Springer

Dear Author,

Here are the proofs of your article.

- You can submit your corrections **online**, via **e-mail** or by **fax**.
- For **online** submission please insert your corrections in the online correction form. Always indicate the line number to which the correction refers.
- You can also insert your corrections in the proof PDF and **email** the annotated PDF.
- For fax submission, please ensure that your corrections are clearly legible. Use a fine black pen and write the correction in the margin, not too close to the edge of the page.
- Remember to note the **journal title**, **article number**, and **your name** when sending your response via e-mail or fax.
- **Check** the metadata sheet to make sure that the header information, especially author names and the corresponding affiliations are correctly shown.
- **Check** the questions that may have arisen during copy editing and insert your answers/ corrections.
- **Check** that the text is complete and that all figures, tables and their legends are included. Also check the accuracy of special characters, equations, and electronic supplementary material if applicable. If necessary refer to the *Edited manuscript*.
- The publication of inaccurate data such as dosages and units can have serious consequences. Please take particular care that all such details are correct.
- Please **do not** make changes that involve only matters of style. We have generally introduced forms that follow the journal's style.
  Substantial changes in content, e.g., new results, corrected values, title and authorship are not allowed without the approval of the responsible editor. In such a case, please contact the Editorial Office and return his/her consent together with the proof.
- If we do not receive your corrections **within 48 hours**, we will send you a reminder.
- Your article will be published **Online First** approximately one week after receipt of your corrected proofs. This is the **official first publication** citable with the DOI. **Further changes are, therefore, not possible.**
- The **printed version** will follow in a forthcoming issue.

Please note

After online publication, subscribers (personal/institutional) to this journal will have access to the complete article via the DOI using the URL: http://dx.doi.org/[DOI].
If you would like to know when your article has been published online, take advantage of our free alert service. For registration and further information go to: http://www.springerlink.com.

Due to the electronic nature of the procedure, the manuscript and the original figures will only be returned to you on special request. When you return your corrections, please inform us if you would like to have these documents returned.

Metadata of the article that will be visualized in OnlineFirst

Please note: Images will appear in color online but will be printed in black and white.

ArticleTitle	Improving automatic robotic welding in shipbuilding through the introduction of a corner-finding algorithm to help recognise shipbuilding parts	
Article Sub-Title		
Article CopyRight	JASNAOE (This will be the copyright line in the final PDF)	
Journal Name	Journal of Marine Science and Technology	

Corresponding Author	Family Name	**Sanders**
	Particle	
	Given Name	**David**
	Suffix	
	Division	Faculty of Technology
	Organization	University of Portsmouth
	Address	Anglesea Road Building, PO1 3DJ, Portsmouth, UK
	Email	david.sanders@port.ac.uk
Author	Family Name	**Tewkesbury**
	Particle	
	Given Name	**Giles**
	Suffix	
	Division	Department of Mechanical and Design Engineering
	Organization	University of Portsmouth
	Address	Anglesea Road Building, Portsmouth, UK
	Email	giles.tewkesbury@port.ac.uk
Author	Family Name	**Ndzi**
	Particle	
	Given Name	**David**
	Suffix	
	Division	Electronic and Computer Engineering
	Organization	University of Portsmouth
	Address	Anglesea Road Building, Portsmouth, UK
	Email	david.ndzi@port.ac.uk
Author	Family Name	**Gegov**
	Particle	
	Given Name	**Alexander**
	Suffix	
	Division	
	Organization	School of Computing
	Address	Buckingham Building, Lion Terrace, Portsmouth, UK
	Email	alexander.gegov@port.ac.uk
Author	Family Name	**Gremont**
	Particle	
	Given Name	**Boris**

	Suffix	
	Division	Electronic and Computer Engineering
	Organization	University of Portsmouth
	Address	Anglesea Road Building, Portsmouth, UK
	Email	david.ndzi@port.ac.uk
Author	Family Name	**Little**
	Particle	
	Given Name	**Andrew**
	Suffix	
	Division	Department of Mechanical and Design Engineering
	Organization	University of Portsmouth
	Address	Anglesea Road Building, Portsmouth, UK
	Email	andrew.little@port.ac.uk

	Received	21 June 2009
Schedule	Revised	
	Accepted	29 June 2011

Abstract

A system that uses a combination of techniques to suggest weld requirements for ships' parts is proposed. These suggestions are evaluated, decisions are made and then weld parameters are sent to a program generator. New image capture methods are being combined with a decision-making system that uses multiple parallel artificial intelligence (AI) techniques. A pattern recognition system recognises shipbuilding parts using shape contour information. Fourier descriptors provide information and neural networks make decisions about shapes. The system has distinguished between various parts, and programs have been generated to validate the approaches used. The system has recently been improved by pre-processing using a simple and accurate corner finder in an edge-detected image.

Keywords (separated by '-'): Robot - Welding - Shipbuilding - Pattern recognition - Locating corners - Image processing

Footnote Information

Journal: 773
Article: 154

Springer
the language of science

Author Query Form

Please ensure you fill out your response to the queries raised below and return this form along with your corrections

Dear Author

During the process of typesetting your article, the following queries have arisen. Please check your typeset proof carefully against the queries listed below and mark the necessary changes either directly on the proof/online grid or in the 'Author's response' area provided below

Query	Details required	Author's response
1.	Please check and confirm whether the corresponding author is correctly identified.	
2.	For equations in this article: We have checked the correct conversion to PDF. Because the material is complex, please double-check the equations in order to ensure the accurate presentation of the data.	
3.	Please provide volume number of Ref. [3], if possible.	

J Mar Sci Technol
DOI 10.1007/s00773-011-0154-x

ORIGINAL ARTICLE

Improving automatic robotic welding in shipbuilding through the introduction of a corner-finding algorithm to help recognise shipbuilding parts

David Sanders · Giles Tewkesbury · David Ndzi · Alexander Gegov · Boris Gremont · Andrew Little

Received: 21 June 2009 / Accepted: 29 June 2011

Abstract A system that uses a combination of techniques to suggest weld requirements for ships' parts is proposed. These suggestions are evaluated, decisions are made and then weld parameters are sent to a program generator. New image capture methods are being combined with a decision-making system that uses multiple parallel artificial intelligence (AI) techniques. A pattern recognition system recognises shipbuilding parts using shape contour information. Fourier descriptors provide information and neural networks make decisions about shapes. The system has distinguished between various parts, and programs have been generated to validate the approaches used. The system has recently been improved by pre-processing using a simple and accurate corner finder in an edge-detected image.

D. Sanders (✉)
Faculty of Technology, University of Portsmouth, Anglesea Road Building, Portsmouth PO1 3DJ, UK
e-mail: david.sanders@port.ac.uk

G. Tewkesbury · A. Little
Department of Mechanical and Design Engineering, University of Portsmouth, Anglesea Road Building, Portsmouth, UK
e-mail: giles.tewkesbury@port.ac.uk

A. Little
e-mail: andrew.little@port.ac.uk

D. Ndzi · B. Gremont
Electronic and Computer Engineering, University of Portsmouth, Anglesea Road Building, Portsmouth, UK
e-mail: david.ndzi@port.ac.uk

B. Gremont
e-mail: david.ndzi@port.ac.uk

A. Gegov
School of Computing, Buckingham Building, Lion Terrace, Portsmouth, UK
e-mail: alexander.gegov@port.ac.uk

Keywords Robot · Welding · Shipbuilding · Pattern recognition · Locating corners · Image processing

1 Introduction

Although some shipyards have used robots for welding steel for 20 years [1, 2], integration of robotic welding presents problems [3]. The low level of repeatable welds within some ships means that, although the quality and speed of robotic welding are acceptable, generation of programs capable of carrying out welding has proved difficult. Many welding robots work primarily in "teach-and-playback" mode, but this further limits flexibility.

Although the superstructure of a ship may be complicated, this may be a complexity of scale; i.e., a ship's superstructure can be a complicated object made from a large number of simple objects, most of which are made from either metal bar (of varying sizes and shapes) or metal plate. Additional items are often cut from metal plate. A small metal crossbeam 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 that uses AI techniques to determine where to weld such parts is being created. New image capture methods are being combined with a decision-making system that uses multiple parallel AI techniques. The proposal uses object-oriented programming techniques to create the framework for the system and uses imaging software to capture and process image data. The final 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 robot program for use on the shopfloor. The whole system is shown in Fig. 2.

Journal : Large 773 | Dispatch : 6-12-2011 | Pages : 8
Article No. : 154 | ☐ LE | ☐ TYPESET
MS Code : 154_JMST-D-09-00030 | ☑ CP | ☑ DISK

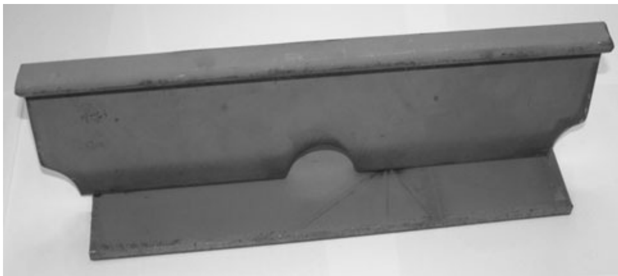

Fig. 1 Metal bar part of a ship (1 m long)

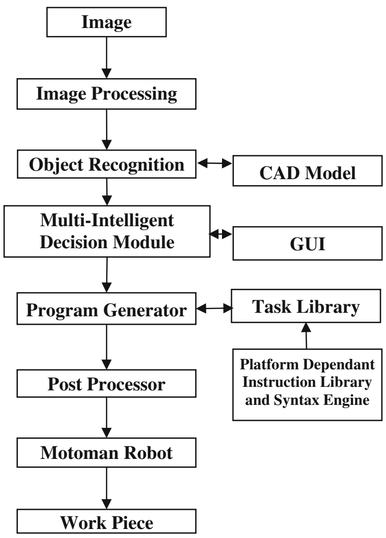

Fig. 2 System flow diagram

To date, the image capture and program generator systems are working, a camera has been mounted above the assembly line at VT Shipbuilding in Portsmouth to capture images (frames), and new image processing and object recognition sub-systems have been successfully created to operate on these images. The decision module is now under construction. New sub-systems have successfully distinguished between various ships' parts by processing shape information so that Fourier descriptors [4] can be extracted and formed into sets for association with training sets so that decisions can be made. This work was described previously [2]. In that work, images were broken into equal segments, which were then 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 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, the data were expressed as phase and magnitude. The modulus of this transformed data was considered in order to discard phase information and thereby operations that affected phase. Descriptors were then invariant (within a small error) under rotation, dilation and translation.

2 Proposed system

This section explains the existing RinasWeld/Motoman system in place at VT Shipbuilding and discusses how additional systems may be integrated with them [5]. The proposed system is discussed, including software systems required, image processing systems and use of multiple artificial intelligence techniques to make decisions.

The RinasWeld/Motoman software systems at VTS work in series to construct viable robot programs. These systems existed before the start of the research. The first system, the computer-aided design (CAD) model interpreter, accepts a CAD model and determines the welds required. This data is fed to the program generator, which re-orientates the weld requirements in line with the real-world orientation of the panel. The program generator then sends any programs sequentially to the robot (normally one program per weld line). Additional software systems could be incorporated into the existing system at the point where the robot programs are sent to the robot system. This is because the transmission protocol at this point is standard transmission control protocol/Internet protocol (TCP/IP) and any programs to be sent can be viewed as text files.

The proposed system in Fig. 2 shows that data will be gathered from a post-processed image. The data will then be combined with the data contained within a CAD model. A multi-intelligent decision module will then use multiple AI techniques to suggest a required weld (Fig. 3). The decision module uses case-based reasoning (solving new problems based on the solutions of similar past problems), a rule-based system (using pre-defined rules to make deductions) and fuzzy logic (a form of multi-valued logic derived from fuzzy set theory to deal with reasoning that is approximate rather than precise). This weld requirement will then be displayed for the operator to check. If the operator rejects the suggestion, the system will learn from that rejection and suggest a different requirement. Assuming that the operator now accepts the requirement, the system will generate a compatible robot program by using the program generator and post-processing systems.

The image processing systems involve detecting edges, line identification and geometric data generation. These data can then be used to identify the different objects within the image. A software package named 'WiT 8.3' by Dalsa Coreco was initially used to reduce the development time of the first prototype image processing systems. This

Journal : Large 773 | Dispatch : 6-12-2011 | Pages : 8
Article No. : 154 | □ LE | □ TYPESET
MS Code : 154_JMST-D-09-00030 | ✔ CP | ✔ DISK

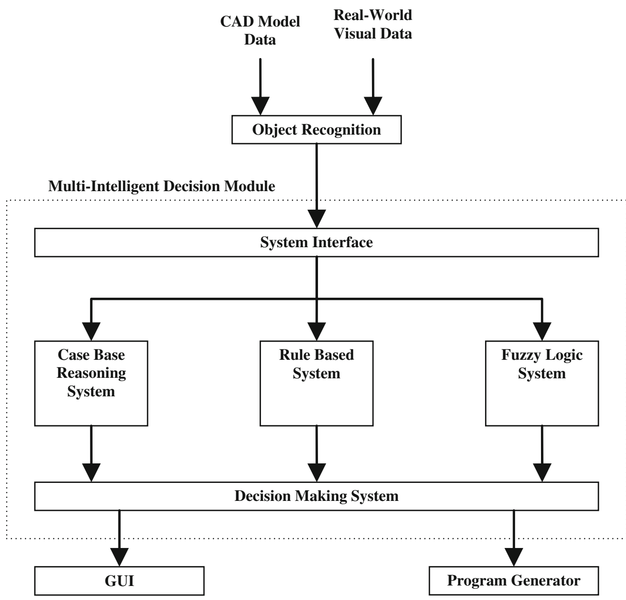

Fig. 3 Multi-intelligent decision module diagram

software had a graphical interface, which was used to create and test prototype algorithms that were exported as VB.net-compatible functions for inclusion within a .net framework software package. In the early prototypes, the image was read, converted to greyscale and then passed through a low-pass filter. The low-pass filter removed some of the noise in the image and reduced the occurrence of small random edges. The image was then operated on by an edge-tracing function which used a Prewitt edge-detection algorithm, and then any edges were collated into a collection of geometric lines. These lines were then overlaid onto the filtered greyscale image for viewing. Later systems used Fourier descriptors [1, 2] and artificial neural networks (ANNs) [6–8], and in the most recent systems described herein new corner-finding algorithms to effectively reduce noise were also introduced.

The many different methods of implementing AI each have their own strengths and weaknesses [9–14]. Some effort has been made in combining different methods to produce hybrid techniques with more strengths and fewer weaknesses. The neuro–fuzzy system which seeks to combine the uncertainty handling of fuzzy systems with the learning strength of ANNs is an example of this. This paper proposes a system using multiple AI techniques to decide on weld requirements for a job. The system will combine real-world visual data captured through the image processing algorithms with the data provided by the CAD model by comparing the expected lines and corners with those in the captured data. It will then use this combined data to present differing AI systems with the same information. These systems will then make weld requirement suggestions to a multi-intelligent decision module (Fig. 3).

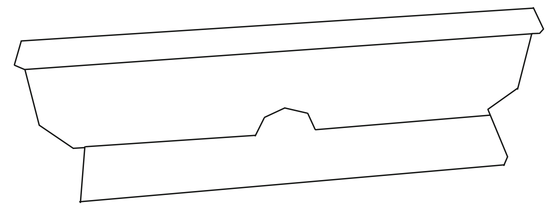

Fig. 4 Image output from edge-detection algorithm after post-processing

This module will evaluate the suggestions and determine the optimum weld path. The suggestions will be passed to the existing robot program generator.

3 Current progress

The current state of the research is that the robot program generation systems have been created and tested. These systems have been used to produce consistent straight-line welds. A simple edge-detection system was created using the WiT software. Figure 1 shows the initial image. Figure 4 shows the edges as detected. The edge detection in this instance is good, as the object can be identified from its perimeter detail. The external perimeter detail is more defined than the internal detail. The work on the AI systems is in its early stages and will be taken further over the next 6 months. During this time the multi-intelligent decision module framework will be completed and combinations of AI techniques will be tested, for example different combinations of rule-based, case-based and fuzzy systems. Meanwhile, improvements have been made to the image processing systems as described herein.

4 Image processing

Information about shape or pattern is held within contours, so Fourier descriptors were applied to the contours of shapes being classified. The edge-detected image in Fig. 4 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 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; rather, it represents a section of length around the contour. Because contours were closed, this implies that there exists a value T such that $f(t + nT) = f(t)$, where nT is the contour length. So, f can be expressed as a complex Fourier series, yielding

$$f(t) = \sum_{-\text{infinity}}^{\text{infinity}} A_n \exp\left[\frac{\mathrm{j}n2\pi t}{T}\right] \tag{1}$$

And Fourier coefficients become : A_n

$$= \frac{1}{T}\int_0^T f(t)\exp\left[\frac{-\mathrm{j}n2\pi t}{T}\right]\mathrm{d}t. \tag{2}$$

For simplicity, the velocity can be such that $T = 2\pi$, and $\frac{1}{T}\int_0^T f(t)\exp[-\mathrm{j}nt]\mathrm{d}t$. These Fourier coefficients depend on the starting point and differ with respect to the 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 $f(t) = f^{(0)}(t + \tau)$.

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

$$y(t) = \sum_0^{2\pi} A_n \exp[\mathrm{j}nt], \tag{3}$$

$$\begin{aligned} A_n &= \frac{1}{2\pi}\int_0^{2\pi} f^{(0)}(t+\tau)\exp[-\mathrm{j}nt]\mathrm{d}t \\ &= \exp[\mathrm{j}n\tau]\frac{1}{2\pi}\int_0^{2\pi} f^{(0)}(t)\exp[-\mathrm{j}nt]\mathrm{d}t \\ &= \exp[\mathrm{j}n\tau]a_n^{(0)}. \end{aligned} \tag{4}$$

Translations, rotations and dilations can be considered as follows:

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

$$f(t) = f^{(0)}(t) + Z = \sum_{-\text{infinity}}^{\text{infinity}} A_n^{(0)} \exp[\mathrm{j}nt] + Z. \tag{5}$$

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 under translation. A_0 is the complex vector indicating the position of the centre of gravity.

Rotation: If the centre of gravity is at the origin, then rotation of the contour function $f(t)$ about the origin by an angle 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, the coefficients of the rotated contour will be $A_n = \exp[\mathrm{j}\varphi]A_n^{(0)}$.

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

5 Extracting Fourier descriptors

The general form of the 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 $A_n^{(0)}$ are the coefficients of the original contour. They are not useful in this form because they contain information on orientation, whereas only shape information is needed. Considering $B_n = A_{1+n+1}{\cdot}A_{1-n}/A_1^2$ and applying rotation, translation and dilation results in an expression that does not contain τ, R or φ. If the coefficient A_0 is not used, then these B_n coefficients are invariant under translation, rotation and dilation. Thus, the coefficients B_n represent the shape (or form). These Fourier coefficients are invariant under translation, rotation and dilation and just represent the shape [2]. ANNs were trained using back-propagation algorithms. Back-propagation is a common method for teaching ANNs to perform a given task, dating back to the late 1960s. Nets were considered to be trained when the error became zero (within pre-set ranges). A number of teaching runs were required before outputs converged.

6 Testing

It is most difficult to differentiate between shapes that are similar. For testing in this part of the work, four metal bar parts were selected as a worst case. The parts were of the type shown in Fig. 1 but of different lengths: 1, 1.25, 1.5 and 2 m. A teaching net was created to take two sets of inputs and two sets of demand vectors. The layout of the ANN was a 5–38–4 pattern, i.e. a layout with five input neurons, 38 hidden neurons and four output neurons. Errors were used to update weights within the ANN. A number of teaching runs were required before outputs converged. After 150 teaching runs, the network gave some suitable outputs. Weights were saved. The application net was combined with the description program and set up to analyze two shapes in different orientations. Tests then involved presenting images (video frames) to the system until a decision was made. In 100 tests using the taught system, the program classified 98 shapes correctly after three frames of video. When presented with two input sets, the system showed a 98% classification rate within three frames.

The training net was then modified to take 3 sets of inputs; the most recent results are presented here. Weights were frozen after 500 test runs, and the outputs are presented in Table 1. The desired outputs for each part are a certainty value of 1 that the part was recognised and two values of 0 to show that the other two parts are rejected as solutions. For each part, the higher the certainty value for

Journal : Large 773 | Dispatch : 6-12-2011 | Pages : 8
Article No. : 154 | ☐ LE | ☐ TYPESET
MS Code : 154_JMST-D-09-00030 | ☑ CP | ☑ DISK

Table 1 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 2 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

that part the better, and the lower the other two values the better.

Programs were tested with 3 different parts of a ship in different orientations. In 100 tests the program classified 97 shapes correctly after three frames. The 3-pattern recogniser achieved 97% classification. Programs were then modified to take 4 training sets and demand vectors. This ran for 2 h, and the outputs observed after 6219 test runs are presented in Table 2.

Over 50 tests, the program classified 44 shapes correctly after three frames. The 4-pattern recogniser worked with 88% classification.

The results were good compared with other systems, but attempts were made to improve the results further by carrying out some post-processing on the edge-detected image. The various sets of outputs are those recorded after teaching.

7 Improving the system

After processing the edge-detected image (Fig. 4) to obtain a clear image using geometrical rules, the edge was sampled. A method published as a short note in the *Proceedings of the IMechE* was used to convert continuous lines into equally spaced line segments and then to polylines by specifying endpoints for each segment [1]. This is shown in Fig. 5. The new sub-systems successfully distinguished between various ships' parts by:

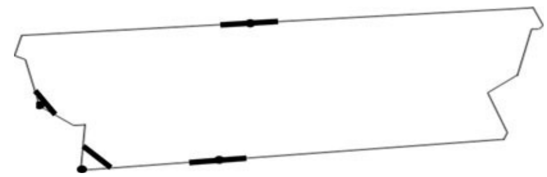

Fig. 5 Sampling points around the edge-detected image

- Edge-detecting the image (Figs. 1, 4)
- Sampling points around the edge-detected image (Fig. 5)
- Calculating distance between endpoints of windows around sampled points
- Taking points with minimum distance to be corners (the shorter bold lines in Fig. 5)
- Using corners and connecting lines to extract Fourier descriptors
- Associating sets of descriptors with training sets
- Deciding.

Points were sampled and corners were detected based on the diagonal length of a segment's bounding box. Interspacing distance was equal to the diagonal of the bounding box divided by a constant M (set to 50). M was determined empirically in this early work by testing a range of values and finding the value that produced the best accuracy; increasing M increased noise, while decreasing M created smoother edges so that some corners were removed.

Points could be sampled once an interspacing distance, S, had been calculated. An empty set was created to store sampled points. Each point was then appended to that set. The distance holder D was initially set to zero. The new algorithm was as follows:

1. The Euclidean distance d between two consecutive points was added to D.
2. If D was less than the interspacing distance S, then i was increment by 1 and step (1) was repeated.

Otherwise

(a) A new point, q, was created, at approximately distance S away from the last sampled point. qx and qy were calculated to achieve a distance $(S - D)/d$ between point $i - 1$ and point i.
(b) q was inserted into the set of sampled points before point i.
(c) Repeat from step (1) without incrementing i until $i >$ |points|.

The new algorithm found corners from this primitive information and from higher-level patterns that determined possible insertions or corner deletions. Firstly, corners were

Journal : Large 773 | Dispatch : 6-12-2011 | Pages : 8
Article No. : 154 | LE | TYPESET
MS Code : 154_JMST-D-09-00030 | CP | DISK

found based on the distance between the beginning of a line segment around a point and the end of that line segment; for example, considering a point at *pi*

$$\text{SEGMENTi} = |pi - W, pi + W|, \qquad ([6])$$

where W is a constant window and $|pi - W, pi + W|$ is the Euclidean distance between points $pi - W$ and $pi + W$.

As the edge of a shape bends at a corner, the SEGMENT of points shortens, and a local minimum SEGMENT is a likely corner. To find an initial corner set, all SEGMENTs were first computed. The median SEGMENT length was found, and a threshold t was set at the median $\times$ 0.9. For each SEGMENT, if the SEGMENT was a local minimum below the threshold t, then the SEGMENT was considered a corner. Line segments around a part all had a window of ±10 points either side of the point being considered (although ±5 were used in practice). Shorter SEGMENTs were found around some points at corners, and those points were considered corners. Points on straighter sections had SEGMENTs that were close to the median SEGMENT length and were not considered to be corner candidates.

After this set of corners was found, some higher-level processing found missed corners and removed false positives. The system checked to see if each consecutive pair of corners passed a line test. This similarity was represented through the ratio of distance(points; a; b) to path $-$ distance(points; a; b). If this ratio was above a set threshold, the segment between points a and b was considered a line. If the part segment between any two consecutive corners did not form a line, then there were additional corners in between. Missing corners were assumed to be approximately halfway between corners. Since these potential corners were below the original threshold t, the threshold was relaxed and the new corner was taken to be the point with minimum SEGMENT. This process of adding corners was repeated until all segments between pairs of consecutive corners were lines.

A check was then conducted on subsets of triplet, consecutive corners. If three corners were collinear, then the middle corner was removed. This process checked and removed false positives. Three consecutive corners were considered collinear if the part segment between the outer corners passed a line test.

Two hundred thirty images of nine different parts of ships that were to be welded were initially used to test the corner finder. A Douglas–Peucker algorithm was implemented along with a simple differentiation algorithm [5]. The algorithms had filters to remove close or overlapping corners. Two measures were used to determine the accuracy of the corner finders: correct number of corners found and an all-or-nothing measure. The first was calculated by dividing the number of correct corners found by the total number of correct corners perceived by observation of each processed image. The second measure checked that only the minimum number of corners to segment a boundary was found (in other words, that the part shape had no false positives or negatives). This was calculated by dividing the number of correctly segmented parts by the total number of parts; it was either correct or incorrect. Results are presented in Table 3.

Table 3 Results for the new system and two other corner finders for comparison

	New system	Douglas–Peucker	Simple differentiation
Corners found correctly	1799	1669	1017
Points wrongly identified as corners	43	115	295
Accuracy	0.98	0.96	0.85
Percentage of lines without any points wrongly identified as corners (%)	87	71	34
Average time per part (ms)	0.8	0.32	1.03

The corner-finding system improved on other corner finders that were considered. Although the new method was slightly slower than the Douglas–Peucker algorithm, the new method found more corners correctly in the images, and wrongly identified fewer points as corners; it gave improved accuracy with all-or-nothing accuracy that was 20% better than that of the Douglas–Peucker implementation. Once corners were identified, the shapes were redrawn so that lines went directly from corner to corner. This removed noise. Fourier descriptors were then extracted from the contours of the shapes being classified.

8 Testing and results for the improved system

The 5–38–4 pattern used in Sect. 6 was reused to compare results. The training net was reset to take 3 sets of inputs and demand vectors. Weights were frozen after 500 test runs, and the outputs are presented in Table 4.

Programs were tested with 3 different shapes in different orientations. In 100 tests the program classified 98 shapes correctly after just one frame, and better than 99 after three frames. Programs were then modified to take 4 training sets and demand vectors. This ran for 6112 test runs. The observed outputs are shown in Table 5. Over 50 tests, the program classified 48 shapes correctly after just one frame and 49 after three frames.

These results were compared with those achieved by the most recently published system for identifying ships' parts [2], using the same shapes for the comparison. The most recently published system used Fourier descriptors on edge-detected shapes without considering corner identification.

Journal : Large 773 | Dispatch : 6-12-2011 | Pages : 8
Article No. : 154 | LE | TYPESET
MS Code : 154_JMST-D-09-00030 | CP | DISK

Table 4 Output from three sets of inputs

Input set	Output	Desired output	Input set	Output	Desired output
1	1	1	3	0.87×10^{-8}	0
	0.1×10^{-6}	0		0	0
	0	0		0.998	1
2	2.7×10^{-6}	0			
	0.998	1			
	4.7×10^{-7}	0			

Table 5 Output from four sets of inputs

Input set	Output	Desired output	Input set	Output	Desired output
1	1	1	3	0.2×10^{-8}	0
	0	0		0	0
	-2.61×10^{-9}	0		1	1
	0	0		-9.54×10^{-8}	0
2	-2.12×10^{-7}	0	4	0	0
	1	1		0	0
	-4.46×10^{-9}	0		-3.31×10^{-8}	0
	-1.3×10^{-7}	0		0.9998	1

With the 2-pattern program, that system only achieved a 98% classification rate within three frames, whereas the current system achieved close to a 100% classification rate with three frames.

The 3-pattern recogniser achieved 97% classification after three frames, but the new system achieved 99% classification.

The 4-pattern recogniser achieved 88% classification after three frames in the most recently published system, while the new system achieved 98% classification after three frames. The new system was significantly better after 3 frames but was radically better after being shown only one picture of a part. Graphs showing percentage accuracy compared with number of frames for distinguishing between three or four different shapes are shown in Fig. 6; lines with squares correspond to the previous prototype system, while circles correspond to the new system.

The graphs in Fig. 6 compare the increase in percentage accuracy as the number of frames considered is increased, for the most recently published system and for the system described here. The vertical axis indicates the percentage accuracy and the vertical axis represents the number of frames. Figure 6a shows the results when trying to identify three different parts, and Fig. 6b shows the results when trying to identify four different parts. Substantial

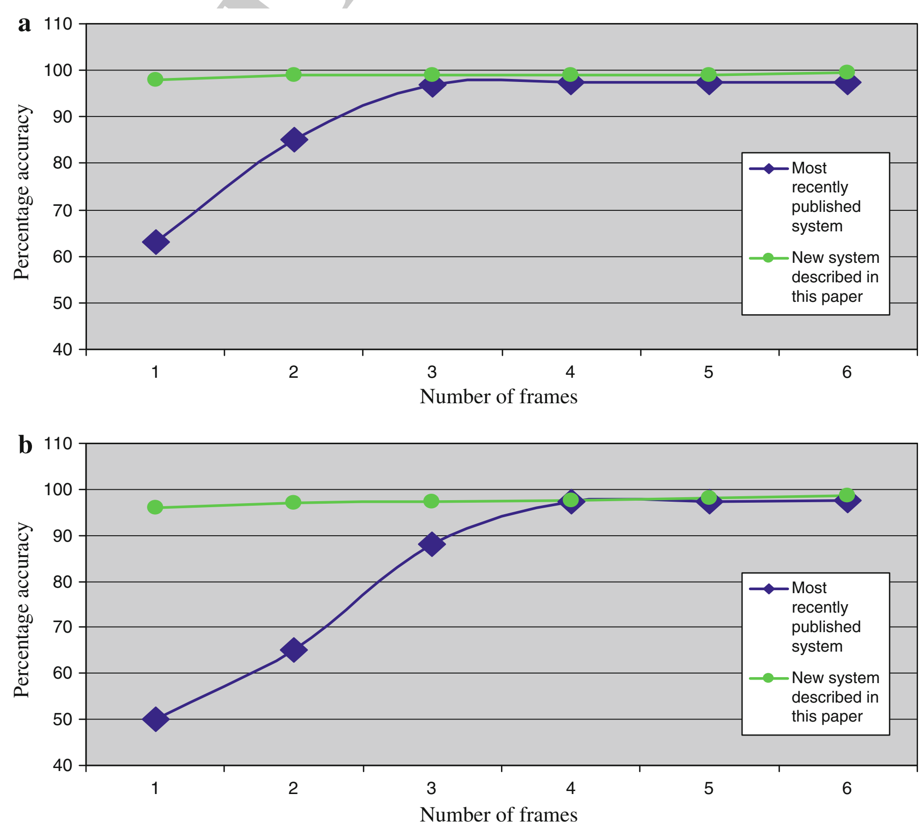

Fig. 6 Comparison of the prototype system with the new system incorporating the corner finder

Journal : Large 773 Dispatch : 6-12-2011 Pages : 8
Article No. : 154 LE TYPESET
MS Code : 154_JMST-D-09-00030 CP DISK

improvement was demonstrated when the new corner finder was added. Tables 4 and 5 show that the improvement was especially significant when more parts needed to be differentiated and when a part needed to be identified quickly (after only one frame).

9 Discussion and conclusions

A proposed system that uses image processing techniques in combination with a CAD model to provide information to a multi-intelligent decision module has been presented. This module will use different criteria to determine a best weld path. Once the weld path has been determined, the program generator and post-processor can be used to send a compatible program to the robot controller. Progress so far has been described.

Different shapes have been successfully identified using a simple pattern recognition system that used an ANN, and that system was improved by using a corner identifier. The system provided shape contour information that was invariant under size, translation and rotation. Since acquiring and processing new images is an expensive task, it is desirable to take a minimal number of additional views, and the new methods quickly and successfully identified parts after only one frame.

The testing used four similar metal bar parts, as differentiating between such similar shapes is a worst case for such testing. If a variety of different types of structural members of a ship had been selected, for example flat metal plates and metal bars joined at corners etc., then they would have been easier to differentiate.

The new system used a rudimentary curvature metric that measures Euclidean distance between two points in a window. These corners were then processed to ensure that every segment between corners was a line and that any extraneous points in the middle of a line segment were removed. The improved accuracy and ease of implementation of this approach can benefit other applications requiring curve approximation, node tracing and image processing, but especially in identifying images of manufactured parts with distinct corners.

The initial results from the whole work suggest that a combination of systems (case-based and rule-based reasoning, fuzzy logic and artificial neural network) could offer the ability to handle the necessary uncertainty whilst still returning a correct weld path (when all/enough factors are known).

References

1. Sanders DA, Lambert G, Pevy L, Tewkesbury GE (2009) Improving robotic welding in the shipbuilding industry through the recognition of shipbuilding parts by pre-locating corners in images. In: Proceeding of 8th International Conference on Computer and IT Applications in the Maritime Industries COMPIT, vol 9, pp 349–363
2. Sanders DA (2009) Recognizing shipbuilding parts using artificial neural networks and Fourier descriptors. Proc Inst Mech Eng Part B J Eng Manuf 223(3):337–342
3. Sanders DA, Rasol Z (2001) An automatic system for simple spot welding tasks. Total Vehic Technol Chall Curr Think 263–272
4. Smach F, Lemaitre C, Gauthier JP, Miteran J, Atri M (2008) Generalized Fourier descriptors with applications to objects recognition in SVM context. J Math Imaging Vis 30(1):43–71
5. Sanders DA, Lambert G, Pevy L (2009) Pre-locating corners in images in order to improve the extraction of Fourier descriptors and subsequent recognition of shipbuilding parts. Proc IMechhE B J Eng Man 223(9):1217–1223
6. Sanders DA, Tewkesbury GE (2009) A pointer device for TFT display screens that determines position by detecting colours on the display using a colour sensor and an artificial neural network. Displays 30(2):84–96
7. Ripley BD (1996) Pattern recognition and neural networks. Cambridge University Press, Cambridge
8. Erwin-Wright S, Sanders D, Chen S (2003) Predicting terrain contours using a feed-forward neural network. Eng Appl Artif Intell 16(5–6):465–472
9. Sanders DA, Graham-Jones J, Gegov A (2010) Improving ability of tele-operators to complete progressively more difficult mobile robot paths using simple expert systems and ultrasonic sensors. Indus Robot J 37(5):431–440
10. Sanders D, Tan YC, Rogers I et al (2009) An expert system for automatic design-for-assembly. Assembly Autom J 29(4):378–388
11. Sanders D (2010) Comparing ability to complete simple tele-operated rescue or maintenance mobile-robot tasks with and without a sensor system. Sens Rev J 30(1):40–50
12. Sanders DA, Lambert G, Graham-Jones J, Tewkesbury GE, Onuh S, Ndzi D, Ross C (2010) A robotic welding system using image processing techniques and a CAD model to provide information to a multi-intelligent decision module. Assembly Autom 30(4):323–332
13. Sanders DA, Tewkesbury GE, Robinson DC (2011) Simple expert systems to improve an ultrasonic sensor-system for a tele-operated mobile-robot. Sens Rev J, Paper No: SR-10-647
14. Sanders D (2009) Introducing AI into MEMS can lead us to brain-computer interfaces and super-human intelligence. Assembly Autom J 29(4):309–312

Journal : Large 773	Dispatch : 6-12-2011	Pages : 8
Article No. : 154	☐ LE	☐ TYPESET
MS Code : 154_JMST-D-09-00030	☑ CP	☑ DISK