Hogg's data is the result of one person walking. The similarity measure in Hogg's paper uses edge points, while Rohr uses edge lines, which also appears more robust. In both cases, the similarity measure is computed for each body part, although Rohr computes a global similarity measure while Hogg finds the best for each body part. Rohr does not use box constraints for the parameters, instead, the search space for the *pose* parameter and the three dimensional location is controlled automatically according to the uncertainties of the parameters. The use of the Kalman Filter provides also more robust and smoother results. Rohr provides no results when the pedestrian does not walk on a plane parallel to the image plane; Hogg provided some results on such a sequence. Rohr has a starting phase that provides starting values for the Kalman Filter computation, using a change detection algorithm and linear regression. Hogg uses one of two procedures, DIFF or HSEARCH, the first based on a difference approach and providing an approximate location of the body, the second providing an approximate location and configuration of the body. However, in a few experiments, some of the parameters have been entered by hand instead of using one of those two procedures.

4.5 Human Motion Recognition

Human motion presents a special challenge because of the amount of possible configurations of the body, seen as an articulated object. In this case the different motions of the body segments need to be determined with respect to each other. Two aspects can be seen to that problem: the recognition of the motion performed by a human, and the discrimination between different people performing the same action. They will be discussed below.

4.5.1 Recognition of Human Movements

Recognition of human motion imply the ability to discriminate between different actions, those actions being actor dependent or not. One of the works will describe how to distinguish between walking, running and skipping actions; the other describes a method for the recognition of tennis strokes.

Using MLDs, Johansson [31] showed that human motion like walking can be recognized within 200ms, less than half of a cycle. The scope of Goddard's thesis [24] is the recognition of 400ms MLDs generated from single actors moving parallel to image plane, using a connectionist approach. The most interesting part of his work consists of the spatial and temporal integration of low-level shape and motion features to form higher level features, and the indexing of high-level models of movements into a database of known models.

Given the set of trajectories of points in a sequence, line segments are extracted by processes outside the network; they are the lowest level features actually used in the network. The goal is to combine line segments together to form legs or arms, then to combine pairs of arms and pairs of legs together to form upper and lower body limbs, and finally to combine upper and lower limbs for the description of the complete motion. However, not only do the arms and legs pairs need to be properly linked in space, their relative motion in time must also correspond to proper body motion. To achieve this level of complexity, a hierarchical system is described, comprising two pathways, the shape pathway and the motion pathway. They both work in similar fashion, in that they start at the segment level; the next level combines segments and is called the component level; the third level combines two components together and is called the assembly level. The shape pathway records variation in length and direction of the segments, while the motion pathways records direction (in a coarser manner) and changes in rotational velocity of the segments. At each level, if a feature is present, its spatial location is determined, and a unit at this location is activated. Components and assemblies are combined in a higher level hierarchy called the scenario hierarchy. Scenarios represent temporal series of events, with information on sequence and duration (the scenario model representation is based on Feldman's paper[20]). The scenario hierarchy is made up of two levels. The lower level integrates shape and motion information from the shape and motion pathways, and the scenarios represent temporally extended movements of the objects. The higher level represents temporally and spatially coordinated combinations of lower-level scenarios. This level is the object level which determines the type of gait that is being represented. An example of lower level scenario is the motion of a pair of legs through one walking gait cycle, while at the highest level, the pair of legs has to be coordinated in time and space with the motion of a pair of arms to describe the motion of the whole body during a walk. The knowledge-base consists in a set of valid scenarios that the system has been "trained" to recognize. In this case, it consists in a scenario for each of three different gaits for one actor (in the first set of experiments), or in an "aggregate" of scenarios for the same three gaits, but taken from four actors; those scenarios are intended to be actor independent.

The discrimination between different tennis strokes was investigated by Yamato et al. [75] using Hidden Markov Models (HMMs). A HMM consists of a set of states $Q = \{q_1, q_2, \ldots, q_l\}$, a set of output symbols $V = \{v_1, v_2, \ldots, v_r\}$, a matrix A whose elements consist of probabilities of transition between every state, a matrix B of output symbol probabilities for each state, and a vector π of initial state probabilities. The model works as follows. The initial state q_i is chosen with a probability π_i. The HMM will change state from q_j to q_k according to probability $A(j,k)$. At each state q_k, only one output symbol v_m is produced with probability $B(k,m)$. For a sequence of length t, t output symbols will be produced.

An image sequence is processed in three steps. In the first step, an observed sequence O of output symbols is derived, each symbol associated to a mesh feature, as computed in chapter 2.6. For n the size of the feature vector (i.e. the number of mesh elements), the space R^n is divided into clusters via pattern classification techniques, and an output symbol v_j is assigned to each of the clusters. In other words, an output symbol v_j is assigned to cluster center c_j. A feature vector will be transformed into the symbol assigned to the closest cluster center using an arbitrary distance measure.

In the second step, sequences are used to train the HMMs. There will be as many HMMs as there are different motions, in this case tennis strokes, to be recognized. During this phase, the parameters $\lambda = \{A, B, \pi\}$ of an HMM are optimized for one particular tennis stroke, i.e. the HMM will generate the sequence of output symbols for that stroke. This is done using the Baum-Welch algorithm (see [75] for reference): the probability values of A, B and π are iteratively refined until

they maximize $P(O \mid \lambda)$. Once the set of parameters λ_i for every HMM i are determined, the learning phase is done.

Finally, the recognition is done in the following manner: given the sequence of observed symbols $O = O_1 O_2 \ldots O_t$, we want to find the HMM j which is most likely to generate the same sequence, i.e. find j such that

$$j = arg\{\max_i (P(\lambda_i|O))\}$$

The likelihood of each HMM is calculated, and the most likely is the one chosen.

Both methods described above generated strong recognition results. Because of the probabilistic nature of the HMMs, along with the characteristics of mesh features, the method used by Yamato et al. [75] is not too sensitive to noise. It is very versatile and the recognition part could be parallelized. Furthermore, is easy to add a new stroke to recognize, by simply training an HMM for a new motion and then adding it to the already trained set of HMMs. However, several preprocessing steps are necessary before feature vector extraction can be performed on each frame. Goddard's approach is very interesting [24]. Each motion is defined as a coordinated sequence of angular velocity changes, which could be seen as another type of motion model. The temporal factor is very important, and, the set of events, for a particular motion, but in the wrong order will not lead to recognition of that motion. Actor-dependent and actor-independent recognition was achieved with his connectionist approach.

4.5.2 Discrimination Between Humans From Their Motion

From our own experience, it is relatively easy to recognize a friend from the way he or she walks, even though this person is at a distance such that the face features are not recognizable. The studies described here can be used in a general way to distinguish between trajectories and can be directly used for motion discrimination as in the previous subsection. However the authors go further and devised a method for recognizing different persons from the subtle differences in the way they walk.

The work of Rangarajan et al. [58] aims at gait and motion discrimination. The authors describe a method that will be able to distinguish between objects having the same shape but different motions and between objects having same motion but different shapes. They base their algorithm on the trajectories of the joints of a human body performing walking motion, i.e. moving light display type of input. Two algorithms are described. The first uses only motion information for matching a pair of trajectories. The second is an extension of the first for matching multiple trajectories, which will ultimately be integrated into an object motion-based recognition system.

MLD type stimuli are used as input; the trajectories are then parametrized using speed and direction, as described in section 2.3. The matching algorithm is based on matching the diffused scale-space of both the speed and direction curves (see section 2.7). The scale-space is computed by repeatedly convolving the input speed or direction signal with a second derivative of Gaussian mask with various σ values. The output is then checked for zero-crossings, indicating discontinuities. The location and potential (the absolute difference between the values where the zero-crossing occurs) of each zero-crossing is stored in a set of arrays, one for each σ value. The set of arrays is organized into a two dimensional table, with the location (frame number) as the x axis and σ as the y axis. The zero-crossings from that table are diffused by convolving with a two dimensional Gaussian of standard deviation equal to 1. The diffusion is necessary because convolution with a Gaussian leads to a delocalization of the zero-crossing relative to the discontinuity. The diffusion ensures that similar trajectories will produce overlapping scale-spaces, and thus increases the robustness

and decreases sensitivity to noise. Scaling is then performed. The matching itself is done by an element by element subtraction of the diffused input and model scale-spaces. The absolute values from the subtraction are stored, and match scores are computed. At the end, two tables containing the results of the subtraction, one table for the speed, the other for the direction, are left. Match score for speed and direction are given as:

$$speed\ score = 1 - \frac{\sum\sum|\epsilon_s(n,\sigma)|}{2 * |\sum\sum\alpha_s(n,\sigma)|},$$
$$direction\ score = 1 - \frac{\sum\sum|\epsilon_d(n,\sigma)|}{2 * |\sum\sum\alpha_d(n,\sigma)|},$$

where ϵ_s and ϵ_d are the arrays containing the element by element subtraction of the input and model, for speed and direction respectively, and α_s, α_d represent model speed and direction. The global match score is the average of speed and direction match scores. The matching of multiple trajectories is an aggregate of simple trajectory matches. This scheme uses also shape information. Motion information is used in each individual component trajectory, as described above, while spatial information is gathered between any pair of components. For spatial information, Euclidean distance between points in each frame is used; a measure for spatial match between trajectory pairs is given in the paper.

Tsai et al.'s method for cyclic motion detection has been extended to object recognition since, in many cases, the trajectory of several points on an object with a predefined motion can identify it. Given the curvature of a trajectory and its computed frequency, one cycle can be isolated and used for recognition, as in Rangarajan et al. [58]. One limitation to that recognition method is that both model and unknown must be aligned with respect to their cycles, but the authors are working on resolving that problem.

4.6 Summary

Several methods were reported that analyzed the motion from a sequence of images in order to recognize the different parts of the body, or to recognize motion in time. They are flexible enough to allow for small differences in shape and/or motions, and can be applied to any type of motion. The tracking methods can distinguish between allowed and non-allowed configurations or postures. The distinction is possible through the description of a body and motion model, which put constraints on configurations and on changes allowed between two consecutive configurations in time. The methods are also relatively robust to noise, since most of the real image sequences were taken in an outdoor environment.

5 Conclusion and Future Directions

Motion-based recognition consists of the recognition of objects or motions directly from motion information extracted from the sequence of images. Knowledge about the object or motion is used to construct models that will ultimately serve in the recognition process. This paper intented to emphasize the process involved in motion-based recognition, and to describe the different methods so far reported. The process consists of mainly two steps, the extraction of motion information from the image sequence and their organization into models, and the matching of an input sequence with a model in a database of models. Most of the information obtained from the images is derived from optical flow or from token extraction and correspondence throughout the sequence, which

determine a trajectory. Trajectories are often parametrized into velocity v_x and v_y, speed and direction, or curvature, in order to get single valued functions as opposed to vector valued functions. Relative motion is an important aspect of our perception of motion. It has been successfully used in the case of articulated motion with the computation of angular velocities. Motion events, like starts and stops, also showed to be useful since they are actually perceived by our visual system. Region-based features extract some motion feature over a region of interest and summarize them with only a few representative values. Matching is often performed through classification techniques, although in several studies specific methods have been developed for that purpose. Several methods were reported, in particular the detection and recognition of cyclic motion, lipreading, gesture interpretation, motion verb recognition and temporal textures classification. Methods for recognition of human motion, like walking and running, labeling and tracking from motion and shape models were also discussed.

There are several problems to be addressed in the future. In the case of multiple objects moving in a scene, proper segmentation of the different objects in the image remains a difficult task. It is sometimes difficult to correctly locate an object in the scene, i.e. to detect features, lines and joints, because of poor contrast, noise, or other similar reasons. In the case of humans, clothing can cause segmentation errors or a tracking program to erroneously estimate the position of a limb, for example. These are reasons why experiments are usually performed in constrained environments or with special apparatus, e.g. glove with contrasting tips [18] or dots placed around the mouth of a speaker in [21] for consistent tracking. If motion-based recognition methods are to be used more widely, feature extraction will have to be performed in noisy environments and without any particular enhancements. For instance, stereo images could be used, where surfaces could help locate the object's shape. Explicit three dimensional models of objects could be used to track them. For example, a three dimensional model of a hand that is tracked in a manner similar to that described for the human body could provide more precise information about a performed gesture.

Perceptual organization of trajectories or spatiotemporal curves is an emerging theme. It has been shown that spatial invariants exist, which permit us to infer three dimensional information from a two dimensional image projection. For instance, it is highly likely that two parallel lines in two dimensions correspond to parallel lines in three dimensions. Those types of invariants applied to motion could be very insightful. For example, two dimensional elliptical trajectories imply a rotation motion in three dimensions; a set of elliptical trajectories with parallel major and minor axes corresponds to the motion of points on a single rotating object in three dimensions. The determination of those types of motion invariants that are reliable and stable, provide a new avenue for this type of research. Similarly, the clustering of spatiotemporal flow curves can provide a representation for coherent motions like a translation or rotation [4]. A hierarchical clustering of these curves can lead to the detection of different objects, their particular motion, their occlusion/disocclusion by one another, and even relative and common motion could be inferred. Thus, dynamic perceptual organization can be a very useful research direction that could lead to very interesting approaches and results.

A significant part of future research will remain application oriented, i.e. the need will dictate the kind of systems that will be developed. Applications will furthermore preferably run in real-time, and hardware solutions will be necessary. Although some of the methods described here are very versatile, recognition of a wide variety of objects and motions remains to be achieved. Such systems would necessitate a large number of features to be extracted, a very general representation and probably more robust and sophisticated matching procedures. Keeping the systems small enough so that they run efficiently and remain manageable thus requires them to be task specific.

We think this trend will remain for some time.

References

[1] K. Akita. Image Sequence Analysis of Real World Human Motion. *Pattern Recognition*, 17(1):73–83, 1984.

[2] M. C. Allmen. *Image Sequence Description Using Spatiotemporal Flow Curves: Toward Motion-Based Recognition.* PhD thesis, University of Wisconsin–Madison, 1991.

[3] M. C. Allmen and C. R. Dyer. Cyclic Motion Detection Using Spatiotemporal Surfaces and Curves. In *Proc. 10th Int. Conf. Pattern Recognition*, pages 365–370, 1990.

[4] M. C. Allmen and C. R. Dyer. Computing Spatiotemporal Relations for Dynamic Perceptual Organization. *Computer Vision, Graphics and Image Processing: Image Understanding*, 58(3):338–351, November 1993.

[5] N. I. Badler. Temporal Scene Analysis: Conceptual Descriptions of Object Movements. Technical Report 80, Dept. Computer Science, Univ. Toronto, February 1975.

[6] N. I. Badler and S. W. Smoliar. Digital Representations of Human Movement. *Computing Surveys*, 11(1):19–38, March 1979.

[7] D. H. Ballard and C. M. Brown. *Computer Vision*, chapter 7. Prentice-Hall, 1982.

[8] C. D. Barclay, J. E. Cutting, and L. T. Kozlowski. Temporal and Spatial Factors in Gait Perception that Influence Gender Recognition. *Perception and Psychophysics*, 23(2):145–152, 1978.

[9] J. L. Barron, D. J. Fleet, S. S. Beauchemin, and T. A. Burkitt. Performance of Optical Flow Techniques. In *Proc. Conf. on Computer Vision and Pattern Recognition*, pages 236–242, 1992.

[10] T. J. Broida and R. Chellapa. Estimating the Kinematics and Structure of Rigid Objects from a Sequence of Monocular Images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-13(6):597–613, 1991.

[11] T. W. Calvert, J. Chapman, and A. Patla. Aspects of the Kinematic Simulation of Human Movement. *IEEE Computer Graphics and Applications*, 2(9):41–50, 1982.

[12] Z. Chen and H.-J. Lee. Knowledge-Guided Visual Perception of 3-D Human Gait from a Single Image Sequence. *IEEE Trans. on Systems, Man, and Cybernetics*, 22(2):336–342, March-April 1992.

[13] C. L. Cheng and J. K. Aggarwal. A Two-Stage Approach to the Correspondence Problem via Forward-Searching and Backward-Correcting. In *Proc. 10'th Int. Conf. on Pattern Recognition*, pages 173–177, Atlantic City, NJ, 16–21 June 1990.

[14] R. Cipolla, Y. Okamoto, and Y. Kuno. Robust Structure form Motion Using Motion Parallax. In *Int. Conf. on Computer Vision*, pages 374–382, 1993.

[15] J. E. Cutting and D. R. Proffitt. The Minimum Principle and the Perception of Absolute, Common, and Relative Motions. *Cognitive Psychology*, 14:211–246, 1982.

[16] T. J. Darrell and A. P. Pentland. Recognition of Space-Time Gestures Using a Distributed Representation. Technical Report TR-197, M.I.T. Media Laboratory Vision and Modeling Group, 1992.

[17] T. J. Darrell and A. P. Pentland. Space-Time Gestures. In *Proc. Conf. on Computer Vision and Pattern Recognition*, pages 335–340, New York, NY, June 15–17 1993.

[18] J. W. Davis and M. Shah. Gesture Recognition. In *European Conference on Computer Vision*, Stockholm, Sweden, May 2-6, 1994.

[19] S. A. Engel and J. M. Rubin. Detecting Visual Motion Boundaries. In *Proc. Workshop on Motion: Representation and Analysis*, pages 107–111, Charleston, S.C., May 7–9 1986.

[20] J. A. Feldman. Four Frames Suffice: a Provisional Model of Vision and Space. *The Behavioral and Brain Sciences*, 8:265–289, 1985.

[21] K. E. Finn and A. A. Montgomery. Automatic Optically-Based Recognition of Speech. *Pattern Recognition Letters*, 8:159–164, 1988.

[22] M. Fukumoto, K. Mase, and Y. Suenaga. Real-Time Detection of Pointing Actions for a Glove-Free Interface. In *IAPR Workshop on Machine Vision Applications*, pages 473–476, December 7–9 1992.

[23] N. H. Goddard. The Interpretation of Visual Motion: Recognizing Moving Light Displays. In *Proc. Workshop on Visual Motion*, Irvine, CA, March 20–22 1989.

[24] N. H. Goddard. *The Perception of Articulated Motion: Recognizing Moving Light Displays.* PhD thesis, University of Rochester, 1992.

[25] 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*, chapter 14. SPIE Optical Engineering Press, June 1992.

[26] K. Gould and M. A. Shah. The Trajectory Primal Sketch: a Multi-Scale Scheme for Representing Motion Characteristics. In *Proc. Conf. on Computer Vision and Pattern Recognition*, San Diego, CA, June 4–8 1989.

[27] D. D. Hoffman and B. E. Flinchbaugh. The Interpretation of Biological Motion. *Biological Cybernetics*, 42:195–204, 1982.

[28] D. C. Hogg. *Interpreting Images of a Known Moving Object.* PhD thesis, University of Sussex, 1984.

[29] R. B. Davis III. Clinical Gait Analysis. *IEEE Engineering in Medicine and Biology Magazine*, pages 35–40, September 1988.

[30] M. Jenkin. Tracking Three Dimensional Moving Light Displays. In N. I. Badler and J. K. Tsotsos, editors, *Proc. Inter. Workshop on Motion: Representation and Perception*, pages 171–175. Elsevier, 1986.

[31] G. Johansson. Visual Perception of Biological Motion and a Model for its Analysis. *Perception and Psychophysics*, 14(2):210–211, 1973.

[32] G. Johansson. Visual Motion Perception. *Scientific American*, pages 76–88, June 1975.

[33] M. P. Kadaba, M. E. Wootten, H. K. Ramakrishnan, K. Hurwitz, and G. V. B. Cochran. Assessment of Human Motion with Vicon. In *Proc. Biomechanics Symp. ASME*, pages 335–338, New York, 1987.

[34] M. Kenner and T. Pong. Motion Analysis of Long Sequence Flow. *Pattern Recognition Letters*, 11:123–131, 1990.

[35] M. Kirby, F. Weisser, and G. Dangelmayr. A Model Problem in the Representation of Digital Image Sequences. *Pattern Recognition*, 26(1):63–73, 1993.

[36] R. Koch. Dynamic 3D Scene Analysis Through Synthetic Feedback Control. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(6):556–568, June 1993.

[37] D. Koller, N. Heinze, and H.-H. Nagel. Algorithmic Characterization of Vehicle Trajectories from Image Sequences by Motion Verbs. In *Proc. Conf. on Computer Vision and Pattern Recognition*, pages 90–95, 1991. Extended version.

[38] R. Kories and G. Zimmermann. A Versatile Method for the Estimation of Displacement Vector Fields from Image Sequences. In *Proc. IEEE Workshop on Motion: Representation and Analysis*, pages 101–106, Charleston, SC, May 7–9 1986.

[39] H.-J. Lee and Z. Chen. Determination of 3d Human Body Postures from a Single View. *Computer Vision, Graphics, and Image Processing*, 30:148–168, 1985.

[40] M. K. Leung and Y.-H. Yang. A Region Based Approach for Human Body Motion Analysis. *Pattern Recognition*, 20(3):321–329, 1987.

[41] M. K. Leung and Y.-H. Yang. Human Body Motion Segmentation in a Complex Scene. *Pattern Recognition*, 20(3):55–64, 1987.

[42] D. Marr and H. K. Nishihara. Representation and Recognition of the Spatial Organization of Three-Dimensional Shapes. In *Proc. R. Soc. London B*, volume 200, pages 269–294, 1978.

[43] G. A. Martin and M. Shah. Lipreading Using Optical Flow. In *Proc. National Conference on Undergraduate Research*, March 1992.

[44] K. Mase and A. Pentland. Lip Reading: Automatic Visual Recognition of Spoken Words. Technical Report 117, M.I.T. Media Lab Vision Science, 1989.

[45] H. P. Moravec. Towards Automatic Visual Obstacle Avoidance. In *Proc. 5'th Intern. Joint Conf. on Artificial Intelligence*, page 584, August 1977.

[46] M. P. Murray. Gait as a Total Pattern of Movement. *American Journal of Physical Medicine*, 46(1):290–333, 1967.

[47] R. C. Nelson and R. Polana. Qualitative Recognition of Motion Using Temporal Texture. *CVGIP: Image Understanding*, 56(1):78–89, July 1992.

[48] B. Neumann and H.-J. Novak. Event Models for Recognition and Natural Language Description of Events in Real-World Image Sequnces. In *Proc. IJCAI-83*, pages 724–726, Karlsruhe, FRG, 8–12 August 1983.

[49] J. O'Rourke and N. I. Badler. Model-Based Image Analysis of Human Motion Using Constraint Propagation. *IEEE trans. on Pattern Analysis and Machine Intelligence*, PAMI-2(6):522–536, November 1980.

[50] D. H. Parish, G. Sperling, and M. S. Landy. Intelligent Temporal Subsampling of American Sign Language Using Event Boundaries. *Journal of Experimental Psychology: Human Perception and Performance*, 16(2):282–294, 1990.

[51] T. S. Perry. Biomechanically Engineered Athletes. *IEEE Spectrum*, pages 43–44, 1990.

[52] E. D. Petajan, B. Bischoff, D. Bodoff, and N. M. Brooke. An Improved Automatic Lipreading System to Enhance Speech Recognition. In *SIGCHI '88: Human Factors in Computing Systems*, pages 19–25, October 1988.

[53] H. Poizner, U. Bellugi, and V. Lutes-Driscoll. Perception of American Sign Language in Dynamic Point-Light Displays. *Journal of Experimental Psychology: Human Perception and Performance*, 7(2):430–440, 1981.

[54] R. Polana and R. C. Nelson. Recognizing Activities. Submitted to CVPR 1994.

[55] R. Polana and R. C. Nelson. Recognition of Motion from Temporal Texture. In *Proc. Conf. on Computer Vision and Pattern Recognition*, pages 129–134, Champaign, IL, June 15–18 1992.

[56] R. Polana and R. C. Nelson. Detecting Activities. In *Proc. Conf. on Computer Vision and Pattern Recognition*, pages 2–7, New York, NY, June 15–17 1993.

[57] H. K. Ramakrishnan and M. P. Kadaba. On the Estimation of Joint Kinematics During Gait. *J. Biomechanics*, 24(10):969–977, 1991.

[58] K. Rangarajan, W. Allen, and M. A. Shah. Recognition Using Motion and Shape. In *Proc. 11'th Intern. Conf. on Pattern Recognition*, volume 1, The Hague, The Netherlands, Aug.30–Sept.3 1992.

[59] K. Rangarajan, W. Allen, and M. A. Shah. Matching Motion Trajectories Using Scale Space. *Pattern Recognition*, 26(4):595–609, April 1993.

[60] K. Rangarajan and M. Shah. Establishing Motion Correspondence. *CVGIP: Image Understanding*, 54(1):56–73, July 1991.

[61] R. F. Rashid. Towards a System for the Interpretation of Moving Light Displays. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-2(6):574–581, November 1980.

[62] K. Rohr. Incremental Recognition of Pedestrians from Image Sequences. In *Proc. Conf. on Computer Vision and Pattern Recognition*, pages 8–13, New York, NY, June 15–17 1993.

[63] K. Rohr. Towards Model-Based Recognition of Human Movements in Image Sequences. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 59(1):94–115, January 1994.

[64] H. Sakoe and S. Chiba. Dynamic Programming Algorithm Optimization for Spoken Word Recognition. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, ASSP-26(1):43–49, February 1978.

[65] I. K. Sethi and R. Jain. Finding Trajectories of Feature Points in a Monocular Image Sequence. *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-9(1):56–73, 1987.

[66] H. Shariat and K. Price. How to Use More than Two Frames to Estimate Motion. In *Proc. Workshop on Motion: Representation and Analysis*, pages 119–124, Charleston, SC, May 7–9 1986.

[67] G. Sperling, M. Landy, Y. Cohen, and M. Pavel. Intelligible Encoding of ASL Image Sequences at Extremely Low Information Rates. *Computer Vision, Graphics and Image Processing*, 31:335–391, 1985.

[68] M. Subbarao. Interpretation of Image Motion Fields: A Spatiotemporal Approach. In *Proc. Workshop on Motion: Representation and Analysis*, pages 157–165, Charleston, SC, May 7–9 1986.

[69] S. Sumi. Upside-Down Presentation of the Johansson Moving Light-Spot Pattern. *Perception*, 13:283–286, 1984.

[70] C.-K. Sung. *Extraktion von Typischen und Komplexen Vorgangen aus einer Bildfolge einer Verkehrsszene*, pages 90–96. Springer-Verlag, 1988.

[71] P.-S. Tsai, M. Shah, K. Keiter, and T. Kasparis. Cyclic Motion Detection. Technical Report CS-TR-93-08, University of Central Florida, Computer Science Dept., 1993.

[72] J. K. Tsotsos. Temporal Event Recognition: An Application to Left Ventricular Performance. In *Proc. 7'th Intern. Joint Conf. on Artificial Intelligence*, volume 2, pages 900–907, Vancouver, Canada, August 24–28 1981.

[73] J. K. Tsotsos, J. Mylopoulos, H. D. Covvey, and S. W. Zucker. A Framework for Visual Motion Understanding. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-2(6):563–573, November 1980.

[74] J. A. Webb and J. K. Aggarwal. Visually Interpreting the Motion of Objects in Space. *Computer*, 14(8):40–46, August 1981.

[75] J. Yamato, J. Ohya, and K. Ishii. Recognizing Human Action in Time-Sequential Images Using Hidden Mardov Model. In *Proc. Conf. on Computer Vision and Pattern Recognition*, pages 379–385, Champaign, IL, June 15–18 1992.

[76] X. Zhunag and R. M. Haralick. Two View Motion Analysis. In *Proc. Conf. on Computer Vision and Pattern Recognition*, pages 686–690, 1985.