CAR-TR-88
CS-TR-1437

DAAK70-83-K-0013
August 1984

TARGET TRACKING BASED SCENE ANALYSIS

Nader Kazor
Center for Automation Research
University of Maryland
College Park, MD 20742

COMPUTER VISION LABORATORY

CENTER FOR AUTOMATION RESEARCH

UNIVERSITY OF MARYLAND
COLLEGE PARK, MARYLAND
20742

CAR-TR-88
CS-TR-1437

DAAK70-83-K-0018
August 1984

TARGET TRACKING BASED SCENE ANALYSIS

Nader Kazor
Center for Automation Research
University of Maryland
College Park, MD 20742

ABSTRACT

Target Tracking and 3-D Scene Analysis are two research areas in Computer Vision which in the past have been considered separately. However, there are many advantages in combining the two problems. One such advantage would be the ability to analyze and build a model of a stationary scene/environment through which dynamic objects move. This is possible through tracking the moving objects and detecting instances of occlusion. This work is based on such an idea and is concerned with the design of an Intelligent Target Tracking System (ITTS) which combines the above two problems into one. In this paper we present an experimental ITTS which generates a perspective and ground map of a stationary environment.

The support of the Defence Advanced Research Projects Agency and the U.S. Army Night Vision Laboratory under Contract DAAK70-83-K-0018 (DARPA Order 3206) is gratefully acknowledged.

1- INTRODUCTION

Target Tracking and Scene Analysis are two research areas in Computer Vision which,in the past, have been dealt with as separate problems. Most target tracking systems [3,5,9,10,18,23,26,27,30] have the relatively modest goal of detecting and identifying moving objects and tracking them, using prediction and correlation techniques, as long as they move unobscured across the field of view. Most of these systems have been developed with a set of stringent real time constraints which, given current hardware technology, constrains them to employ a set of computationally simple algorithms.

Most scene analysis/segmentation systems [11,17,19,25] analyze a given scene based on the static properties of objects such as shape/structure and texture, where the class of objects are generally predefined. These systems tend to be slow whenever the scene is complex, containing many objects.

There are advantages to combining the two problems into one. This paper is based on such an idea and is concerned with the design of an Intelligent Target Tracking System (ITTS). This experimental system tracks targets based not only on models for targets (shape, motion, etc) but also on models of the environment through which the targets navigate and of the sensing system(s) employed to acquire the time-varying images on which the analysis is based.

In this paper we utilize the dynamic occlusion of moving and stationary objects by one another to place bounds on distance and angular extent of location of the stationary objects. Occlusion of a moving object by a stationary

object yields an upper bound on the distance for that particular stationary object, while occlusion of this stationary object by another moving object yields a lower distance bound for it. We detect occlusion using object models and perspective information. Figure 1 illustrates an example where an occluding object, such as a tree, can be detected by tracking a moving object as it moves behind the tree. During the course of tracking the moving object, we are able to segment the stationary scene via the dynamic occlusion. This allows us to construct a perspective map and a scaled ground map of the environment.

The ITTS is composed of a number of processes which can be divided into three processing stages: target recognition , segmentation of time-varying images, and scene model generation.

Target recognition is an interactive process (in the current version of our ITTS) where the distance and orientation of a moving target with respect to the viewer, at a particular instant, is computed from a zoom picture. The segmentation stage is composed of a series of processes, including image differencing, noise cleaning, connected component detection, and region extraction from a wide angle picture.

The scene model generation stage is the main part of the system and is composed of a variety of processes :

1- the "main program" which controls the system,

2- a "matcher" which matches the predicted appearance of moving targets to their actual appearance in a given frame (instance),

3- a "predictor", which given two or more instances of a moving object's appearance, predicts its appearance in the next frame,

4- an "analyzer", which corrects and updates all the data structures based on the closeness of a match between predicted and actual appearance of a moving target,

5- a "map generator" which generates two kinds of maps, a perspective map which segments the scene perspectively into open spaces and stationary portions (where there are stationary objects), and a scaled ground map of the environment which indicates where the moving targets have been observed and where the stationary objects are located.

2- AN EXPERIMENTAL SYSTEM

In this section we present an experimental ITTS. The principal goal is to build a model of the stationary scene by tracking the known moving targets. We accomplish this by detecting instances of occlusion, of which there are two kinds, "occlusion from the front", where a target moves behind a stationary object, and "occlusion from behind", when the target moves in front of the stationary object.

2.1- Problem Domain Definition

The shapes of dynamic objects are modeled as rectangular parallelepipeds of constant gray level (color); their images are modeled as rectangles. Three parallelepipeds of different sizes were used to represent our targets.

A target's mobility was represented by its velocity. An object at any instant of time was represented by its type, size, location, orientation, velocity, trajectory, and information about related (occluding) stationary objects.

Stationary objects were modeled as tall, elongated, rectangular parallelepipeds in 3-D, and as rectangles in 2-D. Their appearances were represented by size and location. The 3-D location of a stationary object is represented by a pair of intervals $[(\alpha_1 , \alpha_2) , (d_1 , d_2)]$ where the true line of sight to the stationary object is included in the interval (α_1 , α_2) and the true range is included in (d_1 , d_2); see Figure 2.

We adopt a two camera system model. One has a wide angle lens (10 mm), viewing the scene from a fixed perspective, and another has a zoom lens (25 mm) capable of panning around the Y axis. Figure 3 illustrates the camera model. Images taken by the first camera are used by the segmentation algorithm. Images taken by the zoom lens camera are used, in conjunction with object models, to determine the distance and orientation of moving objects in a particular frame. The motivation for using a zoom lens camera is that the high angular resolution allows accurate computation of the distance and orientation. It is assumed that all the camera parameters are known.

2.2- Data Representation

Our system includes two different groups of data structures, one symbolic and the other iconic. The symbolic data structure includes a pair of graphs, one for representing the dynamic part of the scene and another for representing the stationary part of the scene see Figure 4. The two are linked to each other at the region and object level. The graph representing the dynamic objects in the scene is interpreted as follows: Analysis of the observed data over a period of F frames has resulted in the detection and tracking of N dynamic objects, where object "i" has been seen for a sequence of X_i frames, and in each frame the object has appeared to consist of one or more dynamic regions.

The interpretation of the stationary component for the same period is as follows : Tracking the N moving objects has resulted in recognition of a

stationary scene (environment) composed of M stationary objects, each consisting of one or more stationary regions; see Figure 4.

It should be noted that the degree of the APRANC node (which specifies the number of regions comprizing the object at each frame) for any moving object in the most recent frame (where the APRANC initially represents a prediction) can be changed due to a detailed analysis of the appearance of that object by the matching process. Of course an APRANC node of degree two represents "middle occlusion" explicitly. For example , the object in Figure 1.c is represented by node "A" of Figure 4, indicating the fact that object A in the first frame was occluded in the middle, and therefore represented by two regions.

Occlusion, in general, is represented in the symbolic data structure by establishing links between the dynamic and stationary region and object nodes. For example the two dynamic regions of Figure 1.c are represented by nodes DYNREG#1 and DYNREG#2 in Figure 4 and the node APRANC #1 represents the fact that this object is occluded in the middle. The stationary region between the two dynamic regions of Figure 1.c is represented by the node STNREG#1. This particular instance of occlusion is represented by the explicit links between STNREG#1 and DYNREG#1 and STNREG#1 and DYNREG#2.

The second symbolic data structure is a two dimensional array containing the range information for the detected stationary objects. Every row of this array represents a detected stationary object with enough information to place a bound on the area in the scene where the stationary object resides. There are bi-directional links between every row of this array and the appropriate DYNOBJ

and STNOBJ nodes of the graph pair (described earlier in this Section). This array is used and updated by the predictor and the analyzer. The ground map generator also uses this array. The third symbolic data structure is also a two dimensional array containing the 3-D information concerning the moving targets, which results from the target recognition stage.

The iconic data structures (maps) are the other class of data structures that are employed. Our ITTS incorporates two different maps, a perspective map and a ground map. The perspective map represents the detected stationary regions and in turn the stationary objects. It also represents the detected visible portion (space) of the environment, indicating the paths traveled (in perspective) by the moving objects.

The visible space content of the perspective map is accumulated by the matching process, and the stationary region portion is accumulated by the predictor and the error analyzer. Figure 14 shows some examples of the perspective map. The light regions represent the visible portions of the scene, while the darker regions represent the stationary portions of the scene (the stationary objects).

The ground map represents a scaled map of the scene, where each target at any given frame (time instant) is represented by its location, trajectory, and velocity. The accumulation of these for any target represents the path along which the target has been moving. The ground map also represents the areas of the ground where the stationary objects reside. This map is constructed by the "map generator". Figure 15 gives some examples of the ground map. The map

represents a 70" by 70" scene and each side of a square in the grid represents approximately 2.5 inches of the real world. The bright vectors represent instances of the moving objects and their velocity , while the darker vectors (in some cases making up polygons) represent regions of the ground where stationary objects have been found to reside. The bottom of Figure 15 indicates the location of the camera (origin of the world and camera coordinate system). The two dark vectors pointing towards the upper sides of the image represent the angular field of view of the wide angle lens.

2.3- Process Definition

This Section describes the processing components of the ITTS individually. Figure 5 illustrates these processes schematically.

2.3.1- Target Detector

Input to this process is a set of gray level images and a *mask frame* (an image of the stationary environment only), all taken by the wide angle lens camera. It processes one image at a time and produces a set of region descriptors (in our case parameters describing a rectangle) representing the dynamic regions. The target detector consists of the following processes:

1- take the difference between an image and the mask frame,

2- apply thresholding to the difference picture,

3- apply shrink and expand operations (noise cleaning) to the image obtained in step 2,

4- apply connected components analysis to the image obtained from step 3,

5- Fit upright bounding rectangles to components remaining after step 4.

6- apply local image differencing to a window surrounding each detected region to refine the estimates of the shapes of the dynamic regions (in this step the thresholding process, after differencing, uses a threshold value much lower than in step 2).

Figure 6 shows an example of the target detector's output.

2.3.2- Target Recognition

In order to generate a model of the stationary scene and approximate areas of the scene where stationary objects reside based on occlusion by/of moving objects, we need to estimate the location and orientation of these moving objects in the scene. This is done based on the known physical dimensions and measured image heights of the moving objects. To better resolve the height of a target in the image we use pictures of the targets taken by a zoom lens camera.

Given any of these pictures, the user interactively segments a face of the target (preferably the one in the center of the image) by positioning four points, two on the bottom edge of the face and two on the upper edge. This will fit a polygon to this particular face as shown in Figure 7. Next we describe the actual estimation of distance.

Figure 8 shows a target in the world coordinate system and its relationship with the image coordinate system. The vertices A,B,C, and D map into the points a,b,c, and d on the image plane, respectively, and from the relation between these vertices and points we can easily compute the distance and orientation. It is easy to show that

$$-Z_0 = \frac{fH}{\Delta y'} \tag{1}$$

where $|Z_0|$ is the desired distance, H is the height of the target and $\Delta y'$ is

$$y_{top}' - y_{bot}' = (y_a - y_c) - x_a\left(\frac{y_b - y_a}{x_b - x_a}\right) + x_c\left(\frac{y_d - y_c}{x_d - x_c}\right). \tag{2}$$

To compute the orientation, p of the planer face, we average the orientation of the top and the bottom edge of the face. The orientation, p, is given by

$$p = \frac{f}{2}\left(\frac{y_d - y_c}{x_d y_c - x_c y_d} + \frac{y_b - y_a}{x_b y_a - x_a y_b}\right) \tag{3}$$

2.3.3- Scene - Model Generation

The components of the scene-model generation process are described in the following subsections.

2.3.3.1- Matcher

This process compares the predicted appearance of dynamic objects for frame n with the actual observed data in the n-th frame. Since both the predicted and actual appearance of the dynamic objects are represented as regions (in the image plane) which in turn are represented by bounding rectangles, the matching is done among these rectangles. These rectangles are represented by the four parameters <c,r,nc,nr> where c = column, r = row, nc = number of columns, nr = number of rows. The first two indicate the location and the other two the size of a rectangle.

The matching process measures the similarity between a predicted appearance $\mathbf{R}_p$ and a region $\mathbf{R}_d$ in the set of dynamic regions in the current frame. The followings steps are taken during the matching of an $\mathbf{R}_p$ with an $\mathbf{R}_d$:

1- Find all $\mathbf{R}_{d_i}$'s from the set of all $\mathbf{R}_d$'s such that
$c_{diff} = | c_p - c_{d_i} | < tvalue_1$, $r_{diff} = | r_p - r_{d_i} | < tvalue_2$,
$nc_{diff} = | nc_p - nc_{d_i} | < tvalue_3$, $nr_{diff} = | nr_p - nr_{d_i} | < tvalue_4$
All the "tvalues" are predefined threshold values.

2- Choose the $\mathbf{R}_{d_i}$ such that c_{diff} , r_{diff} , nc_{diff} , *and* nr_{diff} are minimum for it and $TR_p = TR_{d_i}$ where
TR_p and TR_{d_i} are two trajectories computed as follows:
$TR_p = DIRECTOR (\mathbf{R}_k , \mathbf{R}_p)$;
$TR_{d_i} = DIRECTOR (\mathbf{R}_k , \mathbf{R}_{d_i})$;

where $\mathbf{R}_k$ is the last appearance of the object represented by $\mathbf{R}_p$.

Note : director returns one of the following directions : N , S , E , W , NE , NW , SE , SW based on the comparisons between c,r,nc, and nr.

The above steps describe the matching process between any $\mathbf{R}_p$ and $\mathbf{R}_d$. It should be realized, however, that the process generally matches **n** $\mathbf{R}_p$'s with **m** $\mathbf{R}_d$'s (or vice versa). Depending on the relation between **n** and **m**, different cases may arise. Some of these cases are as follows :

1- Success : This is the case where **n** = **m** and every $\mathbf{R}_p$ is matched to one and only one $\mathbf{R}_d$.

2- An $\mathbf{R}_p$ has no corresponding $\mathbf{R}_d$: This is typically due to an incorrect predicted decomposition; the system had predicted two regions ($\mathbf{R}_{p_1}$ and $\mathbf{R}_{p_2}$) representing an object, but only one $\mathbf{R}_d$ is matched to either $\mathbf{R}_{p_1}$ *or* $\mathbf{R}_{p_2}$ or both, resulting in an ambiguity indicating an incorrect prediction.

Another possible cause of this case is that the object is curr tly totally occluded, but the system had predicted that it would see sor f it. In our set of image frames this case did not occur.

3- An $\mathbf{R}_d$ has no corresponding $\mathbf{R}_p$: There are two possibilities. Either a new object has entered the scene, or an object has been decomposed unexpectedly(middle occlusion).

a) An object is classified as new if either of the following is true:

1- $c_d \leq C+e$	Target entering from LEFT
2- $r_d \leq R+e$	Target entering from BOTTOM
3- $c_d + nc_d \geq NC-e$	Target entering from RIGHT
4- $r_d + nr_d \geq NR-e$	Target entering from TOP

Picture frame $\equiv$ <C,R,NC,NR> and e is some pixel count threshold(i.e. in our case e=50).

If this is the case the count for the dynamic objects is increased by one, and corresponding DYNOBJ, FRAME, APRANC, and DYNREG nodes of Figure 4 are generated. This new rectangle $\mathbf{R}_d$ is painted as visible space in the iconic data structure.

b) If $\mathbf{R}_d$ does not represent a new incoming object, then indeed it must represent part of an existing moving object. We assume that to whichever object, X, this $\mathbf{R}_d$ belongs, the other part of X has already been matched to some other region, say $\mathbf{R}_{d'}$.

The following steps will find an $\mathbf{R}_{d'}$ which in combination with $\mathbf{R}_d$ represents object X :

for k=1,N **do**
Rem : N = # of regions extracted from the current frame.
Find an $\mathbf{R}_{d'}$ which has already been matched to an $\mathbf{R}_{p'}$
if $c_{d'} - (c_d + nc_{d'}) \leq$ *avgtreewidth* **and** $| r_{d'} - r_d | \leq e$
or ,
if $c_d - (c_{d'} + nc_{d'}) \leq$ *avgtreewidth* **and** $| r_{d'} - r_d | \leq e$
then $\mathbf{R}_d$ to point to whichever object the $\mathbf{R}_{d'}$ points to.
endif
endif
endo
endfor

After finding $\mathbf{R}_{d'}$, update the symbolic and iconic data structures to indicate this decomposition of object X. More specifically the degree of the APRANC node is changed from 1 to 2.

4- An $\mathbf{R}_d$ has multiple corresponding $\mathbf{R}_p$'s : This is a special instance of the second case .

5- An $\mathbf{R}_p$ has multiple corresponding $\mathbf{R}_d$'s : This is a special instance of the third case.

2.3.3.2- Predictor

This module predicts the future appearance of a dynamic object (in 3-D and then in 2-D) based on its type and :

1- its current 3-D status (location, orientation, velocity, and trajectory),
2- its previous 3-D status, and,
3- the accumulated information about the stationary world.

The location of a target in the real world is the location of a reference point on that target in the real world. This reference point is chosen to be the intersection of the optical axis of the zoom lens camera system and the center of a particular face of the target. We modeled the targets so that the origin would lie in the center of either a Right or Front face of the target.

If $P_t\,(X_t, Z_t)$ denotes the location of a target in the scene at time t where

$$X_t = d_t \times \sin \beta_t \quad \text{and} \quad Z_t = d_t \times \cos \beta_t \tag{4}$$

d : the distance to the target computed by the procedure of Section 2.3.2

β : the pan angle by which this particular zoom picture was taken,

then P_{t+1} will represent the location of the target at time t+1, and is computed by

$$P_{t+1} = P_t + A\Delta t + B\Delta t^2, \tag{5}$$

where the second and third terms on the right hand side represent velocity and

acceleration/deceleration rate of that target, respectively. Writing (5) in component form yields

$$X_{t+1} = X_t + A\Delta t + B\Delta t^2, \tag{6}$$

$$Z_{t+1} = Z_t + A\Delta t + B\Delta t^2. \tag{7}$$

In order to use (6) and (7) we need to compute the coefficient A and B. It is easy to show that

$$A = \frac{(X_{t-2} - 4X_{t-1} + 3X_t)}{2}, \tag{8}$$

$$B = \frac{(X_{t-2} - 2X_{t-1} + X_t)}{2}. \tag{9}$$

or, equivalently, that

$$A = \frac{(Z_{t-2} - 4Z_{t-1} + 3Z_t)}{2}, \tag{10}$$

$$B = \frac{(Z_{t-2} - 2Z_{t-1} + Z_t)}{2}. \tag{11}$$

Substituting (8) and (9) into (6), and (10) and (11) into (7) gives

$$X_{t+1} = X_t + \frac{(X_{t-2} - 4X_{t-1} + 3X_t)}{2} + \frac{(X_{t-2} - 2X_{t-1} + X_t)}{2}, \tag{12}$$

$$Z_{t+1} = Z_t + \frac{(Z_{t-2} - 4Z_{t-1} + 3Z_t)}{2} + \frac{(Z_{t-2} - 2Z_{t-1} + Z_t)}{2}, \tag{13}$$

and simple algebra yields

$$X_{t+1} = 3X_t - 3X_{t-1} + X_{t-2}, \tag{14}$$

$$Z_{t+1} = 3Z_t - 3Z_{t-1} + Z_{t-2}. \tag{15}$$

It should be realized that (14) and (15) are true for t > 2; for the case where t = 2 the following simpler equations are used:

$$X_{t+1} = 2X_t - X_{t-1}, \tag{16}$$

$$Z_{t+1} = 2Z_t - Z_{t-1}. \tag{17}$$

Equations (14) and (15) or (16) and (17) give the translational components of (5); next we compute the rotational component. Let α_t represent the orientation of a target at time t with respect to the X-axis of the real world coordinate system. Associated with (5) is a rotational component computed by

$$\alpha_t = \alpha_t + A\Delta t + B\Delta t^2, \tag{18}$$

where the second and third terms of the right hand side represent change in orientation and rate of change in orientation respectively. Solving for A and B (as we did for the case of X and Z) and substituting them in (18) we get

$$\alpha_{t+1} = 3\alpha_t - \alpha_{t-1} + \alpha_{t-2} \quad \text{for} \quad t>2, \tag{19}$$

$$\alpha_{t+1} = 2\alpha_t - A_{t-1} \quad \text{for} \quad t=2, \tag{20}$$

which completes the transformation a target will go through from the time t to

t+1. In order to complete the 3-D prediction process, all the vertices of a target (its model) should go through the computed transformations.

Let $P_{w_{t+1}}(X_{w_{t+1}}, Z_{w_{t+1}})$ represent a point (preferably a corner) of a target in the real world at time t+1, and $P_m(X_m, Z_m)$ represent the corresponding point on the target's model. Then the 3-D prediction for a target is computed for every corner of the model according to

$$\begin{bmatrix} X_{w_{t+1}} \\ Z_{w_{t+1}} \end{bmatrix} = \begin{bmatrix} \cos\alpha_{t+1} & \sin\alpha_{t+1} \\ -\sin\alpha_{t+1} & \cos\alpha_{t+1} \end{bmatrix} \begin{bmatrix} X_m \\ Z_m \end{bmatrix} + \begin{bmatrix} X_{t+1} \\ Z_{t+1} \end{bmatrix}. \tag{21}$$

This completes the first step of the prediction process. Next we compute the perspective projection of the points computed by (21). This is accomplished using the following equations :

$$x_{t+1} = \frac{X_{w_{t+1}}}{Z_{w_{t+1}}} f, \tag{22}$$

$$y_{t+1} = \frac{(Y_{w_{t+1}} - L)}{Z_{w_{t+1}}} f. \tag{23}$$

where $p_{t+1}(x_{t+1}, z_{t+1})$ is a point on the image corresponding to $P_{w_{t+1}}(X_{w_{t+1}}, Z_{w_{t+1}})$, L is the camera height, and f is the focal length of the wide angle lens.

The above procedure will give us eight points (the eight corners of a rectangular parallelepiped), to which we next fit the best rectangle, $\mathbf{R}_{t+1}$ which represents a region on the image at time t+1 for which we predict the presence of a particular target.

Having predicted $\mathbf{R}_{t+1}$, we check for possible occlusion, for which if one exists, $\mathbf{R}_{t+1}$ is modified appropriately. To check for possible occlusion, we need to compute the angle along which the target is predicted to be seen. This angle is simply computed using (14) and (15) or (16) and (17) as follows :

$$\beta_{t+1} = \arctan\ (\frac{X_{t+1}}{Z_{t+1}}) \tag{24}$$

Next we compare β_{t+1} with entries in the symbolic data structure representing the range information regarding the detected stationary objects (see Section 2.2). If a match exists, then the target may be occluded depending on the accumulated distance information for that particular stationary object and the predicted distance for this target. If distance comparison yields an occlusion, then $\mathbf{R}_{t+1}$ is modified based on the region $\mathbf{R}_s$ representing the stationary object.

This modification is done in one of the following two ways: either the size of $\mathbf{R}_{t+1}$ gets smaller (the case when the target will be occluded on one side), or $\mathbf{R}_{t+1}$ splits into two pieces $\mathbf{R}^1_{t+1}$ and $\mathbf{R}^2_{t+1}$ (the case where the target will be occluded somewhere in the middle). See Figure 9.

The above modifications would hold if the stationary object was known to have been completely detected. Otherwise, $\mathbf{R}_{t+1}$ and $\mathbf{R}_s$ are modified on the assumption that this stationary object is as wide as possible. For example, if a stationary object (represented by $\mathbf{R}_s$) is partly detected and its observed width is w, then it is possible that we have not yet detected the remaining part of the object, $\mathbf{r}_s$, with a width equal to W-w (if W is the maximal width of any

stationary object). Therefor, $\mathbf{R}_{t+1}$ would be modified as if it was occluded by a region of width W. See Figure 10.

The last step in the prediction process is the updating of the graph pair data structures, range information data structure, and the perspective and ground maps. The graph pair is updated by generating DYNREG node(s), corresponding STNREG node (if any $\mathbf{r}_i$ is predicted), and FRAMES node and labeling all these nodes as P (Prediction). These predictions will be checked against the actual data by the analyzer at time t+1.

2.3.3.3- Analyzer

The analyzer examines the reports from the matching process. If the errors (mismatches) are below some predefined threshold, then the predictions are accepted and all data structures are updated appropriately. Otherwise a search is initiated to determine the cause of the failure.

The search can be best described by giving an example. Figure 11.a shows a predicted target's image X with an associated known part of a stationary object A. Figure 11.b shows the actual appearance of the object. By comparing these two (a and b) and considering the values of different parameters, different conclusions can be made. For example, if P1 = P2 and l1 = l2, then the hypopaper about the stationary object to be extended (perspectively) by width A from the right is wrong, and the actual width of the stationary object is B. Or, if

if $X1 \neq X2$, then we conclude that the previous estimate of this target's velocity is incorrect and should be corrected by the value $X2 - X1$.

The analyzer is also responsible for the detection of unpredicted stationary regions (and in turn stationary objects). This is done as follows : Suppose a predicted region $\mathbf{R}_p$ is matched to a region $\mathbf{R}_d$ of the current frame with an above threshold error. Furthermore assume that one of the two ends of $\mathbf{R}_p$ and $\mathbf{R}_d$ match (i.e. $c_p = c_d$ OR $(c_p + nc_p) = (c_d + nc_d)$) and the length of $\mathbf{R}_p$ is larger than the length of $\mathbf{R}_d$ (i.e. $nc_p < nc_d$) . Then it is hypothesized that there exists a piece of a stationary region (representing part of a stationary object) with width $|nc_p - nc_d|$ located to the left or right of the target, depending on the trajectory of the moving object.

2.3.3.4- Map Generator

The map generator is responsible for constructing/updating the perspective and the ground map. Both of these maps are 510 by 510 8-bit/pixel pictures. The perspective map is constructed and updated by painting regions (rectangles) corresponding to the dynamic regions ($\mathbf{R}_d$'s) and stationary regions ($\mathbf{R}_s$'s) whose parameters are taken from the symbolic graph pair data structure (see Section 3.2). Figure 14 contains some examples of the perspective map.

The map generator is also responsible for the construction and maintenance of the ground map. For every instant of a moving target a vector is drawn representing its location (in 3-D), orientation, and velocity. All of the areas on

the ground (in 3-D) where stationary objects have been detected to reside are also indicated by drawing vectors bounding them. Figure 15 contains examples of the ground map.

2.3.3.5- Occlusion From Behind Detector

This procedure detects all those stationary objects which are occluded by the moving targets, by taking the difference between every dynamic region $\mathbf{R}_t$ corresponding to $DOBJ_{i,t}$ and the same region in the mask frame, resulting in a difference region $\mathbf{R}_{diff}$.

The region $\mathbf{R}_{diff}$ is then thresholded based on the predefined gray level values corresponding to the dynamic and stationary objects, yielding the thresholded region $\mathbf{R}_{thresh}$. If, in fact, there exists a stationary object behind this dynamic object $DOBJ_{i,t}$, then the region $\mathbf{R}_{thresh}$ would contain non-zero elements, to which we then fit a rectangle $\mathbf{R}_s$. $\mathbf{R}_s$ is then entered into the symbolic data structures and the iconic data structures. Examples of this detection are illustrated in Figures 14.a, 14.c, 15.c, and 15.e.

3- EXPERIMENTAL RESULTS

This section includes results of processing a sequence of twenty picture frames by our ITTS. The images are 510 by 510, eight bit pixels. Dark objects represent the stationary objects and bright objects represent the moving targets. There were three different sizes of moving targets, and a variety of stationary objects.

The scene covers an area of approximately 70" by 70". Figure 12 shows the sequence of twenty frames plus the mask frame where, in order to save space, the 510 by 510 images have been clipped into 510 by 175 segments (areas of interest). Figure 13 illustrates the scene layout with the paths along which targets move. Figure 14 shows some snapshots of the perspective map where, again to save space, the images are clipped into 510 by 175 pieces. The bright regions represent the visible portions of the scene (where the targets have been seen) and the dark regions represent the stationary objects. The perspective map changes as more picture frames are processed.

It should be noted that the stationary objects are not represented by their full height; due to the altitude of the camera with respect to the ground plane, the tops of stationary objects cannot interact with dynamic objects. More accurate estimate of heights could be detected if the cameras were looking down at the scene so that the targets at different distances pass behind and are occluded by the stationary objects.

Figure 14.a shows the perspective map after processing one frame. The stationary region detected in this frame is due to the procedure for detecting occlusion from behind (see Section 2.3.3.5 for details); this is also indicated in the ground map, as shown in Figure 15.a. Other instances of this kind of occlusion detection are shown in Figure 14.c and 16.c where in each, the rightmost stationary region is detected by the above procedure.

Figure 12 not only shows sequence of the 20 frames, but also shows the results of the predictor. What is shown in this figure are different frames, with the prediction for that particular frame overlaid on it. For instance, Figure 12.a shows the third frame; the rectangles represent predictions for this frame, obtained in the second frame. Since the predictor in the second frame knew about the existence of the leftmost stationary object in the scene, which would be occluding the topmost moving target at the third frame, two regions were predicted to represent this particular moving target.

Another example of such double region prediction is shown in Figure 12.g for the bottom target. It is evident that this particular prediction is wrong. As can be seen in Figure 15.e representing the ground map (explained fully later in this Section), after processing the tenth frame there is only a maximal distance associated with the second stationary object from the left. At this time, all the ITTS knows about this particular stationary object is that it is located somewhere between the viewer and its maximal distance point. Now, since the predicted location of the moving target is within this interval, the predictor after considering the possibilities of occlusion, predicts two regions. However at the

twelfth frame the analyzer detects this mistake and corrects all data structures.

Figure 15 shows examples of the ground map (as described in Section 2.2) generated by the map generator of Section 2.3.5. Figure 15.a shows the ground map after processing the first frame and Figure 15.j the result after twenty frames, where the difference between them is apparent. It should be noted that one of the goals for our ITTS was producing a map which would resemble and approximate the map shown in Figure 13. This task was accomplished through the generation of the ground map shown in Figure 15.j.

The analyzer is one of the processes responsible for producing the information used in generating the ground map. We now give some examples of how the analyzer works. The first example concerns the case of the predictor's mistake, which was discussed earlier in this Section. At the eleventh frame, the analyzer detects this mistake by realizing that 1) the two predicted R_p's match only one R_d, 2) the R_d bounds the two R_p's, and 3) that the predictor's decision based on the maximal distance for the corresponding stationary object was incorrect. This leads the analyzer to realize that the moving target should be in front of this particular stationary object, yielding a bound on the minimal distance to the stationary object. This is reflected in shortening the vector representing the interval in which the stationary object is located, as shown in Figure 15.f. A similar analysis is done at the sixteenth frame, for the moving object at the bottom.

The analyzer is also responsible for tightening the intervals and regions representing the location of the stationary objects; this is illustrated in Figures 15.a through 15.j.

ACKNOWLEDGEMENTS

I would like to thank all those who have helped me during this work. Many thanks to Dr. Azriel Rosenfeld and special thanks to my advisor Dr. Larry S. Davis. I am grateful to Dr. Allen Waxman, and I am also very thankful to Dr. Takashi Matsuyama. At last but not least, thanks to all members of the CVL and my friends for their help in many ways.

REFERENCES

[1] J.K. Aggarwal, L.S. Davis, W.N. Martin, and J.W. Roach, "Survey : Representation Methods for Three-Dimensional Objects", in L.N. Kanal and A. Rosenfeld, Eds., **Progress in Pattern Recognition 1** , North-Holland, Amsterdam, 1982, pp 377-391.

[2] S.T. Barnard and M.A. Fischler, "Computational Stereo", *Computing Surveys* 14, 1982, pp 553-572.

[3] K.H. Bers, M. Bohner, and P. Fritscne, "Image Sequence Analysis For Target Tracking", in T.S. Huang Ed., **"Image Sequence Processing and Dynamic Scene Analysis"**, Proceedings of NATO Advanced Study Institute, Braunlage/Harz, FRG, June 21 - July 2, 1982, Springer, Berlin, 1983, pp 493-501.

[4] B. Bhanu,"Recognition of Occluded Objects", *Proceedings of the Eighth International Joint Conference on Artificial Intelligence* (IJCAI-83, Karlsruhe, West Germany, August 8-12, 1983), pp 1136-1138.

[5] B. Bhanu, A.S. Politopoulos, and B.A. Parvin, "Intelligent Autocueing of Tactical Targets in FLIR Images", *Proceedings, CVPR '83: IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Washington, DC, June 19-23, 1983), pp 502-503.

[6] C.R. Brice and C.L. Fennema, "Scene Analysis using Regions", *Artificial Intelligent* 1, 1970, pp 205-226.

[7] C. Cafforio and F. Rocca, "The Differential Method for Image Motion Estimation", in T.S. Huang Ed., **"Image Sequence Processing and Dynamic Scene Analysis"**, Proceedings of NATO Advanced Study Institute, Braunlage/Harz, FRG, June 21 - July 2, 1982, Springer, Berlin, 1983, pp 104-124.

[8] R.B. Cate, T.B. Dennis, J.T. Mallin, K.S. Nedelman, M.H. Trenchard, and R.M. Bizzel, "A New Approach to Extraction of Invariant Scene Characteristics", *Proceedings, Trends and Applications, 1983 : Automating Intelligent Behavior - Applications and Frontiers* (Gaithersburg, MD, May 25-26, 1983), pp 210-215.

[9] A.E. Cowart, W.E. Snyder, and W.H. Ruedger, "The Detection of Unresolved Targets Using the Hough Transform", *Computer Vision, Graphics, and Image Processing* 21, 1983, pp 222-238.

[10] Uwe L. Haass, "A Visual Surveillance System for Tracking of Moving Objects in Industrial Workroom Environments", *Proceedings, 6th International Conference on Pattern Recognition* (Munich, FRG, October 19-22, 1982), pp 757-759.

[11] R.M. Haralick, "Image Segmentation Survey", in O.D. Faugeras, Ed., **Fundamentals in Computer Vision - An Advanced Course,** Cambridge University Press, Cambridge, UK, 1983, pp 209-223.

[12] S.M. Haynes and R. Jain, "Detection of Moving Edges", *Computer Vision, Graphics, and Image Processing* 21, 1983, pp 345-367.

[13] M. Herman, T. Kanade, and S. Kuroe, "Incremental Acquisition of a Three-Dimensional Scene Model from Image", IEEE *Transactions on Pattern Analysis and Machine Intelligence* 6, 1984, pp 331-339.

[14] T.S. Huang, Ed., **"Image Sequence Processing and Dynamic Scene Analysis",** Proceedings of NATO Advanced Study Institute, Braunlage/Harz, FRG, June 21 - July 2, 1982, Springer, Berlin, 1983.

[15] R. Jain, W.N. Martin, and J.K. Aggarawal, "Segmentation Through the Detection of Changes due to Motion", *Computer Graphics and Image Processing* 11, 1979, pp 13-34.

[16] R. Jain and H.H. Nagel, "On the Analysis of Accumulative Difference Pictures from Image Sequences of Real World Scenes", IEEE *Transactions on Pattern Analysis and Machine Intelligence* 1, 1982, pp 206-214.

[17] W. Kropatsch, "Segmentation of Digital Images Using a Priori Information about Expected Image Contents", in R.M. Haralick, Ed., **Pictorial Data Analysis** (Proceedings of the NATO Advanced Study Institute on Pictorial Data Analysis, Bonas, France, August 1-12, 1982), Springer, Berlin, 1983.

[18] G.R. Legters Jr. and T.Y. Young, "A Mathematical Model for Computer Image Tracking", IEEE *Transactions on Pattern Analysis and Machine Intelligence* 4, 1982, pp 583-594.

[19] S. Levialdi, "Basic Ideas for Image Segmentation", in O.D. Faugeras, Ed., **Fundamentals in Computer Vision - An Advanced Course,**

Cambridge University Press, Cambridge, UK, 1983, pp 239-261.

[20] H.H. Nagel, "Overview on Image Sequence Analysis", in T.S. Huang Ed., **"Image Sequence Processing and Dynamic Scene Analysis"**, Proceedings of NATO Advanced Study Institute, Braunlage/Harz, FRG, June 21 - July 2, 1982, Springer, Berlin, 1983, pp 2-39.

[21] L.J. Potter, "Velocity as a Cue to Segmentation", IEEE *Transactions on Systems, Man and Cybernetics* 5, 1975, pp 390-401.

[22] L.J. Potter, "Scene Segmentation Using Motion Information", *Computer Vision, Graphics, and Image Processing* 7, 1978, pp 558-584.

[23] B. Reischer, "Target Tracking Methodologies Present and Future", paper no IIA-1, Presented at the Workshop on Imaging Trackers and Autonomous Acquisition Applications for Missile Guidance, Nov. 19-20 1979, Redstone Arsenal, Alabama.

[24] E.M. Riseman and M.A. Arbib, "Computational Techniques in the Visual Segmentation of Static Scenes", *Computer Graphics and Image Processing* 6, 1977, pp 221-276.

[25] A. Rosenfeld, "Segmentation : Pixel-Based Methods", in O.D. Faugeras, Ed., **Fundamentals in Computer Vision - An Advanced Course,** Cambridge University Press, Cambridge, UK, 1983 , pp 225-237.

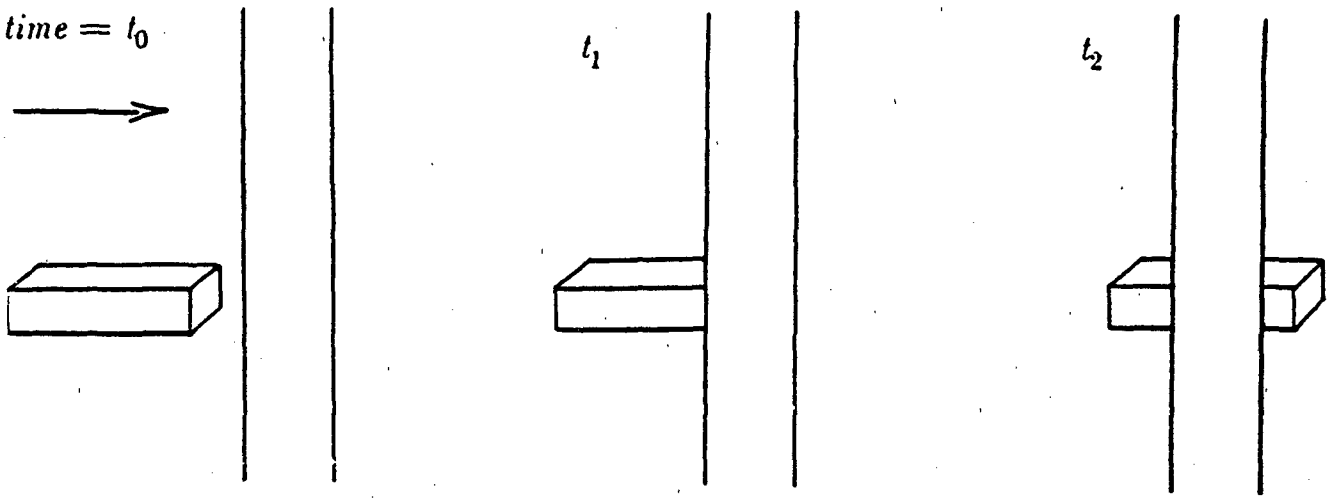
[26] R.J. Schalkoff and E.S. McVey, "A Model and Tracking Algorithm for a Class of Video Targets", IEEE *Transactions on Pattern Analysis and Machine Intelligence* 4, 1982, pp 2-10.

[27] L. Sevigny, G. Hvedstrup-Jensen, M. Bohner, E. Ostevold, S. Grinaker, and J. Dehne, "Discrimination and Classification of Vehicles in Natural Scenes from Thermal Imagery", *Computer Vision, Graphics, and Image Processing* 24, 1983, pp 229-243.

[28] W.B. Thompson, "Combining Motion and Contrast for Segmentation", IEEE *Transactions on Pattern Analysis and Machine Intelligence* 2, Nov 1980, pp 543-549.

[29] S. Yalmanchili, W.N. Martin, J.K. Aggarwal, "Extraction of Moving Object Descriptions via Differencing", *Computer Vision, Graphics, and Image Processing* 18, 1982, pp 188-201.

[30] M. Yamada and S. Ozawa, "Estimation of Detecting Reliability and Tracking Condition for a Picture Tracking System of Moving Targets", in *Proceedings of the Eighth International Joint Conference on Artificial Intelligence* (IJCAI-83, Karlsruhe, West Germany, August 8-12, 1983), pp 945-951.

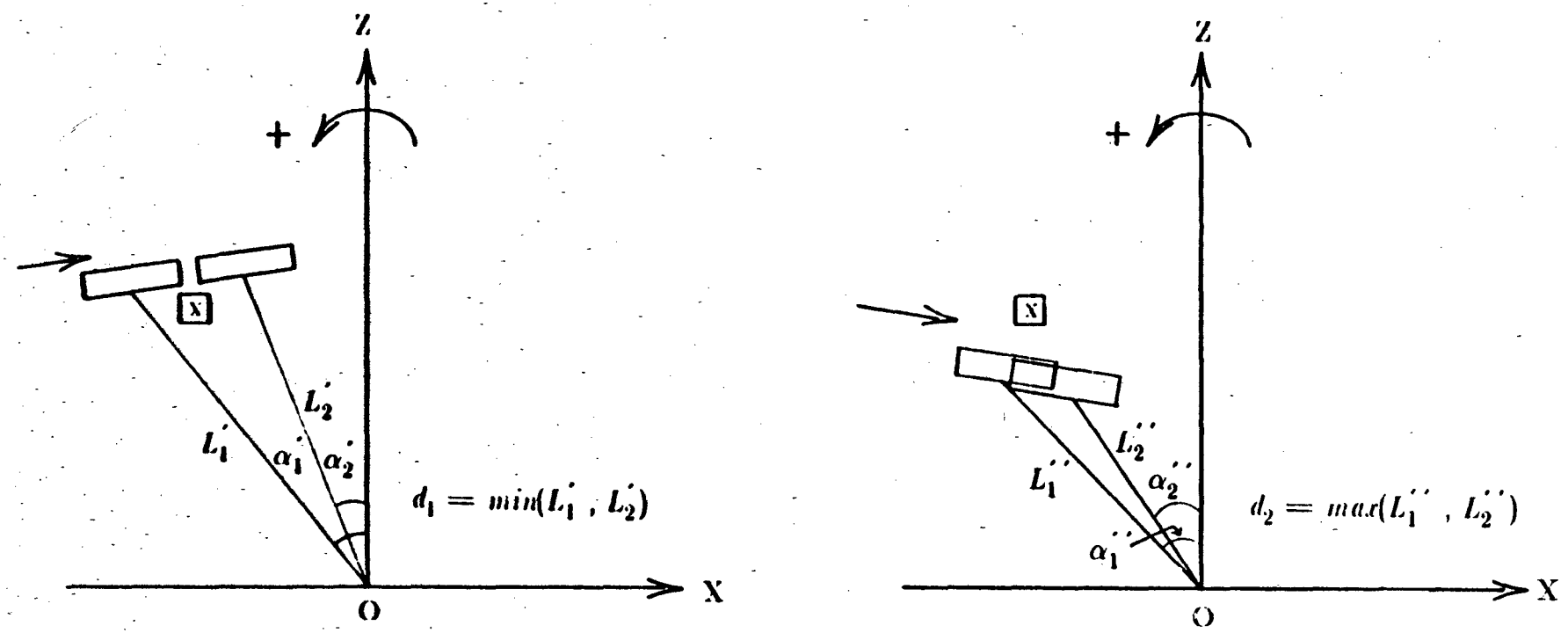


a) no interaction b) target occluded on the right side c') middle occlusion

a') no interaction b') right occlusion c) target occluded in the middle

Figure 1.

a, b, c and ç show instances of a target in 3-D; a', b', and c' show its corresponding regions. One can hypothesize (at or after t_2) that there exists an occluding object somewhere between the viewer and the target.



a) An instance of occlusion from the front

b) An instance of occlusion from behind

Figure 2.

a) Results in detecting the stationary object "x" with a maximal distance equal to d_1, and an angular interval (α_1', α_2'). b) Detection of a minimal distance for the same object "x". (a) and (b) together approximate the location of object "x" by $[(\alpha_1, \alpha_2), (d_1, d_2)]$ where

$$\alpha_1 = min(\alpha_1', \alpha_1'') \text{ and } \alpha_2 = max(\alpha_2', \alpha_2'').$$

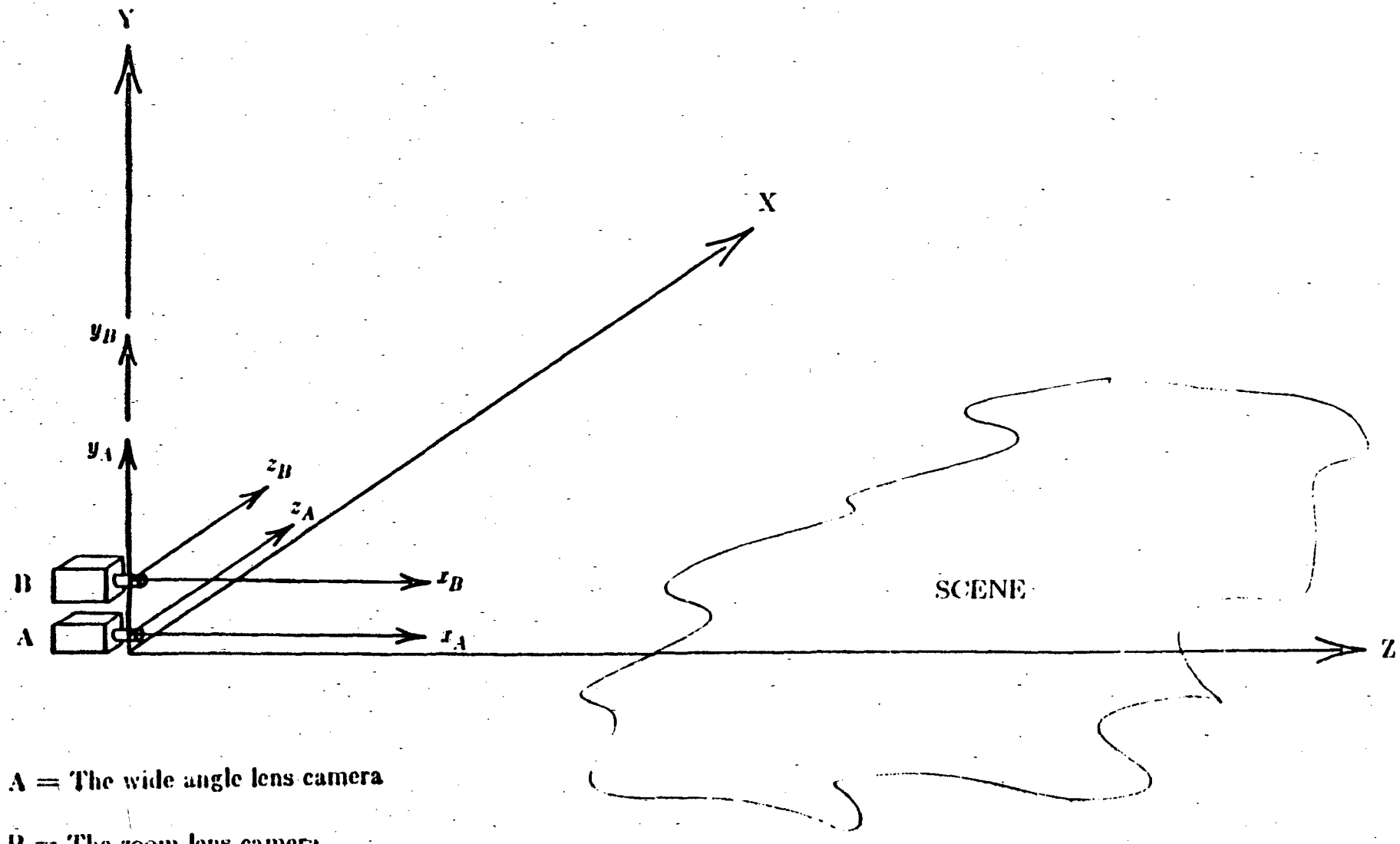
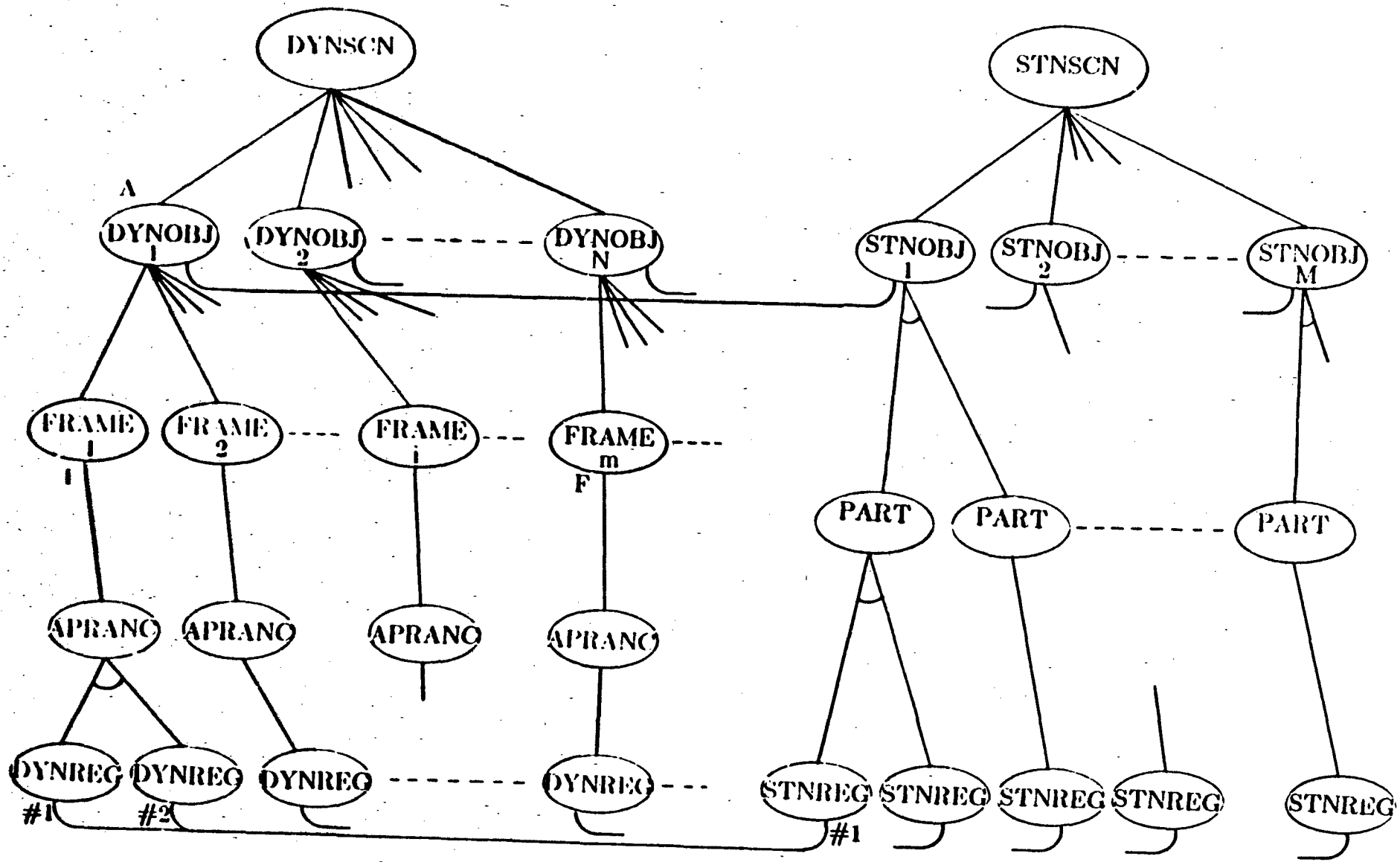


Figure 3.
The camera model.



DYN – DYNamic
STN – STatioNary
SCN – SCeNe
OBJ – OBJect
REG – REGion
APRANC – APpeaRANCe

Figure 4.

The graph pair symbolic data structure.

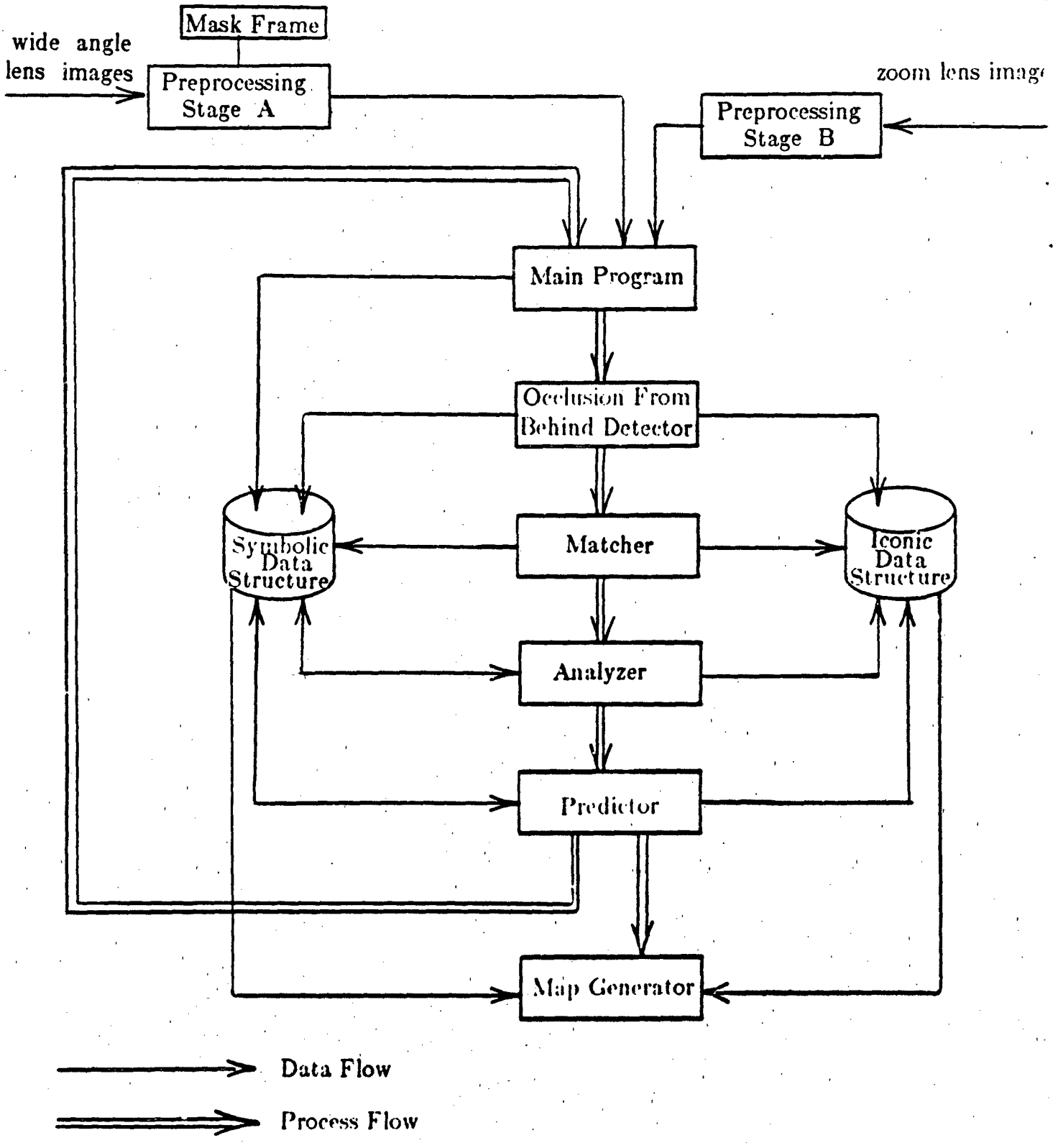


Figure 5. The processing components of our experimental ITTS.

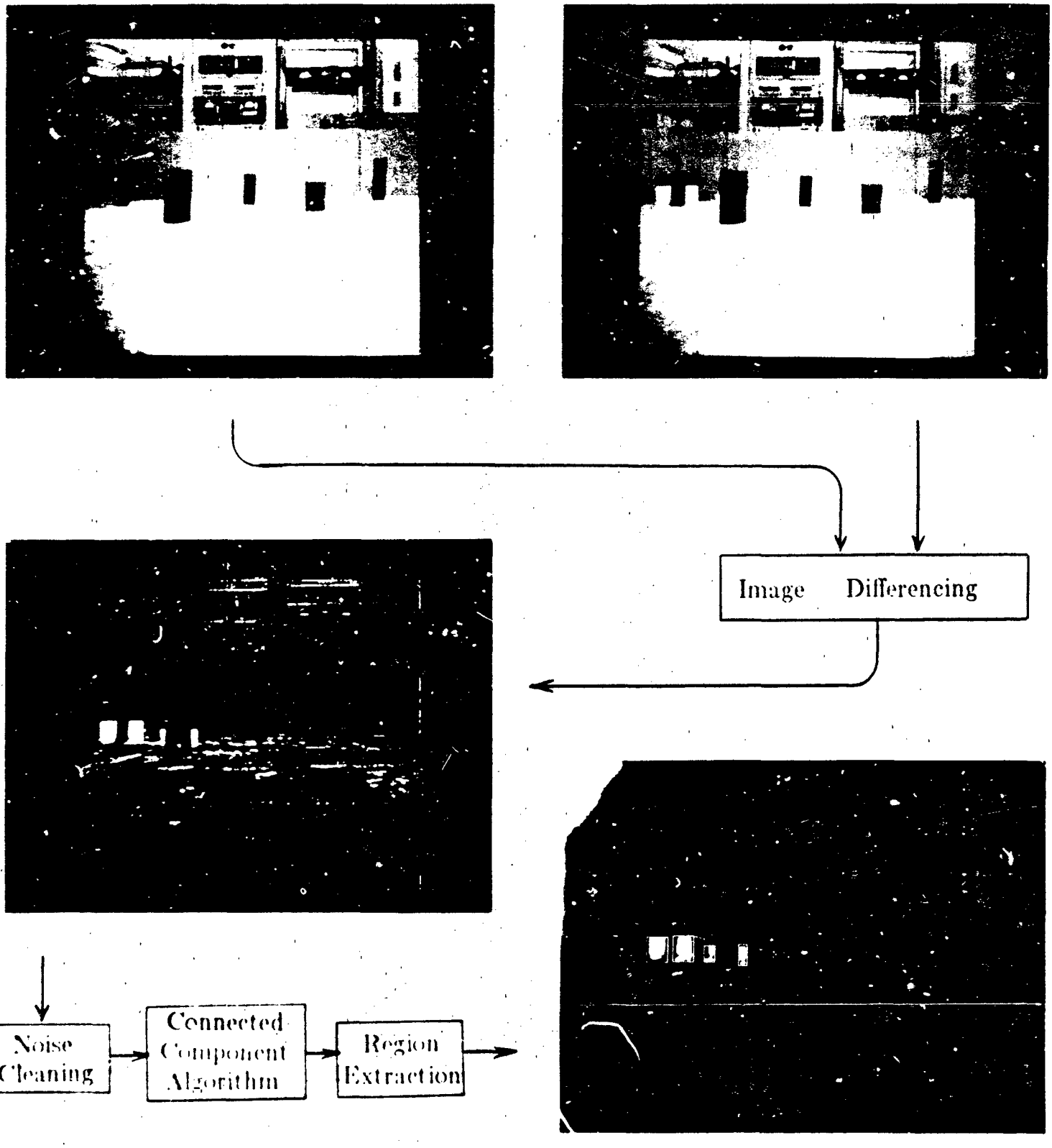


Figure 6.

An example of target detection; preprocessing pictures taken by the wide angle lens camera.

Figure 7.

An example of range detection; a polygon is fitted to a face of a moving target in a picture taken by the zoom lens camera.

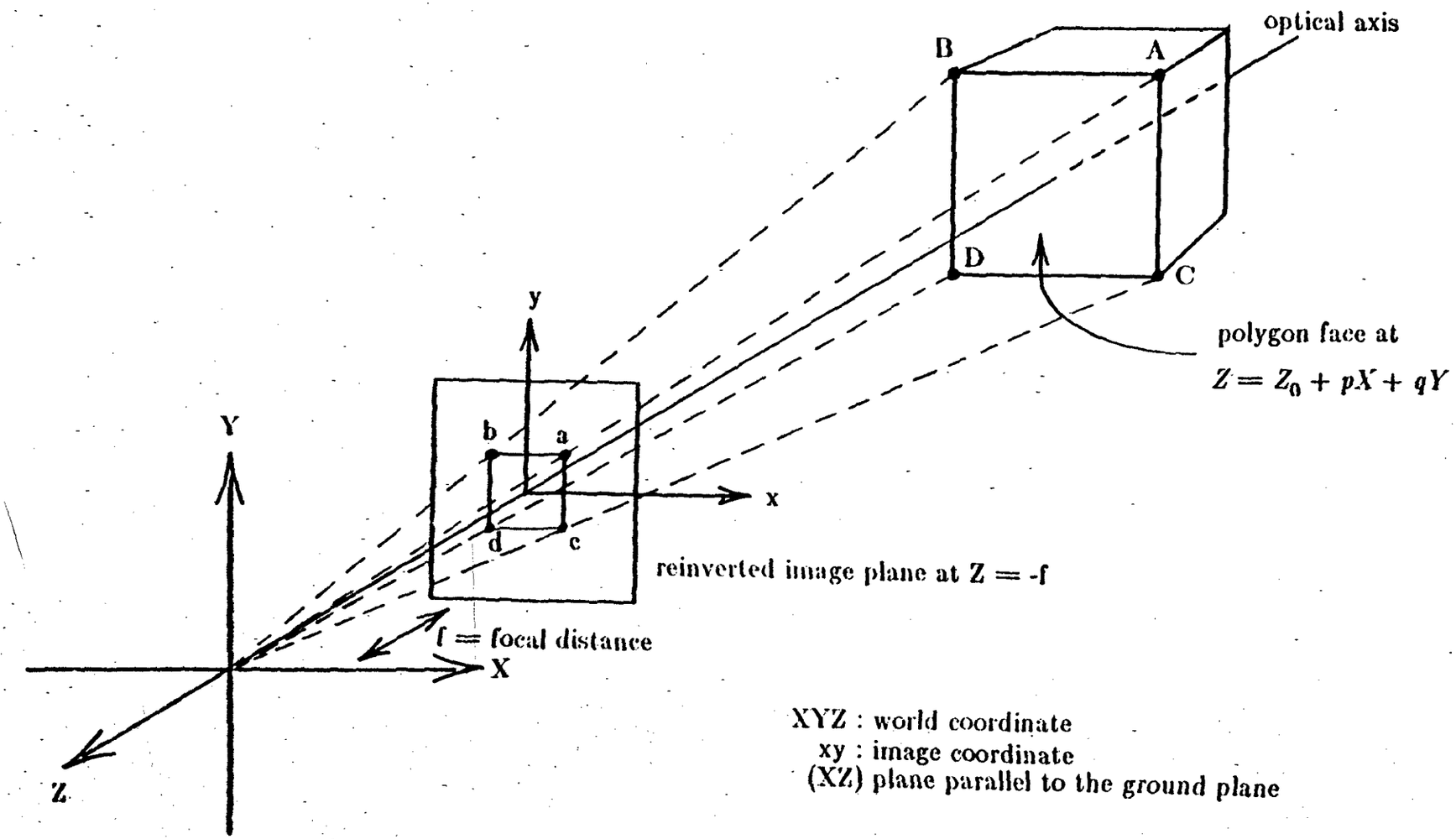
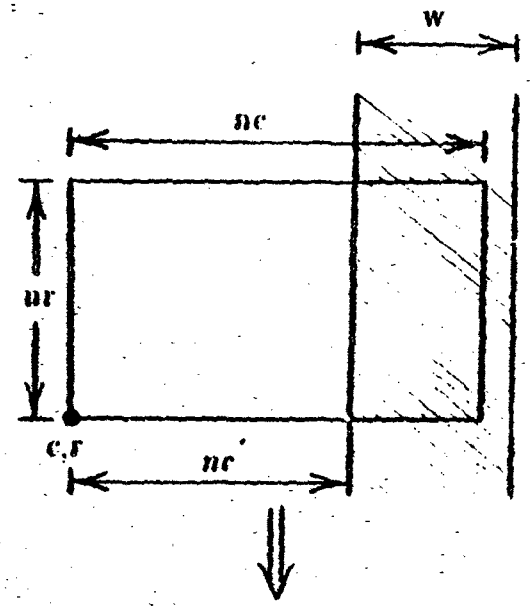


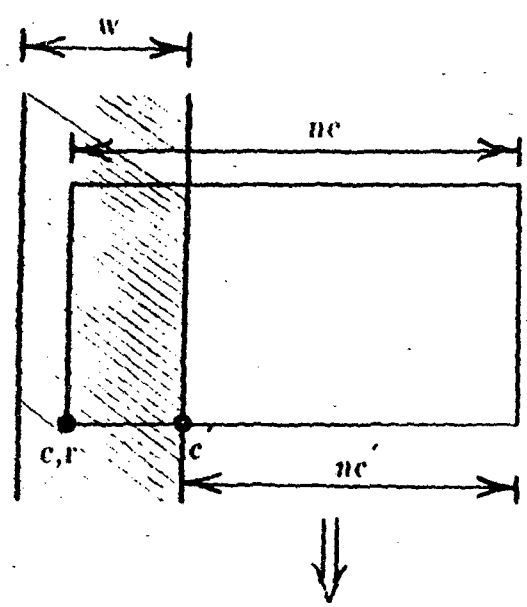
Figure 8.

Relationship between a target's face in 3-D and its image in 2-D.



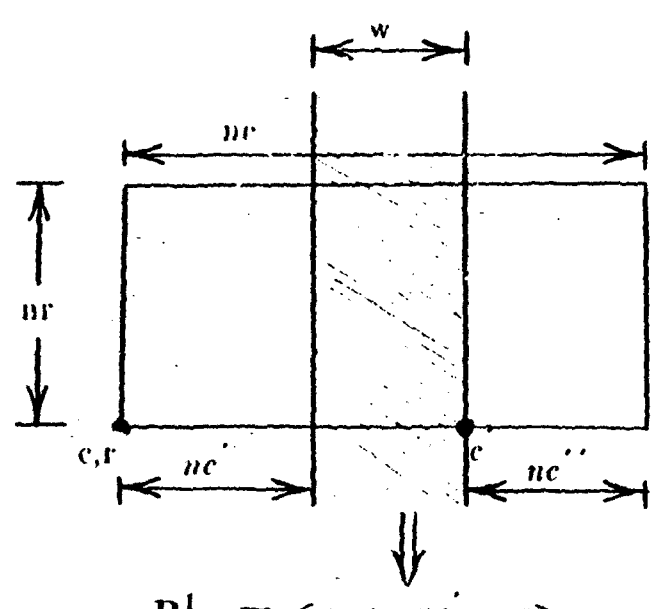
$\mathbf{R}_{t+1} \equiv <c\ ,\ r\ ,\ nc'\ ,\ nr>$

a) An example of right occlusion by a completely detected stationary object.



$\mathbf{R}_{t+1} \equiv <c'\ ,\ r\ ,\ nc'\ ,\ nr>$

b) An example of left occlusion by a completely detected stationary object.



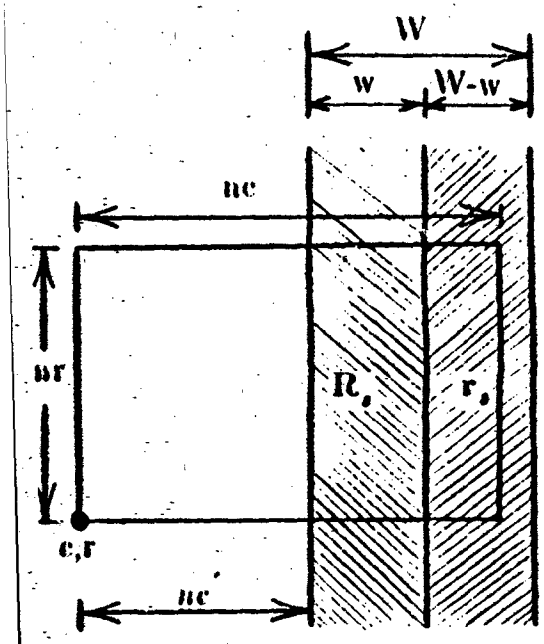
$\mathbf{R}^1_{t+1} \equiv <c\ ,\ r\ ,\ nc'\ ,\ nr>$

$\mathbf{R}^2_{t+1} \equiv <c\ ,\ r\ ,\ nc''\ ,\ nr>$

c) An example of middle occlusion by a completely detected stationary object.

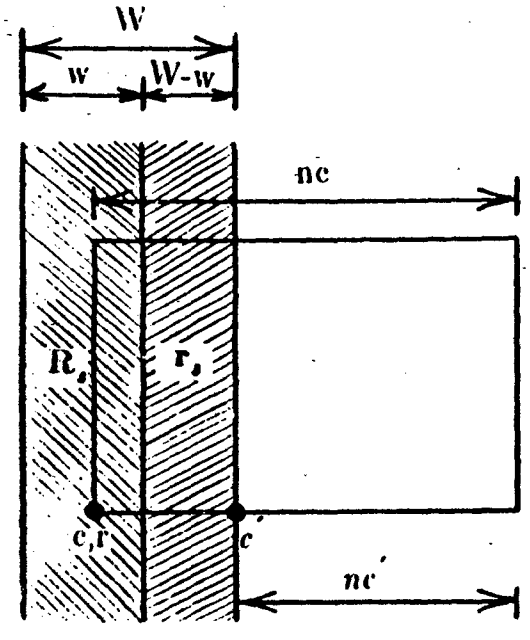
Figure 9 .

Results of modifying a prediction ($\mathbf{R}_{t+1} \equiv <c\ ,\ r\ ,\ nc\ ,\ nr>$) for a moving object based on occlusion by a completely detected stationary object.



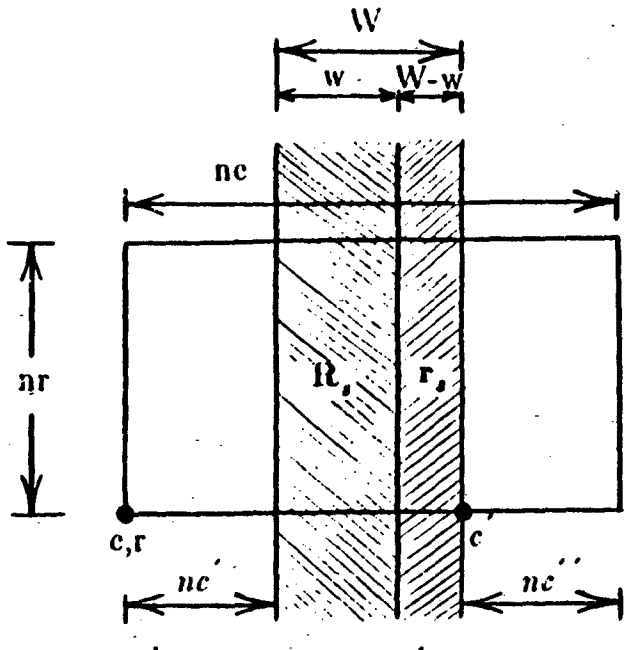
$R_{t+1} \equiv <c, r, nc', nr>$

a) An example of right occlusion by a partly detected stationary object.



$R_{t+1} \equiv <c', r, nc', nr>$

b) An example of left occlusion by a partly detected stationary object.



$R^1_{t+1} \equiv <c, r, nc', nr>$

$R^2_{t+1} \equiv <c, r, nc'', nr>$

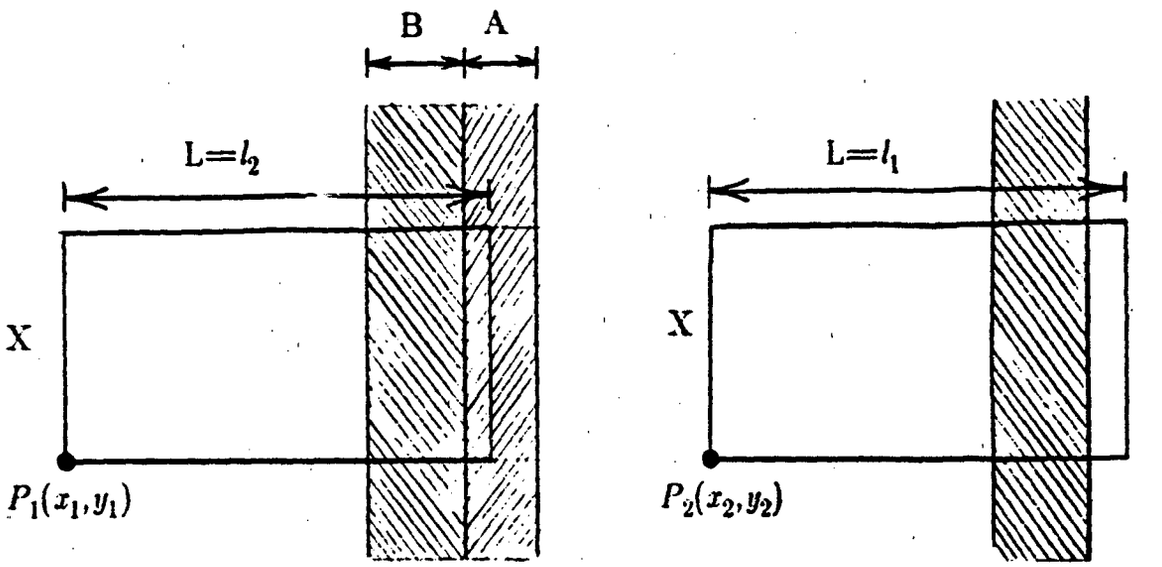
c) An example of middle occlusion by a partly detected stationary object.

detected part of the stationary object

predicted part of the stationary object

Figure 10.

Results of modifying a prediction ($R_{t+1} \equiv <c, r, nc, nr>$) for a moving object based on occlusion by a partly detected stationary object. Notice that R_s represents that part of the stationary object which has already been detected, and r_s represents the remaining part which we predict to exist.



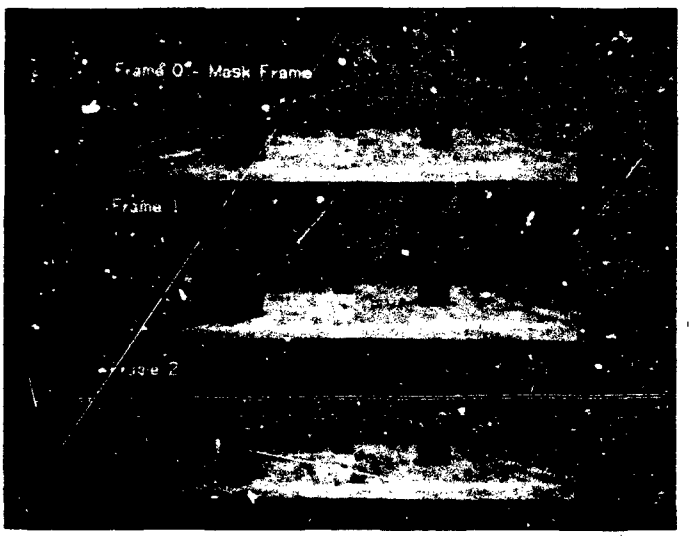
a) predicted apperance (region). b) actual appearance (region).

Figure 11.
An example of an incorrect prediction.

a)

b)

c)



d)

e)

f)

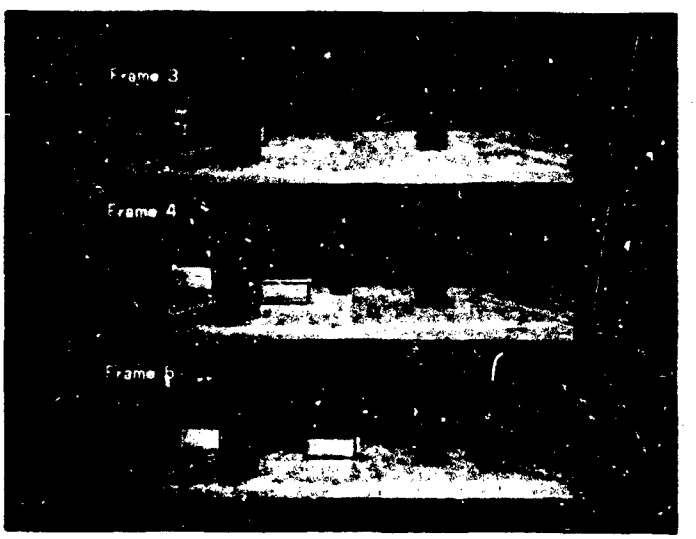


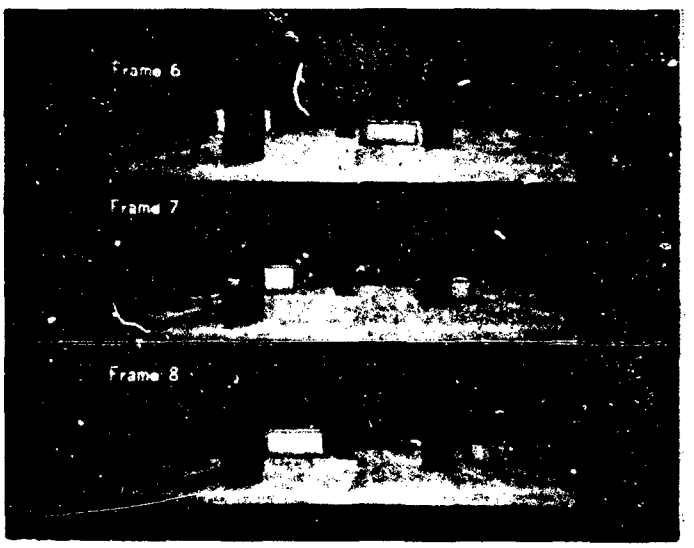
Figure 12.

The sequence of 20 frames (our input data), where the black and white windows represent the predictions for each moving target. a) shows the mask frame, b) through u) show the sequence of 20 frames. It should be noted that at the 9th and 12th frames, the leftmost moving objects are just entering the scene.

g)

h)

i)



j)

k)

l)

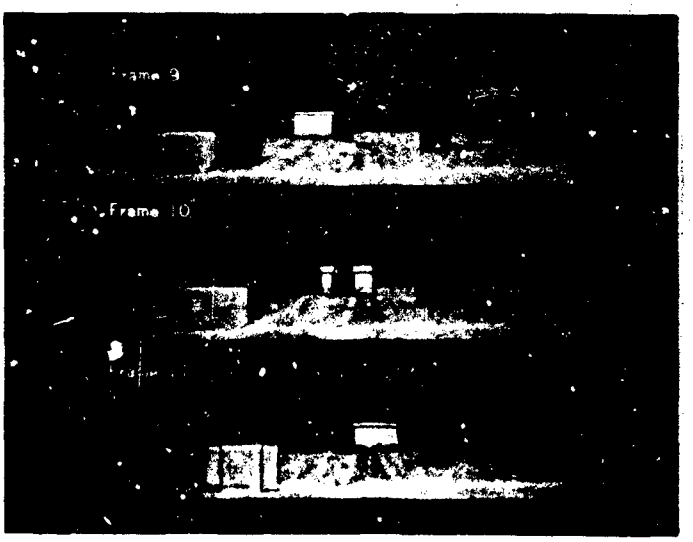
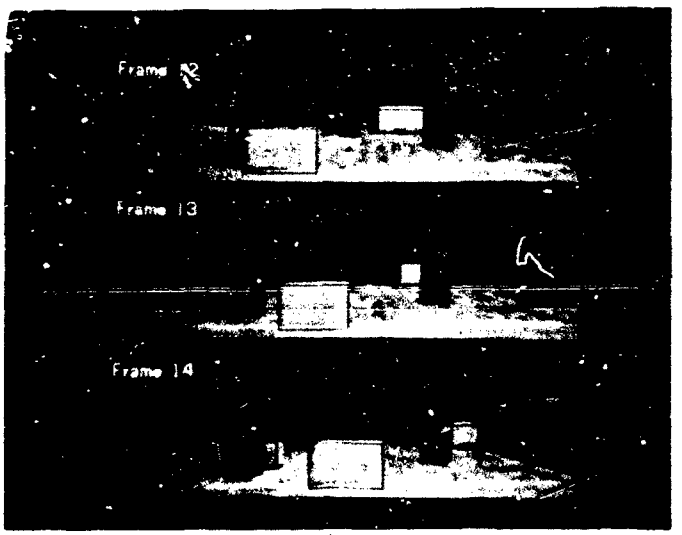


Figure 12 (cont.) -

m)

n)

o)



p)

q)

r)

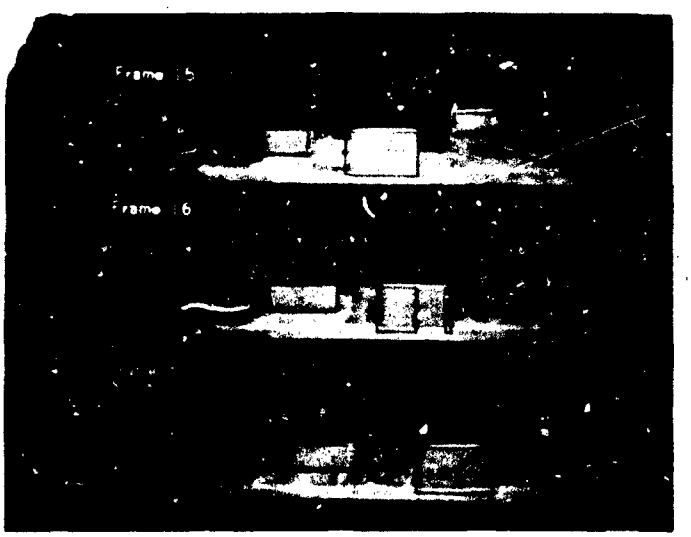


Figure 12 (cont.) -

s)

t)

u)

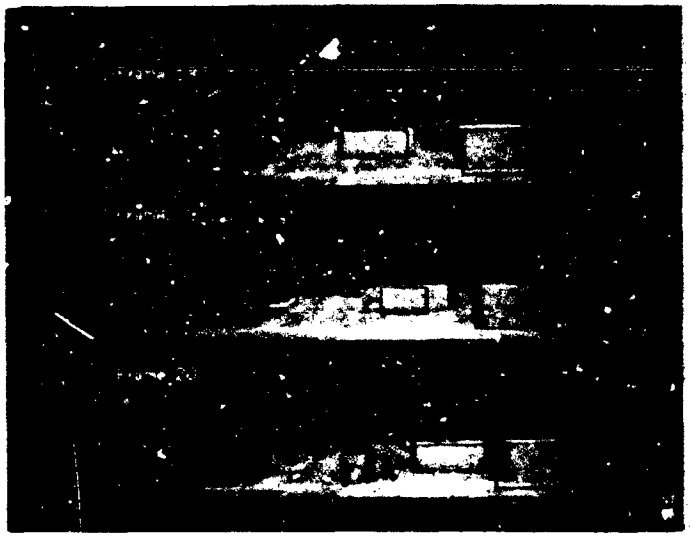


Figure 12 (cont.) -

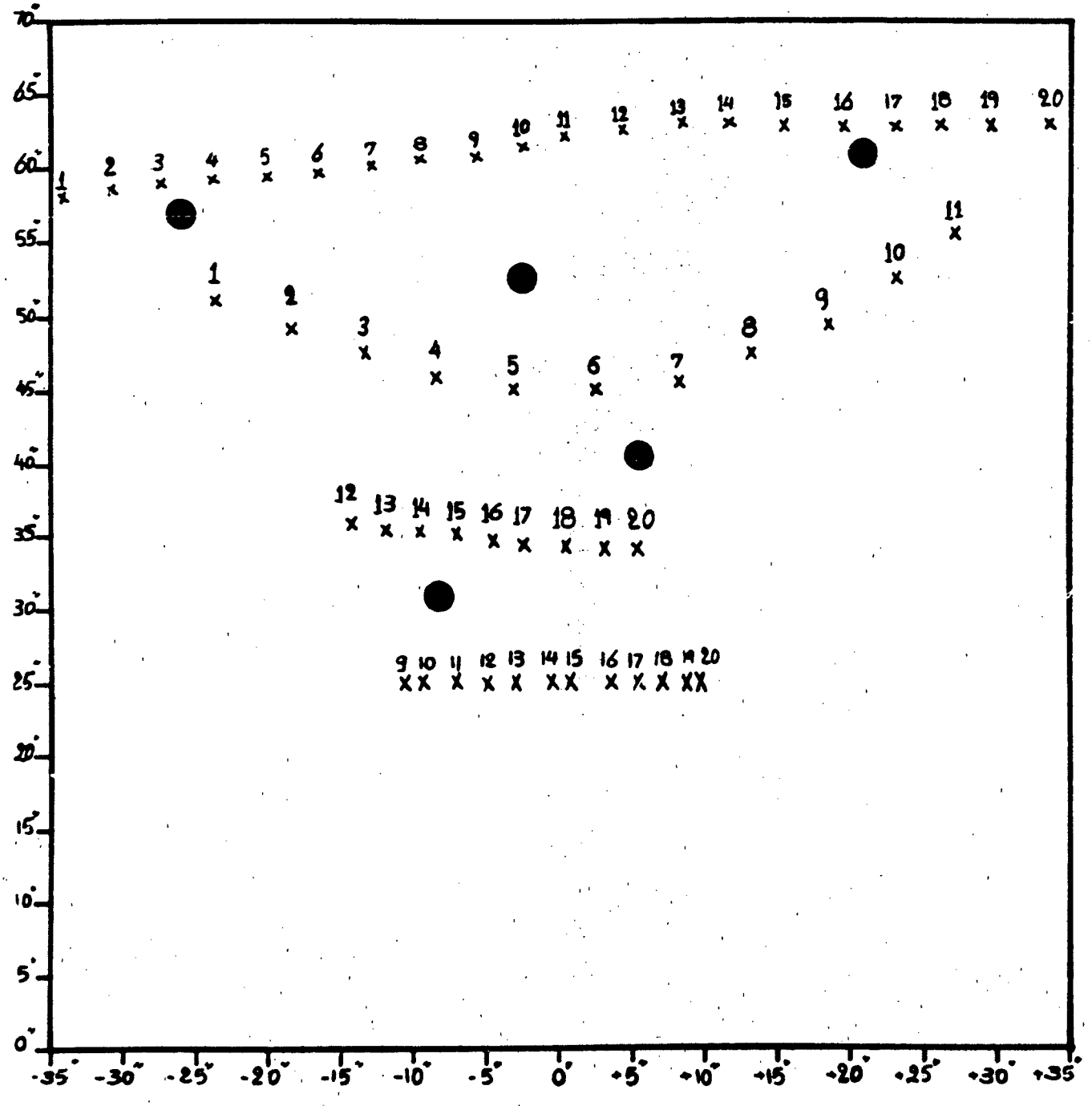


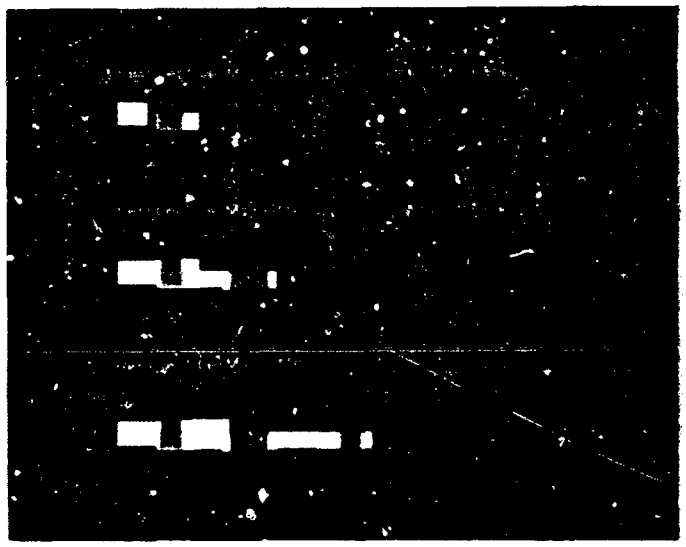
Figure 13.

This figure represents the ground map where all the points are measured. "X" represents location of each moving target with the particular frame number above it. The black circles represent the location of the stationary objects.

a)

b)

c)



d)

e)

f)

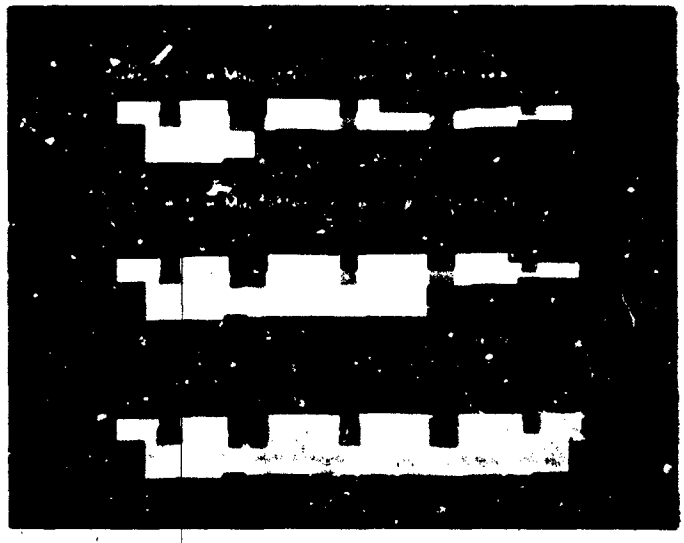
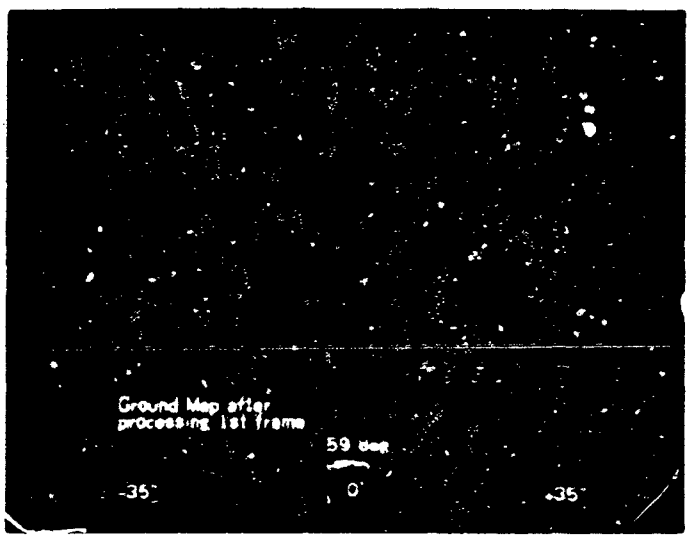
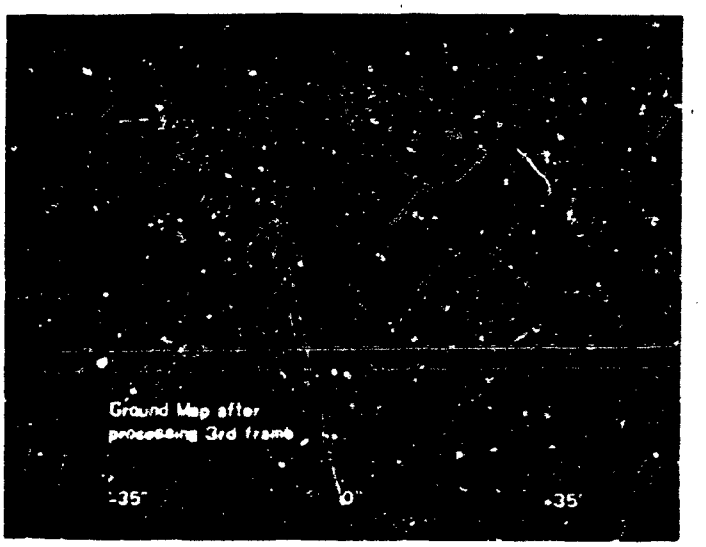


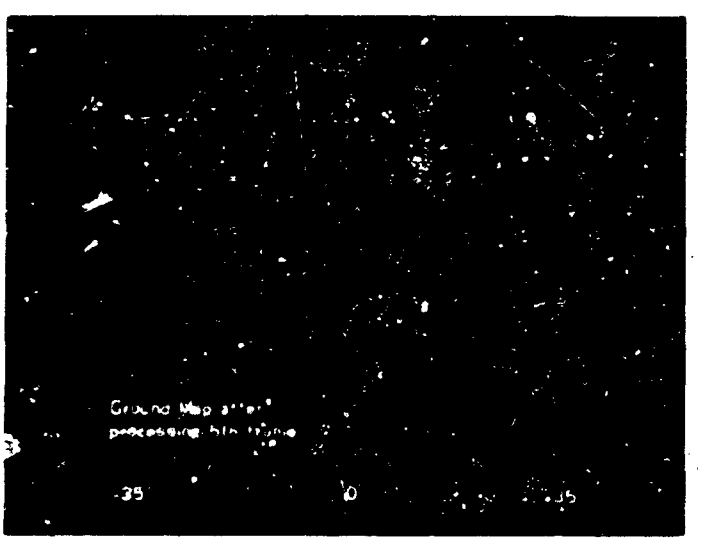
Figure 14. Some snapshots of the Perspective Map.



a)



b)

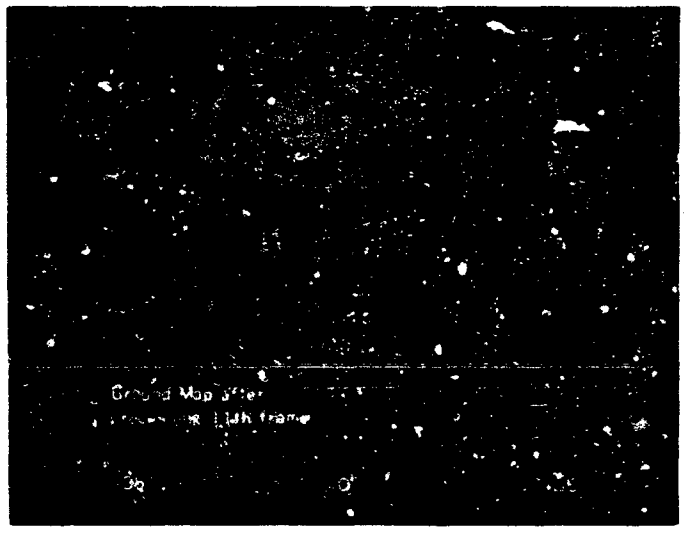


c)

d)

Figure 15. Some snapshots of the Ground Map.

e)

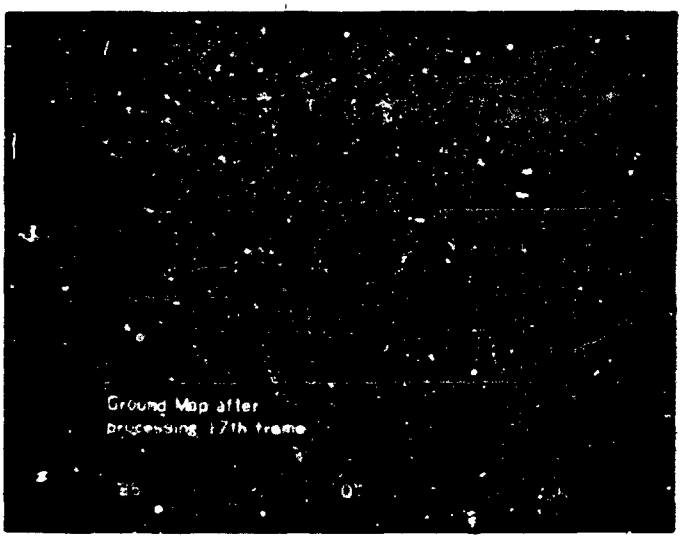


f)

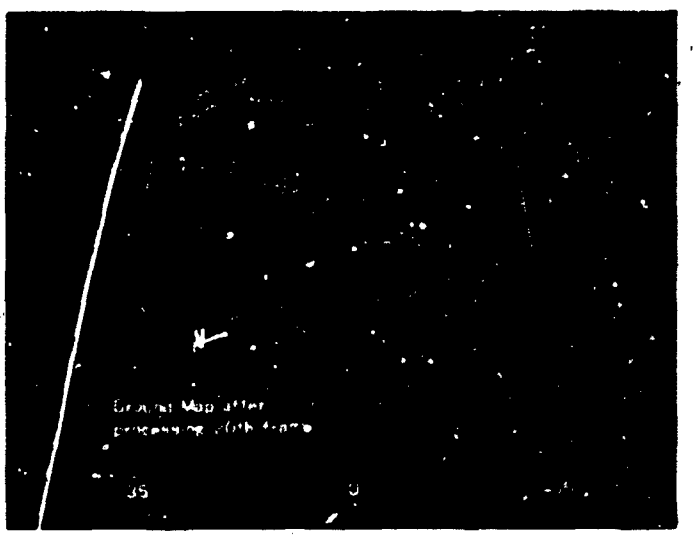
g)

h)

Figure 15 (cont.) -



i)



j)

Figure 15 (cont.) -

END

FILMED

10-85

DTIC