

**FINDING TRAJECTORIES OF POINTS IN
A MONOCULAR IMAGE SEQUENCE**

I. K. Sethi

**Department of Computer Science
Wayne State University
Detroit, MI 43202**

Ramesh Jain

**Department of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, Michigan 48109-1109**

April 1985

CENTER FOR RESEARCH ON INTEGRATED MANUFACTURING

Robot Systems Division

COLLEGE OF ENGINEERING

THE UNIVERSITY OF MICHIGAN

ANN ARBOR, MICHIGAN 48109-1109

TABLE OF CONTENTS

1. Introduction	3
2. Correspondence Problem	7
3. Applications of the Tracking	9
4. Path Coherence and Smoothness of Motion	10
5. Finding Trajectories	14
6. Optimization Algorithm	16
6.1. Algorithm	17
7. Path Coherence Function	19
8. Experiments	20
9. Modified Greedy Exchange Algorithm	22
9.1. Modified GE Algorithm	23
10. Occlusion	24
10.1. Hypothesize and Test Occlusion Algorithm	25
11. Conclusion	27
12. REFERENCES	29

ABSTRACT

Identifying the same physical point in more than one image, the correspondence problem, is vital in motion analysis. Most research for establishing correspondence uses only two frames of a sequence to solve this problem. By using a sequence of frames it is possible to exploit the fact that due to inertia, the motion of an object can not change instantaneously. By using *smoothness of motion*, it is possible to solve the correspondence problem for arbitrary motion of several non-rigid objects in a scene. We formulate the correspondence problem as an optimization problem and propose an iterative algorithm to find trajectories of points in a monocular image sequence. A modified form of this algorithm is useful in case of occlusion also. We demonstrate the efficacy of this approach considering synthetic, laboratory, and real scenes.

Index Terms: Correspondence, motion, structure from motion, path coherence, smoothness of motion.

1. Introduction

The last few years have seen increasing interest in dynamic scene analysis. The input to a dynamic scene analysis system is a sequence of images. As is well known, an image represents a 2-D projection of a 3-D scene at a time instant. A major problem in a computer vision system is to recover the information about objects in a scene from images. This problem can not be solved without some assumptions about the world. A sequence of frames allows one additional dimension to recover the information about 3-Dimensional world that is lost in the projection process. Multiple views of a moving object acquired using a stationary camera may allow recovery of structure of the object [Ull79, TsH81, TsH82, THZ82, Tod82, RaA84]. A mobile camera may be used to acquire information about the structure of the stationary objects in a scene using optical flow [Gib79, Pra81], axial motion stereo [OBJ84] and other methods [Joh76, JeJ84].

The most attractive feature of a dynamic scene, for recovering information, is, however, the fact that by working in a spatio-temporal domain, a vision system may be in position to *hypothesize-and-test* freely and thus employ the principle of least commitment [Mar76, JaH82] to solve complex problems. Many researchers in psychology of vision support the recovery of information from image sequences, rather than an image, representing a scene [Nei76, Gib79, Joh76]. Gibson [Gib79] argued in support of active information pick-up by the observer in an environment. Johansson [Joh76] demonstrated the efficacy of only motion information in recognition of objects using moving light displays. Neisser [Nei76] proposed a model according to which the perceptual processes continually interact with the incoming information to verify anticipations formed on the basis of available information until a given time instant. In computer

vision systems, the efficacy of even noise sensitive approaches, such as difference and accumulative difference pictures, was demonstrated by using hypothesize-and-test mechanisms to analyze complex real world scenes [JaN79]. Though many researchers are addressing the problem of recovering information in dynamic scenes, it appears that due to the legacy of static scenes, most researchers are approaching the recovery problem using just two or three frames of a sequence. This self-imposed restriction results in approaches suitable for *quasi-dynamic* scene analysis, rather than dynamic scene analysis. Since the information recovery process requires constraints about the scene, the analysis based on minimal number of frames rests on assumptions that ignore the most important information in dynamic scenes - the motion of objects.

Structure from motion has attracted significant research efforts recently from researchers working in the field of dynamic scene analysis [RoA80, Tod82, TsH81, THZ82, Ull79, WeA80]. Ullman popularized the *rigidity assumption* in computer vision. This assumption states that any set of elements undergoing a 2-D transformation which has a unique interpretation as a rigid body moving in space should be so interpreted. The rigidity assumption allows recovery of the structure of objects, under certain conditions, in 3 frames.

Another popular approach for the recovery of the structure and motion is using optical flow fields [Gib79, Pra81, JeJ84, MuT85, RiL83], while others try to recover the same information using points in frames. Optical flow is the field of retinal velocities. In computer vision, it is considered as velocity field for all image points. It has been shown that the optical flow contains information about the motion of the observer and the environment. Approaches for the computation [HoS81], and for the recovery of structure [Wax84] have been proposed. Considering the difficulties in computing opti-

cal flow of acceptable quality, some efforts are being made to recover the structure using the characteristics of optical flow, but without computing it [Jai83, OBJ84].

Recently, the trajectory based recovery has attracted some attention [Tod82, SeJ83, WeA82, RaA84, Jen 83]. It has been shown that human visual system requires an extended frame sequence to recover structure of moving patterns [Tod82, RaA84] and that the noise sensitivity of the system improves with increase in number of frames [DLP84]. We believe that the trajectory based approach to the recovery of structure is more suitable for complex scenes. This approach will allow successive refinement of the structure of objects as more frames are observed. Using a first few frames, an initial, though tentative, structure of an object may be hypothesized and then this structure could be refined based on later observations. A major advantage of such an approach will be to free ourselves from assumptions of rigidity and rely more on natural assumptions of motion characteristics.

A major step in the recovery using a token based approach, both in quasi-dynamic and dynamic approaches, is identifying images of a physical point in several frames, usually called the correspondence problem. This paper addresses the problem of establishing correspondence for tokens in a sequence of images. Contrary to most other approaches, we do not try to solve the correspondence problem using just two frames. In two frames, one can use only the location of the tokens and has to make assumptions about the nature of the objects or about the maximum velocity of objects. A longer sequence of frames allows use of velocity information in solving the problem. Jenkins [Jen83] proposed an approach for establishing correspondence in a binocular image sequence. He suggested use of the smoothness of velocity and his approach is influenced by the Gestalt rules. We also believe that the smoothness of velocity is more

general and more powerful compared to the rigidity assumption. As shown in a later section, our approach is similar to Jenkins' in using smoothness, but is very different from his approach in many other respects.

We propose an optimization approach to the correspondence problem and propose an iterative optimization algorithm to find optimal trajectories. This computational approach allows solution of the correspondence problem incrementally. As new frames are acquired, our iterative optimization approach, establishes the correspondence. Thus solution of the correspondence problem also gives us trajectories of points, which, in turn, may be used to obtain structure. Our approach is able to handle limited occlusion and disocclusion also. A limitation of the proposed approach, similar to Jenkins' is that it assumes uniform motion. If due to some reason there is a sudden change in motion, this approach will not be able to detect it and may result in wrong correspondence.

Section 2 discusses approaches used for correspondence with emphasis on Jenkins' approach because of its similarity to our approach. Section 3 gives possible applications of the proposed approach. Path coherence and smoothness of motion, as used in our approach, are discussed in Section 4, and the optimization approach to the correspondence is presented in Section 5. Section 6 gives the Greedy exchange algorithm for establishing correspondence. The issues related to the formulation of the path coherence function are discussed in Section 7 and results of the application of the algorithm are given in the next Section. The basic algorithm has some problems. These problems and modifications to handle occlusion are discussed next. Results for laboratory sequences and real movie (Superman) sequence are presented to show the efficacy of the algorithm.

2. Correspondence Problem

In the recovery of structure from motion, the establishment of the token correspondence has been a major hurdle. Ullman [Ull79] proposed a minimal mapping approach to this problem. His approach was based on the 2-D features of the tokens and the 2-D distance between tokens. His elegant approach is computationally very attractive because of the possibility of implementation using a network of processors. The major problem with the approach is, however, that it fails to consider the fact that most realistic motion analysis must be performed by considering an extended region of space-time. Correspondence based on 2-D distances alone may lead to erroneous results in scenes containing several objects moving in assorted directions and in case of non-rigid motion. In fact, psychological experiments conducted to study the feasibility of this approach, consider only a few points in artificial motion situations; Todd [Tod82] shows that most human observers require several frames to infer the structure of moving objects even in simple experimental situations.

Barnard and Thompson [BaT79] and Prager and Arbib [PrA83] proposed relaxation based approaches to the solution of this problem. In relaxation based methods, all tokens in the frame 2 that are within a spatial neighborhood of a token in the frame 1 are assigned a weight indicating the match strength between the token in the frame 1 and the tokens in the frame 2. It is assumed that only one of these matches is correct. The correct match is obtained by using relaxation. The relaxation process tries to find most consistent matches based on the disparities. The most consistent match implies uniform motion and hence *rigid* objects. These approaches also use 2-D displacements in two frames as a major factor in establishing the correspondence.

These approaches to the solution of the correspondence problem try to solve the problem using tokens in two frames only. In fact, most research in dynamic scene analysis, or time varying image analysis, has been concerned with only 2 or 3 frames, *not with dynamic scenes*. It is not clear, why most researchers in this field have been concerned with recovery of information considering minimal number of points in minimal number of frames, rather than recovery of reliable information (see Brady's comment on Todd's paper in [Bra83]). It appears that for real world problems, like many other problems in dynamic scene analysis, the correspondence problem may be solved more reliably using more than two frames. Due to the self-imposed limitation of working with only two frames, some form of the rigidity assumption was essential for solving the problem. In two frames the most powerful constraint is the spatial uniformity of displacements. The rigidity assumption, with orthographic projections, is very appealing because it guarantees the uniformity of displacements and is domain-independent. If we consider more than two frames in establishing correspondence, however, we may admit non-rigid objects in our analysis. Thus assumptions about the nature of motion can be used to relax those about the nature of objects. In its simplest form, the smoothness of motion is used to relax rigidity of objects.

This can be done using *path coherence* and assuming *smoothness of motion*. In this paper we introduce the notion of *path coherence* and use it for solving the correspondence problem as an optimization problem.

Jenkins [Jen83] proposed a novel approach for combining the solution of correspondence problems for stereo and motion. He argues in support of the smoothness assumption. He is interested, like us, in an approach that is not based on assumptions like rigidity [Ull79], planarity [HoF80], and rotation and translation [WeA82]; but on the smoothness of motion. His notion of smoothness is related to the 3-D motion.

It is based on

1. The location of a given point would be relatively unchanged from one frame to the next.
2. The scalar velocity, or speed of a given point would be relatively unchanged from one frame to the next.
3. The direction of motion of a given point would be relatively unchanged from one frame to the next.

To solve the correspondence problem, he assumed that at all instants, all points were visible in both images of the binocular frame sequence, and the 3-D position and velocity of the points was known at some initial time. As will be shown in the next section, our approach uses smoothness of image motion and uses a monocular sequence. Moreover, we have extended the algorithm to cope with limited occlusion, severe occlusion requires characteristics of objects. Since we are concerned only in points, our domain in this paper may be considered similar to that of Rashid [Ras80] O'Rourke and Badler [ORB80], and Jenkins[Jen83]. We do not address the problem of occlusion analysis in general cases, using other than simple motion characteristics in this paper.

3. Applications of the Tracking

The tracking approach proposed here will play an important role in many different applications. A complete object may be assumed a point and then the proposed algorithms may be applied to those points. This approach will be useful in analysis of scenes containing several moving objects. Some possible applications are traffic analysis, cell motion analysis and others. More important application of the proposed algorithm may be in segmentation of dynamic scenes by grouping points with similar

motion in one object [FIC81]. A nonrigid or jointed object can be segmented by tracking individual points and then grouping these points based on common motion characteristics, such as done by humans as demonstrated by Johansson [Joh76]. This segmentation will give idea about the general structure of an object. For rigid objects, by tracking several points on the object, one may use structure from motion approaches using trajectories [Tod82].

4. Path Coherence and Smoothness of Motion

We exploit the fact that, due to inertia, the motion of a physical entity cannot change instantaneously. If a frame sequence is acquired at a rate such that no dramatic changes take place between two consecutive frames, then for most physical objects no abrupt change in the motion can be observed. Thus a very reasonable assumption for the analysis of real world dynamic scenes is: *the motion of an object at any time instant can not change abruptly. This is path coherence.*

Note that this assumption will be valid for all moving objects; rigid and non-rigid. If the objects collide then some high level process may be required to analyze the motion after the collision. In most other cases, if the sampling rate is high enough then the changes in the motion will be gradual.

The projection of a smooth 3-D trajectory will be a smooth 2-D trajectory in both orthographic and perspective projections. This can be easily verified by considering the projection equations and analyzing first and second derivatives of the projected points. It can be shown that for any continuous variations in the velocity of a point in space, the velocity of the projected point will also be continuous. This fact allows us to study correspondence problem using path coherence in images, rather than in 3-D space as in

[Jen83]. It should be mentioned here that in the pathological case of a trajectory that has in part only motion along the optical axis of the camera, it may appear that the object stops and then restarts. Based only on image information, it may not be possible to resolve this ambiguity. However, if the sampling rate is appropriate, the correspondence algorithm will not have any problem because the notion of the path coherence will still be valid.

Let us consider a trajectory, T_i , for a point P_i in an image sequence. The coordinates of the point are given by a vector X_i in a frame. The coordinates in the k^{th} frame for this point are denoted by $X_{i,k}$. Let us represent a trajectory T_i as :

$$T_i = \langle X_{i,1}, X_{i,2}, \dots, X_{i,n} \rangle \quad (1)$$

where $X_{i,k}$ is the point in the k^{th} frame participating in the i^{th} trajectory. A set of points in the k^{th} frame will be denoted by X^k .

Now let us consider the deviation d_i^k in the path of the point in the k^{th} frame. We define a measure of deviation in the path, and hence a measure of path coherence, as

$$d_i^k = \Psi(\overline{X_{i,k-1} X_{i,k}}, \overline{X_{i,k} X_{i,k+1}}) \quad (2)$$

Where Ψ is a path coherence function. This function may consider direction, magnitude, or both, of the displacements of the point in an image sequence.

We may define the deviation for the trajectory as

$$D_i = \sum_{k=2}^{n-1} d_i^k \quad (3)$$

Note that for translational motion, the difference in the direction will be a suitable measure of deviation; in more complex cases one may need more rigorous measures.

For uniform translational motion in image plane D_i will be zero for a correct trajectory. For an arbitrary trajectory, if the motion is smooth, the value of D_i will be very small.

Let us consider the projection of two points in frames of a sequence, as shown in Figure 1. The points are moving in different directions. The three frames under consideration are at the crossover of the trajectories of the points. The rigidity assumption can not be used in this situation. Moreover, if distances between points in two frames are the basis for finding the trajectory of the points, then one will obtain an abrupt change in the direction of motion from the second frame to the third frame. The path coherence will allow determination of the correct correspondences over three frames and will result in the correct trajectories, as shown in Figure 2. If we are considering a trajectory, then the instantaneous change in the motion of a physical entity may be obtained by comparing the motion vectors from the previous frame and the current frame. Path coherence requires that there should not be abrupt changes in the motion vectors obtained from consecutive frames.

In a scene, several objects may be undergoing random motions. The objects may be rigid, jointed, or arbitrary, the motion of each object, however, will be smooth. Thus, if a dynamic scene is sampled at a rate fast enough to capture all significant events in the scene, then the observed motion of all objects will be smooth trajectories. As suggested by Todd [Tod82], more reliable motion perception may be achieved by considering the primitive units of motion perception to be the global trajectories in an optic array defined over an extended region of space-time.

Now, if there are m points in a sequence of n frames, resulting in m trajectories, then the deviation for all the trajectories should be considered. This deviation is given by

$$D(T_1, T_2, \dots, T_m) = \sum_{i=1}^m \sum_{k=2}^{n-1} d_i^k \quad (4)$$

The problem we are facing is to determine these trajectories. Thus our problem, in its simplest form, may be stated as: *Given m points in each of the n frames of a sequence, determine m trajectories.* Since there are m^n total possible trajectories, only m of which are to be selected, we need a method to judiciously select these m trajectories. We know that each trajectory is smooth, and hence a reasonable approach is to try to maximize the smoothness of the set of trajectories that are selected as the correct set. This approach is consistent with those proposed by [Jen83, RaA84, Tod82]. This approach of maximizing smoothness of motion may be stated as: *If a set of points undergoing a 2-D transformation has a unique interpretation as a set of points following smooth 3-D trajectories, then it should be so interpreted.*

Note that the above smoothness of motion conjecture is an extension of the path coherence assumption. The path coherence assumption for a point in a frame considers its motion from the previous frame to the current frame and from the current frame to the next frame; the smoothness of motion hypothesis extends this notion of *smooth velocity changes* to a longer sequence. It should be emphasized here that this hypothesis gives more importance to the continuity of motion, rather than displacements in just two frames, and considers individual point characteristics for establishing correspondence in a dynamic scene. Clearly, two frames are not enough to apply smoothness of motion, not even to use path coherence. We require at least three frames to compute the change in the motion. For verifying smoothness of motion, more frames will be required.

5. Finding Trajectories

In a general but noise free case, elements may be moving in assorted directions. Suppose that a feature detector, such as a moving corner detector [ShJ84], gives points in each frame. In the problem formulation, let us assume that number of moving points in each frame is m . The problem now is to find trajectories of m points in n frames. We can make general assumptions based only on motion characteristics; not on the nature of objects. These assumptions are

1. An element in a frame can only belong to one trajectory,
2. There should be m trajectories, each containing n points,
3. For each trajectory the deviation should be minimal, and
4. The sum of the deviations for trajectories should be minimal.

The first two assumptions above, are to assign each point to only one unique trajectory. The path coherence assumption requires that the deviation in the path be minimal at every time instant for a point. Path coherence alone may lead to wrong trajectories near the points of intersection of trajectories. The smoothness of motion comes to rescue in such situations; which is presented in the form of the fourth assumption above.

In order to represent the correct correspondence of points on a trajectory, we define an n -dimensional array, C , to represent m valid trajectories. This array will have

$$C(i_1, i_2, \dots, i_n) = 1 \quad (5)$$

for a valid i^{th} trajectory. All other entries will be zero. Thus for a set of trajectories under consideration, m out of m^n entries in the array C will be 1, while the rest will

be 0. In Figure 3 we show points on 4 trajectories in 5 frames. For the trajectories shown in the figure, only the following elements of the C array will have non-zero values:

$$C(1,2,3,4,4)=1$$

$$C(2,1,1,2,3)=1$$

$$C(3,4,4,3,1)=1$$

$$C(4,3,2,1,2)=1$$

Now, we can formulate the trajectory finding method as an optimization problem. The total number of all possible trajectories is m^n , however only m of those are valid. One may consider exhaustively all possible combinations of trajectories to find the correct solution. The exhaustive approach will be computationally very expensive, however. Before we consider the computational approach, let us consider the solution of the problem. According to the hypothesis of the smoothness of motion, we want to select those trajectories from the set of all possible trajectories that maximize the smoothness of the trajectories. Thus we want to select that set which minimizes the function $D(T_1, T_2, \dots, T_m)$, given by

$$D(T_1, T_2, \dots, T_m) = \sum_{x_1 \in X^1} \dots \sum_{x_n \in X^n} \left(\sum_{t=2}^{n-1} \Psi(\overline{X_{t-1} X_t X_t X_{t+1}}) \right) * C(i_1, i_2, \dots, i_n) \quad (6)$$

Subject to the conditions

$$\sum_{x_1 \in X^1} C(i_1, i_2, \dots, i_n) = 1 \quad (7)$$

$$\sum_{X_{i_1} \in X^n} C(i_1, i_2, \dots, i_n) = 1$$

and

$$C(i_1, i_2, \dots, i_n) \geq 0 \text{ for } X_{i_1} \in X^1, \dots, X_{i_n} \in X^n. \quad (8)$$

The set of trajectories, T_1, T_2, \dots, T_m , that minimizes this function should be interpreted as the correct set describing the motion of m points in n frames.

6. Optimization Algorithm

As pointed out in an earlier section, an exhaustive solution of the optimization problem to establish the correspondence is computationally a difficult problem. Although established optimization methods like branch and bound and dynamic programming can be applied to solve our problem; they are inadequate for two reasons. Firstly, the computational requirements of these optimization methods are still quite high and they do not exploit the redundancy of information available in dynamic scenes. Moreover, in our problem, it will be desirable to have an incremental algorithm that assigns points in the $(k+1)^{\text{th}}$ frame to trajectories that have already been obtained up to the k^{th} frame.

We use an iterative optimization algorithm to assign points to proper trajectories. Suppose that initial trajectories are obtained using some criterion, such as closeness alone. Points from a new frame may be tentatively assigned to trajectories using a weak criterion, and then the resulting trajectories can be refined using a stepwise optimization procedure to minimize the overall deviation of the trajectories while establishing correspondence. Like hill-climbing procedures in general, stepwise optimization

procedures guarantee only local optimization. Also, in these procedures, the initial correspondence plays an important role in the final solution. The solution obtained is strongly influenced by the starting point. With these general limitations of stepwise approaches in mind, we propose an algorithm in the following.

6.1. Algorithm

The algorithm proposed here is called *Greedy Exchange Algorithm*. This algorithm will extend trajectories upto $(k + 1)^{th}$ frame, assuming that the trajectories up to the k^{th} frame have already been obtained. The points of $(k+1)^{th}$ frame are assigned to the established trajectories using the nearest neighbors. These tentative trajectories are then iteratively refined using the following method. The value of the criterion D for the tentative trajectories for the $(k+1)^{th}$ frame is

$$D = \sum_{p=1}^m D_p = \sum_{p=1}^m \sum_{q=2}^k d_p^q \quad (9)$$

Now let us consider two trajectories, the i^{th} and the j^{th} , and rewrite the above equation as

$$\begin{aligned} D &= \sum_{p=1; p \neq i, j}^m D_p + \sum_{q=2}^k d_i^q + \sum_{q=2}^k d_j^q \\ &= \sum_{p=1; p \neq i, j}^m D_p + \sum_{q=2}^{k-1} d_i^q + \sum_{q=2}^{k-1} d_j^q + d_i^k + d_j^k \end{aligned} \quad (10)$$

Suppose now we exchange the points from the $(k+1)^{th}$ frame on i^{th} and j^{th} trajectories. Clearly, such an exchange would not affect the first three terms in the above equation. Now, let \underline{d}_i^k and \underline{d}_j^k be the new path coherence measures for the i^{th} and j^{th} trajectories after the exchange. Then, for the above exchange to be profitable, we must have

$$\underline{d}_i^k + \underline{d}_j^k < d_i^k + d_j^k \quad (11)$$

or

$$g_{i,j}^k \equiv d_i^k + d_j^k - (\underline{d}_i^k + \underline{d}_j^k) \quad (12)$$

should be positive.

If we are to select only one such exchange from all possible exchanges, then we should make a decision in favor of the exchange maximizing the gain, i. e. exchange i and j if

$$g_{i,j}^k \geq g_{rs}^k \quad (13)$$

for all possible values of i, j, r , and s . This idea leads to the following greedy exchange algorithm.

1. **Initialization** : For each point in $X^k, k = 1, 2, \dots, n-1$, determine the nearest neighbor in X^{k+1} . Using these nearest neighbors, initialize m trajectories, such that the point X_{i+1}^k of the i^{th} trajectory is the nearest neighbor of X_i^k . In case of multiple nearest neighbors, the decisions are arbitrary and a point in any frame is assigned to only one trajectory.

2. **Exchange loop** : For each frame value $k = 2$ to $n-1$;

a. Compute the gain $g_{i,j}^k$ for $i = 1$ to $m-1$ and $j = i+1$ to m

b. Pick the $i-j$ pair providing the maximum gain.

c. Exchange the points in the $(k+1)^{\text{th}}$ frame on the T_i and T_j trajectories and set the exchange flag on.

3. **Termination** : Check the exchange flag at the end of Exchange loop. If an exchange was made during a frame, go back to exchange loop; otherwise

stop.

Clearly the complexity of the algorithm is of $O(nm^2)$. If we chose our path coherence measure such that the D cannot become negative, then the above procedure will terminate after a finite number of iterations.

7. Path Coherence Function

In selecting a function for path coherence, we used four guiding principles:

1. The function should not be negative.
2. The function should consider the amount of deviation in the direction of motion, not the sense (left or right) of the deviation. Thus, the sign of the angle of deviation should not factor in the computation of the function.
3. The function should respond equally to increases and decreases in speed.
4. If there is no change in the motion characteristics, then the function should be zero.

These four conditions help us in selecting a suitable function to represent path coherence at a point. We used the following function in our experiments:

$$\Psi(X_{i-1}, X_i, X_{i+1}) = w_1 \left[1 - \frac{\overline{X_{i-1}X_i} \cdot \overline{X_iX_{i+1}}}{||\overline{X_{i-1}X_i}|| \cdot ||\overline{X_iX_{i+1}}||} \right] \quad (14)$$

$$+ w_2 \left[1 - \frac{2 \left[||\overline{X_{i-1}X_i}|| \cdot ||\overline{X_iX_{i+1}}|| \right]^{1/2}}{||\overline{X_{i-1}X_i}|| + ||\overline{X_iX_{i+1}}||} \right]$$

where w_1 and w_2 are weights.

As can be easily verified, this function satisfies above four conditions and takes into account, both, the direction and the magnitude of changes in the motion. In fact the first term in the above expression can be considered *directional coherence* and the second term *speed coherence*. Note that the first term is the dot product of the displacement vectors, and the second considers geometric and arithmetic mean of the magnitude of these vectors. The weights w_1, w_2 are experimentally selected to be 0.1 and 0.9, respectively. It should be pointed out here that one may consider acceleration also in the path coherence. This will require extending the notion of the path coherence to more than three frames so that acceleration may be computed. In our current study, no effort was made to account for acceleration.

8. Experiments

We studied the efficacy of the proposed approach considering several synthetic and real scenes. In the synthetic data, we considered assorted motions of points in several frames and added random noise to the position of points. In all cases, excellent results were obtained. Here we present results of two synthetic sequences to show how our algorithm works.

In the first experiment, we generated 4 points in space. These points were undergoing different rotational motions. The proposed algorithm tracked the images of the points and found the trajectories shown in Figure 4.

In the next experiment with the synthetic data, we considered four points in ten frames. A composite frame showing location of points in all frames is shown in Figure 5. All points are moving left to right. As can be seen from the figure, the trajectories of points cross between fourth and sixth frames. During this period the location of points are such that a correspondence scheme based only on the location of points is

likely to give wrong results. We applied our iterative algorithm to this data, using the path coherence function discussed above. The initialization phase resulted in the trajectories shown in Figure 6a. Note that the initialization phase considers only the location of points from frame to frame, and hence it is not surprising that the resulting trajectories are wrong. In fact, these trajectories are completely counter intuitive, when one considers the data. In the first iteration, four corrections are made. The trajectories after the first correction and after the first iteration are shown in Figures 6b and 6c respectively. Note that the first correction removes the abrupt change in direction due to wrong correspondence of fifth frame points on second and fourth trajectories (trajectories are numbered from top to bottom). After the first iteration, the trajectories are improved, but are far from being correct. In fact, not even one trajectory is correct. The trajectories after 2, 3, and 4 iterations are shown in Figures 6d, e, and f, respectively. As can be seen, after each iteration, the trajectories are refined. The fourth iteration results in correct trajectories and the algorithm terminates.

We generated a real sequence in our laboratory to study the efficacy of our algorithm in real scenes. In the scene shown in Figure 7, three blocks are moving such that their trajectories intersect. We manually obtained interesting points on these blocks. We selected several points on each block and applied our algorithm. In order to show the behavior of our algorithm, without unnecessary complications, we show results for the four points on 3 blocks. The initial trajectories are shown in Figure 8a. The first iteration requires four corrections, the trajectories after each correction are shown in Figures 8b, c, d, and e, respectively. The second iteration was enough to give the correct results, resulting in the trajectories shown in Figure 8f. The trajectories for all interesting points were successfully determined by our algorithm and are shown, for six frames, in Figure 9. It required 8 iterations for the convergence to final trajectories.

We applied this algorithm to a real sequence from a movie (Superman). The frames of the movie are shown in Figure 10. We tracked points on the belt and head of three soldiers. These points are shown in Table 1. Our algorithm converged after 4 iterations giving trajectories shown in Figure 11. Though soldiers are moving in different directions, the motion is not really very complex and this allowed the algorithm to converge very fast for this case.

9. Modified Greedy Exchange Algorithm

A close look at the GE algorithm reveals that the correspondence for the first two frames is never altered as the exchange loop operates from the second frame onwards. Thus for the above algorithm to provide correct correspondence, it is essential that the correspondence for the first two frames be correct to start with. Since the initial correspondence is purely based on the closeness of feature points in successive frames, the GE algorithm was found to give poor results for the case where the displacement of the objects was comparable to the object size. Figure 12 shows a sequence generated in our laboratory to study the efficacy of this approach. The objects and their motion were selected to confuse the algorithm. Figure 13 (a) show the trajectories obtained by the GE algorithm for one such laboratory generated sequence. Clearly the source of error in this case is the initial correspondence between the first two frames which is never altered. In order to remove this kind of error source, we modified the GE algorithm. The Modified Greedy Exchange Algorithm (MGE) essentially allows the exchange loop to operate in both directions, i.e. forward and backward. By changing the termination criterion to have a cascade of exchange free loops in either direction, it becomes possible to alter the correspondence in any frame. The steps of the MGE

algorithm are given below.

9.1. Modified GE Algorithm

1. Initialization : Set the forward and backward iteration flags on. for each point in X^k , $k = 1, 2, \dots, n-1$, determine the nearest neighbor in X^{k+1} using these nearest neighbors, initialize m trajectories.
2. Forward Exchange Loop : For each frame value $k = 2$ to $n-1$;
 - a. Compute the gain g_{ij}^k for $i = 1$ to $m-1$ and $j = i+1$ to m .
 - b. Pick the i - j pair providing the maximum gain.
 - c. Exchange the points in the $(k+1)^{th}$ frame on the T_i and T_j trajectories and set the exchange flag on.
 - d. Check the exchange flag at the end of loop. If any exchange was made during the entire pass, set the backward iteration flag on and go back to the beginning of forward exchange loop. Otherwise clear the forward iteration flag and go to the termination check step.
3. Backward Exchange Loop : For each frame value $k = n-1$ to 2 ;
 - a. Go through the exchange loop similar to above with exchanges now being made in the $(k-1)^{th}$ frame.
 - b. Check the exchange flag at the end of loop. If any exchange was made then set the forward iteration flag on and go back to the beginning of loop. Otherwise clear the backward iteration flag and go to the termination check step.
4. Termination check : Check the forward and backward iteration flags. If both are off, then stop; Otherwise go to the (forward/backward exchange) loop for which the corresponding flag is on.

Figure 13(a) shows the trajectories which were obtained after the iterations ended in the forward exchange loop. Figure 13(b) shows the trajectories which were available at the end of iterations in the backward exchange loop. The annoying correspondence of first two frames have been clearly changed. The final and correct trajectories are shown in figure 13(c) which required 22 iterations in the forward direction, 10 iterations in the backward direction, again 8 iterations in the forward direction, and one exchange free iteration in the backward direction.

10. Occlusion

When working with a large sequence of frames, it is possible that some objects may disappear totally or partially. Similarly some new objects may appear in the frame sequence from some intermediate frame onwards. In order to handle the problem of occlusion in an effective way, it is necessary that the system should have much more information than just the number and the location of feature points. For example, consider the case of two rectangles moving linearly in different planes parallel to the optical axis of camera. Suppose the positions of these two rectangles are as shown in Figure 14 (a) in one frame and in the next frame some of the different possibilities are as shown in Figure 14 (b-d) .

For all the cases of the next frame shown above, a corner detector will detect 8 feature points and there is no way of telling whether an occlusion has occurred or not by simply looking at the number of feature points. In fact until and unless some assumption about the rigidity of the feature point configurations are made or the object structural information is available, it is difficult to say that an occlusion has occurred. Thus it is our contention that for any method of correspondence analysis to be successful with respect to occlusion, much more information than just the number

and location of feature points is needed.

Does this imply that the problem of occlusion can not be taken care of when the only information available is the set of moving points? The answer is 'no' as evident by some psychological studies [ShR82, RaA83]. It has been shown that the human visual system is capable of filling-in the missing points by relying on the integration of information over several frames. Shaw and Ramchandran [ShR82] have determined that for the filling-in to take place, it is necessary that at least two frames on either side of the frame with missing points be available. The availability of at least two frames on either side possibly allows the hypothesize-and-test situation where the missing point position is interpolated using the previous two frames and the interpolated value is tested against the subsequent two frames.

With respect to the concepts of path coherence and smoothness of motion, the findings of Shaw and Ramchandran are not surprising in the sense that with five frames, the interpolated value of the feature point appears in all the three local path coherence measures which are integrated to verify the filling-in by checking the overall smoothness of the motion. Thus within the framework of feature point location alone, it is possible to hypothesize and test the occurrence of occlusion with the constraint that any change in the number of feature points signifies occlusion. We have found the following hypothesize and test occlusion algorithm useful in several experiments. The algorithm is discussed with respect to occlusion but can be modified easily to take care of disocclusion.

10.1. Hypothesize and Test Occlusion Algorithm

Assumption: If the number of feature points is same then it implies no occlusion.

There is only one frame in which some points disappear. For simplicity, we assume

that at least on one side of the frame with missing points, there are three frames available.

1. Determine the frame with some missing points. Let it be the p^{th} frame and the number of points in this frame be $m_p (< m)$. Let A represent this set of points with individual points denoted as $A_j, j = 1, 2, \dots, m_p$.
2. Establish the correspondence using MGE algorithm up to $(p-1)$ frames and from $(p+1)^{\text{th}}$ frame to n^{th} frame. As per the stated assumption, it should be possible to obtain correspondence at least on one side, say up to $(p-1)^{\text{th}}$ frame.
3. For each of the m trajectories up to $(p-1)^{\text{th}}$ frame obtained through correspondence, predict the trajectory points for the p^{th} frame using forward interpolation. Let this set of predicted points be denoted by P and let P_i represent the predicted point for the i^{th} trajectory.
4. For each $A_j, j = 1$ to m_p , determine the ordered list of nearest neighbors from the set P of predicted points.
5. From the ordered list of nearest neighbors for every A_j , construct the sets N_1, N_2, \dots, N_m respectively as the set of the first nearest neighbors, the set of up to second nearest neighbor, and ..., set of up to $(m-1)^{\text{th}}$ nearest neighbors. Thus if $P_i \in N_j$, then P_i is at least j^{th} nearest neighbor of one of the points from set A .
6. Count the number of distinct points from the sets N_j 's for $j = 1$ to m . Stop at the set when the count value is same as m_p , say at the j^{th} set. The remaining $(m - m_p)$ predicted points not in the j^{th} set are then selected as candidate points for filling-in. However, if no N_j with count value is m_p , then determine the set N_j with count values such that $\text{count}(N_j) < m_p < \text{count}(N_{j+1})$ predicted points as candidate points for filling-in.

7. Complete the correspondence for n frames by using in the p -th frame the given m , points A and the candidate points from step 6.
8. For the case of multiple candidate sets for filling-in, retain that candidate set which yields the best filling-in measured in terms of D .

We applied the above algorithm to the frame sequence of Figure 12. Points on the 3rd and 4th trajectory from 5th frame were suppressed thus creating occlusion. The correspondence up to the 4th frame was then determined and the points for the 5th frame were then predicted. Table 2 shows the location of 10 actual points from frames and 12 predicted points for frame 5. Table 3 gives the ordered set of nearest neighbors while the set N_1 , N_2 and N_3 are shown in Table 4. The candidate points for filling-in are then any two points from P_4 , P_{11} and P_{12} . Figure 15 (a-c) show the trajectories obtained by using P_4 - P_{12} , P_4 - P_{11} , and P_{11} - P_{12} as filling-in points, respectively.

Comparing these trajectories with the actual trajectory of Fig 15 (d), one important observation can be made and that is - irrespective of the candidate points for filling in, the overall perception of the trajectories in limited occlusion is correct.

11. Conclusion

In this paper we show that by considering an extended frame sequence, one can use coherence in spatio-temporal properties for establishing correspondence to obtain trajectories of objects. Quasi-dynamic approaches for the analysis of motion try to solve all problems in *minimum* number of frames. Obviously the limited amount of data in temporal domain necessitates some assumptions on the nature of objects. More importantly, such approaches require methods that are complex and sensitive to noise.

By considering *dynamic scenes*, we can use techniques that can successively refine the information recovered from the sequence. Such methods are much less sensitive to noise and can be usually performed very fast. The smoothness of motion allows to establish correspondence in case of occlusion of points also, in a limited situation. This is very encouraging because in most approaches to occlusion analysis one requires information about features also.

In this paper, we considered only the correspondence problem. After the trajectories are established, one may try to recover the structure of objects. As shown in [Tod82, SeJ83], one may use parameters of curves representing trajectories to obtain the motion characteristics and structure of objects. It seems that one may use successive refinement in the recovery of the structure also.

The proposed approach assumes that the motion is smooth. Indeed, in many applications, one is required to detect points of discontinuities in motion. The discontinuities in motion play important role in the analysis and verbalization of the motion of objects. We are studying techniques for detecting discontinuities in motion characteristics and their use in describing motion of objects [HaJ84].

It should be mentioned here that in our algorithm, we did not use any characteristics of points. The trajectories were determined using only locations of points in frames. The robustness and computations can be improved by considering pictorial characteristics of points in frames. This additional information may play a very important role in the initialization phase of the algorithm by eliminating unlikely grouping of points, even if they are close to each other.

Acknowledgement

We are thankful to Paul Besl and Susan Haynes for many useful discussions and criticisms during the course of this work.

12. REFERENCES

- [BaT79]
Barnard, S.T. and W.B. Thompson, "Disparity analysis of images," *IEEE Trans. on PAMI*, vol. PAMI-2, 1980, pp. 333-340.
- [Bra83]
Brady, M., "Parallelism in Vision," *Artificial Intelligence*, Vol. 21, 1983, pp.271-283.
- [DLP84]
Donner, J., J.S. Lappin, and G Perfetto, "Detection of three-dimensional structure in moving optical patterns", *Jour. of Exp. Psychology: Human Perception and Performance*, vpl 10. No. 1, pp. 1, 1984.
- [FIC81]
Flinchbaugh, B. E. and B. Chandrasekaran, "A theory of spatio-temporal aggregation for vision", *Artificial Intelligence*, vol. 17, pp. 387-408, 1981.
- [Gib79]
Gibson, J. J., *The ecological approach to visual perception*, Houghton Mifflin, Boston, 1979
- [HaJ84]
Haynes, S. M. and R. Jain, "Low level motion event: Trajectory discontinuities" *Proc. First Conf. on AI Applications*, pp. 251-256, Dec. 1984
- [HoF80]
Hoffman, D.D. and B. E. Flinchbaugh, "The interpretation of biological motion", MIT AI Memo. 608, 1980.
- [HoS81]
Horn, B. K. P. and B. G. Schunck, "Determining optical flow," *Artificial Intelligence* vol. 17, 1981, pp.185-203.
- [Jai83]
Jain, R., "Direct computation of the focus of expansion," *IEEE Trans. on PAMI*, vol. PAMI-5, pp.58-64, 1983.

[JaH82]

Jain, R. and S. Haynes, "Imprecision in computer vision", *IEEE Computer*, Aug. 1982.

[JaN79]

Jain, R. and H.-H. Nagel, "On the analysis of difference and accumulative difference pictures in dynamic scene analysis," *IEEE Trans. PAMI*, vol 1, No. 2, pp.206-214, 1979.

[JeJ83]

Jerian, C. P. and R. Jain, "Determining motion parameters for scenes with translation and rotation," *Proc. Workshop on Motion: Representation and Control*, April 4-6, Toronto, 1983.

[Jen83]

Jensen, M., "Tracking three dimensional moving light displays" *Proc. Workshop on Motion: Representation and Control*, Toronto, pp. 66-70, 1983.

[Joh76]

Johansson, G., "Spatio-temporal differentiation and integration in visual motion perception," *Psych. Research*, vol. 38, pp.379-383, 1976.

[Mar76]

Marr, D., "Early processing of visual information", *Phil. Trans. R. Soc. London, B* 275, pp. 489-524, 1976.

[MuT85]

Mutch, K.M. and W.B. Thompson, "Analysis of accretion and deletion at boundaries in dynamic scenes", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol PAMI-7, pp.133-138, Mar. 1985.

[Nei76]

Neisser, U., *Cognition and Reality*, W.H. Freeman, San Francisco, 1976.

[OBJ84]

O'Brien, N. G. and R. Jain, "Axial motion stereo", *Proc. Workshop on Computer Vision*, Annapolis, 1984.

[ORB80]

O'Rourke, J. and N.I. Badler, "Model-Based Image Analysis of Human Motion Using Constraint Propagation," *IEEE Trans. PAMI*, Vol PAMI-2, pp. 522-536, 1980.

[Pra81]

Prazdny, K., "Egomotion and relative depth map from optical flow", *Biological Cybernetics*, vol. 36, pp. 87-102, 1980.

[PrA83]

Prager, J.M., and M. A. Arbib, "Computing the Optic Flow: The MATCH Algorithm and Prediction," *Computer Vision, Graphics, and Image Processing*, col. 24, 1983, pp. 271-304.

[RaA84]

Ramchandran, V. S. and S.M. Anstis, "Extrapolation of motion path in human visual perception", *Vision Research.*, vol. 23, pp.83-85, 1984.

[Ras80]

Rashid, R.F., "Towards a system for the interpretation of moving light displays", *IEEE Trans Pattern Analysis and Machine Intelligence*, vol. PAMI-2, pp. 574-581, 1980.

[RiL83]

Rieger, J.H. and D. T. Lawton, "Determining the instantaneous axis of translation from optic flow generated by arbitrary sensor motion", *Proc. Workshop on Motion*, Toronto, pp. 33-41, 1983.

[RoA80]

Roach, J.W. and J.K. Aggarwal, "Determining the movement of objects from a sequence of images," *IEEE Trans. on PAMI*, vol. PAMI-2, No.6, Nov. 1980, pp. 554-562.

[SeJ83]

Sethi, I.K. and R. Jain, "Determining three dimensional structure of rotating objects using a sequence of monocular views", *Technical Report*, Dept. of Computer Science, Wayne State University, Detroit, Mi 48202, 1983.

[ShR82]

Shaw, G. L. and V. S. Ramchandran, "Interpolation during apparent motion". *Perception*, vol. 11, pp.491-494, 1982.

[Tod82]

Todd, J. T., "Visual information about rigid and nonrigid motion: A geometric analysis", *J. Experimental Psychology: Human Perception and Performance*, vol. *, pp.238-252, 1982.

[TsH81]

Tsai, R. Y. and T. S. Huang, "Estimating three-dimensional motion parameters of a rigid planar patch", *Proc of PRIP*, pp. 94-97. 1981.

[TsH82]

Tsai, R. Y. and T. S. Huang, "Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces", *Proc. IEEE Conf. Pattern Recog. and Image Processing*, 1982, pp. 112-118.

[THZ82]

Tsai, R. Y., T. S. Huang, and W.L. Zhu, "Estimating three-dimensional motion parameters of a rigid planar patch. II: Singular value decomposition", *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. ASSP-30, pp.525-534.

[Ull79]

Ullman, S., *The interpretation of visual motion* Cambridge, Mass, MIT Press, 1979

[Wax84]

Waxman, A.M., "An image flow paradigm", *Proc. Workshop on Computer Vision*, Annapolis, 1984.

[WeA82]

Webb, J. A. and J. K. Aggarwal, "Structure from motion of rigid and jointed objects," *Artificial Intelligence*, vol. 19, pp.107-130, 1982.

[Wil80]

William, T.D., "Depth from motion in real world scenes", *IEEE Trans. PAMI*, vol. 2, pp.511-516, 1980.

Table 1
Points from the Superman Sequence

Frame	Head1 ; Belt-1	Head-2 ; Belt-2	Head-3 ; Belt-3
1	(73,306) ; (72, 261)	(147,305) ; (140, 240)	(295, 300) ; (290, 256)
2	(79, 302) ; (78, 261)	(154, 302) ; (147, 239)	(291, 296) ; (288, 254)
3	(86, 305) ; (83, 259)	(171, 301) ; (166, 234)	(278, 296) ; (276, 249)
4	(95, 312) ; (93, 261)	(189, 303) ; (185, 236)	(275, 304) ; (273, 253)
5	(95, 309) ; (95, 258)	(200, 303) ; (196, 238)	(266, 303) ; (263, 250)
6	(108, 307) ; (106, 256)	(229, 292) ; (222, 222)	(254, 303) ; (249, 244)
7	(120, 310) ; (119, 254)	(249, 300) ; (242, 228)	(251, 307) ; (243, 240)
8	(129, 313) ; (130, 252)	(256, 298) ; (257, 222)	(243, 307) ; (240, 236)
9	(138, 313) ; (137, 254)	(262, 301) ; (262, 222)	(250, 307) ; (242, 233)
10	(149, 307) ; (147, 245)	(268, 303) ; (271, 223)	(256, 305) ; (245, 229)

Table 2
Actual and Predicted position for frame 5

Trajectory number	Actual Point Position	Predicted Point Position.
1	(389, 195)	(388, 184)
2	(376 ,171)	(360, 159)
3	missing	(313, 249)
4	missing	(276, 239)
5	(289, 214)	(292, 225)
6	(311, 212)	(318, 214)
7	(296, 151)	(275, 150)
8	(317, 149)	(298, 133)
9	(237, 189)	(256, 225)
10	(273, 173)	(299, 227)
11	(304, 262)	(329, 288)
12	(323, 250)	(342, 221)

Table 3
Ordered Sets of Nearest Neighbors for A's from P's

A_1	$\{ P_1, P_2, P_{12}, P_6, P_3, P_{10}, P_5, P_8, P_{11}, P_4, P_7, P_9 \}$
A_2	$\{ P_1, P_2, P_{12}, P_6, P_8, P_{10}, P_5, P_3, P_7, P_4, P_{11}, P_9 \}$
A_5	$\{ P_5, P_{10}, P_4, P_6, P_9, P_3, P_{12}, P_7, P_8, P_{11}, P_2, P_1 \}$
A_6	$\{ P_5, P_{10}, P_5, P_{12}, P_3, P_4, P_9, P_7, P_2, P_{11}, P_1, P_8 \}$
A_7	$\{ P_8, P_7, P_2, P_6, P_5, P_{10}, P_{12}, P_9, P_4, P_1, P_3, P_{11} \}$
A_8	$\{ P_8, P_7, P_2, P_6, P_{12}, P_1, P_5, P_{10}, P_9, P_4, P_3, P_{11} \}$
A_9	$\{ P_9, P_7, P_4, P_5, P_{10}, P_8, P_6, P_3, P_{12}, P_2, P_{11}, P_1 \}$
A_{10}	$\{ P_7, P_8, P_9, P_5, P_{10}, P_6, P_4, P_{12}, P_3, P_2, P_1, P_{11} \}$
A_{11}	$\{ P_3, P_{10}, P_{11}, P_4, P_5, P_6, P_{12}, P_9, P_1, P_7, P_2, P_8 \}$
A_{12}	$\{ P_3, P_{10}, P_{12}, P_6, P_{11}, P_5, P_4, P_9, P_1, P_2, P_7, P_8 \}$

Table 4

N_1	$\{ P_1, P_3, P_5, P_6, P_7, P_8, P_9 \}$	Count = 7
N_2	$\{ P_1, P_2, P_3, P_5, P_6, P_7, P_8, P_9, P_{10} \}$	Count=9
N_3	$\{ P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10}, P_{11}, P_{12} \}$	Count=12

Figure 1 In this figure we show the trajectories of two points. We are concerned with the three points on the crossover of the trajectories. The points in the first, second, and third frames are , , and , respectively.

Figure 2 The correspondence established using 2-D distance and the smoothness of motion are shown using broken and full lines, respectively.

Figure 3 The points on 4 trajectories are shown in 5 frames. This figure is a composite frame showing locations of points in different frames. The points are numbered and the trajectories are shown by lines. The frame number is assumed increasing from left to right. The number shown next to a point indicates that in its frame it is the k^{th} point.

Figure 4 The trajectories obtained by the algorithm for rotation of points in a sequence.

Figure 5 A composite frame shows 4 points in 10 frames. The points move from left to right in this figure.

Figure 6 The trajectories of the points after several iterations and corrections are shown in this figure.

Figure 7 Three frames of a block scene. The blocks were manually moved and the interesting points were also manually obtained.

Figure 8 The trajectories of four points in the block scene. We considered ten frames.

Figure 9 Trajectories for all interesting points in six frames for the block scene.

Figure 10 Three frames from the Superman sequence. The points tracked by the algorithm are on the belt and head of the three soldiers that are running towards the camera.

Figure 11 The trajectories obtained by the algorithm after 4 iterations.

Figure 12 Three frames of a laboratory sequence. In this sequence the objects

are of uniform color and their shape is also similar. At a point in the sequence, points of one object are not visible due to occlusion.

Figure 13 The trajectories obtained by the modified algorithm. The trajectories obtained after the first forward pass, forward and backward passes, and forward, backward, and again forward passes are shown in a, b, and c, respectively.

Figure 14 A frame sequence in which there is occlusion, but the number of vertices remains same.

Figure 15 The trajectories obtained by the occlusion algorithm for the hypothesized candidate pairs P_4-P_{12} , P_4-P_{11} , $P_{11}-P_{12}$ are shown in a, b, and c, respectively. The trajectories in d are actual.

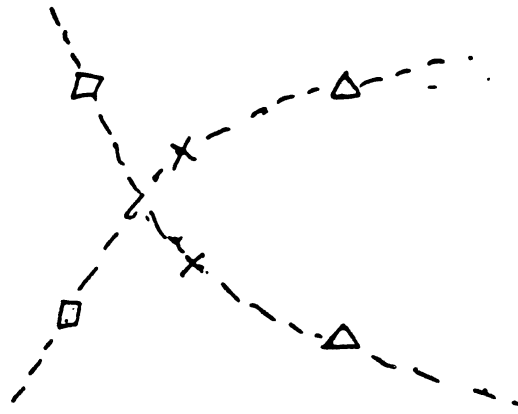


Figure 1

In this figure we show the trajectories of two points. We are concerned with the three points on the crossover of the trajectories. The points in the first, second, and third frames are \square , \times , and \triangle , respectively.

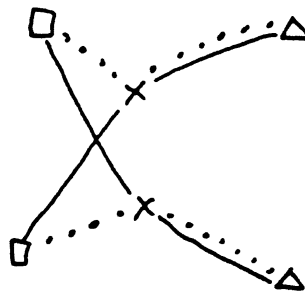


Figure 2

The correspondence established using 2-D distance and the smoothness of motion are shown using broken and full lines, respectively.

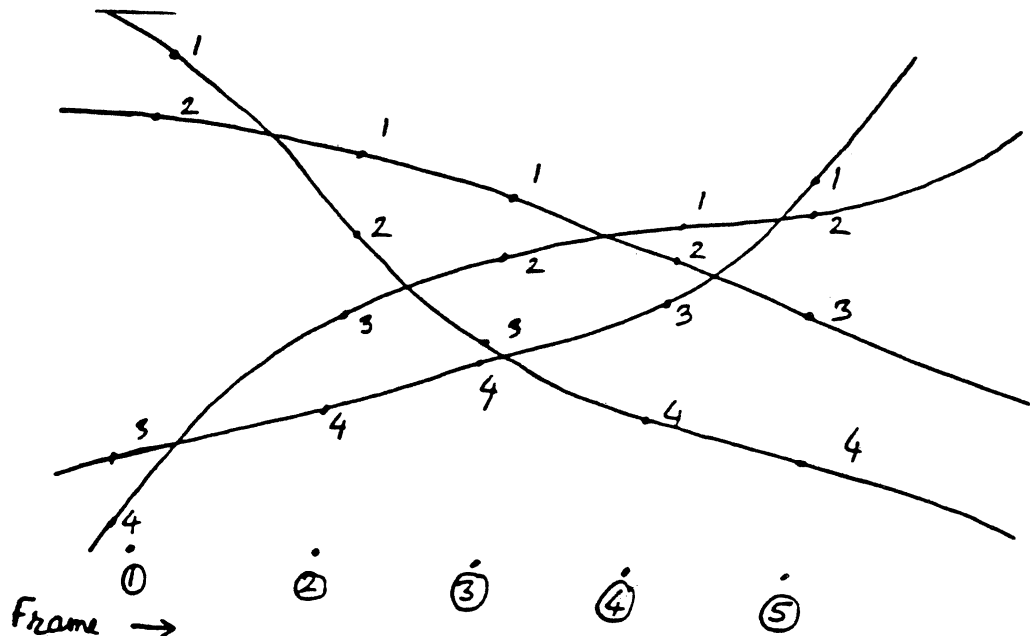


Figure 3 The points on 4 trajectories are shown in 5 frames. This figure is a composite frame showing locations of points in different frames. The points are numbered and the trajectories are shown by lines. The frame number is assumed increasing from left to right. The number shown next to a point indicates that in its frame it is the k^{th} point.

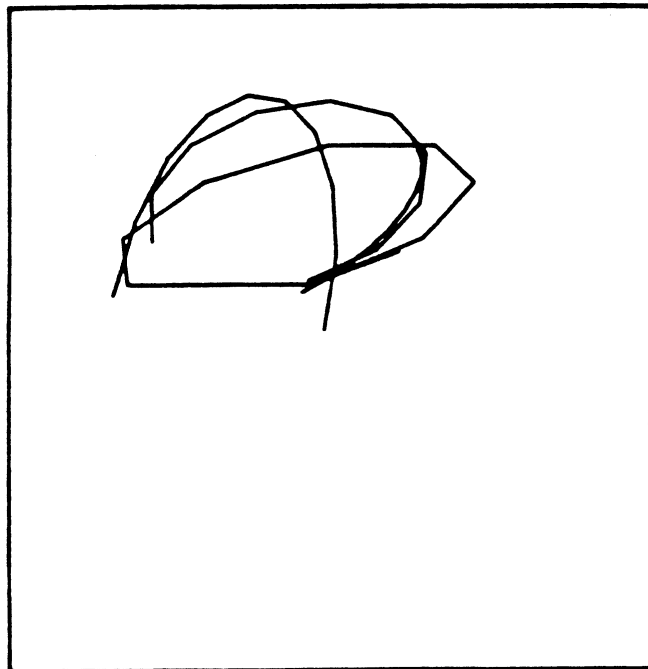


Figure 4 The trajectories obtained by the algorithm for rotation of points in a sequence.

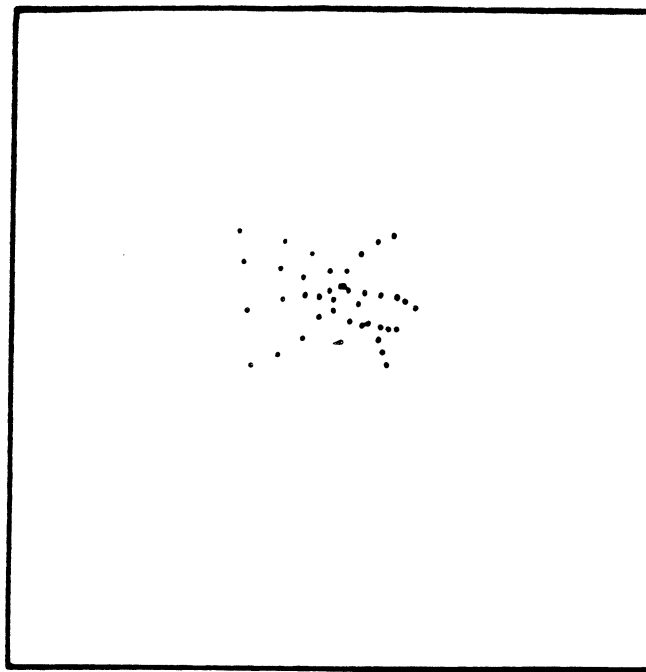


Figure 5 A composite frame shows 4 points in 10 frames. The points move from left to right in this figure.

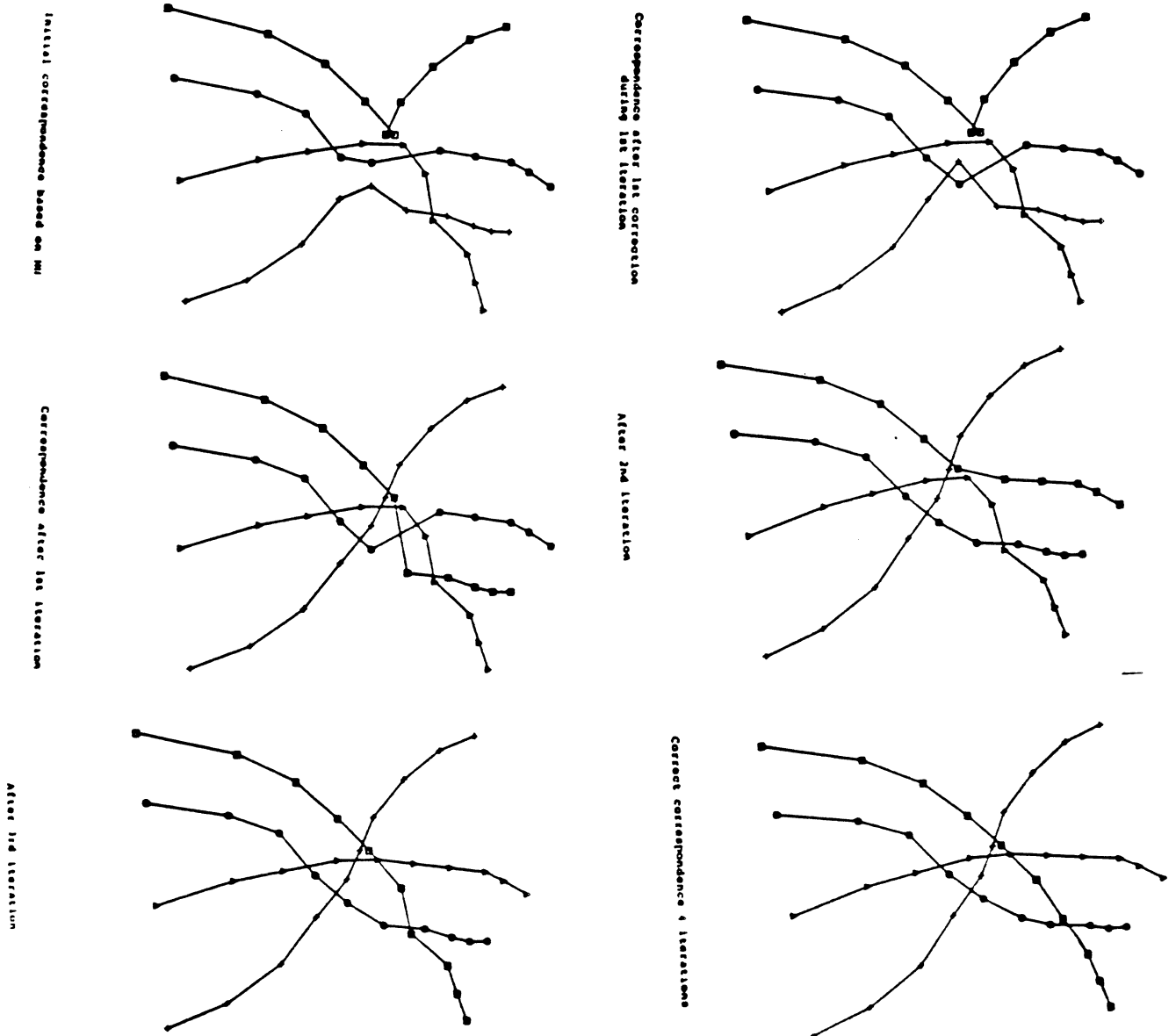


Figure 6 The trajectories of the points after several iterations and corrections are shown in this figure.

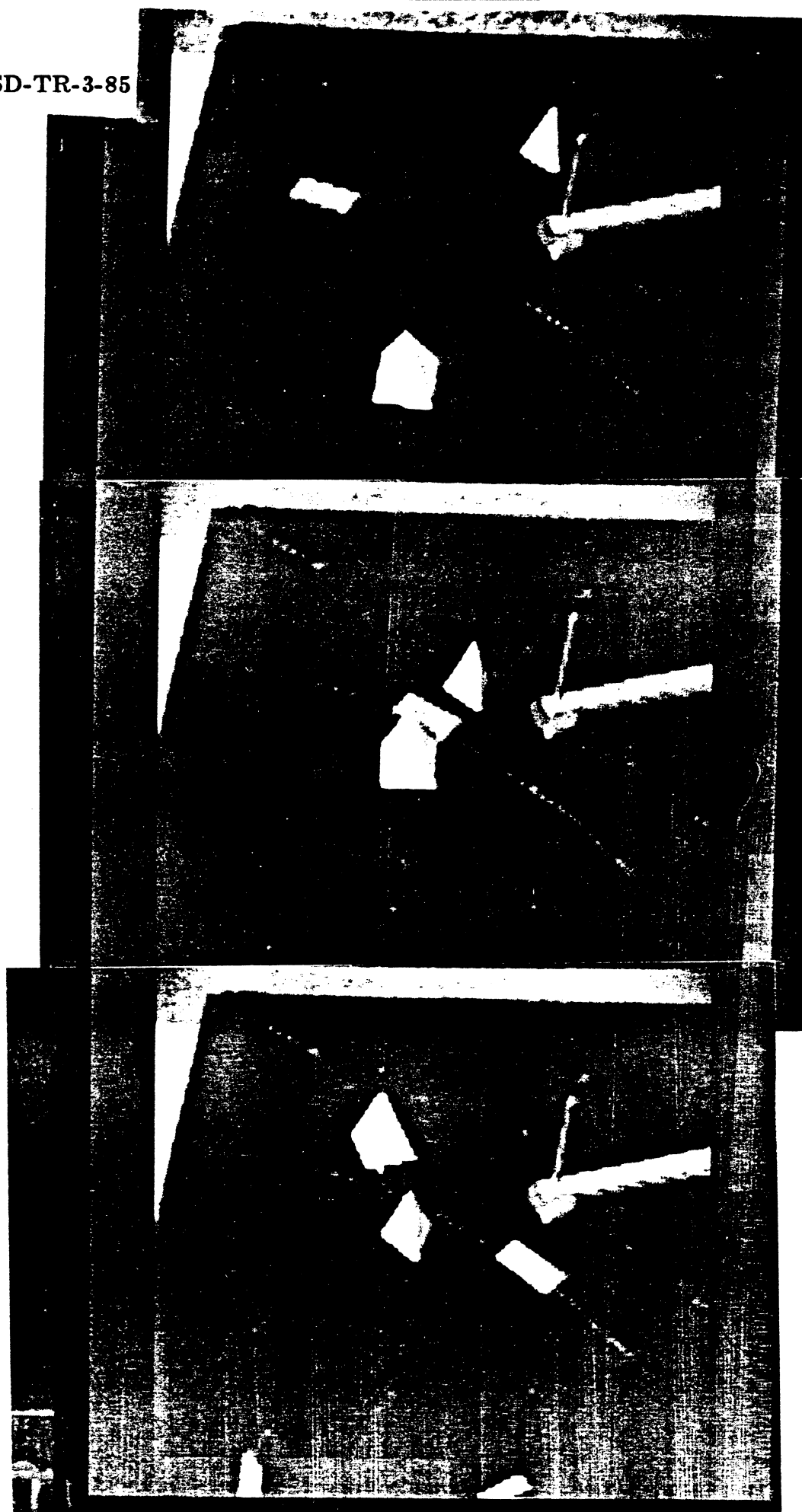
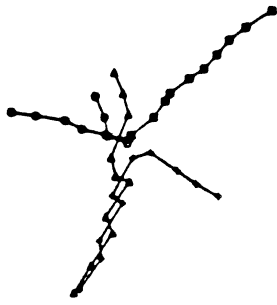
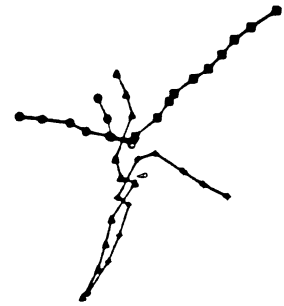


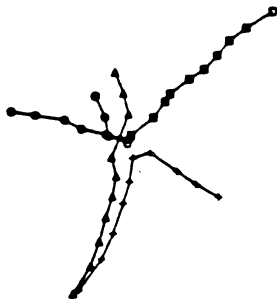
Figure 7 Three frames of a block scene. The blocks were manually moved and the interesting points were also manually obtained.



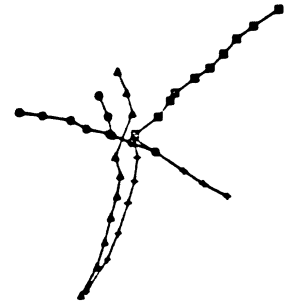
Initial correspondence (21)



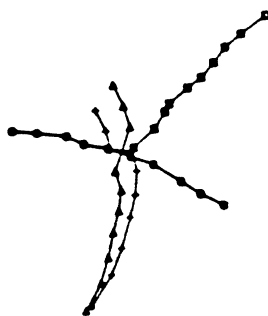
After first correction in 1st iteration



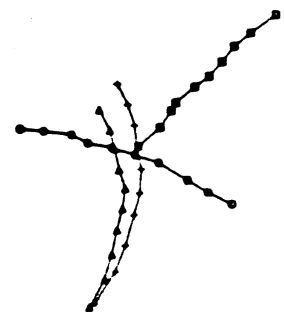
After 2nd correction in 1st iteration



After 3rd correction, 1st iteration



After 1st iteration



After 2nd iteration

Figure 8 The trajectories of four points in the block scene. We considered ten frames.

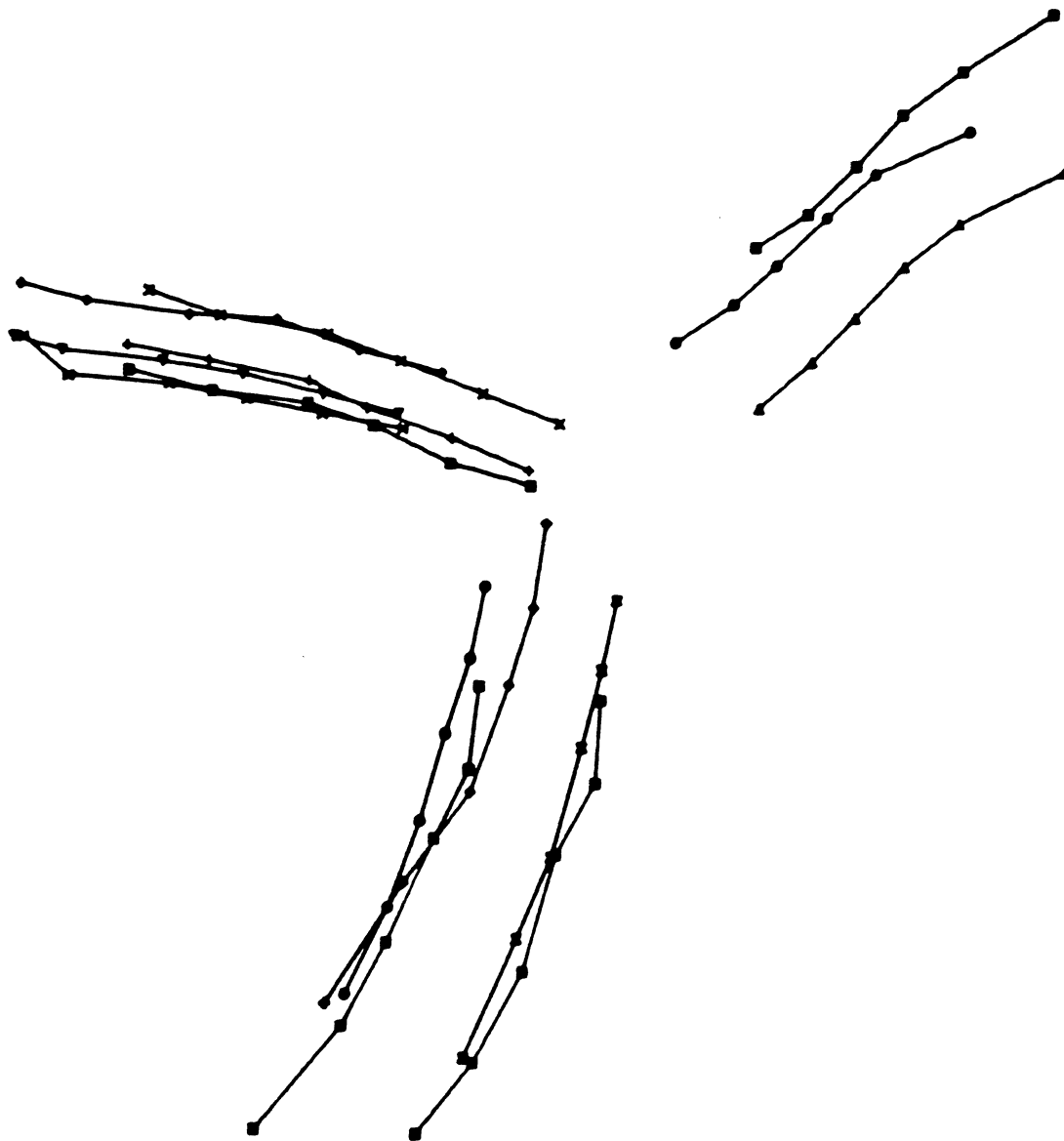


Figure 9 Trajectories for all interesting points in six frames for the block scene.

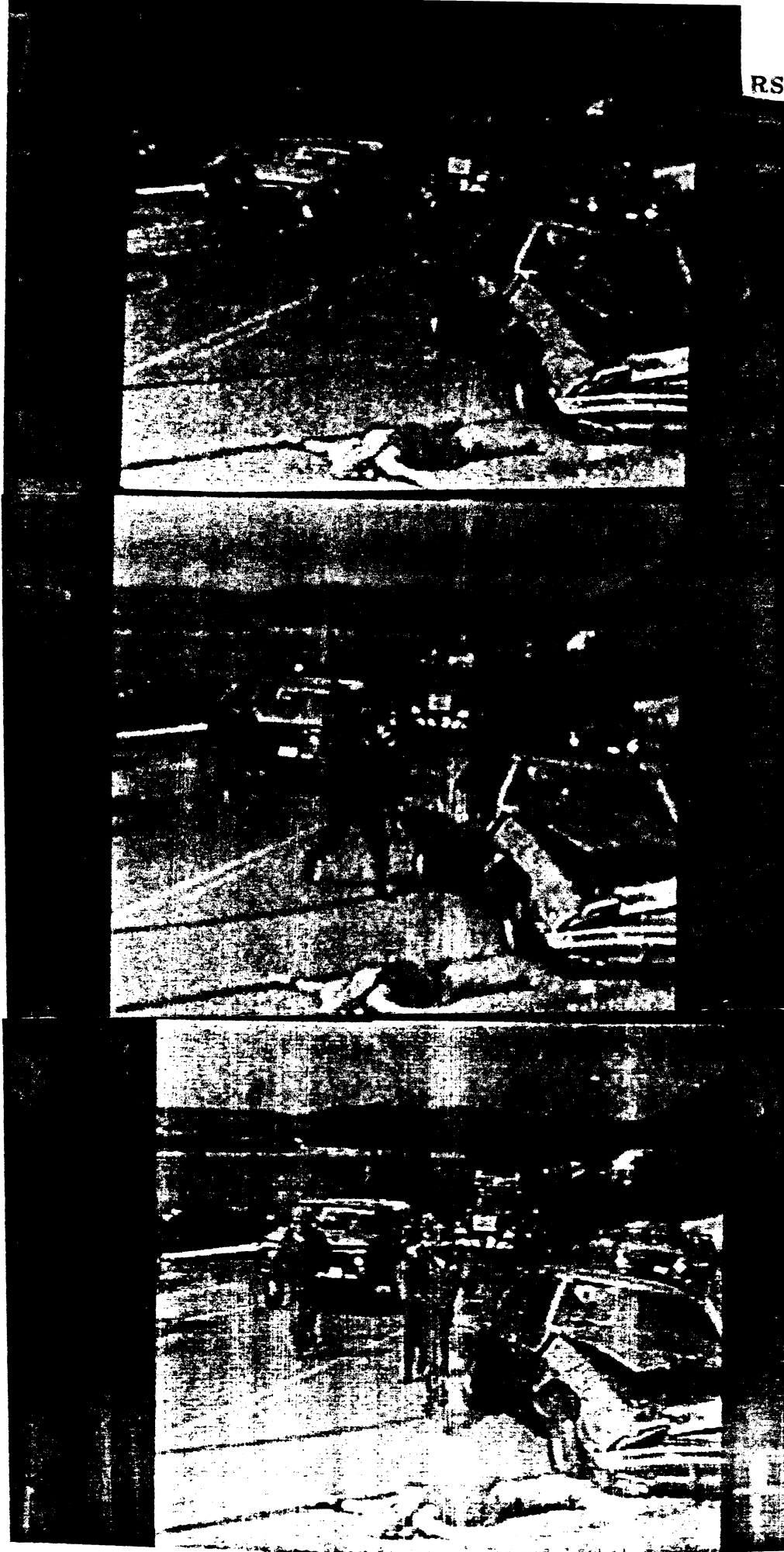


Figure 10 Three frames from the Superman sequence. The points tracked by the algorithm are on the belt and head of the three soldiers that are running towards the camera.

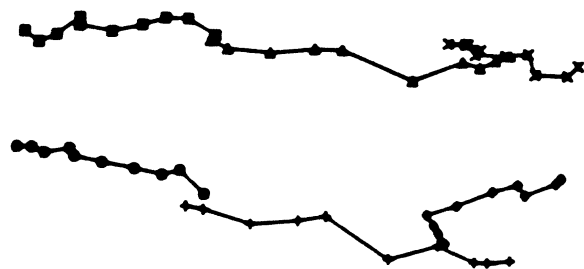
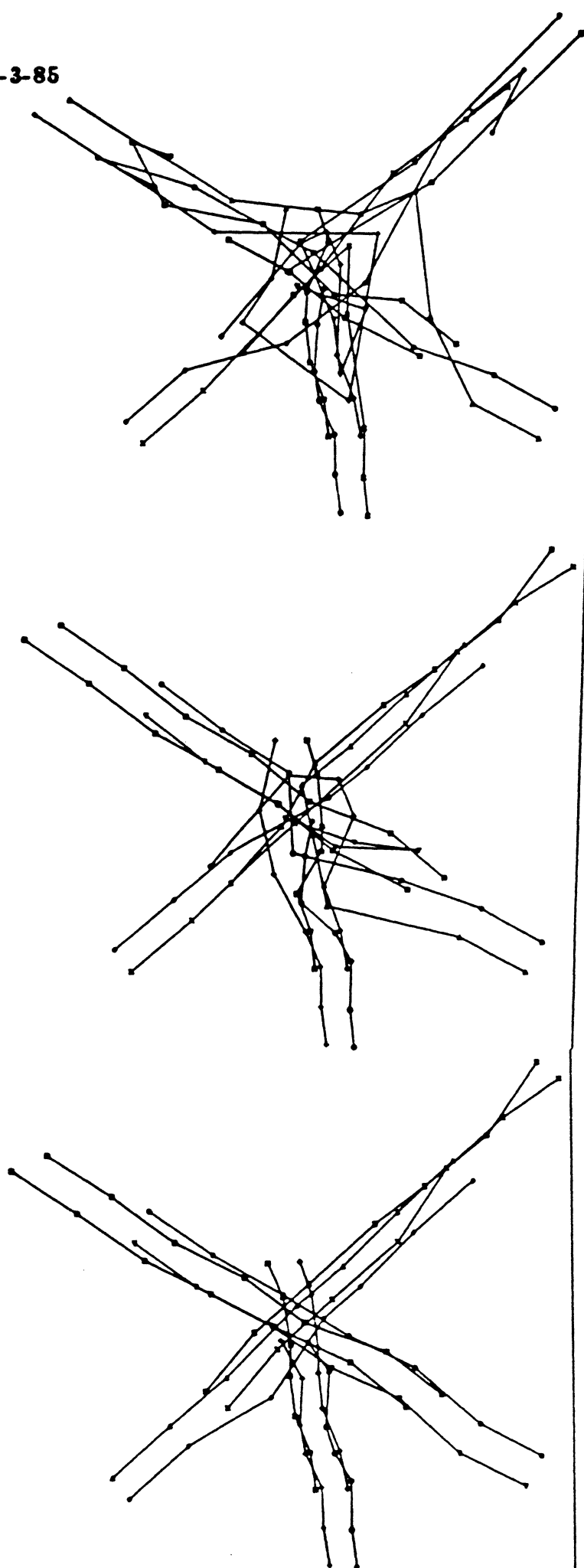


Figure 11 The trajectories obtained by the algorithm after 4 iterations.



Figure 12 Three frames of a laboratory sequence. In this sequence the objects are of uniform color and their shape is also similar. At a point in the sequence, points of one object are not visible due to occlusion. **46**



a

b

c

47 **Figure 13** The trajectories obtained by the modified algorithm. The trajectories obtained after the first forward pass, forward and backward passes, and forward, backward, and again forward passes are shown in a, b, and c, respectively.

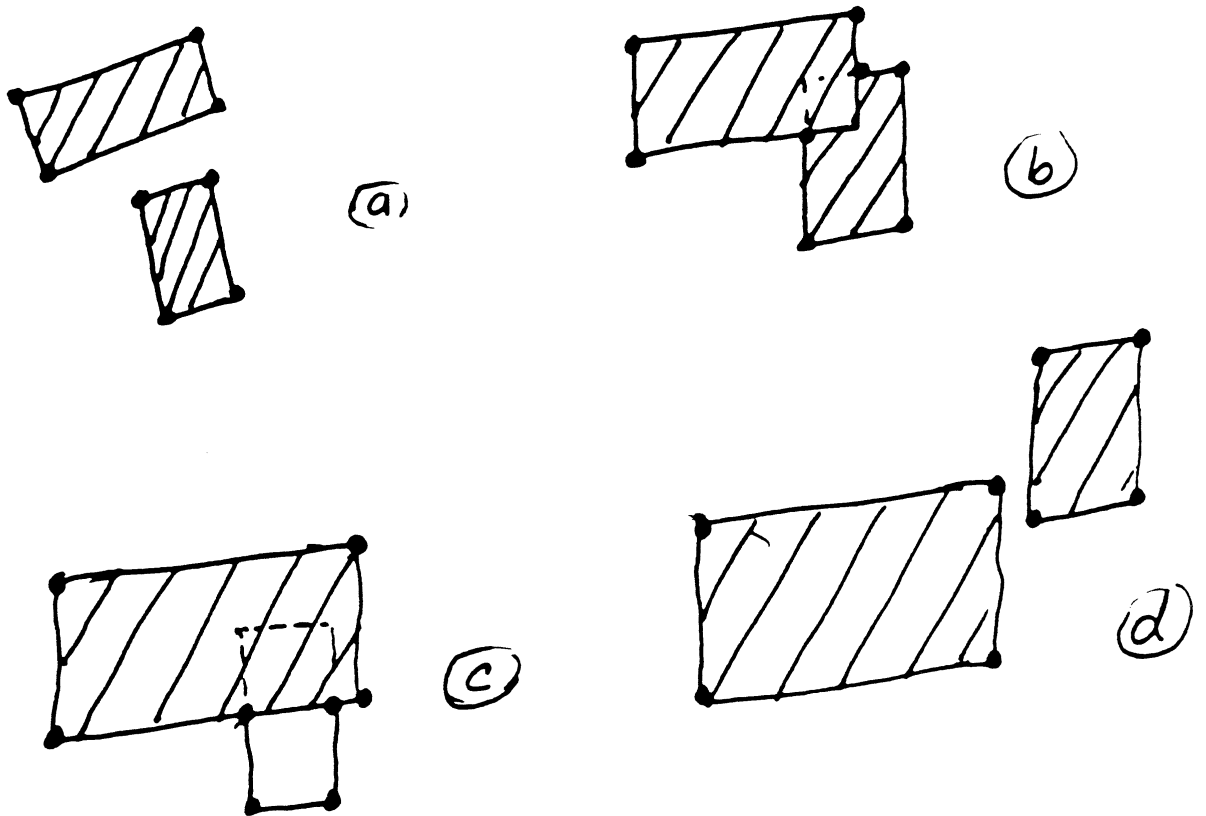


Figure 14 A frame sequence in which there is occlusion, but the number of vertices remains same.

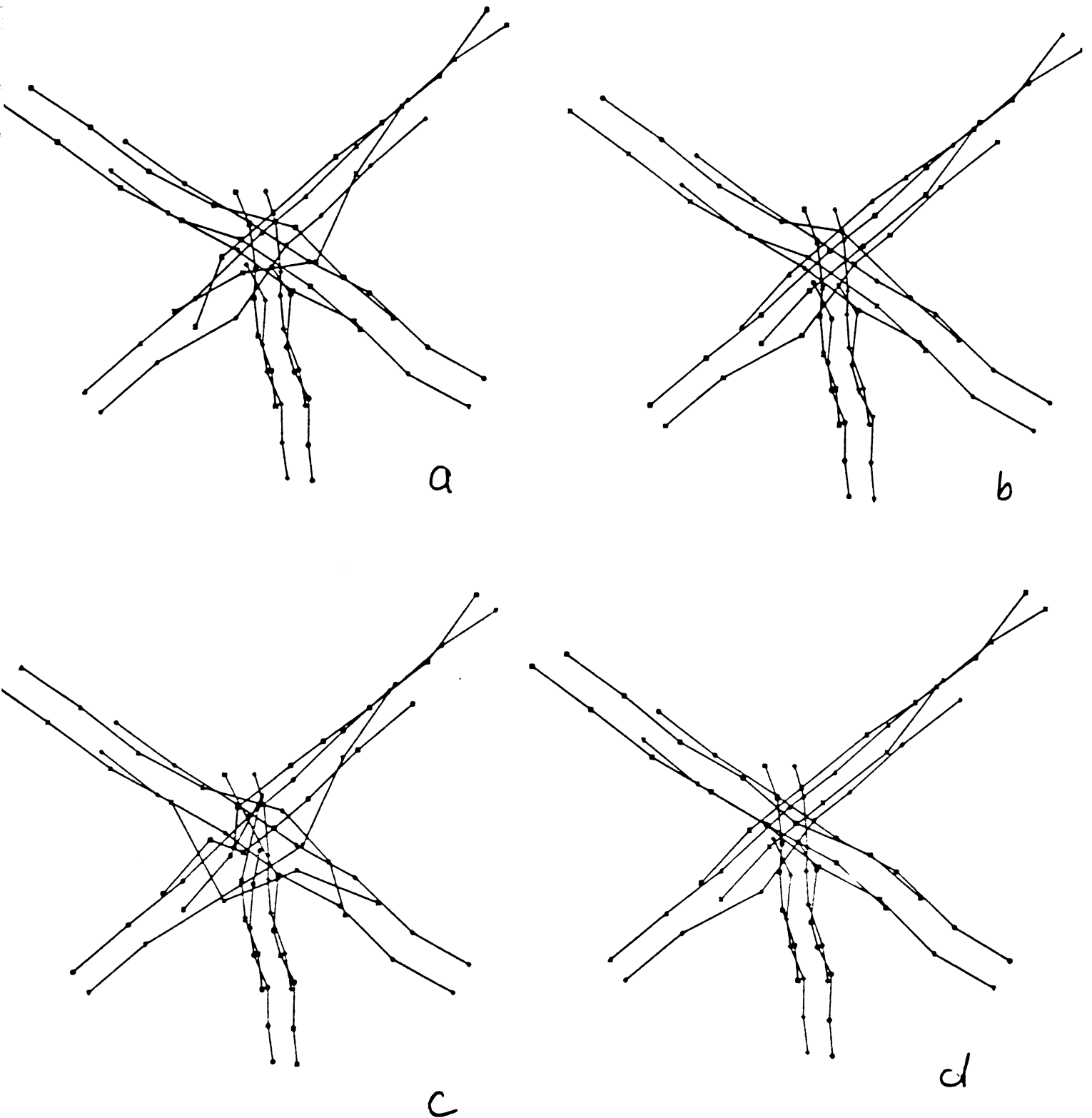


Figure 15 The trajectories obtained by the occlusion algorithm for the hypothesized candidate pairs P_4-P_{12} , P_4-P_{11} , and $P_{11}-P_{12}$ are shown in a, b, and c, respectively. The actual trajectories are given in d.

UNIVERSITY OF MICHIGAN



3 9015 03524 4303