8-1979

Image Analysis of Human Motion Using Constraint Propagation

Joseph O'Rourke
University of Pennsylvania

Norman I. Badler
University of Pennsylvania, badler@seas.upenn.edu

Recommended Citation

Joseph O'Rourke and Norman I. Badler, "Image Analysis of Human Motion Using Constraint Propagation", . August 1979.

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-80-02.

Image Analysis of Human Motion Using Constraint Propagation

Abstract

A system capable of analyzing image sequences of human motion is described. The system is structured as a ·feedback loop between high and low levels: predictions are made at the semantic level, and verifications are sought at the image level. The domain of human motion lends itself to a model-driven analysis, and the system includes a detailed model of the human body. All information extracted from the image is interpreted through a constraint network based on the structure of the human model. A constraint propagation operator is defined and its theoretical,properties developed. An implementation of this operator is described, and results of the analysis system for a short image sequence are presented.

Keywords

motion, time-varying images, human motion, constraint propagation, constraint networks, computer vision

Disciplines

Computer Engineering | Computer Sciences

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-80-02.

MS-CIS-80-2

IMAGE ANALYSIS OF HUMAN MOTION USING CONSTRAINT PROPAGATION

Joseph O'Rourke
Norman Badler

August 1979

Computer and Information Science Department
Moore School of Electrical Engineering/D2
University of Pennsylvania
Philadelphia, PA 19104

Abstract- A system capable of analyzing image sequences of human motion is described. The system is structured as a feedback loop between high and low levels: predictions are made at the semantic level, and verifications are sought at the image level. The domain of human motion lends itself to a model-driven analysis, and the system includes a detailed model of the human body. All information extracted from the image is interpre ted through a constraint network based on the structure of the human model. A constraint propagation operator is defined and its theoretical properties developed. An implementation of this operator is described, and results of the analysis system for a short image sequence are presented.

Index Terms- motion, time-varying images, human motion, constraint propagation, constraint networks, computer vision

TABLE OF CONTENTS PAGE

TABLE OF CONTENTS PAGE

1. INTRODUCTION

Given a sequence of images of a human in motion, a computer system should be capable of following the motion in three dimensions and "understanding" or describing the motion in some form, tasks which are routinely accomplished by humans. We are building a system to perform this analysis of human motion based primarily on *constraint propagation* and *high level prediction*. We will show that together these techniques allow tracking of human motion with a minimum of image analysis. The emphasis in this paper will be on the propagation of constraints, which is shown to be a useful method for interpreting low level knowledge in accordance with a detailed world model.

There is now a sizable body of literature on the analysis of time-varying images, and a number of survey articles (Martin and Aggarwal [33], Nagel [37], Scacchi [50]) have examined the work from the point of view of the *techniques* employed. In order to locate our research effort with the spectrum of previous work, we will classify the research according to the *domains of application* studied. Since the task of building a system to analyze digital image sequences is a difficult one, researchers have been forced to restrict the problem in various ways in order to make it tractable. There are four main dimensions which determine the complexity of a system which analyzes image sequences: (1) the complexity of the objects in the images; (2) the number of such objects; (3) the type of motions the objects execute; and (4) the depth of understanding achieved by the system. As might be expected, there is a trade-off between these complexities in the current research, in that those who tackle the more complex end of one dimension usually compromise on one or more of the other. At one extreme, the objects observed are simple point-like shapes: tachistoscope dots (Ullman [60]), biological cells (Levine [29], Futrelle [15]), and it is in just these cases where many objects can be handled simultaneously. When the objects become less point-like but still remain rigid inflexible bodies, then fewer objects are treated and the analyses become more complicated, usually involving occlusion. Such objects include automobiles (Nagel [36], Jain and Nagel [26], Fennema and Thompson [12]), industrial parts (Neuman [39]), rocks (Eskenazi and Cunningham [11]), polygons (Aggarwal and Duda [1]), and polyhedra (Roach and Aggarwal [46], Chien and Jones [7]). The most complex objects, such as hearts (Schudy [51], Herman and Liu [22], Tsotsos [55], Yachida et al [65]), require complex shape analysis, and are always considered in isolation.

The type of motions which have been studied include: 2D rigid motion without rotation (Levin [29], Futrelle [14], Potter 42], Chow and Aggarwal [8], Aggarwal and Duda [1]); 2D rigid motion with rotation (Martin and Aggarwal [33]); 3D rigid motion without rotation (Roach and Aggarwal [46]); 3D rigid motion with rotation (Nagel [36], Jain and Nagel [26], Fennema and Thompson [12], Wallace and Mitchell [62]); 3D articulated rigid motion with rotation (Badler [2], Rashid [43], Tsuji et al [58]); and various amorphous shape changes (Schudy [51], Herman and Liu [22], Tsotsos [55], Hernan and Jimenez [23]).

Concerning the fourth complexity dimension mentioned above, the depth of understanding achieved by the system is related to the ability of the system to answer questions about the image sequence and the scope of the allowed quesitons. Much of the recent research has been concerned mainly with segmentation and tracking, and so can only field a limited set of questions. However, a few research efforts have attempted deeper descriptions, usually employing a linguistic approach for representing motion concepts (Badler [2], Tsuji et al [57], Herman [20][21], Tsotsos [56]).

The domain we have chosen to examine is that of human motion. The human body is an extremely complex object, being highly articulated and capable of a bewildering variety of motions. Rotations and twists of the body parts occur in nearly every movement, and various parts of the body continually move into and out of occlusion. Therefore our domain is far along the first (object) and third (movement) complexity dimensions. In order to keep the complexity within manageable limits, we will simplify the domain of the second complexity dimension by only considering a single human in an environment devoid of other objects (except for the ground or floor). For the fourth dimension of complexity, we hope to eventually achieve rather deep semantic understanding of human motion (see Badler and Smoliar [5] for an indication of our goals), but the results presented in this paper only show a rather modest understanding.

Most of the work to date in computer analysis of time-varying images has attempted to reach high-level understanding by building on the results of low-level processing. This _bottom-up_ approach is especially suited to the analysis of real-world scenes, where some primitive change detection and region segmentation is usually necessary for any further analysis (for example, Fennema and Thompson [12], Nagel [36], Jain and Nagel [26], Potter [44]). One of the reasons we have chosen to study the domain of human motion is to investigate a _top-down_ approach to analysis.

The human body has a well-defined structure which can be encoded into a model. In our system, we use a model of the human body as a type of detailed frame (Minsky [34]) or schema (Neisser [38], Hayes [19]). All of the information we gather from the image will be interpreted in terms of the model of the body, and the model will be used to predict or anticipate future positions of the body. Low level image processing is relegated to a rather minor role in our system, not because it is unimportant, but because we wish to concentrate on the high-level aspects of motion perception.

In the next section we present an overview of our system, and in Section 3 the human model is briefly discussed. A theoretical basis for constraint networks and propagation is developed in Section 4, and our implementation described. Section 5 presents the results of the motion analysis system on a test image sequence, and future work is outlined in Section 6.

2. SYSTEM OVERVIEW

We will use the term high level to describe the semantic level involving the 3D scene domain and object models, and the term low level to mean the signal level involving the 2D gray-scale picture domain (see Kanade [27]). In terms of this distinction, our system can be described as consisting of four main components or processes: prediction, simulation, image analysis, and parsing. As shown in Figure 1, the prediction component operates at the high level, and the image analysis takes place at the low level. The simulation serves to convert from high to low, and the parsing component interprets low level data as higher level concepts.

Note that the model is depicted as sitting in the middle and influencing all the other components. Each of these four components will now be described in some detail, and the role of the model in each outlined.

A. Image Analysis

The image analysis component is the only process which actually looks at the image. The input to this component is a list of picture areas where various body features are predicted to appear. (The generation of this input will be described later.) Using these predicted regions as a guide, the image analysis module searches the image for certain body features, employing various feature detectors. Notice that the image has not been preprocessed to segment it into regions, or detect edges, or any other such low level processing. This type of processing is done only when needed, and then only within the area predicted for a particular feature. In effect, we are processing the image via "successive glimpses" (Hochberg [24]) similar to human saccades. With each glance, some feature detector is applied within the predicted image region for that feature, and if it is successful, the region within which the feature is understood to lie becomes smaller (than the predicted region). This new knowledge is immediately fed to the constraint propagation mechanism, which infers the spatial consequences of the knowledge. Generally, the result will be a further reduction of the areas where other body features may appear, which reduces the search space for these features.

It could happen, however, that the constraint propagation reveals that the knowledge just passed to it is inconsistent with the previously predicted and/or determined locations of the entire network of features. The usual method of handling such inconsistencies is to initiate some type of backtracking, eventually resulting in some alternate choices made or hypotheses postulated.* Our approach, however, is to simply terminate the analysis of the current image when an inconsistency is detected, and attempt to

*Stahlman and Sussman [54] have developed this idea into an intelligent interactive tool for designing electronic circuits.

recover gracefully by passing as much useful information as possible to the next component, together with an indication that an error occurred. The justification for this approach is: (1) it is difficult to determine exactly which observation or assumption is the cause of the inconsistency, and therefore, (2) backtracking can be very time-consuming, and finally, (3) the current image may be difficult to interpret, but future images may resolve the ambiguities and uncertainties. In effect, we allow for a moment of confusion, and hope that succeeding images will resolve matters.

When the features have been localized to a small enough area (more on this later), or when no further progress can be made in analyzing the image, or when the process is aborted because of inconsistencies, the image analysis component passes its results onto the next stage. The output is in the same form as the input: a list of 3D regions where the various body features have been found to reside. If the analysis was at all successful, the output regions are substantially smaller than the predicted input regions. Informally, the amount of shrinkage represents the system's increase in knowledge from analyzing one image frame.

B. Parsing

Over a number of cycles, the outputs of the image analysis phase constitute a stream of regions in space for each feature of interest. Each region represents the location in which the feature has been found to lie at a particular time. The "parser" fits these location-time streams with piecewise linear functions of time (see O'Rourke [41] for details). The model is used to choose the appropriate variables (linear or angular position, relative or absolute coordinates) to describe the motion of each feature. Each linear piece is considered a movement primitive, in the sense that it describes a uniform continuous motion. Thus the parser converts the 3D spatial regions into primitive movement commands describing "chunks" of motion for the body parts.

Although currently not implemented, we also intend to group together sequences of movement primitives which represent a repetitive pattern, similar to the approach of Badler [2]. Such pattern recognition operating on movement primitives would reach a higher semantical level, and so would be more useful for prediction purposes.

C. Prediction

The prediction component operates entirely at the high level. It receives sequences of primitive movement commands which describe the observed motion, and it projects these commands into the future to predict the position of the body in the next frame. The usual method of extrapolating the commands is to simply continue them without change: if a rotation is being observed, then it is predicted to continue. However, if some repetitive pattern has been recognized by the parser, such as walking and swinging of arms, the continuation of the repetition would be predicted, rather than just a continuation of the current movement.

The ouput of the prediction stage is a set of movement commands which will move the human body model into the predicted position. Note that the position itself is not the basis of the prediction, but the movement primitives. Our assumption is that predictions made at the semantic level will be more accurate and useful than those made at lower levels, and that the movement primitives have more semantic content than the raw positional data. (For a rather different approach to model-based prediction, see Futrelle and Speckert [16].)

D. Simulation

In order to translate the predicted movements into data that can be used by the image analysis component, the movements are simulated by a human movement simulator. This simulator will execute each movement by actually moving the indicated body part as specified by the movement commands. The simulator embodies extensive knowledge of the human body and how it may move. For example, it will not move any limb beyond the limitations of its associated joint, nor will it move one body part through another. The simulator understands about gravity, and will attempt to keep the body model in balance (see Badler et al [3] for further details).

The simulator can also interpret inconsistent commands, in the sense that it will attempt to reach some compromise among the possibly conflicting commands it receives from the prediction stage. This means that the prediction module does not have to take into account the myriad restrictions and details of human movement: the simulator will act as a filter on the commands.

The output of the simulator is a particular positional configuration of the human body. The location of each feature of the body is precisely determined by this position. However, prediction errors arise from two causes: (1) the amount of time the prediction is extrapolating into the future, coupled with the acceleration abilities of the body, and (2) the uncertainty of the previous analysis. To account for these errors, each feature is predicted to lie within some spatial region surrounding its exact location in the positioned model. These regions are then fed to the image analysis component, which uses them to guide the search for the features in the next frame.

3. HUMAN MODEL AND SIMULATOR

A. Human Model

In this section we describe the structure of the human model and the basic capabilities of the human motion simulator. The description of any aspects of the model which bear on constraint propagation will be defered until the next section.

The human model contains all of the system's "world knowledge" about the human body (Badler et al [4]). It is composed of segments and joints linked together into a tree-structured skeleton. A joint is a unique point connecting two segments (sliding joints are not permitted). A segment is an abstract rigid body with an associated embedded coordinate system. Each segment may have a number of joints located at fixed points within its coordinate system. Each segment moves rigidly; the only articulation permitted is at the joints. Our current model consists of 24 segments and 25 joints. The "flesh" or surface of each segment is defined by a collection of graphical primitives located at fixed positions within the segment's coordinate system. Currently we are using spheres as our primitive, resulting in the model shown in Figure 2 (see O'Rourke [40]).

The human model incorportates two fundamental restrictions on the motions it may execute: angle limits and collision detection. Each pair of segments connected by a jont are only permitted to have certain orientations with respect to one another, expressed as limits on angles at the joint. Also, the model includes a method of detecting collisions between non-adjacent segments, which can be used to prevent one segment from passing through another (see Badler et al [4]).

The human model is located within a global coordinate system, which also includes a camera. All of the parameters of the camera model are assumed to be known; only the position and orientation of the human body are unknown (in contrast to the approach of Roach and Aggarwal [46]).

The camera model together with the human body model allows us to take pictures of the model. Together with the simulator described below, this gives us the capability of producing simulated motion sequence films, which we have used as input to our analysis system.

B. Simulator

The simulator moves the human body model in response to certain movement commands based on human movement notations (Badler and Smoliar [5]). These are eventually executed by five basic movement primitives: MOVE, ROTATE, BEND, TWIST, and TOUCH. The simulator accepts a stream of movement commands, and "executes" them by positioning the body in accordance with the commands. Conceptually, the monitor of the simulator sends each command to appropriate joint processors, which then process the commands in parallel. In practice, the commands have to be scheduled according to their scope and the hierarchy of the body, and executed serially. The details of this process are described more fully in Badler et al [3].

Only the most rudimentary capabilities of the simulator are currently used in our analysis system. The most important aspect of the simulator for the purposes of this paper is that it will always position the body in a legal achievable position, and this position is, in some sense, the one which most nearly or naturally achieves the goals of the movement commands driving the simulator.

4. CONSTRAINT PROPAGATION

A. Background

When the image analysis component of our system begins to examine the low level information of the actual image, high level knowledge has already been applied to produce a predicted position of the body. However, if there were no further interaction between the high and low levels, a great deal of power and flexibility would be lost. Every time a body feature is located in the image, the location of other features are constrained by the structure of the human body. We would like to exploit these constraints to aid the image analysis component in finding other features.

Our main tool for employing knowledge of the human body's structure in low level analysis is a method of propagating constraints through a network. The features of the body are connected into a network describing the relationships or constraints between the features. Each time a feature is determined to lie within an area of the image, this constraint is propagated throughout the network, reducing the regions where other body features may appear. The propagation is effected by a reduction operator whose properties are developed in Sections 4.2 and 4.3. Before describing our own work, however, we will first establish a setting for the discussion by a brief review of related literature.

Although there has been some direct work on locating objects in images via constraints (Ballard et al [6], Russel [49]), most of the research on constraint propagation has arisen out of the problem of determining a consistent labeling for a set of units constraining one another in some manner. This problem was recognized and dicussed with various degrees of explicitness in the early papers of Ullman [61], Guzman [17], Fikes [13], Clowes [10], and Huffman [25]. Waltz [63] developed the first system to employ these ideas extensively, and was able to "understand" a blocks world scene by labeling the edges and vertices of the blocks. Waltz developed an elegant "filtering" algorithm to remove the inconsistent labels. Waltz's algorithm removes only what Mackworth [31] calls "arc inconsistencies," that is, inconsistencies between two directly constrained nodes. Montanari [35] studied binary constraint relations in a general algebraic setting, developing the idea of path consistency. Rosenfeld et al [47] further developed the notion of arc and path consistency, and extended these ideas to parallel computation and fuzzy or probabilistic relations, and to relaxation techniques (Zucker [66]). More recently, Freuder [14] has shown how to synthesize the higher-order constraints (beyond path consistency), and Haralick and Shapiro [18] have placed the entire consistent labeling problem into a general setting using look-ahead operators.

The one common assumption of all the above mentioned work is that the set of labels is finite. Thus all of the algorithms developed for finding consistent labelings are methods of limiting the combinatorial search through the space of possible labelings. Removing all of the inconsistent labels has in fact shown to be an NP-complete problem (Montanari [35], Freuder [14]), but the success of Waltz and others (e.g., Shneier [53], Shapira and Freeman [52]) shows that many cases are quite tractable. In our application, the analog to the "unit" or "node" of the labeling problem is a point in 3D space representing the location of a feature, and the analog of a set of "labels" is a region in space (a subset of $\mathbf{R}^3$) where the feature may lie. Thus our set of "labels" is infinite and continuous rather than finite and discrete. Of course one could discretize and bound space to force the allowable positions (and therefore labels) to be finite, but the very large size of the resulting finite sets rules out any direct application of the algorithms developed for finite sets of labels. We will see, however, that similar algorithms can be developed for the continuous case. In fact, in the next section we will develop an operator which achieves the analog of arc consistency for continuous spaces.

B. Theory

As mentioned in the previous section, the "units" of our problem are features of the human body, and the "labels" are regions of space. We will now establish the notation used throughout the remainder of the paper to discuss the continuous constraint propagation problem.

The set of feature indices will be called $J = \{1,2,...,n\}$, and subsets of this index set will be denoted by $I \subseteq J$, with individual subscripts lower case. The cardinality of a set I will be written as $|I|$. The position in 3-space of feature i is $\vec{p}_i$, $\vec{p}_i \in \mathbf{R}^3$, where $\mathbf{R}^3$ indicates 3D Euclidean space. We will follow Freuder [14] in allowing subscripting by index sets, but we want the resulting object to be an ordered k-tuple rather than a set. Thus, if $I = \{i_1,i_2,...,i_k\} \subseteq J$, then $\vec{p}_I = \langle\vec{p}_{i_1},\vec{p}_{i_2},...,\vec{p}_{i_k}\rangle$, where $i_1 < i_2 < ... < i_k$. A set of points for feature i will be written as $P_i = \{\vec{p}_i\}$, and in analogy with single vectors, we will write P_I for an ordered k-tuple of sets $P_I = \langle P_{i_1},P_{i_2},...,P_{i_k}\rangle$.

Our goal is to define and develop the properties of a function which will take an initial set of regions (subsets of $\mathbf{R}^3$) for the features P_J and compute a subset of these regions which satifies a collection of constraint relations. The constraint relations are relations between points or vectors, and will be denoted by r subscripted with the feature indices whose vectors are related by the constraint. We will consider each constraint relation a mapping from the appropriate space into $\mathbf{2} = \{T,F\}$. Thus,

$$r_i \subseteq \mathbf{2}^{\mathbf{R}^3}$$

selects out a subset of $\mathbf{R}^3$ for the point $\vec{p}_i$.

Similarly,

$$r_{ij} \subseteq 2^{\mathbf{R}^3} \times \mathbf{R}^3,$$

and in general,

$$r_I \subseteq 2^{\times}_{I}\,\mathbf{R}^3.$$

A unary constraint simply specifies a subset of $\mathbf{R}^3$ within which a feature may lie. A typical binary constraint is one specifying a range for the distance between two features:

$$r_{ij}(\langle\vec{p}_i,\vec{p}_j\rangle) = T \text{ iff } dmin_{ij} \leqslant |\vec{p}_i-\vec{p}_j| \leqslant dmax_{ij}. \quad (1)$$

A tertiary constraint might express an angular limitation:

$$r_{ijk}(\langle\vec{p}_i,\vec{p}_j,\vec{p}_k\rangle) = T \text{ iff the angle between } \vec{p}_j-\vec{p}_i \text{ and } \vec{p}_k-\vec{p}_i \text{ is } \leqslant \theta_{ijk}.$$

For each constraint r_I, with $|I| = k$, we can define k functions, each of which produces the set of all possible positions of one feature, given the position of all of the other joints. Let $I-i = \{j \mid j \in I \text{ and } i \neq j\}$. For each $i \in I$ define

$$f^i{}_I(\vec{p}_{I-i}) = \{\vec{p}_i \mid r_I(\vec{p}_I) = T\}.$$

For example, the binary constraint r_{ij} (here $I = \{i,j\}$) gives rise to two functions:

$$f^{i}_{ij}(\langle\vec{p}_j\rangle) = \{\vec{p}_i \mid r_{ij}(\langle\vec{p}_i,\vec{p}_j\rangle) = T\}$$

$$f^{j}_{ij}(\langle\vec{p}_i\rangle) = \{\vec{p}_j \mid r_{ij}(\langle\vec{p}_i,\vec{p}_j\rangle) = T\}.$$

Similarly, a unary constraint produces one function of no arguments, and a tertiary constraint has three associated functions, each taking 2-tuples for arguments. Although these constraint functions operate on <u>points</u>, most of our calculations will be based on <u>sets of points</u>. We will therefore generalize the functions to take tuples of sets of points for arguments, as follows:

$$F^{i}_{I}(P_{I-i}) = \bigcup_{\vec{p}_{I-i} \in P_{I-i}} f^{i}_{I}(\vec{p}_{I-i})$$

$$= \{\vec{p}_i \mid r_I(\vec{p}_I) = T \text{ for some } \vec{p}_I \text{ with } \vec{p}_{I-i} \in P_{I-i}\}. \qquad (2)$$

Here the notation $\vec{p}_K \in P_K$ should be read as a shorthand for $\vec{p}_K \in \mathop{X}_{i \in K} P_i$. For a binary relation r_{ij}, the function F^{i}_{ij} is

$$F^{i}_{ij}(\langle P_j\rangle) = \{\vec{p}_i \mid r_{ij}(\langle\vec{p}_i,\vec{p}_j\rangle) = T \text{ for some } \vec{p}_j \in P_j\}.$$

and thus produces, according to constraint r_{ij}, the regions of space in which feature i may lie, given that the feature j is inside P_j. This generalization of the point constraints to sets of points weakens their discriminatory power, but it is a necessary step for the development of constraint propagation on infinite sets.

In general, any one feature i may participate in a number of constraint relations. We will let $C_i(P_J)$ represent the intersection of the constraint regions generated by all constraint functions for feature i:

$$C_i(P_J) = \bigcap_{I \subseteq J} F_I^i(P_{I-i}) \tag{3}$$

It is understood here that not every subset I has a corresponding constraint function F_I, simply because there may not be any constraints relating the features indexed by I. One could consider all such functions to return the entire space $\mathbf{R}^3$.

One simple property of the constraint functions which will be useful later on is their monotonicity. In the case where F is the constraint function for a binary distance constraint of the form illustrated in equation (1), this property simply means that if some point in space can be reached by a link when one end of the link is confined to a region of space, then this point can also be reached if the end of the link is confined to a superset of the region. This is stated formally in the following lemma.

Lemma 1. Each constraint function F_I^i (and therefore each C_i) is montotonic, i.e., if
$P_{I-i} \subseteq P'_{I-i}$, then
$F_I^i(P_{I-i}) \subseteq F_I^i(P'_{I-i})$.

Proof: Let $\vec{p}_i \in F_I^i(P_{I-i})$. Then by the definition of F_I^i, there is some $\vec{p}_I$ with $r_I(\vec{p}_I) = T$ and $\vec{p}_{I-i} \in P_{I-i}$. But since

$P_{I-i} \subseteq P'_{I-i}$, we also have $\vec{p}_{I-i} \in P'_{I-i}$. By

definition of F again, this gives

$\vec{p}_i \in F^i_I(P'_{I-i})$, which establishes the

Lemma. □

We may now define our reduction operator R, which will take an n-tuple of feature regions P_J as an argument, and produce a subset of P_J which is more consistent with the given constraint relations. More precisely, R will intersect each feature's input region with the constraint regions generated by all features related to it, as follows:

$$R(P_J) = P_J \cap C_J(P_J) \qquad (4)$$

This reduction function deletes feature regions which are inconsistent with related neighbors, and so will achieve (after repetition) the equivalent of Mackworth's arc consistency.

We will now establish some simple properties of the reduction function defined by equation (4). The first and most obvious justifies the name "reduction".

Theorem 1. $R(P_J) \subseteq P_J$.

Proof: This follows immmediately from equation (4): $R(P_J)$ is defined as P_J intersected with some set, and therefore the result must be a subset of P_J. □

A second simple but useful property of R is monotonicity.

Theorem 2. R is monotonic: if $P_J' \subseteq P_J$, then

$R(P_J') \subseteq R(P_J)$.

Proof: This follows easily from the definition of R (equation (4)) and Lemma 1, which establishes the monotonicity of the constraint functions. □

Let us call an n-tuple of feature positions $\vec{p}_J = \langle \vec{p}_1, \vec{p}_2, \ldots, \vec{p}_n \rangle$ a configuration. Then define a consistent configuration as a configuration which satisfies all of the constraint relatons. This notion corresponds to the idea of a "consistent labeling" as defined, for example, in Rosenfeld et al [47] or Haralick and Shapiro [18]. It is important that the reduction function not delete any consistent configurations. This is guaranteed by the next theorem.

Theorem 3. If $\vec{p}_J$ is a consistent configuation, and $\vec{p}_J \in P_J$, then $\vec{p}_J \in R(P_J)$.

Proof: Let r_I be one particular constraint relation, with $I \subseteq J$. Since $\vec{p}_J$ is a consistent configuration, $r_I(\vec{p}_I) = T$. Therefore, $\forall i \in I$, $f_I^i(\vec{p}_{I-i}) \supseteq \vec{p}_i$. Because $\vec{p}_J \in P_J$, and from the definition of F_I^i (equation (2)), this implies that $\vec{p}_i \in F_I^i(P_I) \quad \forall i$. Since this is true independent of the particular constraint relation, $\vec{p}_i \in C_i(P_J) \quad \forall i$, which, from equation (4), gives $\vec{p}_J \in R(P_J)$. □

This theorem implies that we can always be assured of including all consistent configurations if we start with the entire space. This is stated in the following corollary.

Corollary. $R^m(<\mathbf{R}^3,\mathbf{R}^3,\ldots,\mathbf{R}^3>)$ includes all consistent configurations for any $m > 1$.

Proof: Immediate from Theorem 2. □

We now look at the effect of applying the reduction function repeatedly.

Theorem 4. $\lim_{m\to\infty} R^m(P_J)$ exists for any P_J.

Proof: This follows immediately from Theorem 1 and the fact that $\emptyset_J = <\emptyset,\emptyset,\ldots,\emptyset>$ is a lower bound for P_J:
$\emptyset_J \subseteq R(P_J) \subseteq P_J$. □

In the case of discrete finite sets of labels, it is possible to prove that the limit in Theorem 4 can be reached after a finite number of applications of R (see for example Theorem 5 in Rosenfeld et al [47]). With infinite sets, this is not necessarily true, and it is important to characterize those constraint problems for which it is in fact true. Given a set of constraint relations, let us call the associated network of the relations the undirected graph with $|J|$ nodes labeled by the feature index set J and an arc connecting i and j iff there is some constraint relation involving both features i and j. Whether or not the limit in Theorem 4 is reached after only a finite number of applications of R depends on whether or not the associated network is a tree.

Under the above definition of associated network, it is clear that any relation involving three or more variables will cause a cycle. Therefore, the network is a tree only when there are just unary and binary constraint relations. Unary constraints can be satisfied by reducing the corresponding regions once, and from then on these constraints have no further effect, so we will restrict our attention to the case where there are only binary constraints (similar to Montanari [35]).

We first need to establish that after one application of R, a leaf node no longer affects its (single) neighbor.

Lemma 2. Let feature 1 be a leaf node of the associated network, so that there is only one constraint relation involving 1. Call this relation r_{1k}, where k is 1's neighbor in the network. Let
$R^m(P_J) = P_J{}^{(m)}$. Then
$F^k_1(P_1{}^{(m)}) \supseteqq P_k{}^{(m)} \quad \forall\, m > 1$. Thus
node 1 can not cause any reduction in node k's region after the first application of R.

Proof: Since feature 1 is only related to feature k, the definition of R (equation (4)) gives

$$P_1{}^{(m)} = P_1{}^{(m-1)} \cap F^1_k(P_k{}^{(m-1)}). \tag{5}$$

Feature k, on the other hand, can be influenced by a number of other features inside the network, and so we will write

$$P_k{}^{(m)} = P_k{}^{(m-1)} \cap F^k_1(P_1{}^{(m-1)}) \cap G \tag{6}$$

where G represents the constraint regions generated from all of the other relations in which k is involved. We are trying to prove that if $\vec{p}_k \in P_k^{(m)}$, then $\vec{p}_k \in F_1^k(P_1^{(m)})$. By (6), $\vec{p}_k \in P_k^{(m)}$ implies

$$\vec{p}_k \quad \in \quad P_k^{(m-1)} \tag{7}$$

and

$$\vec{p}_k \quad \in \quad F_1^k(P_1^{(m-1)}). \tag{8}$$

By definition (equation (2)), equation (8) means that

$$\exists \; \vec{p}_1 \quad \in \quad P_1^{(m-1)} \tag{9}$$

such that

$$r_{1k}(<\vec{p}_1,\vec{p}_k>) = T \tag{10}$$

Equations (7) and (10) together imply

$$\vec{p}_1 \quad \in \quad F_k^1(P_k^{(m-1)}) \tag{11}$$

and equations (9) and (11), together with (5), show that

$$\vec{p}_1 \quad \in \quad P_1^{(m)}. \tag{12}$$

Finally, equations (12) and (10) imply that

$$\vec{p}_k \in F_1^k(P_1^{(m)})$$

which completes the proof. □

With this Lemma, we can easily establish the following theorem.

<u>Theorem 5</u>. If the associated network is a tree of

diameter* d, then R^d is stable, that is,

$R^m(P_J) = R^d(P_J) \quad \forall m > d$.

Proof: The proof will be by induction on d. Suppose d = 0. Then the network consists of a single node and is trivially stable after 0 applications of R, since R is the identity if there are no constraint relations.

Suppose then that the theorem is true for all trees of diameter d, and consider a network of diameter d+1. Apply R once to this network and then remove all leaf nodes (there are some leaf nodes since N is a tree), calling the new network N'. By the Lemma, this removal will not affect the subsequent development of the network N'. Network N' has a diameter of d-1, and so by the induction hypothesis, it will stabilize after d-1 further applications of R. We have now applied R a total of 1+(d-1) = d times, and we are certain that all of the internal nodes of N are stable. It only remains to show that the leaf nodes are also stable.

Let l be a leaf node, and k its only neighbor. We want to prove that the (d+2)nd application of R will not affect node l, i.e., that

$$F_k^l(P_k{}^{(d+1)}) \supseteq P_l{}^{(d+1)}.$$

Now, by the defintion of R (equation (4)),

$$P_l{}^{(d+1)} =$$
$$P_l{}^{(d)} \cap F_k^l(P_k{}^{(d)}).$$

---

*The diameter of a tree is the number of edges in the longest path contained in the tree.

Thus, $\vec{p}_1 \in P_1^{(d+1)}$ implies

$\vec{p}_1 \in F_k^1(P_k^{(d)})$, but since node k

stabilized after d applications,

$P_k(d) = P_k^{(d+1)}$ and

$\vec{p}_1 \in F_k^1(P_k^{(d+1)})$. □

If there is a single loop within the associated network of a constraint problem, then it is possible that there is no finite m for which R^m is stable. To prove this it is sufficient to show an example. The network shown in Figure 3 will never stabilize: each application of R will clip off one piece of one of the four regions, spiraling inwards in a manner reminiscent of a golden section construction.

Let us define the solution tuple S_J for a constraint network to include all of the consistent configurations:

$S_J = \langle S_1, S_2, \ldots, S_n \rangle$ with

$S_i = \{\vec{p}_i \mid$ there is some consistent configuration with an ith component $\vec{p}_i\}$.

If our initial set of regions include S_J, then repeated applications of the reduction function R will always produce a superset of S_J, but we have no guarantee that the supersets will be at all close to S_J. It would be useful if there were a method of approaching S_J arbitrarily closely.

If the reduction function is applied to a single configuration, rather than a set of points, then it acts as a consistency or solution verifier, in the sense that a consistent configuration will remain unchanged while an inconsistent configuration will have at least one of its components reduced to $\emptyset$. If our spaces were discrete and finite, then applying R to every possible n-tuple of points would precisely delimit the solution tuple S. In the case of continuous spaces, we can improve (or at least not worsen) our superset of S by fracturing the regions into pieces. If this fracturing process were carried to the limit, it would be equivalent to testing each n-tuple of points individually. This idea is employed by all the consistent labeling algorithms for finite sets; see, for example, Rosenfeld et al [47] or Haralick and Shapiro [18].

We first define the notion of <u>combinatorial partition</u> recursively:

(1) The set $\{P_J\}$ is a combinatorial partition of P_J.

(2) If $Q = \{Q_1, Q_2, \ldots\}$ is a combinatorial partition of P_J, then a new combinatorial partition can be constructed from Q as follows: identify all tuples which have a common ith component P_i; call this set of tuples Q'. Let $P_i = P'_i \cup P''_i$, with $P'_i \cap P''_i = \emptyset$. Let Q'/P'_i denote the set of all tuples in Q' but with P'_i replacing each ith component, and similarly for Q'/P''_i. Then the following set is also a combinatorial partition of P_J: $Q - Q' \cup Q'/P'_i \cup Q'/P''_i$.

Thus each fracturing of a region for a feature into two pieces requires adding all possible combinations of each piece with all the other regions.

Theorem 6. If the reduction function is applied to each member of a combinatorial partition of P_J and the results unioned, this union will be a subset of $R(P_J)$. Moreover, any consistent configuations in P_J will remain in this union. More precisely, if $Q = \{Q_1, Q_2, \ldots, Q_k\}$ is a combinatorial partition of P_J, and if the solution tuple is included in P_J, $S_J \subseteq P_J$, then

$$S_J \subseteq R(Q_1) \cup R(Q_2) \cup \ldots \cup R(Q_k) \subseteq R(P_J).$$

Proof: By Theorem 2, the function R is monotonic, and since each Q_i is a subset of P_J by the definition of combinatorial partition, it follows that

$$\bigcup_{Q_i \in Q} R(Q_i) \subseteq R(P_J).$$

The remainder of the theorem follows from Theorem 3, which states that R never deletes a consistent configuration which is already present, and the observation that any particular configuration must be a member of one of the tuples in the combinatorial partition. □

C. Implementation

The implementation of a constraint propagation network turns on the choice of a primitive for representing regions of space. A review of the main definitional equations of the last section (equations (2), (3), and (4)) shows that only two operations are performed on spatial regions: generation of the constraint regions via the functions F_I^i, and intersection of two regions. We have chosen to use <u>orthogonal</u> rectangular boxes as a primitive volume element: all the faces of the boxes are parallel to either the x-y, y-z, or x-z planes of a fixed Cartesian coordinate system. This primitive element is crude in many ways, but it has a number of distinct advantages:

(1) A single box B_i can be represented succinctly: 6 numbers are sufficient, 3 for the coordinates of each of two opposite corners:

$B_i ::= \langle(xmin_i,ymin_i,zmin_i),(xmax_i,ymax_i,zmax_i)\rangle$.

(2) The intersection of two boxes is again a box. This is a crucially important property, and is not shared by any other simple volume primitive.

(3) The intersection of two boxes can be easily computed: it requires only taking 6 maximums or minimums. If $B_3 = B_1 \cap B_2$, then in the notation above,

$xmin_3 = \max(xmin_1,xmin_2)$ and

$xmax_3 = \min(xmax_1,xmax_2)$, and similarly for y and z.

(4) Any closed subset of $\mathbf{R}^3$ can be represented as a union of rectangular boxes.

In order to compute the effect of the reduction function

R (equation (4)), we must be able to compute the constraint regions C_i for each joint i (equation (3)), which in turn depends on the constraint functions F_I^i associated with each constraint relation r_I. At this point we will specialize our analysis to binary constraint relations r_{ij}. Binary constraint relations (and not higher order relations) have the useful property that their associated constraint functions are homeomorphic with respect to the union operation, as stated in the following lemma.

Lemma 3. If $P_j = P_j' \cup P_j''$, then $F_j^i(P_j) = F_j^i(P_j') \cup F_j^i(P_j'')$.

Proof: We will first show that an element of the left hand side of the above equation must also be an element of the right hand side. Let Let $\vec{p}_i \in F_j^i(P_j)$. Then by definition of F_j^i, there exists a $\vec{p}_j \in P_j$ such that $r_{ij}(<\vec{p}_i,\vec{p}_j>) = T$. Since $P_j = P_j' \cup P_j''$, this $\vec{P}_j$ must be an element of P_j' or P_j''; let us say that $\vec{p}_j \in P_j'$. Then, again by definition of F_j^i, $\vec{p}_i \in F_j^i(P_j')$, and so is an element of the right hand side of the equation.

The other direction of the proof is similar. Let $\vec{p}_i$ be a member of the right hand side, say $\vec{p}_i \in F_j^i(P_j')$. Then there exists a $\vec{p}_j \in P_j'$ such that $r_{ij}(<\vec{p}_i,\vec{p}_j>) = T$. But since $P_j' \subseteq P_j$, $\vec{p}_j \in P_j$, and therefore

$\vec{p}_i \in F^i_j(P_j)$. □

This lemma permits us to concentrate on defining the constraint functions for a single box; the value of F operating on a region described as a union of boxes can be computed by unioning the results of F on each individual box.

We will specialize the discussion again, this time to a particular binary relation describing the distance between two points:

$$r_{ij}(\langle\vec{p}_i,\vec{p}_j\rangle) = T \text{ iff } dmin_{ij} < |\vec{p}_i-\vec{p}_j| < dmax_{ij}.$$

If B_j is a box, then the constraint function associated with r_{ij} is

$$F^i_{ij}(B_j) = \{\vec{p}_i \mid \exists \vec{P}_j \text{ such that } dmin_{ij} < |\vec{p}_i-\vec{p}_j| < dmax_{ij}\},$$

and represents the region of space which is reachable by rods with an end fixed inside B_j, where the lengths of the rods are between $dmin_{ij}$ and $dmax_{ij}$. Unfortunately, this constraint region is not rectangular, but rather has some spherical surface sections. We can, however, make a conservative rectangular approximation, as is illustrated in Figure 4. The details of the computation of this approximation can be found in O'Rourke [43].

Once we have a method of generating the constraint regions via the F functions, and an algorithm for intersecting boxes, the reduction function R can be simply implemented as follows:

(1) For each joint i, compute $C_i(P_J)$ by intersecting together $F^i_{ij}(P_j)$ for all constraint relations involving joint i.

(2) For each i, intersect P_i with $C_i(P_J)$; the result is the new value of P_i.

Normally the reduction function is applied repeatedly until the regions stabilize, that is, until a fixed point is reached. Theorem 5 guarantees that if the constraint network is a tree, then the number of applications can be easily computed. If there are cycles in the network, however, then some criteria must be applied to stop the iteration loop. We use a simple tolerance check, coupled with a maximum on the number of permitted repetitions. We have yet to encounter a case which required more than 15 repetitions to stabilize within tolerance, and so slow convergence does not appear to be a problem.

Figure 5 shows five "snapshots" of a portion of the constraint network of the body during constraint propagation. The first figure shows a stable network, and the succeeding figures follow the propagation caused by a shrinkage in the left wrist region as a result of image analysis. Eventually, the joints at the left elbow, shoulder, and clavicle, the center shoulder and neck, and the right clavicle and shoulder, are all affected by this change. After five applications of the reduction function R, the network is again stable.

6. RESULTS

In this section we present results of the complete analysis system operating on a short motion sequence.

A. Description of Test

Each image of the test sequence is 100 by 100 pixels, with 256 gray levels of resolution. The frame rate is nominally 5 frames/second. The images were produced with the human model and the human motion simulator. The segments representing the hands, feet, and head are colored a lighter shade of gray than the remainder of the body, giving the images something of the character of moving light displays (Rashid [45]). This is to enable a very simple type of "feature detection" based on the gray value of regions. It is recognized that this is not a very realistic feature detector (although the hands and face often stand out because they are flesh-colored), but it will serve to illustrate the functioning of the system. Each joint of the body is considered a "feature," even though many of them (such as the waist) have no outstanding visual characteristics. Only the hand, foot, and head joints are explicitly searched for in the image.

There is one computational strategy used in the image analysis component which has not been previously described. The silhouette of the figure in the image is used as a "cookie cutter" on the predicted feature regions as the first step of image analysis (see Weiler and Atherton [64]). A rectangular cover is computed for the figure, and this is extended in depth to produce a collection of boxes within which all body features must lie. This cover is then intersected with the predicted box for each feature, clipping them to project within the silhouette.

B. Results of Test

Figure 6 presents the input and output of the system for 10 frames, every other frame for the first 20 frames (4 seconds) of a test sequence. The images were produced by rotating the left arm, left leg, and right arm at various rates, and bending the torso towards the right and the head towards the left. Adjacent to each input image in the Figure is shown the output of the image analysis phase for that frame. Although the camera is viewing the human figure head on, the boxes are shown at an angle to illustrate their three dimensionality. Also, the centroids of each joint's collection of boxes are connected by dotted lines to show the network structure.

Initially (time=0.0) the arms and legs are all vertical, and at time=0.2, it can be seen that movements of the wrists and left ankle have been detected. No movement has been detected in the knees or elbows, but when the simulator is commanded to move the joints to the detected positions, it finds it necessary to move the elbows and left knee in order to reach the position. Thus these joints are predicted to move, and are properly tracked in later frames.

For each frame, after the outline of the figure has been used as a "cookie cutter" on the predicted regions, and the constraint propagation has stabilized the network, the system decides whether the regions for certain features are already tight enough, or whether further analysis is needed. If the latter, then the feature detector is called and examines the image in the area covered by the feature's boxes, and any improvements in the feature's location are propagated via the constraint network. The example described in Section 4 and shown in Figure 5 is taken from the left hand analysis at time=1.6.

The bend of the torso evident in the input images is a bit too subtle for the program to detect initially. Instead it tilts the head sidewards and dips the right shoulder. Eventually, however, the right hand pulls all the joints over, finally producing a torso bend at time=2.6. Actually, the system bends the torso too much, which causes some confusion in the head area (time=3.0 to 3.8), but in later frames (not shown) the torso straightens up somewhat.

Since all the motion in the input sequence was produced by rotation and bend commands to the simulator, and since the parser only worked with rectilinear motion (no angular representations), the program's description of the motion is inevitably not as parsimonious as it could be. Nevertheless, under the limited capabilities, the description is reasonable. Figure 7 shows the findings of the parser for one joint, the left wrist, together with the true position of that joint in the input images.

C. Discussion

The example described above is a very simple test case: the motions exhibited are very limited -- no gross body movement, no motion in depth, and no occlusion. We feel, however, that more complex motions will be adequately handled by the same basic system. Gross body motion will not be difficult when all features are described in relative coordinate systems. Motion in depth requires a proper use of perspective projection; the boxes will then become cone-shaped objects. Occlusion will necessitate use of the collision detection aspect of the simulator as well as the constraint network.

The example was also a simple test in that the figure in the image and the internal model matched dimensions exactly, since the images themselves were made from the internal model. A less precise match can be accomodated very naturally by the constraint propagation mechanism. The link lengths between each pair of joints can be assigned a minimum of, say, the 5th percentile length among a population pool, and a maximum of the 95th percentile length. Then the constraint propagation will naturally relax, after a number of cycles, to the true link lengths of the input figure, as long as they lie between the 5th and 95th percentiles.

Even though the test sequence is simple, it does illustrate that the motion can be tracked without examining the entirety of each image. Note that at no point in the analysis do we difference two input images, or produce a picture of the model and subtract it from an image frame, or any other such expensive image processing technique. In fact, the results of this section were obtained by only looking at about 20 percent of the pixels in each image frame.

7. CONCLUSIONS AND FUTURE WORK

We have described a computer system capable of analyzing image sequences of human motion. The system operates as a feedback cycle between high level predictions and low level verifications and analysis. All computations and inferences are conducted in a three dimensional space; two dimensions are only used while accessing the image. The system is driven by a detailed model of the human body. The constraints implied by the body model are encoded into a constraint network which can propagate location information between various parts of the body.

One area which we have yet to explore fully is the use of Theorem 6 to further reduce the regions of features in the constraint network. Occasionally, a region fractures into two rather distinct pieces, usually along the same line of sight but separated in the depth dimension. In these cases, a sizable reduction in the network may result from partitioning the region into two pieces and propagating with each separately, as justified by Theorem 6. Major improvements my also arise from exploiting the various _resolution hierarchies_ within the system. The human model can be freed from its current fixed structure by defining a body part hierarchy, such that, for example, the arm includes the upper and lower arms and the hand, and the hand includes the fingers (see Clarke [9] and Marr and Nishihara [32] for similar ideas). The system can then switch to the appropriate level of detail, depending on the accuracy of its predictions and the desires of the user. Similarly, the effective coarseness of the image grid size may be altered in certain regions by sampling the pixels within the region rather than looking at every one, perhaps according to a dithering pattern (Lippel [30]) through the time dimension. This will effectively implement a pyramid data structure for the image (Kelley [28], Uhr [59], Rosenthal [48]). There is also a natural motion description hierarchy, in that a concept such as "walk" is composed of lower-level motion descriptions such as "raise thigh" and "bend knee," corresponding to the straight line fits now produced by our parser. Implementing these hierarchies so that the system can dynamically switch between levels will effectively realize an attention/focus mechanism which shares a number of characteristics with human perception (see O'Rourke [41]). We are currently invesitgating these issues as part of an effort towards developing an image analysis system which can understand American Sign Language.

REFERENCES

[1] Aggarwal, J. K., and Duda, R. O., "Computer Analysis of Moving Polygonal Images," IEEE Transactions on Computers, Vol. C-24, pp.966-976, Oct. 1975.

[2] Badler, N.I., "Temporal Scene Analysis: Conceptual Descriptions of Object Movements," Ph.D. Disseratation, University of Toronto, TR 80, Feb 1975. also University of Pennsylvania Technical Report 80.

[3] Badler, N.I. , O'Rourke, Joseph, Smoliar, Stephen W., Weber, Lynne, "The Simulation of Human Movement by Computer," Movement Project Report Number 14, Computer Science Department, University of Pennsylvania, Sept 1978.

[4] Badler, N.I., O'Rourke, J., and Tolzis, H., "A Spherical Representation of a Human Body for Visualizing Movement," IEEE Proceedings, October 1979.

[5] Badler, N.I., and Smoliar, S.W., "Digital Representations of Human Movement," ACM Computing Surveys, Vol. 11, No. 1, pp.19-38, March 1979.

[6] Ballard, D. H., Brown, C. M., and Feldman, J. A., "An Approach to Knowledge-Directed Image Analysis," in Computer Vision Systems, ed. Allen R. Hanson and Edward M. Riseman, Academic Press, NY, pp.271-281, 1978.

[7] Chien, R.T., and Jones, V.C., "Acquisition of Moving Objects and Hand-Eye Coordination," IJCAI-75, (Tbilisi, Georgia, USSR), pp.737-741, Spetember 1975.

[8] Chow, C. K., and Aggarwal, J. K., "Computer Analysis of Planar Curvilinear Moving Images," IEEE Transactions on Computers, Vol. C-26, pp.179-185, Feb 1977.

[9] Clarke, J.H., "Hierarchical Geometric Models for Visible Surface Algorithms," Communications of the Association for Computing Machinery, Vol. 19, No. 10, pp.547-554, 1976.

[10] Clowes, M., "On Seeing Things," Aritficial Intelligence, Vol. 2, pp.79-116, 1971.

[11] Eskenazi, R., and Cunningham, R., "A Random Access Picture Digitizer, Display, and Memory System," IJCAI-77, (Cambridge, Massachusettes), pp.769-770, August 1977.

[12] Fennema, C.L., and Thompson, W.B., "Velocity Determination in Scenes Containing Several Moving Objects," Computer Graphics and Image Processing, Vol. 9, No. 4, pp.301-315, April 1979.

[13] Fikes, R.E., "REF-ARF: A System for Solving Problems Stated as Procedures," Artificial Intelligence, Vol. 1, pp.27-120, 1970.

[14] Freduer, E.C., "Synthesizing Constraint Expressions," Communications of the Association for Computing Machinery, Vol. 21, No. 11, pp.958-966, November 1978.

[15] Futrelle, R. P., "GALATEA: Interactive Graphics for the Analysis of Moving Images," Proc. IFIP-74, (J. Rosenfeld, ed.), North-Holland, Amsterdam, pp.712-716, 1974.

[16] Futrelle, R.P., and Speckert, G.C., "Extraction of Motion Data by Interactive Processing," Proceedings of IEEE Conference on Pattern Recognition and Image Processing, pp.405-408 , June 1978.

[17] Guzman, A., "Decomposition of a Visual Scene into Three-Dimensional Bodies," Proceedings of AFIPS Fall Joint Computer Conference, Vol. 33, pp.291-304, December 1968. also in Computer Methods in Image Analysis, J.K. Aggarwal, R.O. Duda, and A. Rosenfeld, Eds., IEEE Press: New York, 1977, pp. 324-337.

[18] Haralick, R.M., and Shapiro, L.G., "The Consistent Labeling Problem: Part I," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-1, No. 2, pp.173-184, April 1979.

[19] Hayes, Philip J., "Mapping Input onto Schemas," Technical Report TR29, Department of Computer Science, University of Rochester, June 1978.

[20] Herman, M., "How to Understand a Sad Stick Figure: A Cognitive Theory and Computer Model," University of Maryland Report TR-657, MCS-76-23763, May 1978.

[21] Herman, M., "Understanding Stick Figures," Report TR-603, University of Maryland, November 1977.

[22] Herman, G.T., and Liu, H.K., "Dynamic Boundary Sruface Detection," Computer Graphics and Image Processing, Vol. 7, No. 1, pp.130-138, February 1978.

[23] Hernan, M.A., and Jimenez, J., "Automatic Analysis of Movies in Fluid Mechanics," Workshop on Computer Analysis of Time-Varying Imagery, Abstracts (IEEE), University of Pennsylvania, pp.134-135, April 1979.

[24] Hochberg, J., "The Representation of Things and People," in Art, Perception, and Reality, by E.H. Gombrich, J. Hochberg, and M. Black, Johns Hopkins University Press, pp.47-94, 1972.

[25] Huffman, D.A., "Impossible Objects as Nonsense Sentences," Machine Intelligence, B. Meltzer and D. Michie, Eds., Vol. 6, New York: Halsted Press, pp.295-323, 1971.

[26] Jain, R., and Nagel, H.-H., "On the Analysis of Accumulative Difference Pictures from Image Sequences of Real World Scenes," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-1, No. 2, pp.206-214, April 1979.

[27] Kanade, T., "Region Segmentation: Singal vs. Semantics," 4th International Joint Conference on Pattern Recognition, pp.95-105, 1977.

[28] Kelley, M.D., "Edge Detection in Pictures by Computer Using Planning," in Machine Intelligence 6, B. Meltzer and D. Michie, Eds., American Elsevier, 1971.

[29] Levin, M. D., and Youssef, Y. M., "An Automatic Picture Processing Method for Tracking and Quantifying the Dynamics of Blood Cell Motion," Report No. 78-4R, Dept. of Electrical Engineering, McGill University, February 1978.

[30] Lippel, B., "Two- and Three-Dimensional Ordered Dither in Bi-Level Picture Displays," Proceedings of SID, Vol. 17/2, pp.115-121, 1976.

[31] Mackworth, A., "Consistency in Networks of Relations," Artificial Intelligence, Vol. 8, pp.99-118, 1977.

[32] Marr, D., and Nishihara, H.K., "Representation and Recognition of the Spatial Organization of Three-Dimensional Shapes," MIT AI memo 377, August 1976.

[33] Martin, W. N., and Aggarwal, J. K., "Dynamic Scene Analysis: The Study of Moving Images," Dept. of CS & EE, University of Texas at Austin,Technical Report No. 184, Jan 1977.

[34] Minsky, M., "A Framework for Representing Knowledge," in The Psychology of Computer Vision, P.H. Winston, Ed.,McGraw-Hill, pp.211-277, 1975.

[35] Montanari, U., "Networks of Constraints: Fundamental Properties and Applications to Picture Processing," Information Sciences, Vol. 7, No. 2, pp.95-132, 1974.

[36] Nagel, Hans-Helmut, "Formation of an Object Concept by Analysis of Systematic Time Variations in the Optically Perceptible Environment," Computer Graphics and Image Processing, Vol. 7, No. 2, pp.149-194, Apr 1978.

[37] Nagel, Hans-Hellmut, "Analysis Techniques for Image Sequences," Proceedings of the Fourth International Joint Conference on Pattern Recognition, (Kyoto, Japan), pp.186-211, Nov 1978.

[38] Neisser, Ulric, Cognition and Reality: Principles and Implications of Cognitive Psychology, W. H. Freeman & Co., San Francisco, 1976.

[39] Neuman, B., "Interpretation of Imperfect Object Contours," Procededing of the International Joint Conference on Pattern Recognition, (IJCPR-78), 1978.

[42] O'Rourke, J., "Representation and Display of Three Dimensional Objects with Spheres," Computer Science Dept., Univeristy of Pennsylvania, UP-MS-CIS-77-77, August 1977.

[41] O'Rourke, J., "An Optimal Incremental Algorithm for Finding All Straight Lines Consistent with a Set of Data," Techincal Report, Computer Science Dept., University of Pennsylvania, August 1979.

[42] O'Rourke, J., "An Image Analysis System for Human Motion," Computer Science Department, University of Pennsylvania, (to appear 1980).

[43] O'Rourke, J., "Propagation of Constraints Through a Linked Network," Technical Report, Computer Science Department, University of Pennsylvania, Jan 1979.

[44] Potter, J.L., "Scene Segmentation Using Motion Information," Computer Graphics and Image Processing, Vol. 6, pp.558-581, 1977.

[45] Rashid, R.F., "Lights: A System for the Interpretation of Moving Light Displays," Workshop on Computer Analysis of Time-Varying Imagery, Abstracts (IEEE), University of Pennsylvania, pp.52-54, April 1979.

[46] Roach, J. .W., and Aggarwal, J. K., "Computer Tracking of Objects Moving in Space," IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Volume PAMI-1, Number 2, pp.127-135, April 1979.

[47] Rosenfeld, A., Hummel, R., and Zucker, S., "Scene Labeling by Relaxation Operations," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-6, No. 6, pp.420-433, June 1976.

[48] Rosenthal, D., "An Inquiry Driven Computer Vision System Designed for Use of Multi-Spectral and Multi-Resolution Image Data," Computer Science Dept., University of Pennsylvania, December 1978.

[49] Russel, Daniel M., "Where Do I Look Now? Modeling and Inferring Object Locations by Constraints," Proceedings of Conference on Pattern Recognition and Image Processing, (PRIP79), pp.175-183, Aug 1979.

[50] Scacchi, Walt, "Visual Motion Perception by Intelligent Systems," Proceeding of Conference on Pattern Recognition and Image Processing, (PRIP79), pp.646-652, Aug 1979.

[51] Schudy, R.B., "Towards an Anotomical Model of Heart Motion as Seen in Cardiac Ultrasound Data," Workshop on Computer Analysis of Time-Varying Imagery, Abstracts (IEEE), University of Pennsylvania, pp.87-89, April 1979.

[52] Shapira, R., and Freeman, H., "The Cyclic Order of Vertices as an Aid in Scene Analysis," Communications of the Association for Computing Machinery, Vol. 22, No. 6, pp.368-375, June 1979.

[53] Shneier, M.O., "Recognition Using Semantic Constraints," IJCAI-77, (Cambridge, Massachusettes), pp.585-589, August 1977.

[54] Stallman, Richard M., and Sussman, Gerald Jay, "Forward Reasoning and Dependency-Directed Backtracking In a System for Computer-Aided Circuit Analysis," MIT AI Memo No. 380, Sept 1976.

[55] Tsotsos, John K., "Knowledge-Base Driven Analysis of Cinecardioangiograms," IJCAI-5, (Cambridge, Massachusettes), p.699, August 1977.

[56] Tsotsos, John K., "Some Notes on Motion Understanding," IJCAI-5, (Cambridge, Massachusettes), p.611, August 1977.

[57] Tsuji, S., Morizono, A., Kuroda, S., "Understanding a Simple Cartoon Film by a Computer Vision System," IJCAI-5, pp.609-610, 1977.

[58] Tsuji, S., Osada, N., Yachida, M., "Three Dimensional Movement Analysis of Dynamic Line Images," Workshop on Computer Analysis of Time-Varying Imagery, Abstracts (IEEE), University of Pennsylvania, pp.20-22, April 1979.

[59] Uhr, L., and Douglass, R., "A 'Recognition Cone'

Perceptual System: Brief Test Results," IJCAI-77, (Cambridge, Massachusettes), p.597, August 1977.

[60] Ullman, Shimon, The Interpretation of Visual Motion, MIT Press, Cambridge, Massachusetts, 1979.

[61] Ullman, J.R., "Associating Parts of Patterns," Information Control, Vol. 9, pp.583-601, 1966.

[62] Wallace, T.P., and Mitchell, O.R., "Real-Time Analysis of Three-Dimensional Movement Using Fourier Descriptors," Workshop on Computer Analysis of Time-Varying Imagery, Abstracts (IEEE), University of Pennsylvania, pp.32-33, April 1979.

[63] Waltz, David, "Understanding Line Drawings of Scenes with Shadows," in The Psychology of Computer Vision, P. H. Winston, Ed., McGraw-Hill: New York, pp.19-95, 1975.

[64] Weiler, K., and Atherton, P., "Hidden Surface Removal Using Polygon Area Sorting," SIGGRAPH 77 Proceedings, Vol. 11, No. 2, pp.214-222, Summer 1977.

[65] Yachida, M., Ikeda, M., and Tsuji, S., "Efficient Analysis of Noisy Dynamic Pictures Using Plan," Workshop on Computer Analysis of Time-Varying Imagery, Abstracts (IEEE), University of Pennsylvania, pp.90-92, April 1979.

[66] Zucker, S.W., "Relaxation Labeling and the Reduction of Local Ambiguities," Proceedings of the 3rd International Joint Conference on Pattern Recognition, (IJCPR-3), pp.852-861, November 1976.

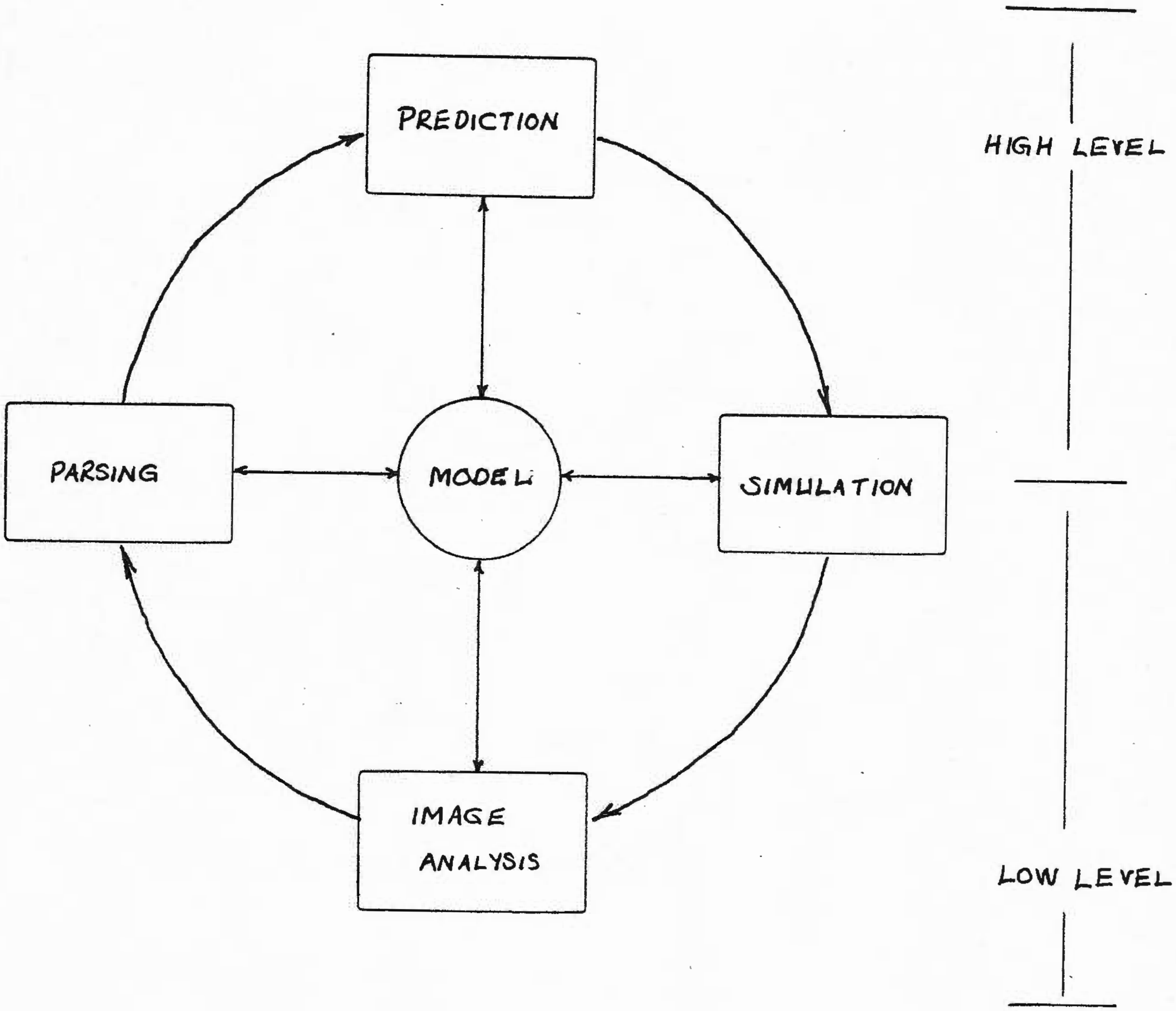


Fig. 1 System Components. The prediction component operates at the high level, the image analysis is conducted at the low level, and parsing and simulation components function to translate information between the levels.

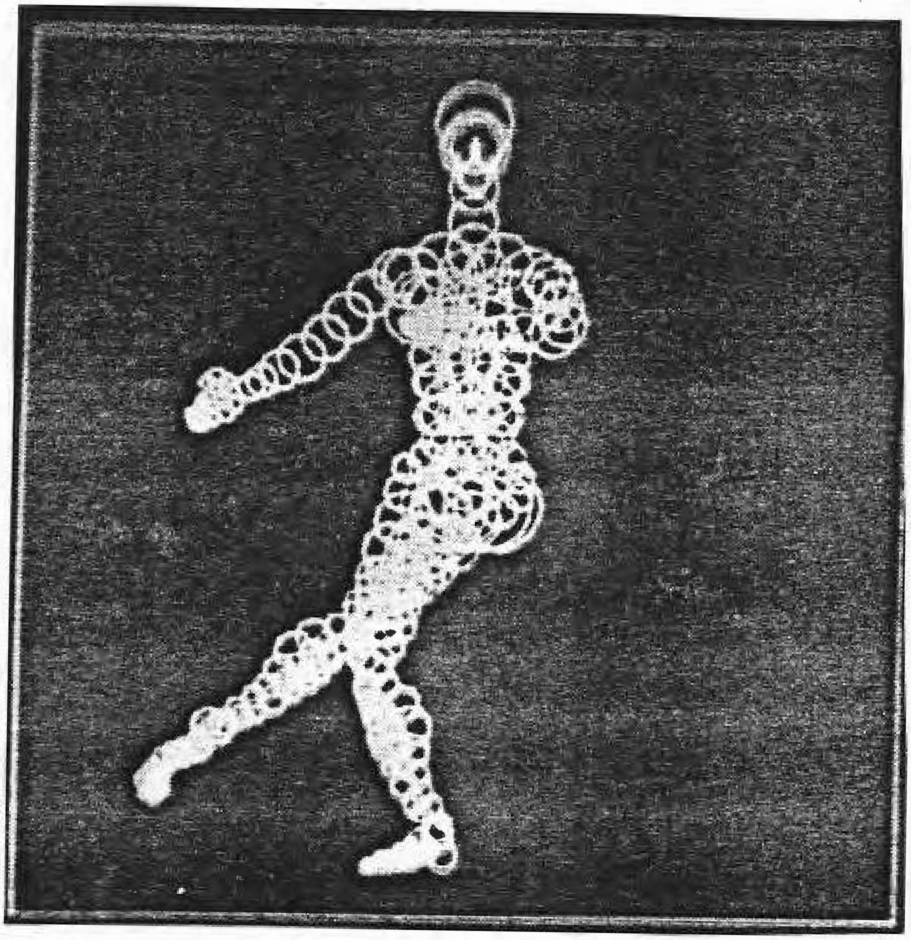


Fig. 2 The current human model, consisting of 24 segments, 25 joints, and 585 spheres.

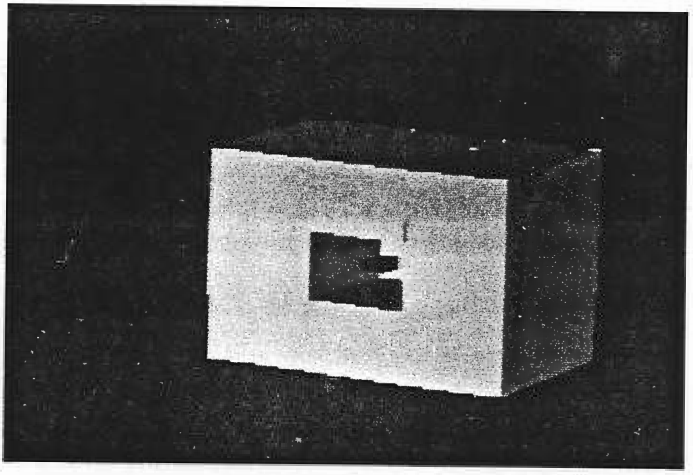
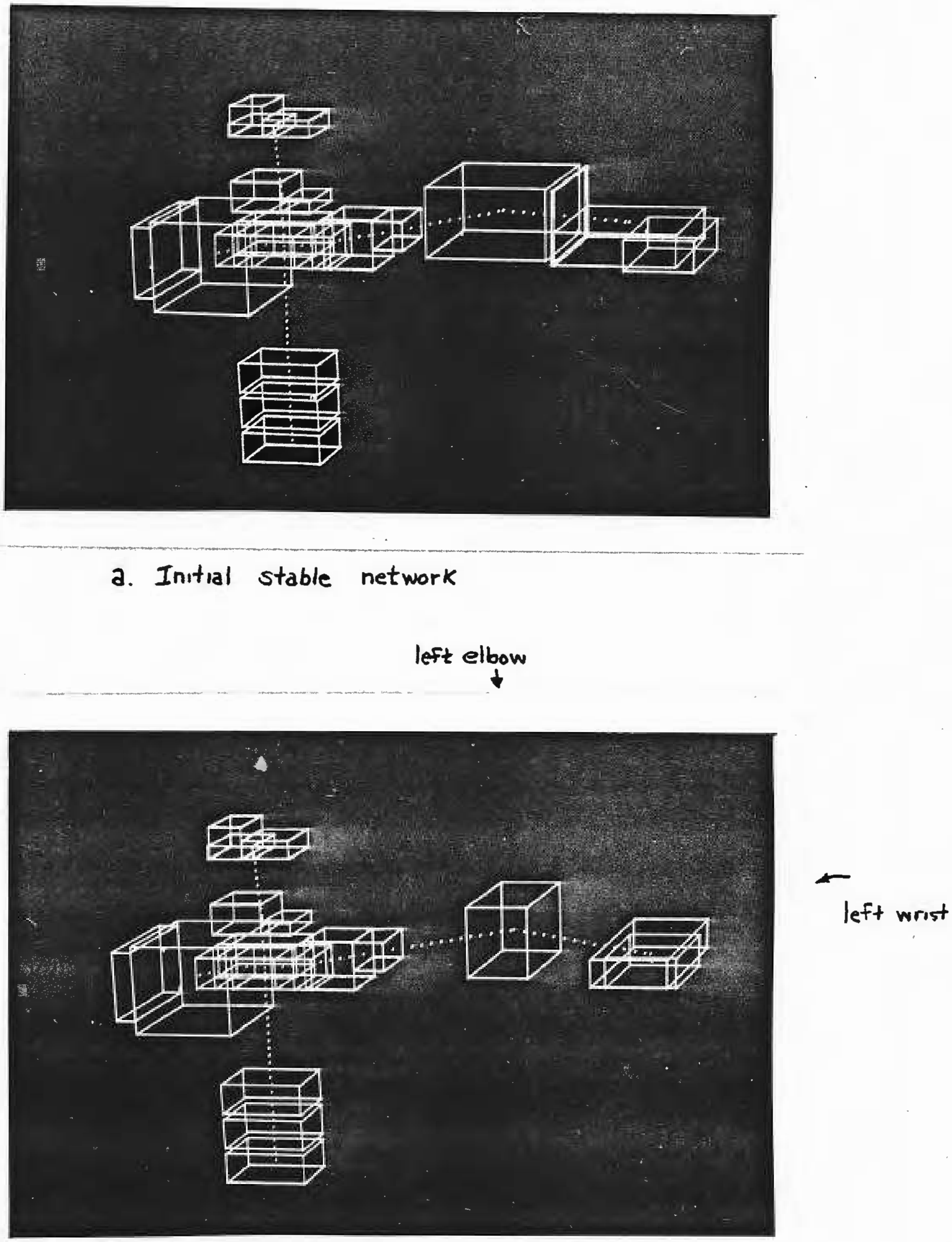


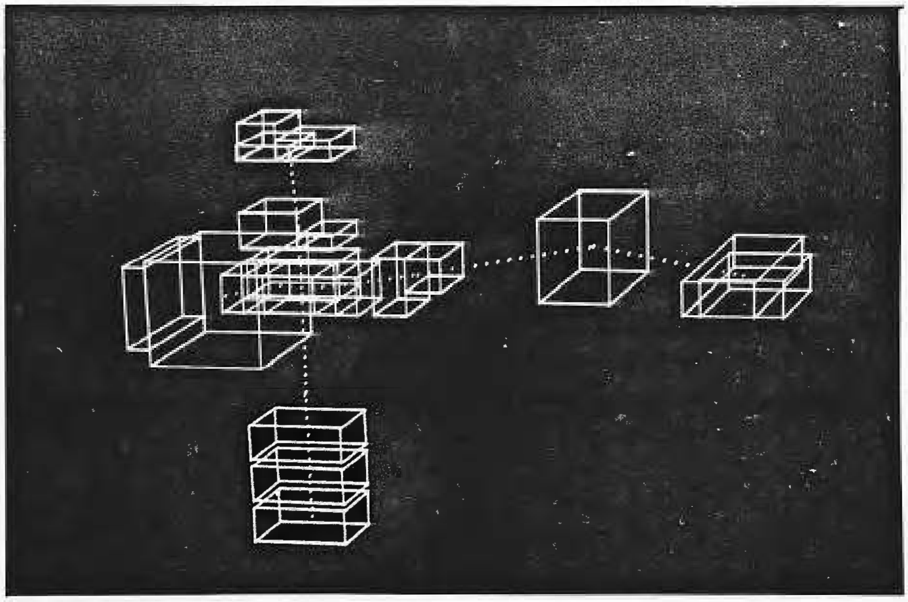
Fig. 4 Constraint region for box. The front face has been cut away for illustration purposes.



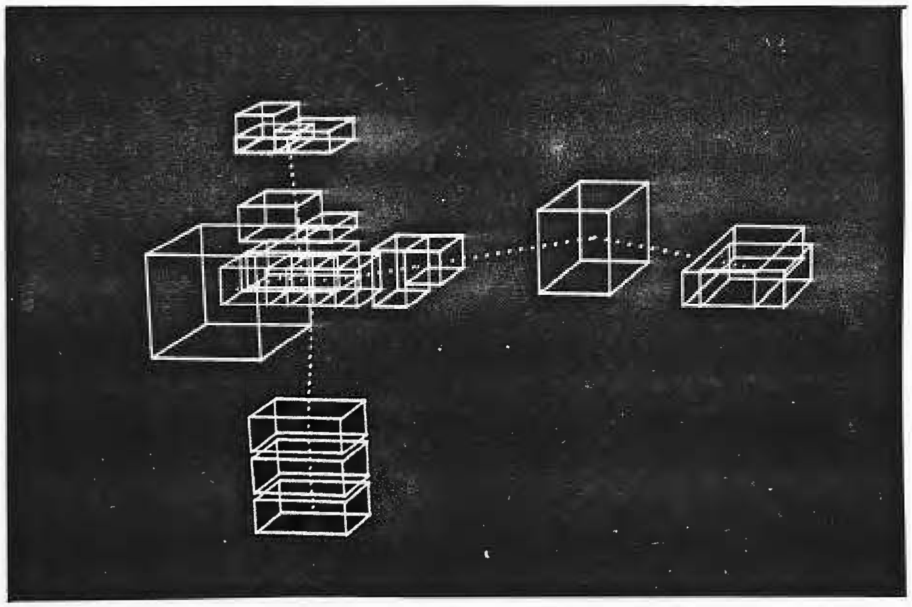
a. Initial stable network

b. Wrist and elbow regions reduced

Fig. 5 Example of propagation through network.

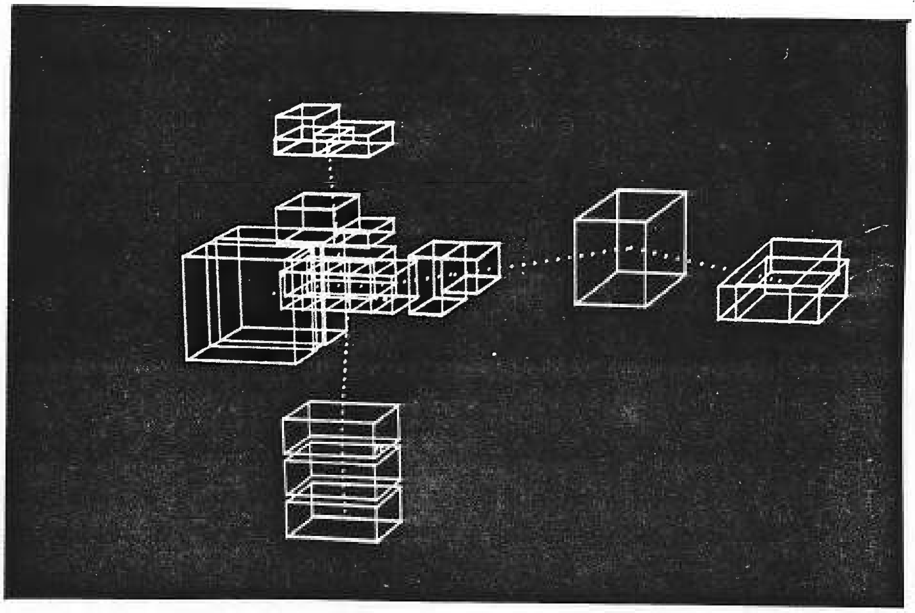


c. Left shoulder region reduced



d. Right shoulder region reduced

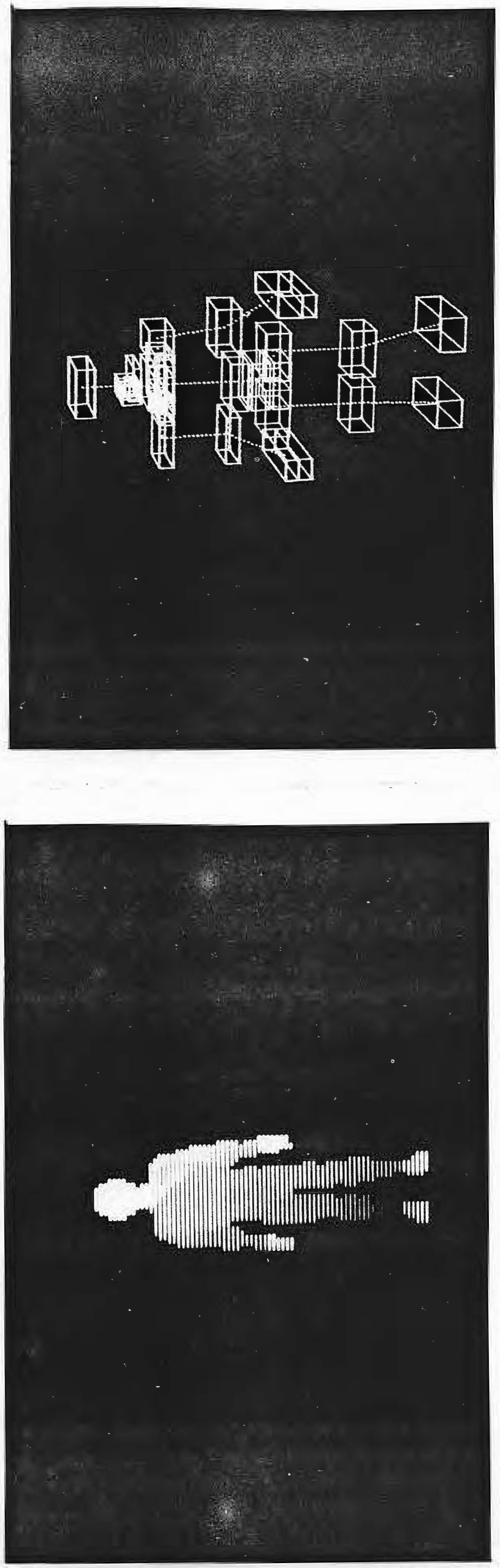
Fig. 5 (continued)



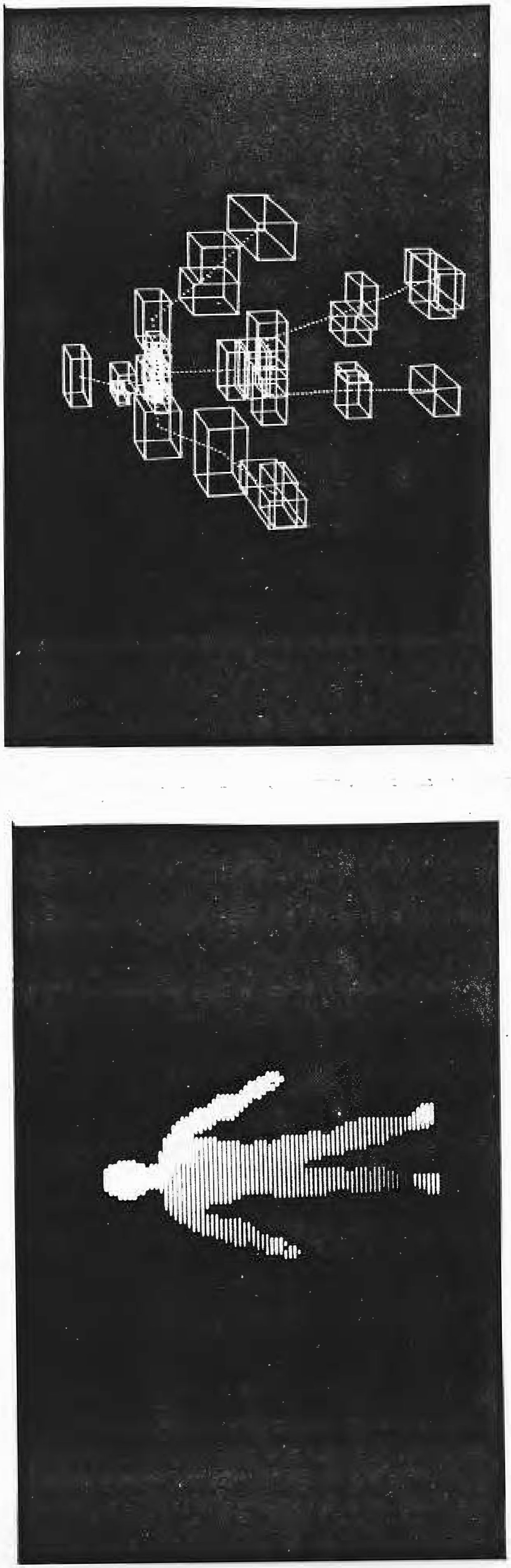
Right shoulder

e. Right shoulder region fractured.
Network is again stable.

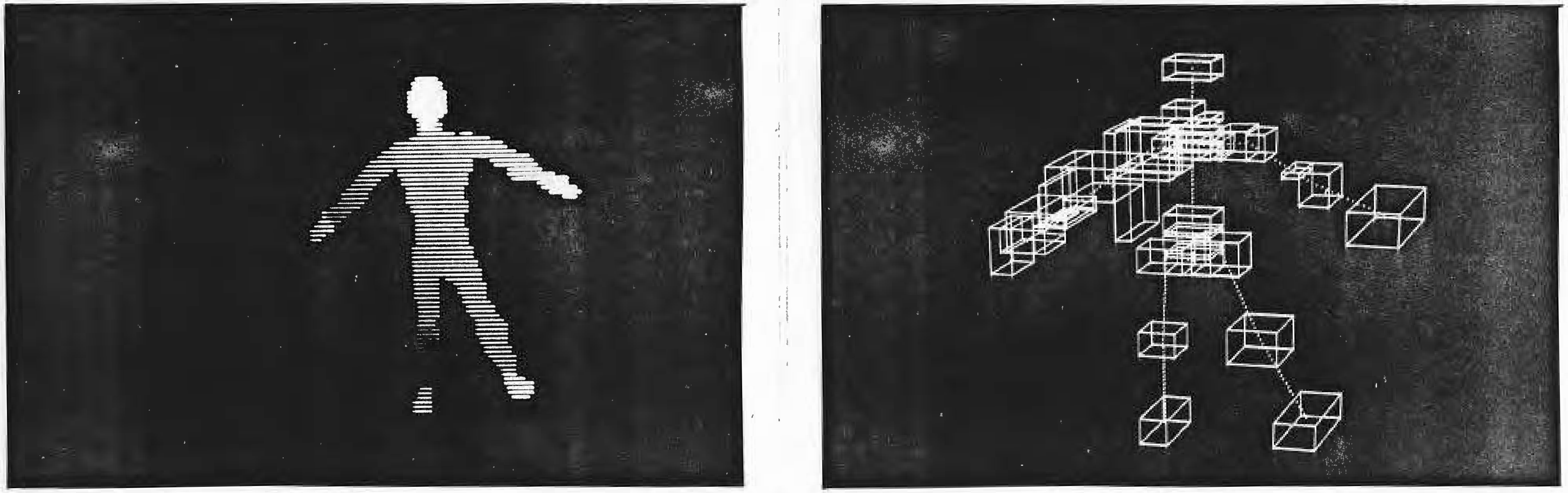
Fig. 5 (continued)



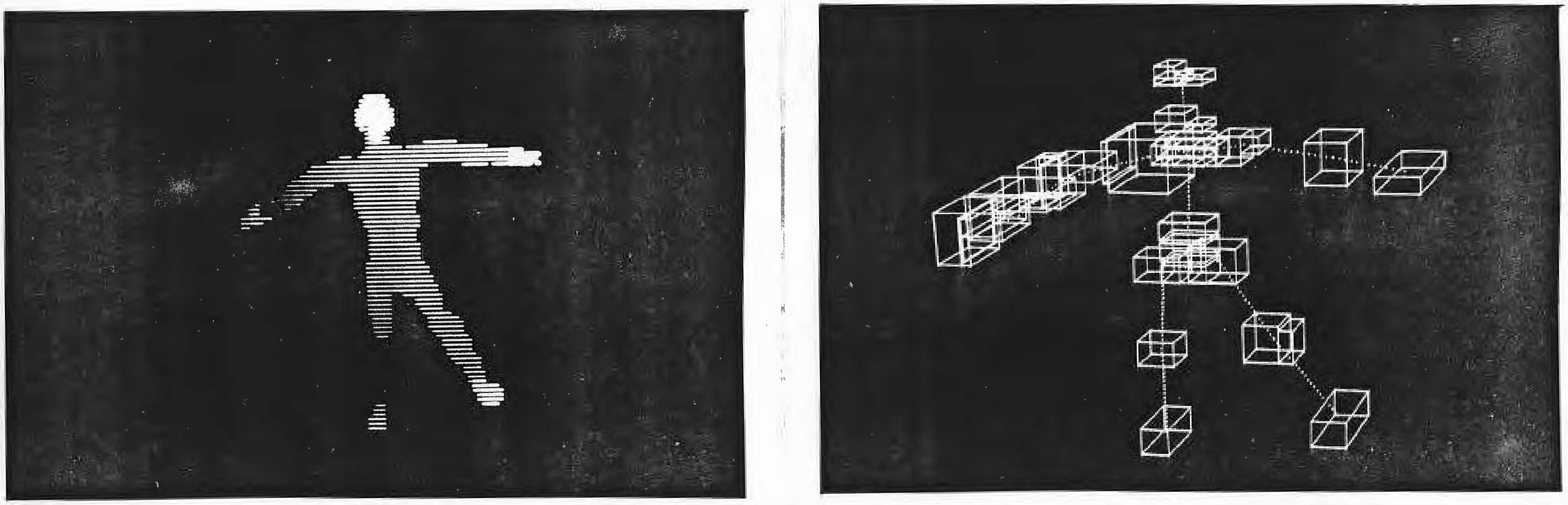
time = 0.2



time = 0.6



time = 1.0



time = 1.4