JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 23, No. 4, July–August 2000

Neural-Network-Based Autonomous Star Identification Algorithm

Jian Hong* and Julie A. Dickerson†
Iowa State University, Ames, Iowa 50011

Most of the existing autonomous star identification algorithms use direct-match algorithms that prestore the star feature vectors in a database. During recognition, the measurements are compared with the reference feature vectors in sequence or by using binary-tree search. The computation time for the star recognition with a traditional model-based system is high, and it increases as the number of the feature patterns in the database increase. We propose an autonomous star identification algorithm using fuzzy neural logic networks. This is a parallel star identification algorithm with fast training speed. The simulation results based on the SKY2000 star catalog (Myers, J. R., Sande, C. B., Miller, A. C., Warren, W. H., and Tracewell, D. A., "The SKY2000 Master Star Catalog," ***AAS/AIAA Space Mechanics Symposium*****, AAS, San Diego, 1997, pp. 1–16) show that the proposed system can achieve both high recognition accuracy and fast recognition speed. Errors due to star magnitude measurement imprecision can also be minimized.**

Introduction

STAR observation is widely used by spacecraft for attitude determination. Star sensors measure star magnitude and star coordinates in the spacecraft frame. The measurements are then compared with a reference star catalog to obtain the attitude information of the spacecraft. Traditional methods use a priori attitude estimates to identify observed stars. However, the coarse attitude may not always be available. Budget restrictions are also pushing spacecraft designers to reduce the redundant sensor hardware. This means that autonomous star identification is becoming more important. With autonomous star identification, system failures can be reduced, and the spacecraft is less dependent on ground communication links.

Many of the existing autonomous star identification systems use direct-match algorithms that store the star feature vectors in a database.[1,2] During recognition, the measurements are compared with the reference feature vectors in sequence[2] or by using binary-tree search,[3] until a correct match is found. The computation time for star recognition with direct-match algorithms is high, and it increases as the number of the feature patterns in the database increases.

Fuzzy logic and neural networks can help deal with these problems. The parallelism of neural networks can help speed pattern search, and fuzzy logic can help deal with measurement uncertainty. However, star recognition is very different from other pattern recognition problems in that the number of star patterns is quite large. For example, the number of star triplet patterns in the submillimeter wave astronomy satellite (SWAS) catalog is 17,533. Therefore, when applying neural-network models to star recognition, network complexity and recognition speed are two major concerns. Many neural-network models are restricted by these limitations. For example, the Hopfield network has found successful application in object recognition.[4] The Hopfield network requires full connections among the nodes, which leads to N^2 nodes; in this case it leads to $17{,}533^2$ nodes. This quickly leads to very large and complex networks. The number of patterns stored in a Hopfield network is limited by the number of neurons in the network.

The neural logic network (NLN) is a neural-network model that combines the strength of both neural networks and rule-based systems.[5] We propose an autonomous star identification system that uses a fuzzy NLN. The algorithm used by the fuzzy NLN for pattern matching is called supervised clustering and matching (SCM).[6] During training, a specific hidden node learns and encodes the input pattern only when both the input and output patterns match well enough. This gives the algorithm fast training capabilities. With the fuzzy NLN, no a priori attitude knowledge is required. The simulation results based on data from the SKY2000 master star catalog[7] show that fuzzy NLN has the advantage of fast learning speed and high accuracy in star pattern recognition. Other fast learning methods such as radial basis function networks were explored as part of this work.[8] However, these networks took a long time to train and had low accuracy rates because there are many similar patterns in the star triplets and the function defined by the input–output pairs is not a continuous function.

NLNs

The NLN is a neural-network model that combines the strength of both neural networks and rule-based expert systems. Artificial neural networks are computational models that have the capability of learning and adaptation, but they cannot reason with symbolic languages. Rule-based expert systems simulate human problem solving process in the form of heuristic rules, but they can not adapt to the new inputs. An NLN combines the two systems to incorporate human problem solving skills with the area of pattern recognition and information evaluation.[5]

An NLN builds up its inference engine with the neural network as its underlying structure, so that it can simulate human decision making behavior as well as learn adaptively. In star identification, the NLN is used as a neural-network model, where parallel pattern learning and recognition is most important.

There are three types of NLNs: Boolean NLN, three-valued NLN, and fuzzy NLN.[6] In a Boolean NLN, 0 is used to denote true and 1 false. In a three-valued NLN, {(1, 0), (0, 1), (0, 0)} are used as the truth values, where (1, 0) denotes true, (0, 1) denotes false, and (0, 0) denotes unknown. In a fuzzy NLN, the truth value domain A is defined by[6]

$$A = \{(x, \bar{x}) : x \in [0, 1],\ \bar{x} \in [0, 1],\ x + \bar{x} = 1\} \tag{1}$$

where x represents the on response and $\bar{x} = 1 - x$ is the complement of x, representing the off response. Complement coding preserves both the on response and the off response to an input vector. It shows if a feature is present or absent and to what degree. This helps the fuzzy NLN handle measurement uncertainty.

The structure of a fuzzy NLN for star pattern recognition resembles a three-layer feedforward neural network as seen in Fig. 1. The hidden nodes are called cluster nodes because one hidden node may

Received 3 June 1998; revision received 26 October 1999; accepted for publication 2 November 1999.

*Graduate Student, Department of Electrical and Computer Engineering; currently Member of Technical Staff, Motorola Semiconductor Systems Division, 1301 East Algonquin Road, Schaumburg, IL 60196.

†Assistant Professor, Department of Electrical and Computer Engineering.

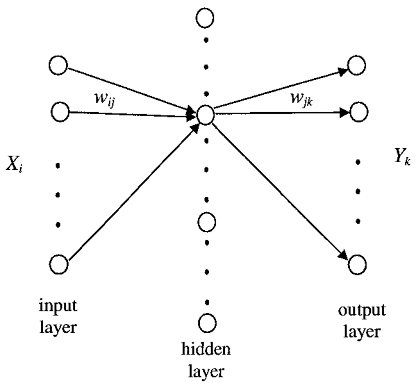

Fig. 1 Structure of a fuzzy NLN.

be used to encode more than one pattern. Each input and output pattern can be represented by

$$X = [(x_1, \bar{x}_1), (x_2, \bar{x}_2), \ldots, (x_M, \bar{x}_M)] \tag{2}$$

$$Y = [(y_1, \bar{y}_1), (y_2, \bar{y}_2), \ldots, (y_N, \bar{y}_N)] \tag{3}$$

where $(x_i, \bar{x}_i)$ denotes the truth value of the ith input node and $(y_j, \bar{y}_j)$ denotes the truth value of the jth output node. The weights w_{ij} and w_{jk} represent templates or prototypes of the input and output patterns, respectively.

The fuzzy NLN is trained with the SCM algorithm.[6] This algorithm is derived from the adaptive resonance associative map,[9] which is a model from the supervised adaptive resonance theory (ART) network family.[10] During training, a cluster node is chosen if both its input and output weight templates match the input and output patterns to a certain degree. The match similarity between the input and the jth hidden node is measured as follows[6]:

$$m_J^x = \frac{\sum_{i=1}^{M}\left(w_{Ji}^x x_i - \bar{w}_{Ji}^x \bar{x}_i\right)}{\left|W_J^X\right|_2 |X|_2}, \qquad m_J^y = \frac{\sum_{i=1}^{M}\left(w_{Ji}^y y_i - \bar{w}_{Ji}^y \bar{y}_i\right)}{\left|W_J^Y\right|_2 |Y|_2} \tag{4}$$

where m_J^x and m_J^y are the match degrees for input and output patterns. Here $(w_{Ji}^x, \bar{w}_{Ji}^x)$ and $(w_{Ji}^y, \bar{w}_{Ji}^y)$ are the Jth weights for the input and output layer, respectively. In Eq. (4), $w_{Ji}^x x_i$ is the correlation measurement between the input pattern x_i and the input weight template w_{Ji}^x, and $\bar{w}_{Ji}^x \bar{x}_i$ is the correlation measurement of their complements. If

$$\sum_i \left(w_{Ji}^x x_i - \bar{w}_{Ji}^x \bar{x}_i\right)$$

is large, then the input pattern is similar to the input weight template. This measurement is then normalized by the multiplication of the L2 norm of the input and weight vectors. The L2 norm function $|\bullet|$ is defined as

$$|X|_2 = \sqrt{\sum_i \left(x_i^2 + \bar{x}_i^2\right)} \tag{5}$$

The match degrees m_J^x and m_J^y must both exceed preset recognition thresholds for the input and output layer, $\rho_x \in [0, 1]$ and $\rho_y \in [0, 1]$, for learning: $m_J^x \geq \rho_x$ and $m_J^y \geq \rho_y$. If both the input and output matches exceed the thresholds ρ_x and ρ_y, the cluster node learns the input and output pattern by updating its weight vectors as follows:

$$\left(w_{Ji}^x, \bar{w}_{Ji}^x\right)^{\text{new}} = (1 - \beta_x)\left(w_{Ji}^x, \bar{w}_{Ji}^x\right)^{\text{old}} + \beta_x(x_i, -\bar{x}_i) \tag{6}$$

$$\left(w_{Ji}^y, \bar{w}_{Ji}^y\right)^{\text{new}} = (1 - \beta_y)\left(w_{Ji}^y, \bar{w}_{Ji}^y\right)^{\text{old}} + \beta_y(y_i, -\bar{y}_i) \tag{7}$$

where $\beta_x \in [0, 1]$ and $\beta_y \in [0, 1]$ are the learning rates for the input and output layer, respectively.

If the node does not satisfy the recognition thresholds ρ_x and ρ_y, the network selects another node. If no existing cluster nodes satisfy the recognition thresholds ρ_x and ρ_y, a new cluster node is added to encode the given input and output pattern. The advantage of using two different thresholds is to allow the network to respond properly to different pattern recognition problems. Higher values can impose stricter matching criteria, which partition the input set into finer categories. Lower vigilance values tolerate greater mismatch, which will result in coarser recognition. In star recognition, we used high input and output thresholds because very similar triplet features may correspond to different triplet patterns due to measurement errors.

System Design

Because a single star does not have enough distinctive features to classify it from thousands of other stars, successful star identification algorithms use features generated by a group of stars. Previous studies have shown that groups of three stars work well for star identification.[3,11] We use star triplets for feature extraction in our system, then the features are sent to the fuzzy NLN to get the recognition result.

Feature Selection

Star features that can be used for recognition include star magnitude, angular separation, the group geometry, or the variations of these features, such as the square of the separation. Star magnitudes can be difficult to determine accurately because different star spectral types can cause shifts in intensity measurements unique to particular imaging systems,[7] and star magnitudes can change over short periods of time. Data from the X-ray Timing Explorer (XTE) mission showed that the observed magnitude difference could be greater than 1.0 relative to the predicted magnitude.[12] After calibration and catalog corrections, this error has been reduced to 0.3 orders of magnitude.[13] Therefore, star magnitude is not always a very reliable parameter for recognition. The angular separation has small variation and can be measured with high accuracy. Therefore, we use the angular separation between the stars in a triplet as the feature vector. The geometry of the triplet is determined by reordering the stars according to their magnitudes. The procedure to generate the triplet feature vector is as follows:

1) Choose a reference star from the central portion of the field of view (FOV).

2) Select the two brightest neighbors within the FOV. For the SWAS CT-601 star tracker,[14] the FOV is 8 × 8 deg.

3) Reorder the triplet from brightest to faintest according to star magnitude, that is,

$$M_1 < M_2 < M_3 \tag{8}$$

Calculate the angular separations between each pair of stars, which is denoted by d_{ij}. Therefore, the input feature vector to the fuzzy NLN network is $\langle d_{12}, d_{23}, d_{31}\rangle$, where d_{12}, d_{23}, and d_{31} are the distances between the three vertices, as shown in Fig. 2.

System Structure

The reference star catalog used by our research is the SWAS star catalog, which is a run catalog created from SKY2000 master catalog.[7] The original SWAS star catalog contains 33,379 stars. This

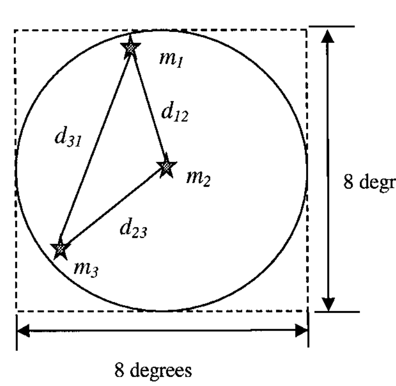

Fig. 2 Star triplet features.

catalog can be downloaded from http://fdd.gsfc.nasa.gov/attitude/skymap.html. Considering only those stars whose magnitudes are brighter than 6.5, the total number of triplet patterns for a sensor with an 8 × 8 deg FOV is 17,533. With this large number of patterns, it is inefficient to train a single neural network. The solution is to divide the problem into smaller NLNs that can be trained more efficiently.

From analysis, we know that the angular separation is a more reliable parameter than star magnitude. The histograms of the three angular separations of the triplet (Figs. 3–5) show that d_{12} (the distance between star1 and star2 in a triplet) is the most evenly distributed attribute; therefore, we used d_{12} to select which NLN to use. During recognition, direct match is first used on d_{12} to decide which fuzzy NLN to use, then the corresponding NLN will give the classification result, as shown in Fig. 6.

Although angular separation is more reliable than star magnitude, with many patterns distributed in a small range, two different patterns may have very similar feature vectors. To improve the identification accuracy, the star magnitudes are used for verification of the final recognition results. The one among the first two winners that has a smaller magnitude error between the observed stars and reference stars is considered to be the final winner. The check on magnitude is especially necessary, when the truth values are very close.

The range for d_{12} is [0.0031, 7.855] deg. Because the CT-601 is a 512 × 512 pixel charge-coupled device CCD imager with an FOV

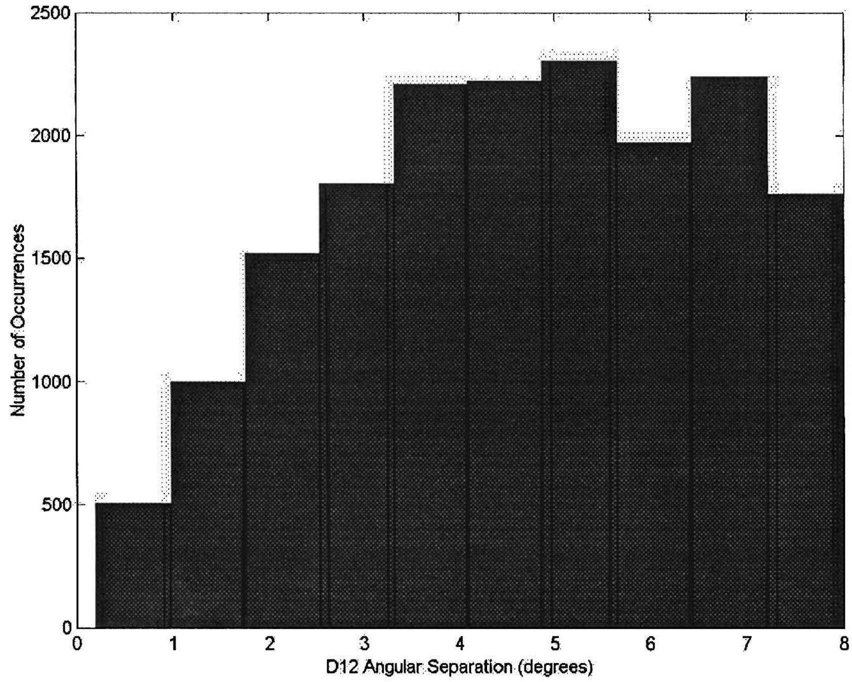

Fig. 3 Histogram of d_{12}.

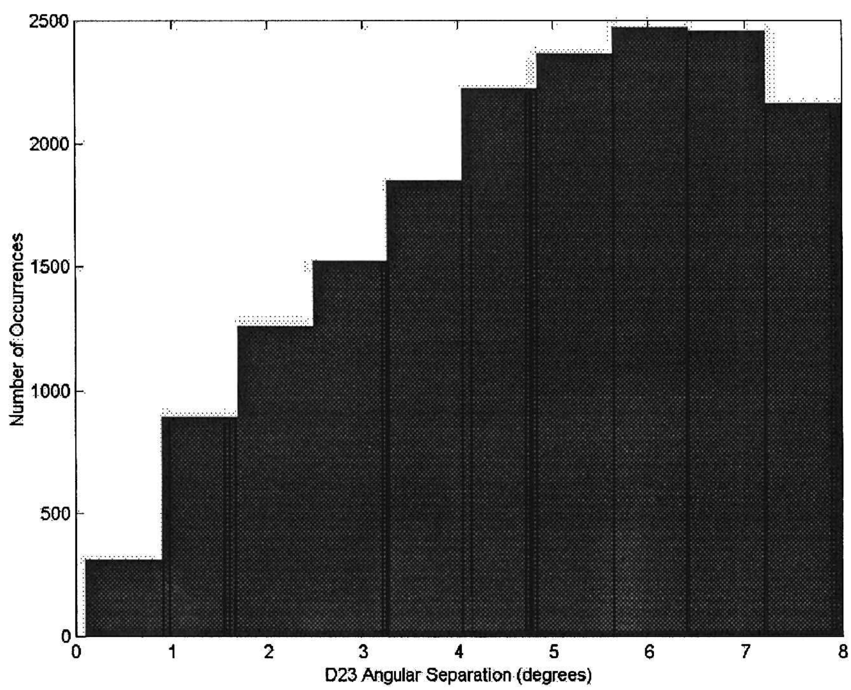

Fig. 4 Histogram of d_{23}.

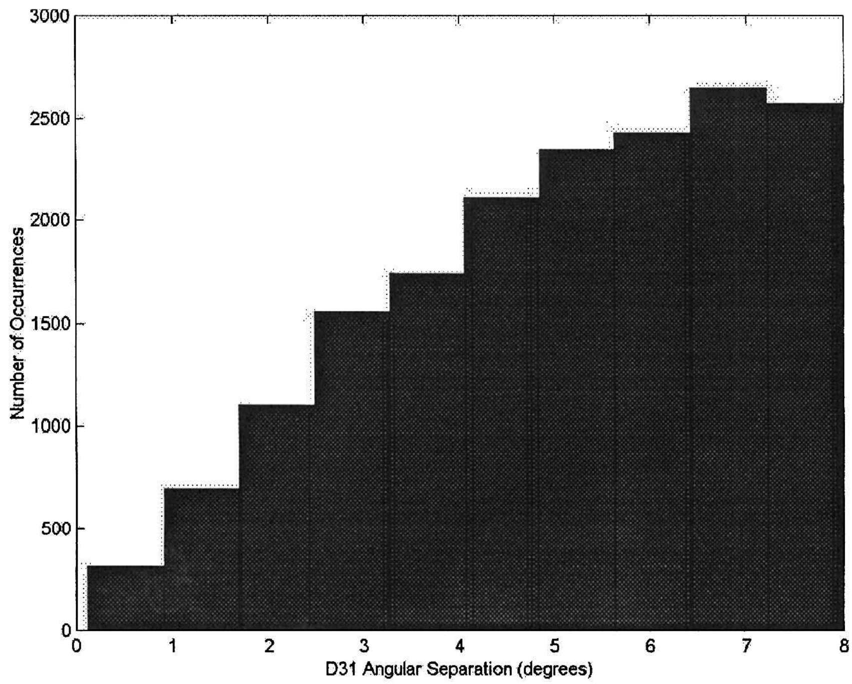

Fig. 5 Histogram of d_{31}.

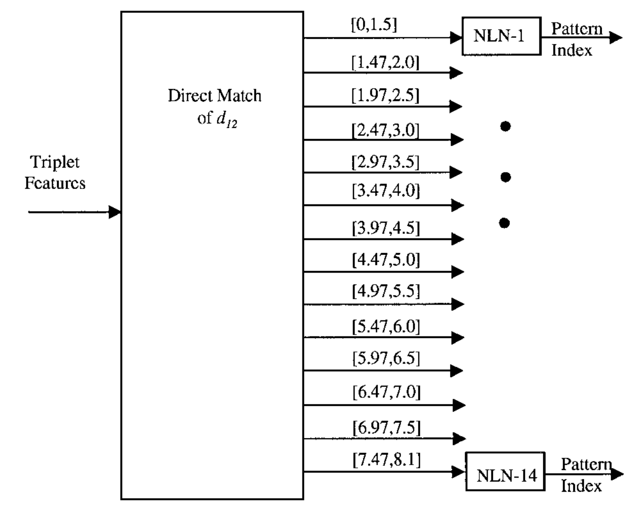

Fig. 6 System structure for star recognition: triplet measurement d_{12} selects which subnetwork NLN to use; numbers in brackets give the range of d_{12} in degrees.

of 8×8 deg, we choose 0.1 deg as the overlap of each interval of d_{12} for error tolerance. The number of triplet patterns in each NLN is (from NLN-1 to NLN-14): 1102, 847, 1126, 1253, 1200, 1498, 1574, 1432, 1616, 1389, 1264, 1476, 1452, and 1198. This is a very quick and efficient method for dividing the total triplet patterns, which also maintains the error tolerance performance. We used an overlap of 0.03 deg (108 arc-s) in this study. This number comes from the sensor accuracy of 60 arc-s (Ref. 14) for unprocessed star data and the star catalog position with a maximum uncertainty of 1.8 arc-s (Ref. 7).

Because of the overlap between each two nearby intervals of d_{12}, one triplet may fall into two NLNs. For example, assume that the original angular separation d_{12} is 2.47 deg. However, it is measured as 2.49 deg. This pattern will fall into two NLNs, NLN-3 and NLN-4, which correspond to the d_{12} intervals of [1.97, 2.50] and [2.47, 3.0], respectively. The correct pattern belongs to NLN-3. The hidden node truth value T_j is calculated:

$$T_j = \frac{\sum_i \left(w_{ji}^x x_i - \bar{w}_{ji}^x \bar{x}_i \right)}{\left| W_j^x \right|_2 |X|_2} \tag{9}$$

Table 1 Magnitude difference of star triplets[a]

Range of magnitude difference	Number of star triplets $\lvert M_1 - M_2 \rvert$ (%)	$\lvert M_2 - M_3 \rvert$ (%)	$\lvert M_1 - M_3 \rvert$ (%)
[0, 0.1]	935 (5.33)	713 (4.07)	6 (0.0003)
[0, 0.2]	1927 (10.99)	1396 (7.96)	50 (0.29)
[0, 0.3]	2993 (17.07)	2109 (12.03)	147 (0.84)
[0, 0.4]	4037 (23.03)	2765 (15.77)	270 (1.54)
[0, 0.5]	4950 (28.23)	3452 (19.69)	406 (2.32)
>0.5	2691 (15.35)	7098 (40.48)	16,654 (94.99)

[a] M_1, M_2, and M_3 are the magnitudes of the brightest, the second brightest, and the faintest star in a triplet.

The hidden node with the maximum truth value T_j is selected. Here, w_{ji}^x is the weight connecting the ith input node and the jth hidden node and x_i is the ith input feature value. $\boldsymbol{W}_j^x$ is the weight vector connecting input vector with the jth hidden node. $\boldsymbol{X}$ is the input vector. T_j is a similarity measurement between the input pattern and the stored pattern. A larger value of T_j means that the input is more similar to the stored pattern. If the truth values are very close, then the magnitudes of the possible star triplets are used to decide which pattern is correct.

Star Identification Algorithm

The magnitude measurement errors have an indirect influence on the performance of the fuzzy NLN because they can cause the star triplet to be improperly ordered. Histograms of the absolute magnitude difference between each two stars in a triplet are shown in Figs. 7–9. Table 1 lists the statistical information of the star magnitude difference based on the histograms.

Table 1 shows the number of potential magnitude errors that would result in a wrongly ordered triplet. For example, if star1 or star2 has a magnitude error of up to 0.3, then the probability of misordering the first two stars in a triplet can be as large as 28.17%. The influence on the order of the first and third star drops to 0.85% when the magnitude error is 0.3. This is a worst-case analysis because the magnitude errors for most stars are relatively small.

A solution for this problem is to present the triplet patterns in different orders to the recognition system if the star magnitudes are close. For example, assume the original feature vector is $\langle d_{12}, d_{23}, d_{31} \rangle$ (mode I); we present it to the recognition system and get solution I. Then we present the feature vector $\langle d_{12}, d_{31},$

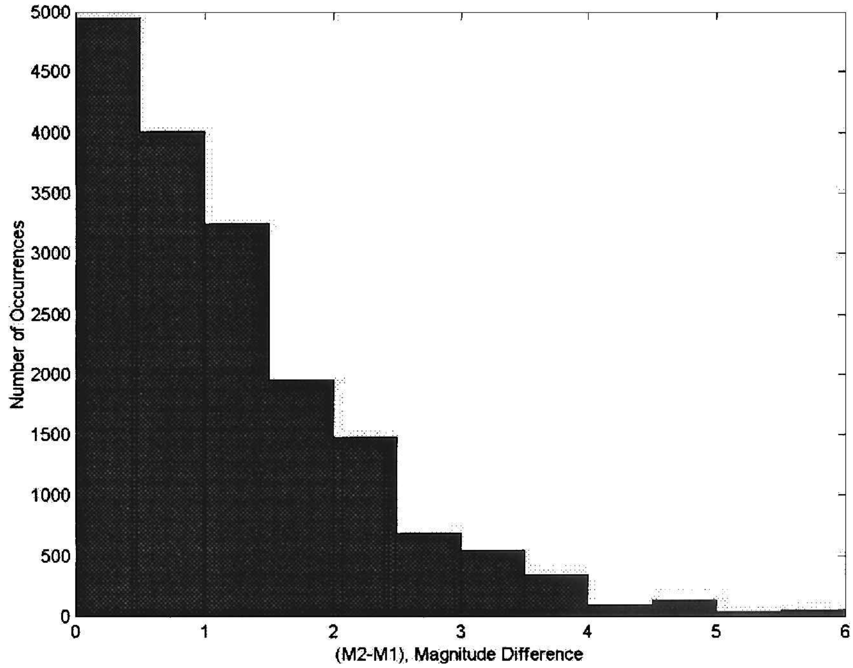

Fig. 7 Histogram of $|M_1 - M_2|$, the absolute magnitude difference between the first and second brightest stars in a triplet.

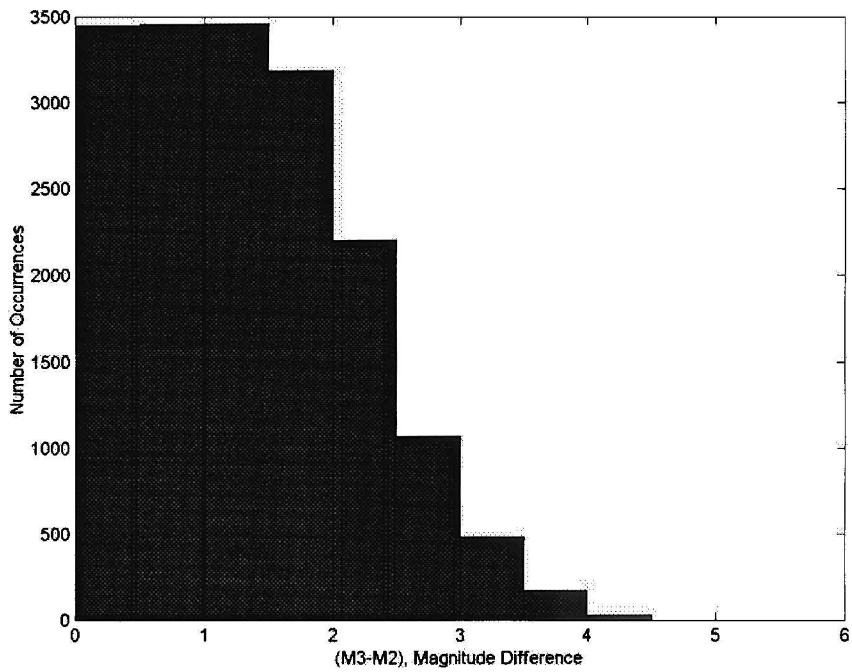

Fig. 8 Histogram of $|M_2 - M_3|$, the absolute magnitude difference between the second and third brightest stars in a triplet.

$d_{23}\rangle$ (mode II) to the system, in case star2 is actually brighter than star1 (thus we switch star1 with star2), and get solution II. Next, we reorder the triplet features to $\langle d_{31}, d_{23}, d_{12}\rangle$ (mode III) and get solution III. This result is possible if star3 is in fact brighter than star2. This gives six modes for one triplet. The other three modes are $\langle d_{31}, d_{12}, d_{23}\rangle$, $\langle d_{23}, d_{31}, d_{12}\rangle$, and $\langle d_{23}, d_{12}, d_{31}\rangle$. The choice of the final solution depends on the maximum hidden node truth value. The solution with the maximum hidden node truth value is considered to be the first winner. This procedure minimizes the influence of the magnitude error on the system performance.

Based on the analysis and system design described, we propose the algorithm for autonomous star identification as follows:

1) Generate a triplet for the reference star that is in the central portion of the FOV. Reorder the triplet according to the star magnitude.

2) Calculate the angular separation for the triplet to get the feature vector: $\langle d_{12}, d_{23}, d_{31}\rangle$.

3) Choose one of the six modes for identification.

4) Use the direct match on d_{12} to choose a proper NLN for identification.

5) If more than one NLN is used, calculate the hidden node truth value of each NLN. Choose the NLN with the largest hidden node truth value.

6) If all of the modes have been tested, choose the two modes with the maximum hidden node truth values. Otherwise, go back to step 3.

7) For each of the chosen modes, compare the first and second maximum hidden node truth values. If the difference is less than a threshold $T_c = 7 \times 10^{-6}$ or $T_c = 7 \times 10^{-2}$, calculate the mean square error of the magnitudes between the observed stars and reference stars. Choose the one with the smallest error as the winner.

Table 2 Results for autonomous star identification

Magnitude error	Distance error, arc-seconds	Number misclassified, $T_c = 7 \times 10^{-6}$	NLN % correct, $T_c = 7 \times 10^{-6}$	Number misclassified, $T_c = 7 \times 10^{-2}$	NLN % correct, $T_c = 7 \times 10^{-2}$
±0	±30.6	2	99.98	00	100
±0.25	±30.6	587	96.6	578	96.7
±0.50	±30.6	1160	93.4	553	96.8
±0	±61.2	11	99.9	46	99.7
±0.25	±61.2	618	96.5	337	98.1
±0.50	±61.2	1157	93.4	594	96.6
±0	±121.4	204	98.8	263	98.5
±0.25	±121.4	768	95.6	508	97.1
±0.50	±121.4	1301	92.6	1104	93.7

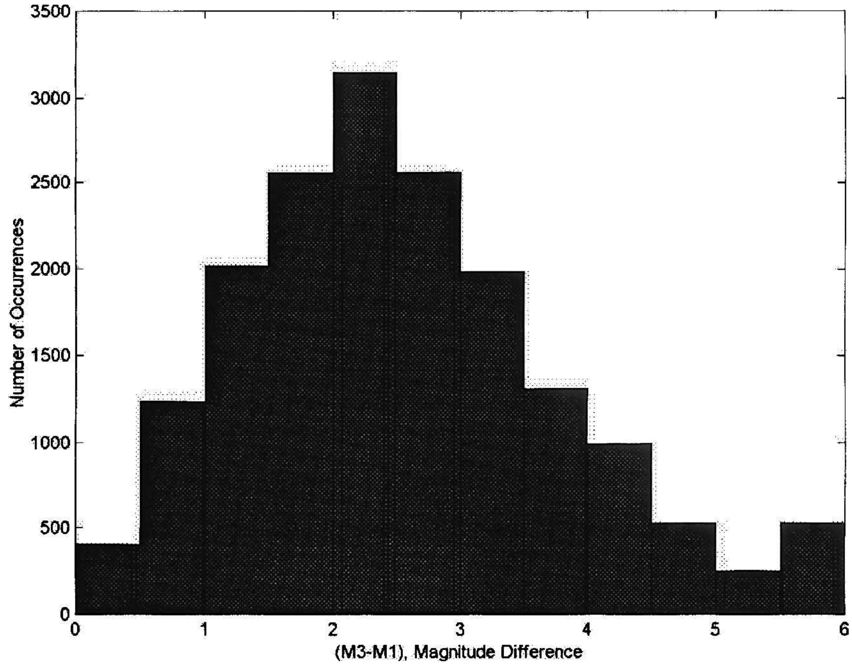

Fig. 9 Histogram of $|M_1 - M_3|$, the absolute magnitude difference between the first and third brightest stars in a triplet.

8) Calculate the output of the network as

$$(y_i, \bar{y}_i) = f^s\left(\sum_j w^y_{ji} t_j, \sum_j \bar{w}^y_{ji} \bar{t}_j,\right) \tag{10}$$

where $(t_j, \bar{t}_j) = (1, 0)$ if it is the maximum hidden node, otherwise $(t_j, \bar{t}_j) = (0, 0)$. $(Y, \bar{Y})$ is the output truth vector for all values of y_i. The slope threshold function f^s is defined by

$$f^s(a, \bar{a}) = \begin{cases} (1, 0) & \text{if} \quad a - \bar{a} \geq 1 \\ (0, 1) & \text{if} \quad a - \bar{a} \leq -1 \\ (a, 1 - a) & \text{otherwise} \end{cases} \tag{11}$$

This binary representation of $(Y, \bar{Y})$ gives the index number of the star triplet.

Simulation Results

The simulation is done in three steps. First, generate a reference catalog of star triplets from the SWAS star catalog. Second, train the fuzzy NLNs using triplet feature vectors. Third, test the system using simulated observation data.

The procedure for creating a triplet catalog from SWAS star catalog is shown in Fig. 10. In the SWAS star catalog, flag7 is the trackability near-neighbor flag. This flag maps the angle to the nearest star either brighter than or up to 4.0 magnitude fainter. Flag8 is the identifiability near-neighbor flag. It maps the angle to the nearest star within 1.0 magnitude. In Fig. 10, the comparison for flag7 and flag8 is to ensure that a reference star has no neighbors within 0.1 deg that are within 4 magnitudes of the reference star brightness, according to the CT-601 star tracker's manual.[13] This ensures that the star parameters can be measured correctly by the star tracker.

After obtaining the triplet feature vector, the second step is to train the fuzzy NLNs. The procedure for training the fuzzy NLN is introduced in the second part of this paper. The inputs to each of the fuzzy subnets are the angular separations, and so there are three input nodes. The hidden nodes are dynamically created during the training. Because there are 17,533 triplet patterns, each pattern can be encoded with 15 binary digits:

$$N = \lfloor \log_2 17533 \rfloor + 1 = 15 \tag{12}$$

The number of output nodes for each fuzzy NLN is 15. Other parameters for training fuzzy NLN are

$$\beta_x = 0.15, \qquad \beta_y = 0.15, \qquad \rho_x = 0.9, \qquad \rho_y = 1.0 \tag{13}$$

We used small learning rates β_x and β_y to get more stable learning results. For star recognition, high input and output vigilance values ρ_x and ρ_y were used to ensure high recognition accuracy.

The feature vector used to train the fuzzy NLN consists of 25 training samples for each pattern. The training samples are generated by adding uniformly distributed random noise to the original feature vectors. The intensity of the noise is up to 2% of the maximum feature value. The training time for each NLN was about 1 h on SGI INDY/IRIX 6.2.

The maximum error for angular separation is set to ±121.4 arc-seconds in the simulation of the SWAS flight catalog data. The maximum error for the magnitude is set to ±0.5. The recognition results are shown in Table 2. From Table 2, we can see

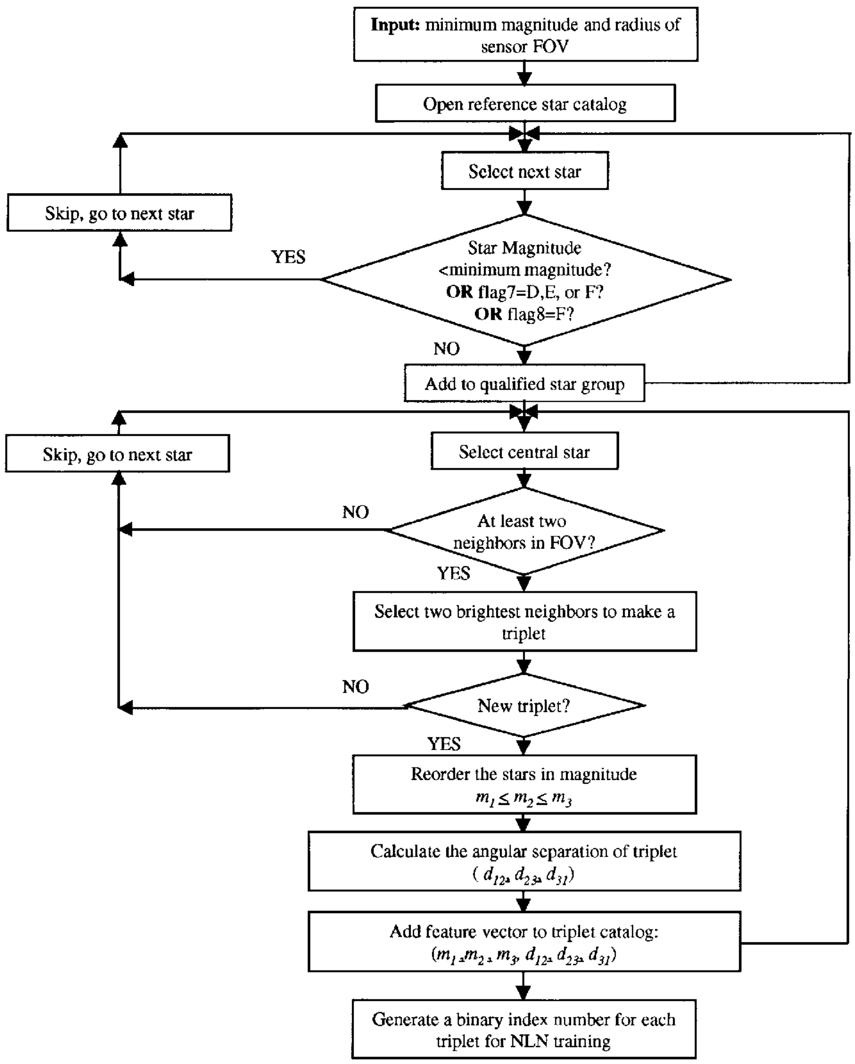

Fig. 10 Diagram for generating triplet catalog.

that even when the magnitude error is up to 0.5 and angular separation error is up to 121.4 arc-seconds, the proposed system has recognition accuracy higher than 95%. The recognition is also very fast. The major computations for the average NLN include $6 \times (3 \times 1253 \times 2 + 1253 \times 15 \times 2) = 270{,}648$ multiplications, where 1,253 is the average number of stored patterns, 3 is the number of input nodes, 15 is the number of output nodes, and 2 is for the complement coding. The storage required for the system includes the weights connecting the input layer with the hidden layer and the weights connecting the hidden layer with the output layer for each NLN. The fuzzy NLN algorithm compares very well with other star recognition algorithms. This system can use a much smaller FOV (8×8 deg) and does not require specialized hardware, unlike the neural-network star identification algorithm of Williams et al.,[15] which requires a FOV as wide as 20×20 deg. The accuracy of the system is comparable to other star recognition systems.[2,3,11] This algorithm is more robust to error measurements than the direct-match algorithms.

The effect of different threshold values in step 7 of the algorithm is also shown. In cases where the magnitude error and the distance error are high, smaller values of T_c allow more solutions to be tested.

Finally, we used some sample attitude maneuver data from the XTE mission to test the system performance. The data was from 6 Jan. 1996. The star sensors were set to measure four stars at a time. Most of the data included the same stars; however, there were 113 distinct measurements in the data file. The fuzzy NLN star recognition system used the brightest three stars to construct the star triplet. The star recognition system recognized 106 of the star triplets. Some of the identification errors were because some of the stars that were measured by XTE were not in the SWAS star catalog. Other errors were caused by the third and fourth measured stars having very similar magnitudes.

Conclusions

In our research, we proposed an autonomous star identification system that uses fuzzy NLNs. The system performance was successfully tested using the star catalog from the SWAS. Compared with the direct-match algorithms, this system can obtain both high accuracy and fast recognition speed when a large star catalog is used. When new patterns are added to the catalog, the NLN can learn the new pattern by adding another cluster node to the network. In contrast, the binary-tree search algorithms need to update the entire tree. With the proposed algorithm, the influence of magnitude errors can also be minimized. Therefore, the proposed system shows great promise for autonomous star identification.

With a large reference catalog, the memory required to store the weight vectors for the NLNs is high. The total memory required for this study with 17,533 triplets is over 3 MB. Although the recognition speed is not slowed, further work will be done to reduce the amount of memory needed for implementation onboard a spacecraft.

Acknowledgments

This work was sponsored by NASA Goddard Space Flight Center under Contract NAG53918. The authors would like to express their appreciation to Mark Woodard and David Tracewell of the Flight Dynamics Division at the NASA Goddard Flight Space Center for their valuable assistance in this research.

References

[1]Daniel, W. K., Correll, T. E., and Anderson, M. O., "Development of a Direct Match Technique for Star Identification of the SWAS Mission," *NASA/Goddard Space Flight Center Flight Mechanics Symposium*, NASA, Greenbelt, MD, 1995, pp. 97–111.

[2]Van Bezooijen, R. W. H., "True-Sky Demonstration of an Autonomous Star Tracker," *Proceedings of the Society of Photo-Optical Instrumentation Engineers*, Vol. 2221, 1996, pp. 156–168.

[3]Quine, B. M., and Durrant-Whyte, H. F., "A Fast Autonomous Star-acquisition Algorithm for Spacecraft," *Control Engineering Practice*, Vol. 4, No. 12, 1996, pp. 1735–1740.

[4]Young, S. S., "Object Recognition Using Multilayer Hopfield Neural Network," *IEEE Transactions on Image Processing*, Vol. 6, No. 3, 1997, pp. 357–372.

[5]Tan, C. L., Quah, T. S., and Teh, H. H., "An Artificial Neural Network that Models Human Decision Making," *Computer*, Vol. 29, No. 3, 1996, pp. 65–70.

[6]Tan, A. H., and Teow, L. N., "Inductive Neural Logic Network and the SCM Algorithm," *Neurocomputing*, Vol. 14, No. 3, 1997, pp. 157–176.

[7]Myers, J. R., Sande, C. B., Miller, A. C., Warren, W. H., and Tracewell, D. A., "The SKY2000 Master Star Catalog," *AAS/AIAA Space Mechanics Symposium*, AAS, San Diego, CA, 1997, pp. 1–16.

[8]Hong, J., "Autonomous Star Identification with Neural Networks," M.S. Thesis, Iowa State Univ., Ames, IA, May 1998.

[9]Tan, A. H., "Adaptive Resonance Associative Map," *Neural Networks*, Vol. 8, No. 3, 1995, pp. 437–446.

[10]Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., and Rosen, D. B., "Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps," *IEEE Transactions on Neural Networks*, Vol. 3, No. 5, 1992, pp. 698–712.

[11]Ketchum, E. A., and Tolson, R. H., "Onboard Star Identification Without A Priori Attitude Information," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 2, 1995, pp. 242–246.

[12]Lee, M. H., "A Simplified Pattern Match Algorithm for Star Identification," *Flight Mechanics/Estimation Theory Symposium*, NASA Conf. Publ. 3333, NASA Scientific and Technical Information Branch, Greenbelt, MD, 1996, pp. 3–14.

[13]Sande, C. B., Braooveanu, D., Miller, A. C., Home, A. T., Tracewell, D. A., and Warren, W. H., "Improved Instrumental Magnitude Prediction from Version 2 of the NASA SKY2000 Master Star Catalog," *Proceedings of the American Astronomical Society/Goddard Space Flight Center 13th International Symposium on Space Flight Dynamics*, AAS, San Diego, CA, 1998, pp. 741–754.

[14]"User's Manual for the SWAS CT-601 Star Tracker," Ball Aerospace, Systems Div., Boulder, CO, Feb. 1994.

[15]Williams, K. E., Strikwerda, T. E., Fisher, H. L., Strohbehn, K., and Edwards, T. G., "Design Study: Parallel Architecture for Autonomous Star Pattern Identification and Tracking," *Advances in the Astronautical Sciences*, Vol. 93-102, Feb. 1993, pp. 43–57.