

# Precision of a radial basis function neural network tracking method

Jay Hanan,<sup>\*</sup> Hanying Zhou, Tien-Hsin Chao

Jet Propulsion Laboratory (JPL), 303-300, 4800 Oak Grove Drive, Pasadena, CA USA 91109

## ABSTRACT

The precision of a radial basis function (RBF) neural network based tracking method has been assessed against real targets. Intensity profile feature extraction was used to build a model in real time, evolving with the target. Precision was assessed against traditionally measured frame-by-frame measurements from the recorded data set. The results show the potential limit for the technique and reveal intricacies associated with empirical data not necessarily observed in simulations.

**Keywords:** Autonomous Tracking, Neural Network, Target Recognition, Data Reduction.

## 1. INTRODUCTION

Automatic visible light tracking applications have employed binary filters or threshold templates to follow high- or low-intensity objects within the field of view. If linear processors were employed, the tracking method was kept simple to allow for necessary speed requirements under most applications. Optical processing was available for performing tracking procedures more advanced than binary in real time. An example of high speed real-time tracking technology is the Grayscale Optical Correlator (GOC) developed at JPL.<sup>1</sup> The GOC uses an advanced algorithm to identify the object of interest reducing false alarms, providing a robust tracker. While the advancing speed of linear processors has not allowed digital computers to overtake optical processors, recent advances have freed the researcher to attempt advanced tracking algorithms at video frame rates. In addition, software implementation versus hardware implementation results in applications and application times which are less costly than previously attainable.<sup>2</sup> Typical frame rates for off-the-shelf products range from 1-10 fps using NTSC frames with some specialty products reaching 30-60 fps. These higher rates take full advantage of the typical video camera sensor.

Tracking capability can be broken up into categories of precision. The simplest category tracks the object at the scale of the entire object. Relevant accuracy is dependent on true or false identification of the object and its general location. This has been attempted using Radial Basis Function (RBF) Neural Net (NN) tracking algorithms.<sup>3,4,5</sup> While such tracking technology has many useful applications, coordinates associated with such tracking algorithms may move across the object with time at multiple frequencies creating an unstable trajectory. Many trajectory observations require higher resolution than the scale of the object tracked. In order to obtain this resolution without a complex, slow technique, a RBF NN tracking algorithm was provided refined feedback allowing more accurate tracking. An analysis of the useful refinements and resulting improvements in tracking follows.

Generally an automated tracking method is graded on its ability to identify the existence and position of the intended target. Tracking is considered successful if the object is identified quickly without a large number of false identifications.<sup>6</sup> However, depending on the nature of the target, the position given by the tracking method may be difficult to grade. For example, objects intended for tracking usually consist of several pixels and show multiple orientations. The position could be defined as a center of mass, a leading edge, or through a specific tag. To reduce uncertainty, an automated tracking method was tested against on real targets from recorded video. The recorded video is part of a data base of manually tracked objects used as a testing reference.<sup>7</sup> Here a tracking method based on a

---

<sup>\*</sup>Corresponding author; electronic mail: Jay.C.Hanan@jpl.nasa.gov; phone 818-354-4820

Radial Basis Function (RBF) Neural Network (NN)<sup>3</sup> was analyzed and compared to the manual results with particular attention to the precision given by the tracking method.

## 2. METHODOLOGY

One of the advantages of NN application to tracking is the tolerance for change in the target morphology. This NN tracking method continuously learns so that, as the target evolves, tracking continues. Another powerful capability of this architecture is the ability to track an object without preconceived knowledge. Input to the network is the initial morphology of the object when it is first presented—taken from the first frame in a sequence of frames increasing with time. How the network calculates a final coordinate and calculates when the object has changed significantly influence its ability to accurately track the object. These calculations were examined using the following methods.

### 2.1 Measurement Description

To eliminate effects of multiple feature extraction techniques or varying vector lengths, a common profile based feature extraction technique was used for all frames. In addition, the region of interest was consistently maintained at  $32 \times 32$  pixels<sup>2</sup> with the region of search kept to  $100 \times 100$  pixels<sup>2</sup>. Thus, the study is applicable to other tracking scenarios regardless of vector length or feature extraction method.

To calculate the performance of each tracking method, the coordinates of the tracking result were compared against the coordinates of the manually measured position in pixels. For each tracked object, a position or tag was located on the object and used as a reference for the position of that object. For each frame this tag was manually recorded as a reference to compare with automatically obtained trajectories. In the case of most objects, the nose or leading edge, of the object was used as the tag (see Figure 1). Reference coordinates,  $(X_{ref}, Y_{ref})$ , for the tag were recorded in pixels along the Cartesian axis of the  $640 \times 480$  pixel<sup>2</sup> frame,  $i$ . The absolute difference between the NN tracked coordinates from the reference coordinates were recorded by component.

$$|(X_{NN}, Y_{NN})_i - (X_{ref}, Y_{ref})_i| = (\Delta X, \Delta Y)_i \text{ For } i = 1, 2, \dots, 300$$

An average and standard deviation of the sum of this difference  $(\Delta X, \Delta Y)_i$  over all  $i$  was calculated for each tracked sequence. Improvements in a particular tracking method is reflected in a decrease of these values.

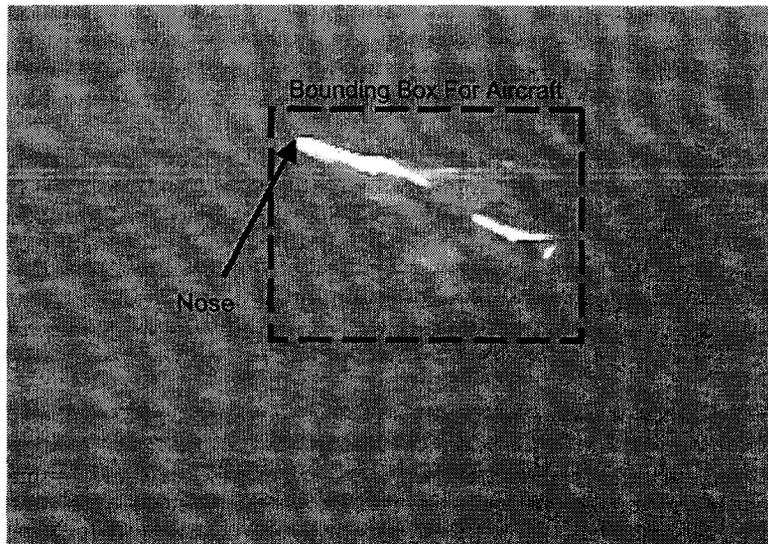


Figure 1 Example frame of sequence used to test network tracking aircraft.

## 2.2 Coordinate Optimization

One decision area analyzed for effect on the tracked object includes the averaging method for the set of positive recognitions of the object given by the network. Since in general the network returns multiple valid identification results for each scan of an object in a frame, the coordinates of the individual results may be sorted and averaged to improve the relevance of the resulting final coordinates. Four methods were tested on the sequences of frames including:

- 1) Average
- 2) Median
- 3) Average weighted by radius
- 4) Average weighed by age of the neuron

The *average* provided a coordinate including equal influence from all positive identifications of the object. The *median* returned the coordinate in the middle of the set of possible coordinates. The *average weighted by radius* weighted the average according to the distance from the center of the firing neuron. The *average weighted by age of the neuron* similarly weighted the average according to the number of frames the neuron had been in existence (younger neurons carried more weight).

Finally the effect of including counterexamples in the tracking procedure was investigated. Counterexamples were automatically selected from the positions around the trained position. Counterexamples reduce the radius of the relevant neuron(s) by setting all distances less than the counterexample distance from the neuron as background.<sup>3</sup> For the test sequences, the coordinates for the automatic counterexamples were varied from 1 to 3 pixels away from the center of the positive example. The displacement value,  $D$ , represents a component of the distance from the positive example the counterexample was taken. Four automatic counter examples were taken, one for each quadrant around the tag in regions ( $X+/-D$ ,  $Y+/-D$ ).

To establish a baseline (or default method) to compare the results from several methods, the following standard method was used:

- Average all results provided by all neurons recognizing the object.
- Add a neuron to the network at the location of this average when less than five results are given by the network in a frame.
- Let the initial radius of all neurons remain throughout the processing sequence (no counterexamples).

## 2.2 Auto-Tracking Simulation

To determine the maximum potential of the automatic tracking system, a recognition engine was constructed to simulate the tracking network. Simulation of automatic tracking was extended to highlight the source of errors generated by automatic tracking. Simulations using a fixed number of neurons starting with one neuron—also used as the input for the automatic tracking test—up to 20 neurons spread evenly throughout the sequence were tested (See Table 1). Neurons were dedicated according to the coordinates provided by the manual tracking measurements. The spacing between manually committed neurons remained constant as the number of neurons increased. The performance provided by the simulation was expected to outperform the automatic tracking system since more “expert” information was provided to the network.<sup>3</sup> Thus a useful performance limit may be assigned to automatic tracking.

### 3. RESULTS

Tests were performed first on a target with high symmetry. This first target was tracked equally well by the simulation and the automatic tracker which both showed excellent agreement with the manual tracking results (Figure 2). Variation of the tracking method did not show significant improvement for the high symmetry target.

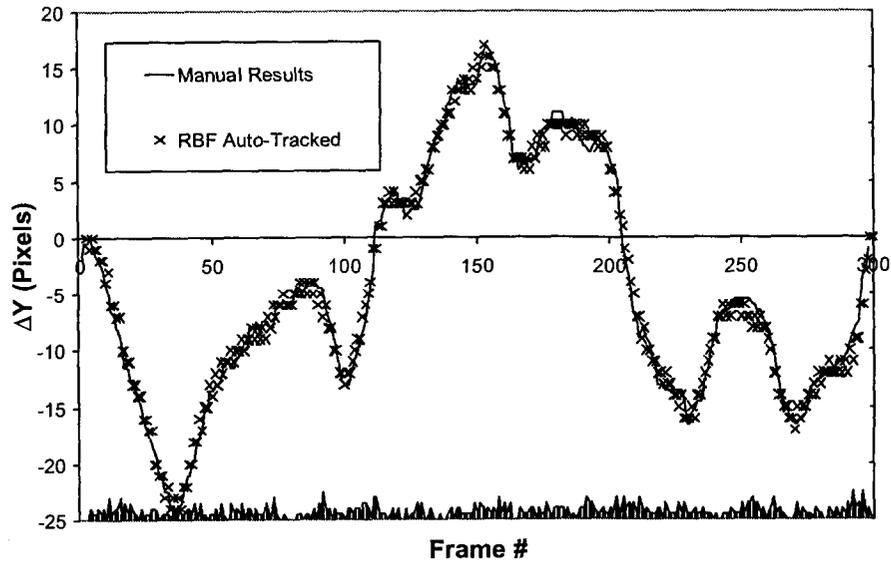


Figure 2. An example of the tracking results for the high symmetry target. The manually measured trajectory along the y-axis is shown by a line with an "x" marking the position identified by the automatic tracker. The difference between the two remains small throughout the sequence as shown by the difference curve along the lower edge of the graph.

A more complex target (as shown in Figure 1) provided an opportunity to observe deviation between the simulated and automatic tracing results. Using the default method, the aircraft was tracked first using the entire aircraft and second using only the nose of the aircraft. The resulting trajectories are shown on the graphs in Figure 3.

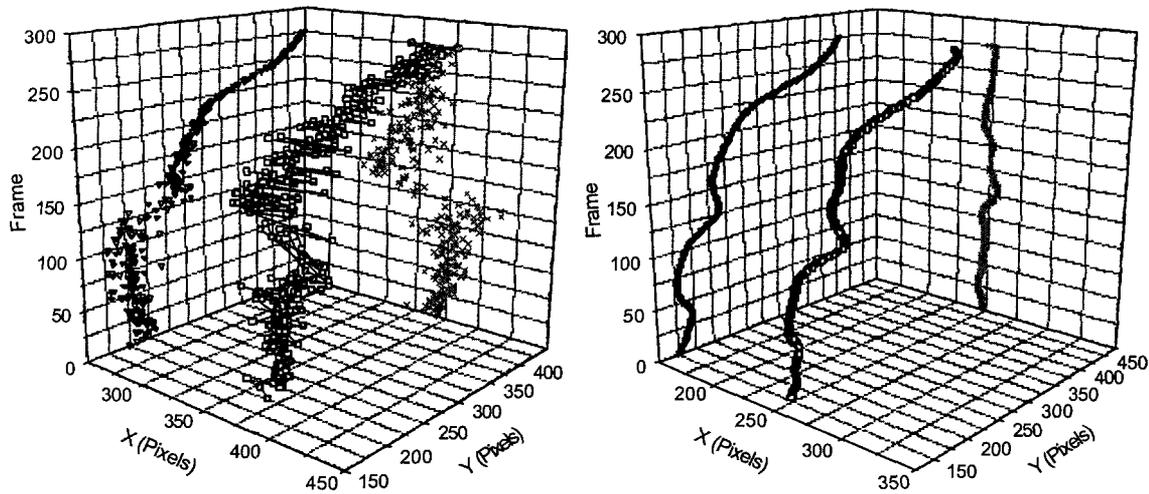


Figure 3. Trajectory (square symbols) obtained from tracking the entire aircraft as input to the network (left) versus limiting the network to tracking only the nose of the aircraft (right) using the default method. Projections of the X (green "x"s) and Y (blue triangles) components of the trajectories are also shown.

For the more complex target, the simulation provided an upper limit to the possible accuracy of the automatic target tracking methods. The performance of the simulation for automatic target tracking of the complex target is summarized in Table 1.

Table 1. Comparison of the difference between simulated and manually tracked nose coordinates for the complex target sequence for varying size networks. Continuity only occurs if all frames in the sequence are successfully processed.

Network Size (neurons)	Average (pixels)	Standard Deviation (pixels)	Maximum Deviation (pixels)	Continuity
1	2.7	2.2	14	No
4	2.9	1.3	9	No
7	2.5	1.2	8	Yes
19	2.7	1.3	8	Yes

The simulation shows the best performance when 7 neurons are dedicated to the nose category of the NN. This corresponds to providing a new expert example to the network once every 50 frames. For sequential processing, increasing the number of expert neurons reduces the speed of the recognition. Increasing beyond 7 shows no significant improvement in performance and in some cases slightly decreases the accuracy (see Table 1). In addition, while 1 neuron shows a good average performance over the sequence, its maximum deviation toward the end of the sequence is significant and does not provide continuous recognition of the nose.

### 3.1 Final coordinate calculation

By varying the simulation method according to the options described in section 2.1 the effects of each method may be compared for ultimate application to the automatic tracking procedure. Using the 7 neuron simulation and an automatic counterexample with a 1 pixel offset, the four coordinate calculation methods were compared (See Table 2).

Table 2. Comparison (in pixels) of the difference between 7 neuron simulated and manually tracked nose coordinates for the complex target sequence for four possible methods.

	<b>Average</b>	<b>Median</b>	<b>Radius Weighted</b>	<b>Youth Weighted</b>
<b>Average</b>	2.3	2.9	2.5	2.4
<b>Standard Deviation</b>	1.3	1.4	1.2	1.3
<b>Maximum Deviation</b>	8	9	7	8

The method with the least deviation was weighting by the radial distance (column 3 Table 2); however, two other methods, average and youth weighted, show less difference with the manual results in spite of their larger maximum and standard deviations. The difference between methods is slight (less than 1 pixel), but could become significant if sufficient automatic tracking was conducted.

### 3.2 Automatic Counterexamples

As observed by comparison of Table 1 and Table 2, counterexamples have an effect on the recognition provided by the NN. However, the significance of counterexamples is better explored under automatic tracking conditions. Using the average sorting method, the change in results when counter examples are used is shown in Table 3.

Table 3. Effect of increased displacement of counterexamples for automatic tracking. The displacement value,  $D$ , represents the distance from the trained example the counterexample was taken. All units are pixels.

<b>D</b>	<b>Average</b>	<b>Std. Deviation</b>	<b>Max. Deviation</b>
None	2.6	2.2	13
1	3.5	2.0	14
2	3.1	1.9	13
3	3.1	1.9	13

The relatively low values observed when no counterexample was used reflects a good choice for the initial neuron radius. In addition, as can be seen in Table 3, the standard deviation decreases with the use of counterexamples and has a stronger influence than a change in method represented earlier in Table 2.

### 3.3 Auto-Tracking vs. Simulation

The trajectory for the auto tracking results summarized in Table 3 by the 2 pixel displacement were compared with the predicted trajectory given by the 7 neuron simulation with the average weighted by radius (Figure 4). For the comparison, the  $\Delta X$  and  $\Delta Y$  were summed for both the auto tracking trajectory and the simulated trajectory. The difference for these sums at each frame is plotted in Figure 4. This figure clearly shows the auto-tracking trajectory agrees with the simulated trajectory for the first 200 frames. Beyond 200 frames, the tracking becomes unstable and begins to deviate from the exact target position increasing in error.

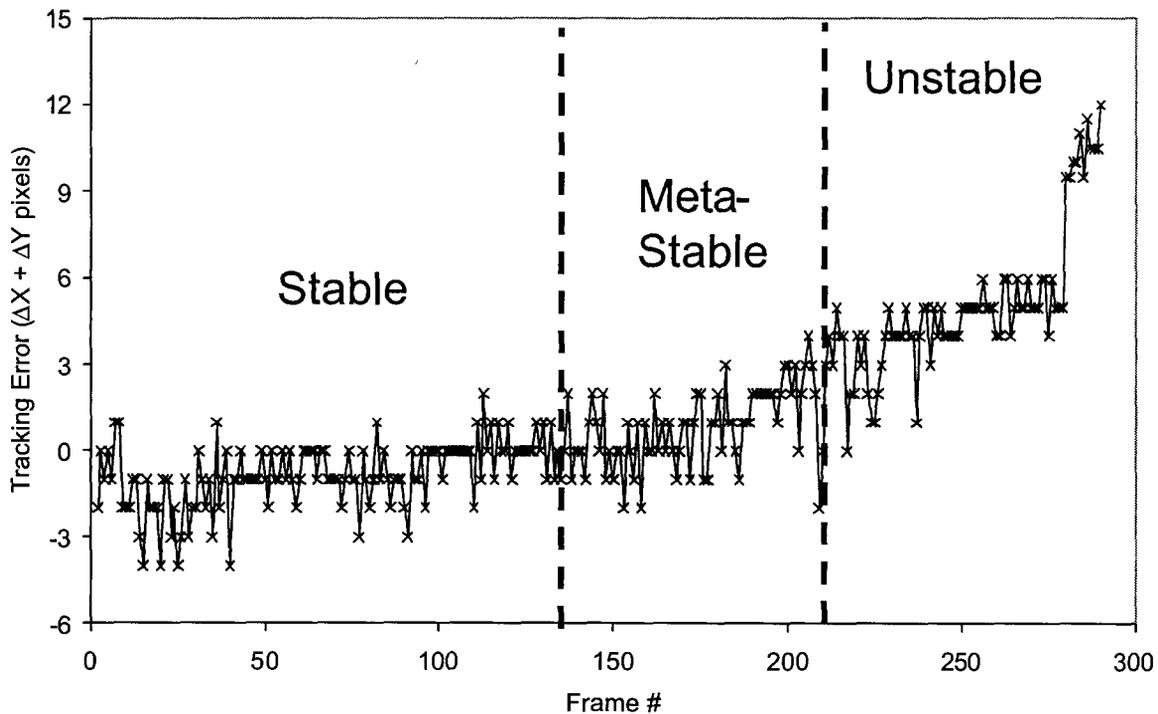


Figure 4. Comparison of auto tracking results with simulated tracking results. First  $\Delta X$  and  $\Delta Y$  are summed for both the auto tracking result and the simulated results. The difference, plotted here, reveals the differences between the auto tracking trajectory and the simulation.

Tracking stability persists over many frames. Eventually erroneous neurons begin entering the network and, by frame 250 in Figure 4, push the tracked position 4 pixels away from the originally intended coordinate. The instability is partially due to rotation of the tracked object and a build up of error associated with neurons too far removed from the original example. Further examination of the instability and methods to overcome it are being explored.<sup>5</sup>

#### 4. CONCLUSION

A radial basis function neural network based automatic tracking method was tested against real targets from digital video. The tracking results compared favorably to precise manual tracking measurements and approached the theoretical limit simulated for the trajectories tested. From the results of precision measurements several conclusions were obtained. Initially, high symmetry targets did not benefit from improved averaging or sorting methods applied to the possible neural network tracking results. Second, tracking a subset of a complex object improves the resulting trajectory measurement. In addition, simulation of the tracking method provided an upper limit to the accuracy. In the case of a complex object the average standard deviation was reduced to 1.2 pixels. As opposed to the high symmetry object, weighting with respect to radial distance provided the best decision for the complex object position. Finally, autonomous addition of counterexamples reduced the standard deviation during tracking but may propagate unwanted errors over extended times.

#### ACKNOWLEDGEMENTS

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

## REFERENCES

- 
- <sup>1</sup> T. Chao, Hh. Zhou, G. F. Reyes, J. C. Hanan, "Real-time automatic target recognition using a compact 512 x 512 grayscale optical correlator," *SPIE: Optical Pattern Recognition XIV*, (2003).
  - <sup>2</sup> P. Sauvan-Magnet, "IMAQ and LabVIEW Automate the Study of Eye Motion," *User Solutions: Physiological Testing*, www.ni.com (1997).
  - <sup>3</sup> M.T. Musavi, W. Ahmed, K.H Chan, K.B. Faris, D.M. Hummels, (1992), "On the training of radial basis function classifiers," *Neural Networks*, **5**, 595-603.
  - <sup>4</sup> R. D. Lamm, D. F. Specht, "Real-Time Reinforcement Learning For Target Tracking," *ANNIE* (2001).
  - <sup>5</sup> J. C. Hanan, T. Chao, *et al.* "Optimized radial basis function neural net automatic tracking algorithm" *NASA, NTR. In progress* (2003).
  - <sup>6</sup> Z. Zhao, "Failure Criterion of Concrete under Triaxial Stresses using neural Networks," *Computer-Aided Civil and Infrastructure Engineering*, **17** (2002) 68-73.
  - <sup>7</sup> C. E. Tapp, J. Kilmer, WSMR NM. *Personal Communication*, (2002).