Recognizing shipbuilding parts using artificial neural networks and Fourier descriptors

D A Sanders

Department of Mechanical and Design Engineering, University of Portsmouth, Anglesea Road Building, Portsmouth PO1 3DJ, UK. email: david.sanders@port.ac.uk

The manuscript was received on 5 October 2008 and was accepted after revision for publication on 1 December 2008.

DOI: 10.1243/09544054JEM1382

Abstract: A pattern recognition system is described for recognizing shipbuilding parts using artificial neural networks and Fourier descriptors. The system uses shape contour information that is invariant of size, translation, and rotation. Fourier descriptors provide information, and the neural networks make decisions about the shapes. A brief review of the current state of the art is included, and results from testing show that the system distinguished between various shapes and proved to be a valid approach.

Keywords: pattern recognition, object recognition, shipbuilding, neural network, Fourier descriptor

1 INTRODUCTION

Although some shipyards have used robots for welding steel for 20 years [**1**], the integration of robotic welding presents problems. Low levels of repeatable welds within some ships means that, although the quality and speed of robotic welding are acceptable, the generation of programs capable of applying welds has proved difficult. Many welding robots work primarily in ‘teach-and-playback’ mode, sometimes using digital pens to program them [**2**, **3**], but teach-and-playback limits flexibility.

Although the superstructure of a ship may be complicated, it can have a complexity of scale. A ship’s superstructure can be a complicated object made from a large number of simple objects. Most are made from either metal bar (of various sizes and shapes) or metal plate. Additional items are often cut from metal plate. A small metal cross-beam from a ship is shown in Fig. 1. It is 1 m long, although size is largely irrelevant within the camera’s field of vision.

A new automated welding system is being created that uses artificial intelligence (AI) techniques to determine where to weld these types of part. The system will use a combination of AI techniques to suggest weld requirements. Suggestions will be evaluated and decisions made regarding weld(s). These parameters will be sent to a program generator to produce a custom robot program for the shop floor. The whole system is shown in Fig. 2. To date, the image capture and program generator systems are working. A camera mounted above the assembly line at VT Shipbuilding in Portsmouth captured images (frames), and this paper concerns the new image-processing and object recognition subsystems that operate on the images. The decision module is now under construction.

New subsystems successfully distinguished between various ship’s parts as follows:

(a) by processing shape information so that Fourier descriptors can be extracted;
(b) by extracting descriptors;
(c) by associating sets of descriptors with training sets and reaching decisions.

Images were broken into equal segments and the segments represented as complex numbers by referring coordinate points to a random starting point. Fourier descriptors were extracted by transforming object descriptions into the frequency domain.

Since data points around the contour were expressed as complex number values and not as complex functions of length, the usual complex form of Fourier series was of little use. As contours were sampled, discrete Fourier transforms (DFTs) were considered but were replaced by more efficient fast Fourier transforms (FFTs). Once transformed, then data were in the form $A_n = \exp(\mathrm{j}n\tau)A_n^{(0)}$, expressed as phase and magnitude. Here $A_n^{(0)}$ is the

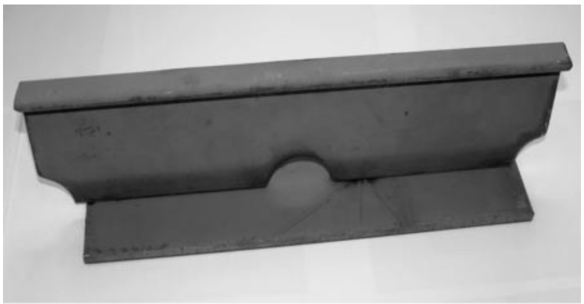

Fig. 1 Photograph of a metal bar part of a ship (1 m long)

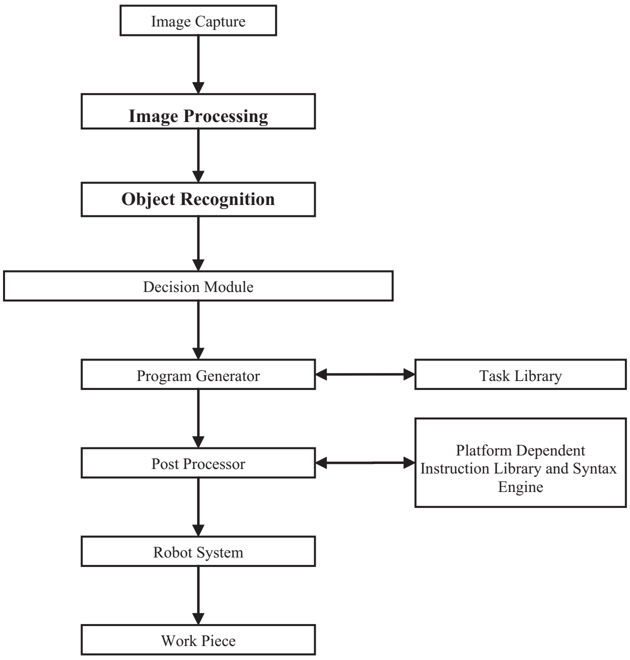

Fig. 2 System flow diagram

same as $|A_n|$. The modulus of this transformed data was considered in order to discard phase information and, consequently, to discard operations that affected the phase. The algorithm was $B_n = A_{n+1}A_{n-1}/A_1^2$. Descriptors were now invariant (within a small error) for rotation, dilation, and translation.

2 BACKGROUND

Much is known about feature recognition, Fourier descriptors, and artificial neural networks (ANNs). Much has been completed in laboratories and simple environments but little has been applied to automating shipbuilding processes or to practical work in noisy environments such as shipyards. A brief review of these areas with respect to shipbuilding is included here to place the work in this paper into a wider context.

2.1 Feature recognition

Recording an image of a ship's part is simple, but recognizing what that image portrays requires comprehension [**4**]. Feature recognition is a first step in translating an image of a ship's part into welding instructions. Methods in the literature for automated feature recognition tend to match structures identified in a part representation with some pattern in a knowledge base, often using if–then rules [**5**]. A disadvantage is that they cannot easily deal with features that cannot be matched with known patterns.

Pattern recognition techniques were originally posed as statistical problems, derived from work in discriminant analysis and applying Bayes theorem [**6**]. Studies analysing developed automatic feature recognition methodologies are reported in references [**5**] and [**7**] to [**10**], and pattern recognition based on Bayesian networks specifically for ships is reported in reference [**11**].

2.2 Fourier descriptors

Describing shapes is essential for pattern recognition. Shape description techniques can be divided into boundary-based and region-based techniques. Region-based techniques consider whole objects while boundary-based techniques concentrate on boundary lines. Boundary-based methods are more popular because shape classifications are based on contour features.

Many integral transforms can be used as feature extractors, for example the general integral [**12**], Mellin [**13**], cross-correlation [**14**], Radon [**15**], and Fourier–Mellin [**16**] transforms. Fourier–Mellin descriptors have tended to perform better than others in noisy conditions (such as shipyards) but are not translation-invariant. Properties of DFTs are analogous to continuous Fourier transforms. Power spectra of DFTs are invariant under cyclic translation of the input vector. Fourier-based methods can be applied efficiently using FFTs. That was the selected method, and shape information was processed so that Fourier descriptors could be extracted.

Fourier descriptors characterize object shapes in a frequency domain. Shape-based objects can be classified using conventional Fourier descriptors, generic Fourier descriptors, or wavelet Fourier descriptors. Duan *et al.* [**17**] recently used Fourier descriptors for automatic object and image alignment, combining Fourier descriptors and iterative closest-point computation in combination with a distance matrix to detect pairs of edges and so to align images. Generalized Fourier descriptors have been described by Smach *et al.* [**18**].

2.3 Artificial neural networks

Previous work on active recognition differs in object representation, information combination, and future planning. Invariant pattern recognition is complicated [**12**] and classification processes can be, firstly, invariant feature extraction or, secondly, feature classification. Feature classification can be achieved using ANNs.

ANNs typically have inputs and outputs, with processing within hidden layers in between. Inputs are independent variables and outputs are dependent variables. ANNs are flexible mathematical functions with configurable internal parameters. To represent complicated relationships accurately, these parameters are adjusted through a learning algorithm. In 'supervised' learning, examples of inputs and corresponding desired outputs are simultaneously presented to networks, which iteratively self-adjust to represent accurately as many examples as possible [**19**].

Once trained, ANNs can accept new inputs and attempt to predict accurate outputs [**20**]. To produce an output, the network simply performs function evaluation. The only assumption is that there exists some continuous functional relationship between input and output data. A general introduction to multilayer feedforward networks has been given by Hinton [**21**].

Rare uses of ANNs for object recognition in ships are the algorithms by Pan *et al.* [**22**] and Park *et al.* [**23**] that recognize ship noise. Recent ANN applications to other ship applications are collision force and avoidance [**24**], fault prediction [**25**], roll stabilization [**26**], manoeuvring and course tracking [**27**, **28**], propeller performance, and ship control [**29–31**].

3 IMAGE PROCESSING

Information about shape or pattern is held within contours, and so Fourier descriptors were applied to the contours of shapes being classified. Figure 3 shows an edge detection image from Fig. 1.

The image was processed to produce closed line shapes so that no lines were left open and hanging. Contours were assumed to be closed curves in complex space. An arbitrary point moving around the contour generated a complex function f. If the point moved around the contour at a constant velocity v, then at every time t a complex number c was defined such that $c = f(t)$. t is not necessarily real time; it represents a section of length around the contour. Because contours were closed, it implied that there existed a value T so that $f(t + nT) = f(t)$, where nT was the contour length. Therefore f can be expressed as a complex Fourier series. This gives

Fig. 3 Image output from the edge detection algorithm

$$f(t) = \sum_{-\infty}^{\infty} A_n \exp\left(\frac{\mathrm{j}n2\pi t}{T}\right)$$

and the Fourier coefficients become

$$A_n = \frac{1}{T}\int_0^T f(t)\exp\left(\frac{-\mathrm{j}n2\pi t}{T}\right)\mathrm{d}t$$

For simplicity, the velocity can be such that $T = 2\pi$, and

$$A_n = \int_0^{2\pi} f(t)\exp(-\mathrm{j}nt)\,\mathrm{d}t$$

These Fourier coefficients depend on the starting point and differ with respect to a parameter τ along the contour, so that for each τ there is a set of Fourier coefficients of the function $f(t) = f(t + \tau)$. If $f(t) = f^{(0)}(t)$, then other functions around the contour will be $f(t) = f^{(0)}(t + \tau)$. Also

$$y(t) = \sum_0^{2\pi} A_n \exp(\mathrm{j}nt)$$

The index (0) refers to a specific contour function, and so the resulting Fourier coefficients become

$$\begin{aligned} A_n &= \frac{1}{2\pi}\int_0^{2\pi} f^{(0)}(t + \tau)\exp(-\mathrm{j}nt)\,\mathrm{d}t \\ &= \exp(\mathrm{j}n\tau)\,\frac{1}{2\pi}\int_0^{2\pi} f^{(0)}(t)\exp(-\mathrm{j}nt)\,\mathrm{d}t \\ &= \exp(\mathrm{j}n\tau)\,a_n^{(0)} \end{aligned}$$

Next, translations, rotation, and dilation are considered.

3.1 Translation

If $A_n^{(0)}$ is a set of Fourier coefficients from a contour function, then translation by a complex vector Z results in a contour function expressed in the inverse Fourier series

$$f(t) = f^{(0)}(t) + Z = \sum_{-\infty}^{\infty} A_n^{(0)} \exp(\mathrm{j}nt) + Z$$

Therefore the Fourier coefficients of the translated contour are $A_n = A_n^{(0)}$ for n (where not equal to zero) and $A_n^{(0)} + Z$ for $n = 0$. All coefficients except A_0 are invariant of translation. A_0 depicts the complex vector indicating the position of the centre of gravity.

3.2 Rotation

If the centre of gravity is at the origin, then a rotation of the contour function $f(t)$ about the origin with an angle of φ produces another function $f(t)$ where $f(t) = \exp(\mathrm{j}\varphi)\, f^{(0)}(t)$. With $f(t)$ expressed as the inverse Fourier transform, coefficients of the rotated contour will be $A_n = \exp(\mathrm{j}\varphi)\, A_n^{(0)}$.

3.3 Dilation

Similarly, dilation of the contour by the scale factor R creates Fourier coefficients of the form $A_n = RA_n^{(0)}$.

4 EXTRACTING FOURIER DESCRIPTORS

The general form of Fourier coefficients of a contour after translation, rotation, and dilation is $A_n = \exp(\mathrm{j}n\tau)\ R\ \exp(\mathrm{j}\varphi)A_n^{(0)}$, where the coefficients $A_n^{(0)}$ are coefficients of the original contour. They are not useful in that form because they contain information on orientation, and shape only is needed.

Consider $B_n = A_{1+n+1}A_{1-n}/A_1^2$. Applying that expression after rotation, dilation, etc. results in an expression that does not contain τ, R, or φ. If coefficient A_0 is not used, then these B_n coefficients are invariant under translation, rotation, and dilation. Thus B_n coefficients represent shape (or form).

5 BACK-PROPAGATION

Back-propagation has been described in references [**32**] and [**33**]. Errors were generated at outputs of ANNs by comparing actual with desired outputs. The error was expressed in terms of ANN weights and weights updated to minimize error. Nets were considered trained when the error became zero (within preset ranges). Considering two cascaded nodes, if errors generated at outputs were expressed as $E = \frac{1}{2}(y_j^s - d_j)^s$, then the error with respect to output y_j^s would be $\mathrm{d}E/\mathrm{d}y = y_j^s - d_j$. The error in terms of weights was $\mathrm{d}E/\mathrm{d}W_j^s$. Expanding the equation gives

$$\frac{\mathrm{d}E}{\mathrm{d}W_{kj}^{Ps}} = \frac{\mathrm{d}E}{\mathrm{d}X_j^s}\frac{\mathrm{d}X_j^s}{\mathrm{d}W - kj^s}$$

$$= \frac{\mathrm{d}E}{\mathrm{d}X_j^s} j_j^{s-1}$$

$$= \frac{\mathrm{d}E}{\mathrm{d}y_j^s}\frac{\mathrm{d}y_j^s}{X_j^s} y_j^{s-1}$$

$$\frac{\mathrm{d}E}{\mathrm{d}W_{kj}^s} = \frac{\mathrm{d}E}{\mathrm{d}y_j^s} f(X_j^s) y_j^{s-1}$$

where $f(X_j^s)$ is the differential of the neuron function and y_j^{s-1} is the output from the node in the next layer back. If layer weights were not on the output layer, then the first step was to find the error of the output of the next layer back. In other words, $\mathrm{d}E/\mathrm{d}y_k^{s-1}$ is the error.

Once again by expanding the equation, it is found that

$$\frac{\mathrm{d}E}{\mathrm{d}y_j^{s-1}} = \frac{\mathrm{d}E}{\mathrm{d}X_j^s}\frac{dX_j^s}{\mathrm{d}y_j^{s-1}}$$

$$= \frac{\mathrm{d}E}{\mathrm{d}X_j^s} W_{kj}^s$$

$$= \frac{\mathrm{d}E}{\mathrm{d}y_j^s}\frac{\mathrm{d}y_j^s}{\mathrm{d}X_j^s} W_{kj}^s$$

so that $\mathrm{d}y_j^s/\mathrm{d}x_j^s = fX_j^s$. Therefore, if $(\mathrm{d}E/\mathrm{d}y_j^s)fX_j^s = \delta_j^s$, the total error for the layer $s-1$ would be

$$\frac{dE}{dy_j^{s-1}} = \sum \delta_j^s W_{kj}^s$$

If this is applied to the general equation for the error with respect to the weights, then

$$\frac{\mathrm{d}E}{\mathrm{d}W_{kj}^{s-1}} = \frac{\mathrm{d}E}{\mathrm{d}y_j^{s-1}} f(X_j^{s-1}) y_j^{s-2}$$

These errors were used to update weights within the ANN. A number of teaching runs were required before outputs converged. A teaching net was created to take two sets of inputs and two sets of demand vectors. The layout was a 5–38–4 pattern. After 150 test runs the network gave the outputs shown in Table 1.

Table 1 Output from two sets of inputs

Input set	Output	Demand	Input set	Output	Demand
1	1 -5.99×10^6	1 0	2	6.05×10^{-5} 0.9999	0 1

6 TESTING

Weights were saved. The application net was combined with the description program and set up to analyse two shapes in different orientations. In 100 tests the program classified 98 shapes correctly after three frames of video. The two-pattern program operated with a 98 per cent classification rate within three frames.

The training net was then modified to take three sets of inputs and demand vectors. Weights were frozen after 500 test runs and the outputs are shown in Table 2.

Table 2 Output from three sets of inputs

Input set	Output	Desired output	Input set	Output	Desired output
1	1	1	3	9.87×10^{-8}	0
	1.2×10^{-6}	0		4.6×10^{-7}	0
	7.5×10^{-7}	0		0.99999	1
2	3.86×10^{-6}	0			
	0.9998	1			
	5.69×10^{-7}	0			

Table 3 Output from four sets of inputs

Input set	Output	Desired output	Input set	Output	Desired output
1	1	1	3	1.2×10^{-8}	0
	0	0		-1.9×10^{-8}	0
	-3.71×10^{-9}	0		1	1
	-4.48×10^{-8}	0		-9.32×10^{-8}	0
2	-1.92×10^{-7}	0	4	0	0
	1	1		-2.22×10^{-8}	0
	-7.46×10^{-9}	0		-3.14×10^{-8}	0
	-1.11×10^{-7}	0		0.9999	1

Programs were tested with three different shapes in different orientations. In 100 tests the program classified 97 shapes correctly after three frames. The three-pattern recognizer worked with 97 per cent classification. Programs were then modified to take four training sets and demand vectors. This ran for 2 h and after 6219 test runs the observed outputs are shown in Table 3. Over 50 tests the program classified 44 shapes correctly after three frames. The four-pattern recognizer worked with 88 per cent classification.

7 CONCLUSION

Different shapes were identified using a simple pattern recognition system that utilized an ANN and shape contour information that was invariant of size, translation, and rotation. Systems distinguished between shapes and proved to be a valid approach to recognition. Since the acquisition and processing of new images are expensive, it is desirable to take a minimum number of additional views [**34**], and this method successfully identified parts after three frames. Potential errors derived from contour sampling resolution, noise, and ANN size and architecture. These are being considered in ongoing work.

REFERENCES

1 **Jacobsen, N. J.** Robot welding of hatch coamings for large container ships. *Ind. Robot*, 2007, **34**(6), 456–461.

2 **Pires, J. N., Godinho, T.,** and **Araújo, A.** Using digital pens to program welding tasks. *Ind. Robot*, 2007, **34**(6), 476–486.

3 **Sanders, D. A., Urwin-Wright, S. D., Tewkesbury, G. E.,** and **Gremont, B.** Pointer device for thin-film transistor and cathode ray tube computer screens. *Electron. Lett.*, 2005, **41**(16), 17–18.

4 **Hilker, E.** Artificial intelligence: a review of current information sources. *Collection Bldg*, 1986, **7**(3), 14–30.

5 **Babic, B., Nesic, N.,** and **Miljkovic, M.** A review of automated feature recognition with rule-based pattern recognition. *Computers Industry*, 2008, **59**, 321–337.

6 **Ripley, B. D.** *Pattern recognition and neural networks*, 1996 (Cambridge University Press, Cambridge).

7 **Jain, A. K., Duin, R. P. W.,** and **Mao, J. C.** Statistical pattern recognition: a review. *IEEE Trans. Pattern Analysis Mach. Intell.*, 2000, **22**(1), 4–37.

8 **Saunders, D.** Progress in machine intellegence. *Industrial Robot – an International Journal*, 2008, **35**(6), 485–487.

9 **Han, J. H., Pratt, M.,** and **Regli, W. C.** Manufacturing feature recognition from solid models: a status report. *IEEE Trans. Robotics Automn*, 2000, **16**(6), 782–796.

10 **Owodunni, O.** and **Hinduja, S.** Evaluation of existing and new feature recognition algorithms. Part 1. Theory and implementation. *Proc. Instn Mech. Engrs, Part B: J. Engineering Manufacture*, 2002, **216**, 839–851.

11 **Wang, Q. J., Gao, X. G.,** and **Chen, D. Q.** Pattern recognition for ship based on Bayesian networks. In Proceedings of the Fourth International Conference on *Fuzzy systems and knowledge discovery* (Eds J. S. Lei, J. Yu, and S. G. Zhou), 2007, pp. 684–688 (IEEE, New York).

12 **Wood, J.** Invarient pattern recognition. *Pattern Recognition*, 1996, **29**(1), 1–17.

13 **Li, Y.** Reforming the theory of invariant moments for pattern recognition. *Pattern Recognition*, 1992, **25**, 723–730.

14 **Pintsov, D.** Invariant pattern recognition, symmetry and Radon transforms. *J. Opt. Soc. Am. A*, 1989, **6**(10), 1544–1554.

15 **Helgason, S.** *The Radon transform*, 1980 (Birkhauser, Boston, Massachusetts).

16 **Kagcyu, S., Ohnishi, N.,** and **Sugie, N.** Augmented multilayer perceptron for rotation- and scale-invariant handwritten numeral recognition. In Proceedings of the 1991 IEEE International Joint Conference on *Neural networks*. 1991, vol. 1, pp. 54–59 (IEEE, New York).

17 **Duan, W., Kuester, F., Gaudiot, J. L.,** and **Hammami, O.** Automatic object and image alignment using Fourier descriptors. *Image Vision Computing*, 2008, **26**, 1196–1206.

18 **Smach, F., Lemaitre, C., Gauthier, J. P., Miteran, J.,** and **Atri, M.** Generalized Fourier descriptors with applications to objects recognition in SVM context. *Mathl Imaging Vision*, 2008, **30**(1), 43–71.

19 **Erwin-Wright, S., Sanders, D.,** and **Chen, S.** Predicting terrain contours using a feed-forward neural network. *Engng Applic. Artif. Intell.*, 2003, **16**(5–6), 465–472.

20 **Urwin-Wright, S., Sanders, D.,** and **Chen, S.** Terrain prediction for an eight-legged robot. *Robotic Systems*, 2002, **19**(2), 91–98.

21 **Hinton, G. E.** How neural networks learn from experience. *Scient. Am.*, 1992, **267**, 144–151.

22 **Pan, X. Q., Li, X. L., Lu, Y., Cao, Y. C.,** and **Zhao, Y.** Research on the algorithm of object recognition of ship

noise based on auditory feature and pattern-matching. In Proceedings of the 2006 IEEE International Symposium on *Intelligent signal processing and communication systems*, 2006, pp. 283–286 (IEEE, New York).

23 **Park, K. C., Lee, P. H., Park, J. N.,** and **Yoon, J.-R.** Neural network approach for automatic classification of ship-radiated noise. *Jap. J. Appl. Physics*, 2006, **45**(5B), 4859–4861.

24 **Liu, Y. H., Du, X. M.,** and **Yang, S. H.** The design of a fuzzy-neural network for ship collision avoidance. In Proceedings of the Fourth International Conference, Lecture Notes in Computer Science, vol. 3930, Guangzhou, People's of Republic of China, August 2006, *Advances in machine learning and cybernetics*. pp. 804–812 (Springer-Verlag, New York).

25 **Yang, G., Yang, G.,** and **Wu, X. P.** Fault prediction of ship machinery based on gray neural network model. In Proceedings of the IEEE International Conference on *Control and automation*, 2007, Vols 1–7, pp. 1641–1644 (IEEE, New York).

26 **Alarcin, F.** Internal model control using neural network for ship roll stabilization. *J. Mar. Sci. Technol.*, 2007, **15**(2), 141–147 (Chinese Electronic Periodical Services, Taiwan).

27 **Ebada, A.** and **Abdel-Maksoud, M.** Prediction of ship turning manoeuvre using artificial neural networks (ANN). In Proceedings of the Fifth International Conference on *Computer applications and information technology in the maritime industries*, 2006, pp. 127–145.

28 **Mizuno, N., Kuroda, M., Okazaki, T.,** and **Ohtsu, K.** Minimum time ship maneuvering method using neural network and nonlinear model predictive compensator. *Control Engng Practice*, 2007, **15**(6), 757–765.

29 **Roddy, R. F., Hess, D. E.,** and **Faller, W. E.** Neural network predictions of the 4-quadrant Wageningen B-screw series. In Proceedings of the Fifth international Conference on *Computer applications and information technology in the maritime industries*, Leiden, The Netherlands, 2006, pp. 315–334 (TU Delft, Delft).

30 **Hess, D. E., Roddy, R. F.,** and **Faller, W. E.** Uncertainty analysis applied to feedforward neural networks. In Proceedings of the Fifth International Conference on *Computer applications and information technology in the maritime industries*, Leiden, The Netherlands, 2006, pp. 296–314 (TU Delft, Delft).

31 **Sui, J. H., Lin, Y. J.,** and **Ren, G.** A genetically optimized fuzzy neural network for ship controllers. In Proceedings of the IEEE International Conference on *Mechatronics and automation*, 2006, Vols 1–3, pp. 1367–1371 (IEEE, New York).

32 **Sanders, D. A., Haynes, B. P., Tewkesbury, G. E.,** and **Stott, I. J.** The addition of neural networks to the inner feedback path in order to improve on the use of pre-trained feed forward estimators. *Math Computers Simulation*, 1996, **41**(5–6), 461–472.

33 **Saunders, D.** System specification 2. *Microprocessing and Microprogramming*, Sept. 1993, **38**(1–5), 833–834 (Elsevier Science, Amsterdam).

34 **Deinzer, F., Denzler, J.,** and **Niemann, H.** Viewpoint selection. Planning optimal sequences of views for object recognition. In *Computer analysis of images and patterns* (CAIP 2003), Tenth International Conference, Groningen, The Netherlands, August 2003, pp. 65–73 (Springer-Verlag, Berlin).