N91-20656

Autonomous Proximity Operations Using Machine Vision For Trajectory Control and Pose Estimation

Dr. Timothy F. Cleghorn
Mail Code PT4
NASA/Johnson Space Center
Houston, TX. 77058

Dr. Stanley R. Sternberg
Machine Vision International, Inc.
912 N. Main Street
Ann Arbor, Michigan, 48104

Abstract

A machine vision algorithm has been developed which permits guidance control to be maintained during autonomous proximity operations. At present this algorithm exists as a simulation, running upon an 80386 based personal computer, using a ModelMATE CAD package to render the target vehicle. However, the algorithm is sufficiently simple, so that following off-line training on a known target vehicle, it should run in real time with existing vision hardware. The basis of the algorithm is a sequence of single camera images of the target vehicle, upon which radial transforms have been performed. Selected points of the resulting radial signatures are fed through a decision tree, to determine whether the signature matches that of the known reference signature for a particular view of the target. Based upon recognized scenes, the position of the maneuvering vehicle with respect to the target vehicle can be calculated, and adjustments made in the former's trajectory. In addition, the pose and spin rates of the target satellite can be estimated using this method.

INTRODUCTION

In order to perform a rendezvous and docking operation in space, it is necessary to determine the attitude and attitude rates of the target vehicle, as well as the relative position and trajectory of the maneuvering craft with respect to that target vehicle. These parameters are obtained currently by using Shuttle astronauts' eyes to guide the maneuvering craft to the desired position so that a grapple with the Shuttle Remote Manipulator System,(RMS), can be performed by a crew member. In the future, it will be desirable to perform these operations with increasing degrees of autonomy; particularly satellite servicing, and Lunar and Martian orbiter rendezvous. In order to do this, a full array of sensors will be required; however it is likely that vision will remain as the major source of input data. One of the chief drawbacks of any sensing system based upon vision data is the sheer number of those data, with the correspondingly long computation times required to process the input. It is therefore very important to develop methods of data compression which permit analyses in keeping with the time scale defined by the characteristic motions of the target/sensor system in question. An algorithm has been developed which permits small errors or drifts in trajectory to be identified and corrected, based upon the view of the target vehicle as seen by a single camera on a maneuvering craft. This algorithm is demonstrated on a PC computer with EGA or VGA graphics.A CAD/CAM system, (ModelMATE, by Generic Software, Inc.), has been used to model the target vehicle. Current vision hardware includes Imaging Technology's PC-Vision frame grabber mounted in a COMPAQ 286, and a Sony XC-57 CCD camera. This is scheduled to be upgraded to an ASPEX PIPE machine attached to a Sun 4in the near future. High fidelity graphics models will be included, and solid models will also be employed. Figure 1 illustrates one view of the target, a (somewhat fanciful) Hubble Space Telescope. It is assumed that the target object is located within the field of view of the camera, and that the target is recognized by the system; i.e., target identification is not the issue, although the techniques described herein could well be used for that purpose also. This algorithm utilizes the radial signatures of a sequence of images to determine a calculated position and trajectory for the maneuvering craft.

The complete program consists of two parts: an off-line training phase, and a series of run-time calculations, as the maneuvering craft approaches the target vehicle. The training phase presupposes the existence of an accurate three-dimensional CAD model of the target vehicle, and typically runs for two days on an 80386 type computer for the level of accuracy used in this work. The training phase consists of the building of decision trees which permit the association of a radial signature of the target's image with an angular orientation of the target vehicle with respect to the maneuvering craft. Details of the training process will be presented in the next section.

Following the off-line training, a "desired" rendezvous trajectory is selected. It is assumed that the angular orientation of the target craft is known to within an accuracy of about 20 degrees at some initial time t0. An angular normalization is made around the camera-target axis to align the image axes with those used during the training phase. Radial signatures of successive images are extracted as the maneuvering vehicle attempts to fly its desired trajectory, and these signatures are normalized to correspond to those used during the training phase. Points on these radial

signatures are fed into a decision tree to determine whether the camera "recognizes" the view. It is normal that for each image,several adjacent views are recognized. Based upon the linear extent of an image compared to a reference image, the apparent distance between the camera and the target can also be calculated. Thus a sequence of images generates a "point cloud", through which a curve or apparent trajectory can be fit. This permits the next segment's trajectory to be predicted, and corrections to be made to drive it closer to that which was planned originally. In addition, or as an alternative, is possible to calculate the target vehicle's attitude and attitude rates. These are necessary parameters for an autonomous docking to be performed.

PROCEDURE

Reference Frame Construction

During both the training and production phases of the algorithm, the relative positions of the target and observing crafts are defined by constructing a geodesic sphere around the target. This virtual sphere is attached to the target vehicle, and the observing craft moves on or outside of the surface. If the observing craft moves inside of the geodesic, a new sphere must be constructed in order to account for distortion. It will be assumed that the geodesic encloses the entire target vehicle. For each node, or line intersection on the sphere's surface, a characteristic view is stored. Actually, using a relatively new technique which will be discussed below, the critical information for a given node is compressed to be only a few numbers, typically six to eight. These numbers are stored in a hierarcical decision tree for each node on the sphere. The geodesic sphere is constructed by repeated bisections of a regular icosahedron, (a twenty-sided polydedron). Each surface of the icosahedron is an equilateral triangle. By connecting midpoints of the edges of each triangle, four new triangles are constructed. If the icosahedron is considered to be the zeroth order sphere, the number of surfaces on an ith order sphere is given by:

$$1) \qquad nfaces_i = nfaces_0 * 4^i$$

where $nfaces_0 = 20$

In terms of the i-1 order geodesic,

$$1a) \qquad nfaces_i = 4 * nfaces_{i-1}$$

Similarly, the number of edges of an ith order geodesic is given by:

$$2) \qquad nedges_i = 1.5 * nfaces_i$$

Each triangle on the surface of the geodesic has three edges, each one of which is shared by one adjacent triangle, hence the factor 1.5. The number of vertices, or nodes is given by:

$$3) \qquad nnodes_i = nnodes_{i-1} + nedges_{i-1}$$

where $nnodes_0 = 12$ for the zeroth order icosahedron.

The density of nodes will determine both the accuracy of the pose calculation and the computer time required for training. It was found that a third order geodesic, with 642 nodes and 1280 faces was a good compromise between accuracy and computing time.

Signature Construction

Having established a coordinate frame, it is necessary to find those parameters which will identify a view of the target uniquely from any location within the space on or outside of the surface of the geodesic. Binary thresholding permits the most rapid computation. In addition to providing the radial signature of the target vehicle, as described below, the binary image allows calculation of the distance of the camera from the target. During training, the areal extent, Aref, of the target image is recorded for each of the 642 nodes. The linear distance, from the centroid of the target image to the camera is given by:

$$4) \qquad d_{calc} = d_{ref} * \mathrm{sqrt}\,(A_{ref}/A_{obs})$$

where d_{ref} is a reference distance, (the radius of the geodesic), and A_{obs} is the observed area of the target image.

Both the training and the on-line or production portions of the program utilize the radial transform to reduce the raw data from the image of the target vehicle to a level which can be dealt with by an AT-class machine. The implementation of the radial transform is a fairly straight-forward procedure, which has been coded in C in order to conform to several available hardware machine vision systems. The transform itself consists first of locating the centroid of the binary image of the target vehicle. Care must be taken to insure that the binary image outline corresponds to the grey level outline of the vehicle, and in fact one future project will be the development of software to permit the binary image to be reconstructed should this correspondence fail due to lighting or other problems. Following location of the centroid, the radial distances to the outermost edge of the binary image is measured. The simulation demonstration uses 294 radial measurements, corresponding to the 294 vertical bins on an EGA graphics screen. The hardware implementation for the PC-Vision board uses 360 radial bins, starting at East,(bin 0), and running counterclockwise. The radial signature of the target is obtained by plotting these distances as a function of bin number. (Figures 2a-b).

Decision Tree Construction

The 294 or 360 bins still represent too large a number of data to analyze,either during the training or on-line phases of the program. For each of the 642 nodes

of the geodesic we wish to have no more than about 10 characteristic features which will identify the node. It is assumed that the relative angular position is known approximately, so that there is no ambiguity between polar symmetric nodes. Additionally, a statistical approach is taken: it is desired that each node's state be classified correctly between 95% and 98% of the time.

For the training phase, each of the 642 nodes is labeled. The camera is assumed to be located on the surface of the geodetic. In order to train for a specific node, the radial signature of that node, plus those for a number of surrounding points are obtained. The surrounding points are selected to be the mid-points of the edges, up to three edge lengths away from the central node, (Figure 3). It is desired that all views up to and including one edge length's distance from the central node be recognized, and all views between one and three edge lengths not be recognized. As can be seen in Figure 3, 73 radial signatures are extracted for each node, of which 19 are "in", and 54 are "out". This operation can be thought of as applying a series of perturbations to the target object. The views seen by the camera will "wobble" about a central axis.

It is desired to select those particular radial bins which will identify the view from the given node most rapidly. A decision tree must be constructed, the terminal branches of which label the node as being "in" or "out", at some level of certainty. There are two general types of classifiers which can be used to separate a data set into components. These include single stage classifiers, such as Bayes linear and quadratic classifiers, Fisher's linear classifier, thresholding the principal feature, or thresholding a component. All of these classify the data into two or more classes in a single step. The present work uses a new technique of classification called a hierarchic classifier. This method can be described as a binary decision tree, in which each terminal branch represents one pattern class, and the non-terminal nodes of the tree represent a collection of classes. The root node represents the entire collection of classes. When an unknown datum enters the hierarchic classifier at the root node, a decision rule associated with the root node is applied to it to determine the next node to which it should go. This process is repeated until a terminal node is reached. Each terminal node has an associated class to which that datum is assigned.

In order to implement a hierarchic classifier a decision rule must be constructed for each node of the tree. A decision rule is a single-stage classifier, such as any one of the types mentioned above. The simplest of these is that which thresholds a component of the data. Thus the construction of the entire decision tree involves three steps: choosing the decision rules at each node of the tree, finding different ways of branching from a non-terminal node to its child nodes, and finding the termination condition for the branching process. The branching condition at each non-terminal node is based on a criteria of minimum entropy or minimum classification error. At each node of the tree, consider a threshold for each data component for all samples of the data. This threshold partitions the data into two classes, those with component values less than the threshold, and those with values greater. The entropy is then computed for left and right partition classes. If the decision rule is effective, these values will be significantly different. If L_i is the number of feature vectors in category i classified to the left child, and R_i is the number classified to the right, the entropy H_i is defined as:

$$5)\quad H_i = L_i * \ln(L_i/L) + R_i * \ln(R_i/R) - (L_i + R_i) * \ln((L_i + R_i)/(L + R))$$

where $L = L_i$, and $R = R_i$. The index i takes on the values "on" and "off".

The entropy is computed for all components of the data and for all thresholds that can partition the data into two classes at each node of the decision tree. The threshold and the component which gives the minimum entropy are considered to be the appropriate ones for that node.

The branching process is terminated when one of the following conditions is met. If the number of samples falls below a certain minimum, the entropy calculation is meaningless. If all samples at a particular node fall in one category, the branching process is stopped, and the class of the node is assigned to that category. Also, if the entropy calculated by equation (1) falls below a certain minimum, there is no significant difference between right and left partitions. In this case, the right and left children are merged into one node. It was found that by using these criteria to determine when to terminate the branching process, the view recognition accuracy was consistently within the desired 95 and 98 percent rate.

The decision tree can be represented in the computer as a series of if-then-else statements. Consider a set of data with three components, (r1, r2, r3). Five samples have component values as follows:

Sample	r1	r2	r3	Category
s1	0.6	1.0	1.0	2
s2	0.4	1.0	0.8	1
s3	0.6	1.0	0.8	2
s4	0.6	1.2	0.8	1
s5	0.6	4.4	0.8	2

Table I

The categories are assigned here simply as left child or right child at the terminal node. Figure 4 illustrates the resulting decision tree. The thresholds are given for each non-terminal node, and the resulting classification appears at the terminal node for each sample. The advantages of the decision tree approach are first that it identifies which components are important, and second, it is faster than the single-stage classifier

techniques once the training phase has been completed. It also can be expressed readily in an Expert System format:

```
IF (r2 < = 1.1) {
  IF (r3 < = 0.9) {
    IF (r1 < = 0.5)
      ASSIGN Category = 1 ; terminal node left
    ELSE
      ASSIGN Category = 2 ; terminal node right
    }
  ELSE
    ASSIGN Category = 2
  }
ELSE
  {
    IF (r2 < = 2.3)
      ASSIGN Category = 1
    ELSE
      ASSIGN Category = 2
  }
```

Figure 5 illustrates two decision trees constructed for separate nodes on the geodesic for the Hubble Space Telescope. "T" stands for a terminal node. If the view is recognized, the value assigned to the terminal node is 1; otherwise it is 0. The radial vector is the first number in the inequality, the threshold value is the second. Thus "2 #156 < = 455" can be read as "If radial vector #156 has a value less than or equal to 455, then...." The initial integer "2" refers to the level within the decision tree. There are two points to observe in Figure 5. First, once a terminal node has been encountered, the calculation is finished. This speeds up the algorithm considerably. Second, note that one of the decision trees is quite long compared to the other. To understand physically what is occurring, consider a thin flat plate of somewhat irregular shape. If viewed nearly edge on, a slight wobble or perturbation will cause the outline or signature of the plate to change significantly. However, if viewed from a point nearly perpendicular to the plate, the same amount of wobble will change the outline or signature only slightly. Thus some viewing directions are vastly simpler than others to identify. The price paid to use the hierarchic classifier is that a decision tree must be constructed for each of the 642 nodes on the geodesic surface. The total time needed to do this was about two days, using an AT-class machine.

Decision Tree Application

In the preceding section, the procedures used to train the classifier have been discussed. Following the training, the second phase of the algorithm takes place, namely its application using images from unknown directions. It must be assumed however, that the target vehicle's pose is known to about 20 degrees at the initial time t_0; otherwise the time it takes to locate a group of recognized "on" nodes will exceed that which it generally takes for the pose to change to some new, and still undetermined value.

There are two initial corrections which must be applied to each of the images. The first of these, the distance correction, has already been discussed, (equation 4). In some cases it was necessary to add a correction for the difference in focal length between the reference and the flight images. The distance equation then becomes:

4a)

$$d_{calc} = d_{ref} * \mathrm{sqrt}(A_{ref}/A_{obs}) * (\mathrm{image\_focal\_length} / \mathrm{reference\_focal\_length})$$

The other initial correction is for rotation about the line-of-sight between the crafts. Again, this assumes an approximately known initial pose.

Having made these corrections, the radial signature of the unknown image is is extracted, and applied to the decision trees of all of the nodes in the neighborhood of the approximate position on the geodesic. If in fact the camera lies somewhere within this region, some of the nodes should recognize the view, that is, they should be "turned on". One of the major advantages of the hierarchic classifier approach is that with several of the nodes being activated simultaneously, if one or two should be missed, the position can still be calculated. Thus an element of robustness against bad lighting conditions, reflections and background is built into the method. Using the distance correction obtained from equation (4), a calculated position in three dimensional space is found for each "on" node. For each image, there are typically five such points. As the maneuvering craft moves with respect to the target, the process is repeated, with new images generating new points, forming what is referred to as a point cloud along the trajectory of the maneuvering craft. The position of the maneuvering vehicle is then calculated using a multi-dimensional minimization procedure called the "Downhill Simplex" algorithm. For a discussion of this method see Press, et al, 1988. This can be thought of as analogous to a four dimensional best fit through the point cloud. The orbits of the maneuvering vehicle were calculated in segments, in order to be able to determine how far that craft was from the desired path. For the cases of circular or spiral rendezvous, one radian segments were chosen. This permitted drift errors to be detected, and the path to be adjusted before the errors became too great. Thus new paths were planned for successive segments, allowing the maneuvering craft to stay close to the desired trajectory.

In addition to the circular and spirial trajectories, this was done using an actual Space Shuttle V-Bar approach trajectory. This required using equation (4a) to determine the distance correction, and image distortion also became a serious problem. As for the circular and spiral cases, it was possible to correct for distortion to some extent by constructing a virtual geodesic with a smaller radius, even to the point of enclosing just a portion of the target vehicle. This relearning obviously becomes very expensive computationally, and really defines one of the limits of usefulness of the algorithm.

In addition to being able to calculate the trajectory for the maneuvering craft, it is possible to calculate the attitude or pose, and attitude rates for the target vehicle. In fact, if the two vehicles are at constant distance from each other in some global coordinate system, the attitude/attitude rate calculation is entirely equivalent to the maneuvering vehicle trajectory determination. The six numbers describing the pose and spin of the target vehicle are needed for an autonomous docking or grappling to occur. There-

fore, the hierarchic classifier approach has a much wider potential application than was originally intended.

CONCLUSIONS

A new method of determining the trajectory of a maneuvering craft with respect to a target vehicle has been described. This method utilizes a hierarchic classifier with input data from a single camera, to calculate either the trajectory of the maneuvering craft, or to determine the pose and spin parameters of the target vehicle, or both. The advantages of this method are that it is faster during on-line calculations than the single-stage classifier methods, it is robust with respect to partial or noisy input data, and it identifies the important components of the target image. The algorithm also runs on commonly available computer systems.

Currently, the algorithm exists as a simulation demonstration, with some pieces having been ported to a hardware machine system. It is planned to continue this porting process, and demonstrating the algorithm using physical models, as well as actual images of satellites in space. This latter will permit testing of the robustness of the algorithm; both Earth and space backgrounds will appear in the images, as well as shadows and reflections on the target vehicle.

ACKNOWLEDGEMENTS

This work is the direct result of SBIR Contract NAS 9-17814 with Machine Vision International, of Ann Arbor, and subcontract to the University of Washington, in Seattle. Without the full and generous cooperation of personnel from both organizations, this work would not have been possible. Special thanks go to Mr. William Dargel, of MVI, and to Professors Robert Haralick and Linda Shapiro, of the University of Washington.

REFERENCES

Press, William H., Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling, "Numerical Recipes in C", Chapter 10, p305ff Cambridge University Press, 1988

Sternberg, Stanley R., "Integrated Computer Vision Research for SpaceConstruction -- FINAL REPORT Vol I", Machine Vision International, Inc., Ann Arbor, Michigan, November, 1989

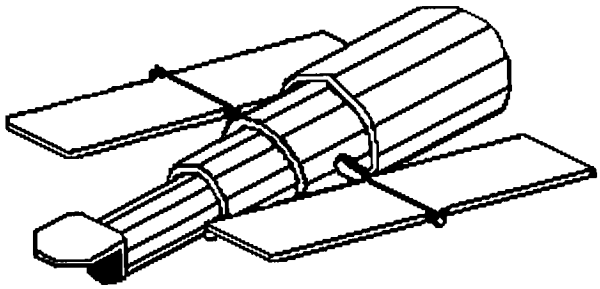

FIGURE 1

Space Telescope

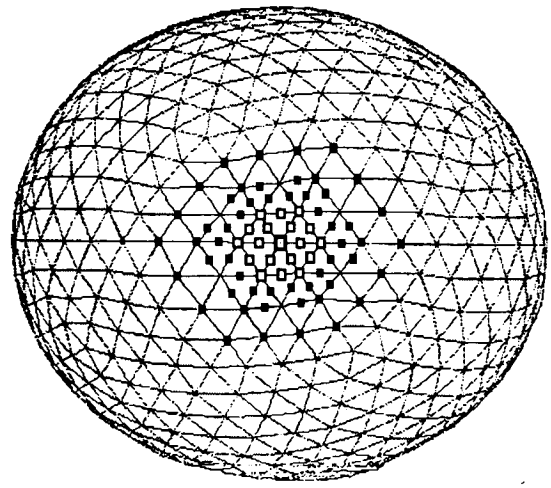

FIGURE 3

Training view positions for a node. There are 19 "ON" positions (open boxes), and 54 "OFF" positions (closed boxes) for training each of the 642 nodes on the geodesic.

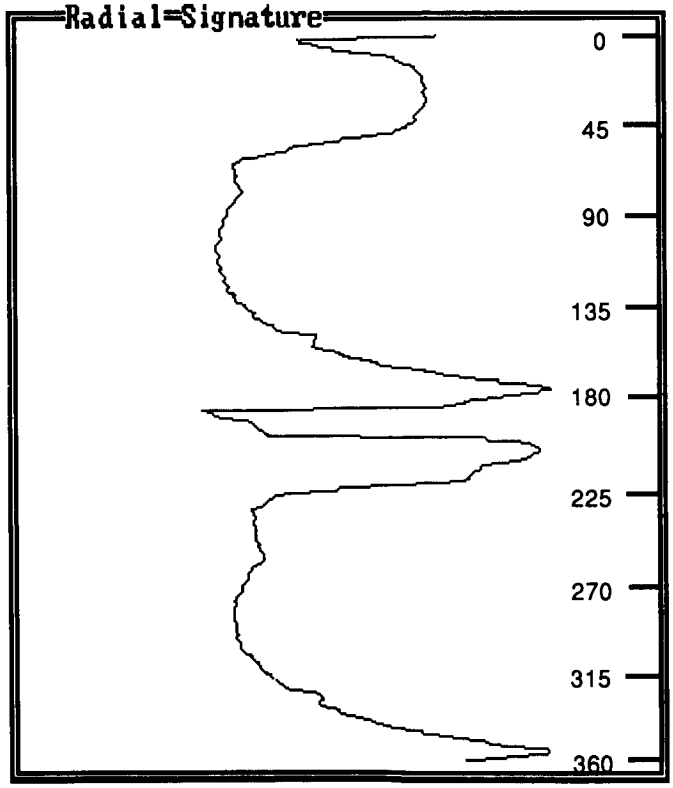

Figure 2a

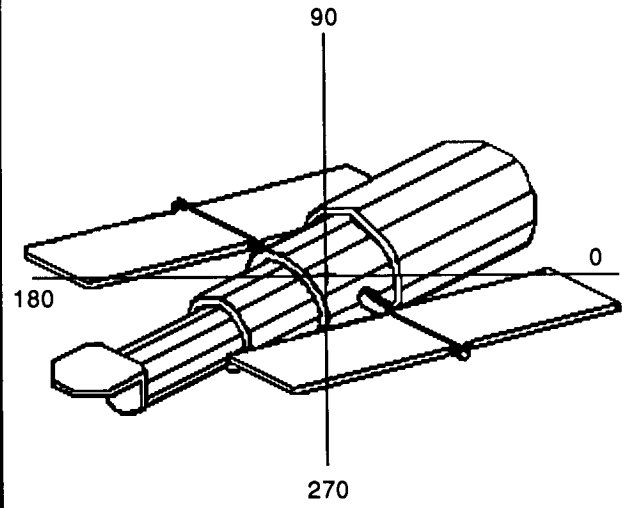

Figure 2b

The radial signature, (Fig 2a), is obtained from the binary image of Figure 2b, by measuring the radial distance from the centroid (+) to the outermost edge of the object.

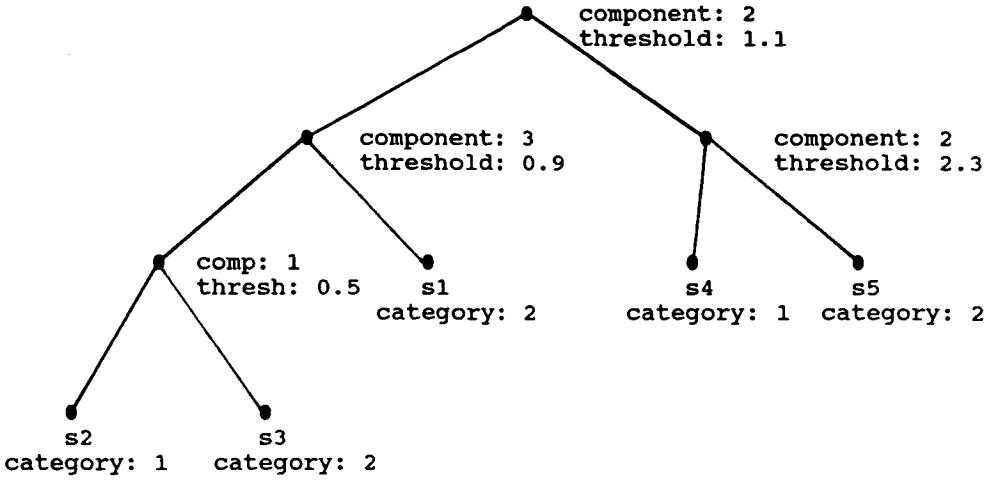

Figure 4

A decision tree for the data in Table I is illustrated. Left branches represent component values less than the threshold at a branching node, whereas right branches represent component values above the threshold. The sample is assigned to the category (left or right) at the terminal node.

```
  Node   12

0 #187 <= 486
  1 #259 <= 459
    2 #7 <= 1293
      3 T 1
      3 T 0
    2 #18 <= 983
      3 #69 <= 380
        4 T 1
        4 T 0
      3 T 1
  1 #189 <= 349
    2 T 1
    2 T 0
```

```
                Node 13

0 #149 <= 830
  1 T 0
  1 #176 <= 616
    2 #150 <= 1146
      3 #170 <= 515
        4 T 0
        4 #293 <= 1044
          5 #31 <= 480
            6 T 0
            6 #117 <= 415
              7 #182 <= 477
                8 T 0
                8 T 1
              7 T 1
          5 T 0
      3 T 0
    2 #286 <= 1444
      3 T 0
      3 #176 <= 660
        4 #0 <= 995
          5 T 1
          5 T 0
```

Figure 5

Two decision trees used in the operational phase. One is short, representing a relatively unambiguous view of the target, whereas the other is long, which indicates that the view from that node is difficult to recognize. The node numbers do not indicate relative locations of the two views.