ROBOT TRACK RECOGNITION USING NEURAL NETWORK

Sukkar, F.

Faculty of Informatic Engineering.University of Aleppo.
P.O.Box 6502, Aleppo, Syria.

ABSTRACT

This paper addresses the design and implementation of a neural network architecture for improving the performance of Robot track recognition.

An approach for track images where the correspondence of a subset of boundary points to a data model is simultaneously determined. The data field has more analysis features than any other, for both raw data and the trained neural network's solution. Global shape parameters derived from the variation of track points in a training set are used to model the image.

A formulation, based on this prior knowledge and the edge information of the track image, is employed to the image boundary with its subset points in correspondence with boundaries in the training set.

Finally this paper presents the results of the Robot tracks of this architecture, integrated with the best-studied learning algorithm. A number of experiments are performed on Robot track images to evaluate the approach, including the validation of the dependence of the method on image quality. The proposed neural network has fast training algorithm, less error, better recognition, and generalises well for new track images.

1. INTRODUCTION

Neural networks are named after the cells in the human brain that perform intelligent operations. The brain is made up of billions of neuron cells. Each of these cells is like a tiny computer with extremely limited capabilities; however, connected together, these cells form the most intelligent system known. Neural networks are formed from hundreds or thousands of simulated neurons connected together in much the same way as the brain's neurons.

Just like people, neural networks learn from experience, not from programming. Neural networks are good at pattern recognition, generalisation, and trend prediction [1]. They are fast, tolerant of imperfect data, and do not need formulas or rules. Neural networks are trained by repeatedly presenting examples to the network. Each example includes both inputs (information to make a decision) and outputs (the resulting decision, prediction, or response).

The network tries to learn each of examples in turn, calculating its output based on the provided inputs. If the network output doesn't match the target output, Intelligent algorithm corrects the network by changing its internal connections. This trial-and-error process continues until the network reaches specified level of accuracy [2]. Once the network is trained and tested, new input can be given to the network, and it will produce a correct output. Neural network architecture is largely a matter of identifying which data is input, and what to predict, assess, classify, or recognise.

Interest in the development and study of image recognition has increased recently in response to both the Artificial Intelligence and industries, growing interest in massively parallel solutions to certain classes of problems. [3] implements learning of behaviours, again in simulation first, and then “transplanted” to a real vision.

The physically grounded approaches to the development of intelligent autonomous vision has been demonstrably successful in programming individual images [4].[5] implements a backpropogation based network to generalise over similar states. The architecture makes the integration of behaviours both incremental and extensible. [6] builds an explicit world model, and seeks a policy that maximises average reward.

Recent work in learning has shown that this learning paradigm offers much to those seeking to develop both behaviours and image functions for track recognition [7]. Numerous sources, particularly [8] discusses the difficulties of scaling up learning to the state and action spaces present on a physically realised image. [9] demonstrates significantly improved performance in the image domain by enhancing the traditional learning paradigm with multiple competitive agents.

The remainder of the paper is organised as follows. Section 2 descrides track model, and how it has been designed for capturing the features. Section 3 reports on the use of neural network to model different tracks. Section 4 demonstrates the effectiveness of the new technique. The paper concludes with Section 5.

2. TRACK MODEL

2.1 Track Points

To extract data to represent the tracks whose boundaries (for instance) are shown in Figure 1. Each track shape should be represented as a set of data points. The input model is then based on the positions of the Robot on the aligned images, and the main variation of the points on the Robot track.

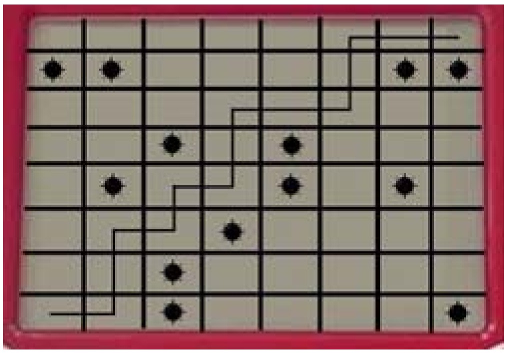

Figure 1.

2.2 Capturing the features

The critical points on the boundary are usually easily indented features, such as high curvature points, sharp corners, etc. Equally spaced points are interpolated between the critical points Figure 2. The training set points are aligned to minimise a weighted sum of squares of distances between equivalent points on tracks.

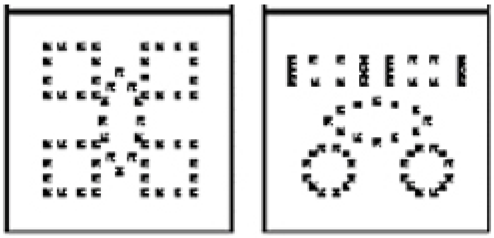

Figure 2. Critical points

Figure 3 shows analytic track image with its point model of the boundary 4 critical points, (large dots), shown with interpolated points (small dots);

An initial sparse set of points on the study track are determined based on proximity and shape. The interpolated points are then generated by the shortest iformation paths between the initial points and then labeling the interpolated points equally spaced along these tracks. Combinations of vectors, one for each track, can move the modeled image points anywhere in the image. Any shape can also be approximated using the mean shape and a weighted sum of deviations obtained from the tracks.

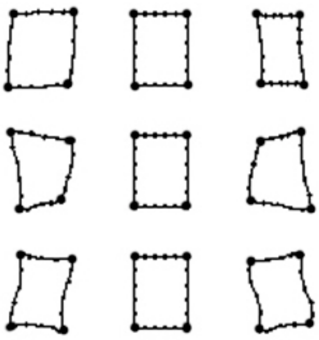

Figure 3. Analytic track model

3. NEURAL NETWORK ARCHITECTURE

Learning is a machine intelligence paradigm, closely related to dynamic programming, and complementary to supervised learning, that learns to achieve a goal through trial-and-error interactions with an environment.

Ideally, if track images by group are chosen completely at random, or are chosen to minimise overlapping strategies, we can hope for an improvement in convergence speed that is linear with the number of images. Since the actions of tracks learning together probably do in fact overlap a great deal, a sub-linear improvement in learning can be found, but it seems that some improvement in convergence speed is guaranteed since more actions will be tried from more states.

3.1 Network Sizing

The number of inputs and outputs of the neural network are determined by the number of parameters that are used during compression of the input and output track vectors. The number of hidden units however could not be readily determined. Figure 4 shows how the error decreases when number of hidden layers and number of output neuron increase.

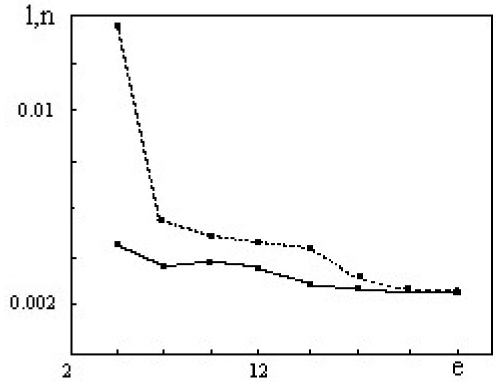

Figure 4. Error versus hidden layer and output neurons

3.2 Algorithm

The algorithm starts with segmented track images, which can be determined automatically. Software procedure can read the image and transform the track into real datapoints. Input data file is readily applied to the neural network.

The image details the track processing. A simple algorithm for the detection of track is used. The images are reduced to a 20*20 datapoints entered directly into the neural network. Pixel image processing is performed on the smaller image.

The count of pixels in each of the 20 columns is totaled (represented as a histogram at the bottom of the image) and flattened into various values. These values represent the amount in the right, center, and left of the image's track of view.
A threshold value of pixels per image is used, after the threshold has been met, the desired direction is the screen portion with the highest number of pixels. As a real-time control program, the performance of this application is a crucial design criterion. The overall performance of this application in controlling and communicating with the image is found by running the control loop and timing its duration. Details are illustrated in Figure 5.

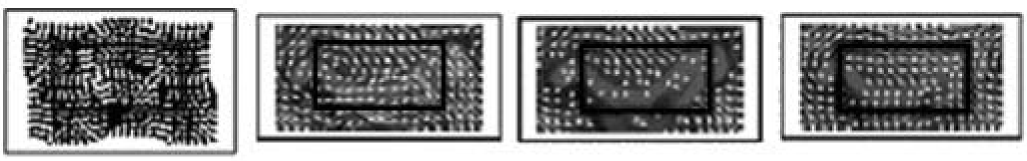

Figure 5. Track image processing

4. RESULTS

The neural network helps filter the data into meaningful results. While performance is notably improved at the beginning of learning by the parallel architecture, in the long run, the performance of the independent tracks is significantly better.

First, data is collected to illustrate how each track image is gaining additional experience from the other images exploring most important track features. Figure 6 shows different track models and valuable datapoints obtained for training.

Figure 6. Datapoints for different track models

The images are run in the ring to train, and then run with learning and random actions turned off to gauge the effectiveness of the neural network.

The images are modeled by 72 available tracks: 400 values for each track, Decision epochs are varied in experiments between 150-200. This, of course, implies that the actions taken by an image are not always the ideal ones.

The learning rate was set to 0.7 throughout the experiments, while the momentum parameter was set to 0.02. To ensure coverage of the output response, the wetights of the network were chosen at random. However, this form of random exploration is advantageous since it means that we do not have to artificially impose a division of the track image's execution time into training and running phases, and prevents the track from becoming "trapped" in a simple cyclic set of actions for an extended period.

The following graph Figure 7 presents the evolution of the learning of the network with respect to time. Plotted at each point in time the number of bumps the network suffered averaged over the next minute. While performance is notably improved in the beginning of learning by the parallel architecture, in the long run the performance of the independent neurons is significantly better.

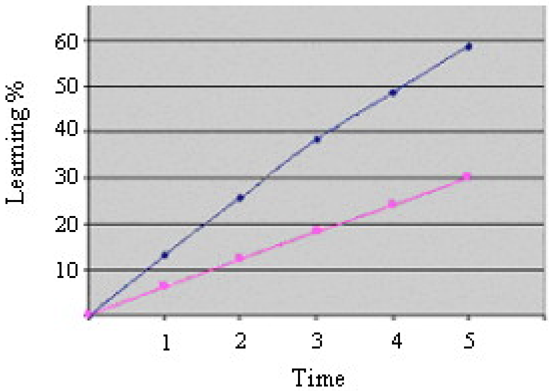

Figure 7. Learning performance

The following graph Figure 8 presents the evolution of the learning of the images with respect to time for different learning rates, shows how the number of experiences each track image gets is improved by the parallel architecture.

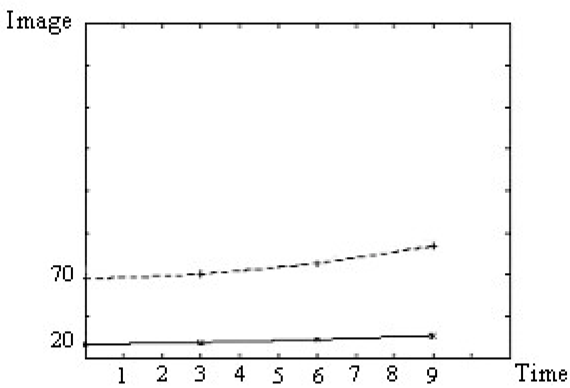

Figure 8. Image experience

Experiment are carried out where the number of hidden units of the neural network is varied and the training error is measured when the inputs are not compressed and when they are compressed. The resultant graph is shown in Figure 9. From Figure 9, the following observations can be obtained: 1) In learning stage, the training error tends to oscillate, i.e., as the training error tends to increase and decrease and increase again as is varied. The training error would decrease monotonically with the time. 2) As the number of tracks increases the oscillation in the training error tends to be damped in the sense that the oscillation is less pronounced as the number of tracks increases. Observe that for the error values are almost the same. 3) The most useful observation is that as input data code increases the difference in the training errors in both the domains tend to converge to almost the same value. This is very useful because training using compressed data is much faster than with uncompressed data. Also, the errors are almost identical in both.

Figure 9. Error reduction

Further experiments are carried out to ascertain how significant the improvement is in the short term. Figure 10 shows the average of 5 runs: This graph confirms the previous results: the neural network architecture improves performance in the many tracks when datapoints are increase. All results are good within reasonable ranges. Note, error curves are not symmetric due to the non-linearity.

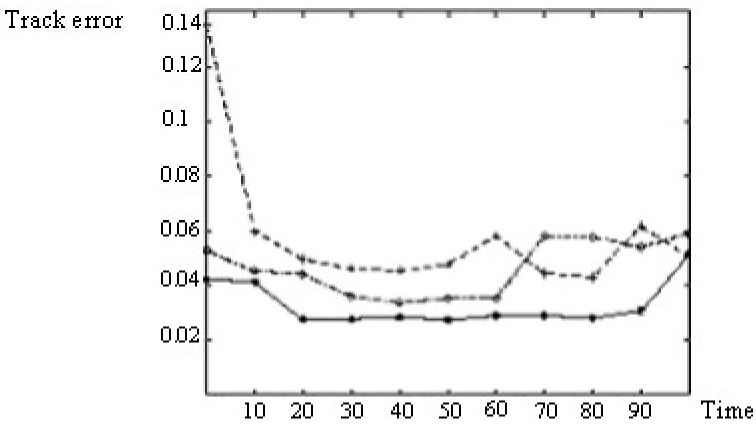

Figure 10. Neural network performance

5. CONCLUSION

The contribution of this paper is that to extend techniques to provide a more robust method for Robot track image recognition. The use of neural network allows to adjust the weighting between the input prior knowledge of the neural network and the image track information based on the image quality and the reliability of the training set. Algorithm based on a training set derived track image datapoints for the prior shape and pose parameters are used during training process.

In order to show the important role of the new technique, a series of experiments are carried out. During recall, the recognition is made for the track images, resulting in a better optimum. In addition, the new technique is faster, accurate and less error is made. Preliminary investigations into the effectiveness of the technique show high promise, and work will be continued in this area.

6. REFERENCES

[1] D. Brett, A. Hill and C. J. Taylor, A method of 3D surface correspondence and interpolation for merging shape examples, Image and Vision Com-puting, Vol. 17, no. 8, pp. 635-642, 1999.

[2] K. Jain, Y. Zhong, S. Lakshmanan, Ob-ject matching using deformable templates, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 18, no. 3, pp. 267-278, 1996.

[3] D.T. Pham, and M.F.Sukkar, Supervised adaptive resonance theory neural network for modelling dynamic systems, IEEE SMC, Vancouver, Canada, Vol. 3, pp. 2500-2505, (1995).

[4] D.T. Pham, and X.Liu, Neural networks for identification, prediction and control, Springer-Verlag, London, Berlin and Heidelberg, (1995).

[5] F. L. Bookstein, Landmark methods for forms without landmarks: Morphometrics of group dif-ferences in outline shape, Medical Image Analysis, Vol. 1, no. 3, pp. 225-243, 1997.

[6] N. Duta and M. Sonka, Segmentation and inter-pretation of MR brain images using an improved knowledge-based active shape models, Informa-tion Processing in Medical Imaging, pp. 375-380, Spring-Verlag Berlin Heidelberg New York, 1997.

[7] T. F. Cootes, C. J. Taylor, D. H. Cooper and J. Graham, Active shape models { their train-ing and application, Computer Vision and Image Understanding, Vol. 61, no. 1, pp. 38-59, 1995.

[8] T Fukuda, and T Shibata. Theory and applications of neural networks of industrial control systems, IEEE Transactions on Industrial Electronics, Vol.39, No.6, pp.472-503, (1992).

[9] X. Zeng, L. H. Staib, R. T. Schultz and J. S. Duncan, Segmentation and measurement of the cortical from 3D MR images, First Inter-national Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 519-530, LNCS 1496, Springer, Berlin, October 1998.