Omni-directional Motion: Pedestrian Shape Classification using Neural Networks and Active Contour Models

Ken Tabb, Neil Davey, Rod Adams & Stella George

e-mail: {K.J.Tabb, N.Davey, R.G.Adams, S.J.George}@herts.ac.uk
Computer Science Department, University of Hertfordshire, England

Abstract

This paper describes a hybrid vision system which, following initial user interaction, can detect and track objects in the visual field, and classify them as human and non-human. The system incorporates an active contour model for detecting and tracking objects, a method of translating the contours into scale-, location- and resolution-independent vectors, and an error-backpropagation feedforward neural network for shape classification of these vectors. The network is able to generate a confidence value for a given shape, determining how 'human' and how 'non-human' it considers the shape to be. This confidence value changes as the object moves around, providing a motion signature for an object. Previous work has accommodated lateral pedestrian movement across the visual field; this paper describes a system which accommodates all angles of pedestrian movement on the ground plane.

Keywords: Snake, Active contour model, Pedestrian, Human, Shape classification, Neural network, Omni-directional, Axis crossover vector, Ground plane.

1. Introduction

Object classification is a common requirement of computer vision and target-based tracking systems. Different techniques exist for estimating an object's position and location within an image [1, 2, 3], which can generally be divided into Marr's low- and high- level categories [4], or combined active vision techniques. Low level 'data-based' vision techniques are only able to identify the shape of the object. Higher level 'model-based' techniques are able to estimate what type of object is being tracked from that shape based on apriori model information, but this process typically places much computational load on the task due to the necessary template matching and / or model manipulating stages of the technique which, by necessity, have to operate at runtime. Similarly, active vision techniques which combine both low and high level techniques are subject to the same pitfalls, although the process is often less computationally intensive due to heuristic information gathered from the low level technique which decreases the high level process' search space [3, 2].

In this paper we present a computationally cheap, and reasonably accurate technique for detecting and tracking moving objects, and for determining whether or not those objects are human. The technique involves three stages. Firstly an active contour model [5] is used to detect and track an object in a sequence of images and to obtain, for each frame, the object's shape as a contour. Secondly, axis crossover vectors [6] are used to re-represent the contour as a scale-, location-, resolution- and control point rotation-invariant vector. Finally a feedforward error backpropagation neural network is used to classify the axis crossover vector as 'human' or 'non-human'.

The particular issue examined in this paper is the extent to which the technique described in [7] allows the classification of human shapes to be undertaken when the motion of the human is in an arbitrary direction with respect to the viewer.

2. Detecting and Tracking Moving Objects

In order to detect objects in the visual field, an active contour model is employed. Our model is based on the Fast Snake model [8] as it was the most suitable for the purposes of automated pedestrian detection [9], although in theory any active contour model would fit into the technique we present.

The video image is preprocessed using a series of image convolutions [8, 9] involving motion detection, blob removal, edge enhancement, and gradient normalisation (Fig. 1). This in turn makes the active contour model's task simpler by reducing potential local minima present in the energy space.

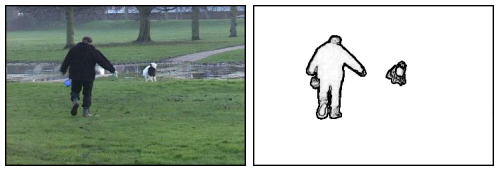

Figure 1. Preprocessing the image improves the performance and accuracy of the active contour model. [Left] The video image prior to preprocessing. [Right] The same image following preprocessing; most of the image has been removed with the exception of the 2 moving objects and slight artefacts around the borders of the objects.

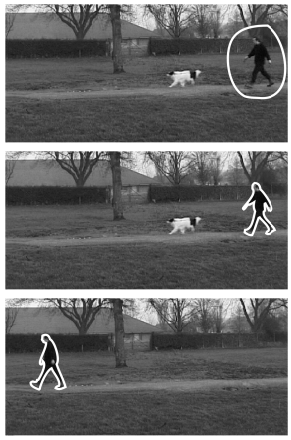

Figure 2. An active contour model detecting and tracking a pedestrian. The active contour is itself operating on preprocessed versions of the video frames, (Fig. 1), but is shown on the raw video frames here for visualisation purposes.

The user places an initial contour around an object of interest early in the video sequence and the active contour model closes in on the object until it relaxes around the object (Fig. 2). It does this by minimising a predetermined energy function which attracts the snake towards areas of interest in the image. Once the active contour has minimised its energy and settled in a particular position within the image, it uses this position as a starting point in the next frame of the video, from which it once again minimises its energy to achieve a new position in the new frame. Using this iterative approach the active contour is able to continue tracking an object through a video sequence. The result is a sequence of contours which depict the object's silhouette during the video frames (Fig. 3).

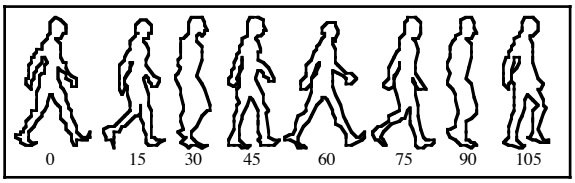

Figure 3. Resultant contours following the tracking of a pedestrian through a video sequence using an active contour model. The frame numbers are shown below each frame.

At the end of each frame of the video, the active contour is stored as a vector of (x,y) coordinates, one for each control point on the contour. These (x,y) coordinates are measured in the image's coordinate system, thus the vector not only includes the number and order of control points on the contour, but also their location in the image, and subsequently the size of the contour.

3. Classifying an Object's Shape

Active contour models are unable to provide any higher level information concerning the class of object detected, which is often of use in the domains of surveillance and target-based tracking. Consequently, an additional stage is needed to analyse the contour's shape and classify the type of object being tracked. As this extra task is vector classification, a neural network is an appropriate tool for this purpose.

In order to use a neural network to discriminate differences in shape, the shapes have to be represented in an appropriate manner. Here, an axis crossover vector [6] is used to map contours onto fixed-length vectors in such a way that the representation is scale-, location-, resolution- and control point rotation-invariant. Fig. 4 shows how the mapping takes place.

Each element of the axis crossover vector is fed in as input to a corresponding input neuron (Fig. 5). Therefore to use the axis crossover representation it is

necessary to determine the appropriate number of axes, and therefore the length of the representational vector, as this dictates the size of the input layer. Previous experiments showed optimal performance for classifying human and non-human contours when using 16 axes [6].

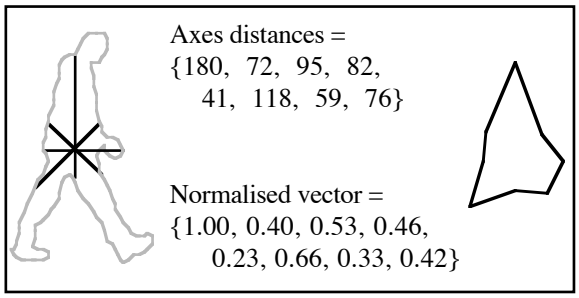

Figure 4. The axis crossover representation. [Left] A contour has several axes projected from its centrepoint to its edges. [Middle] The distance (in pixels) from the centre to the furthest edge, along those axes, are stored in a vector, which is then normalised. [Right] The axis crossover vector depicted as a polygon, indicating the shape information which is retained in the vector.

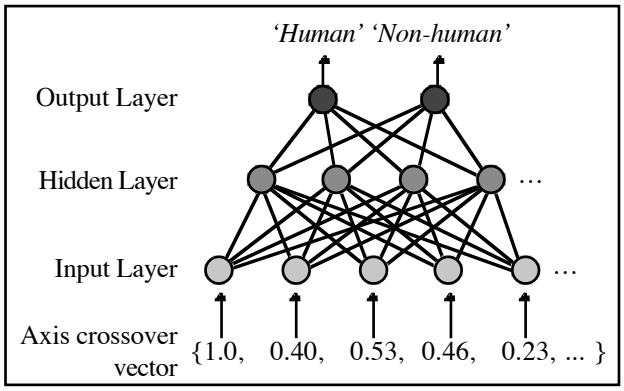

Figure 5. The neural network architecture. The axis crossover vector feeds into the input layer of the neural network. The two output units are trained to fire mutually exclusively, to indicate that the given shape is either 'human' or 'non-human'.

A feedforward neural network with one hidden layer of 13 units, found from experimental investigation [6], was trained using error backpropagation to classify axis crossover vectors as 'human' or 'non-human', as shown in Fig. 5. The network has two outputs which were trained to fire mutually exclusively, so that the confidence in a classification can be calculated as the difference between the two outputs. If only one output unit were present, then it becomes difficult to determine whether a midrange output was caused by the contour containing both 'human' and 'non-human' qualities, or by it containing neither.

4. Experiments & Results

Previous experiments using the axis crossover vector representation showed that, with a training set containing only computer generated (CG) images of humans and other shapes, a neural network could accurately classify previously unseen examples of both real and CG objects [7]. All objects in these experiments were moving orthogonally to the camera, i.e. from side to side, as in Fig. 3.

In addition to correctly classifying unseen examples of previously seen object classes (human, horse, dog), the network was able to classify a previously unseen object class (velociraptor) as being neither very human nor very non-human. Prior to classifying the velociraptor shapes the network had only experienced one bipedal class during training (humans); all non-human classes in the training set were of quadrupeds (horses, dogs) or were of inanimate objects (trees, cars etc.). It is therefore promising that the 16-axis crossover vector could encode sufficient shape information that the network could segregate different biped classes.

It was interesting to test this same network's capabilities of classifying objects whose motion is at a fixed arbitrary direction to the camera, as in for example the pedestrian in Fig. 1. To this end, a test set was formed which contained unseen examples of both real and CG humans and non-humans moving in fixed arbitrary directions on the ground plane. This showed, unsurprisingly, that the neural network found it progressively more difficult to correctly classify objects as their direction of motion moved away from that of their corresponding object class' examples in the training set. Thus, objects moving directly away from or towards the camera, were poorly classified (Fig. 6). Also of note is that classification of the unseen velociraptor object class was also affected by this phenomenon.

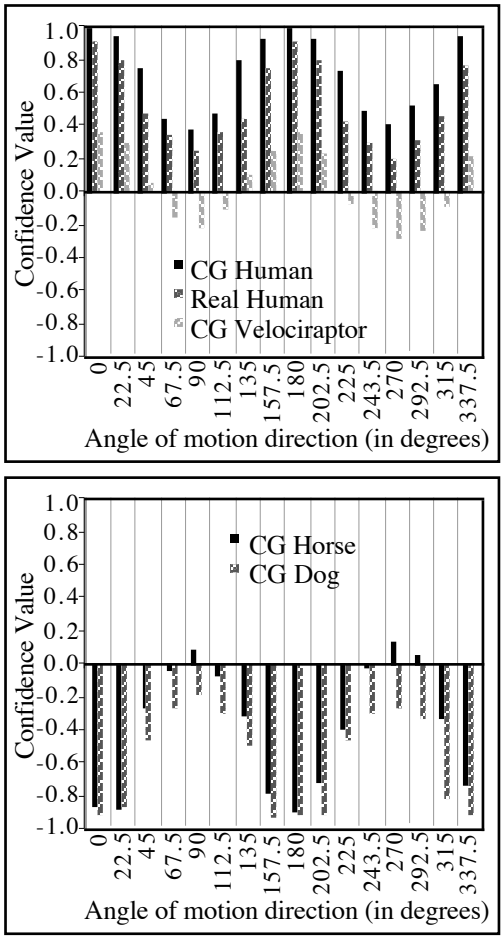

Figure 6. Neural network classification of objects moving in fixed arbitrary directions. When the object is moving at 0° it is moving from the left side of the image to the right (as in Fig. 3); when the object is moving at 90° it is facing the camera. Each value on the graph is the mean classification of 20 different unseen examples of the given object class at the given angle. A confidence of +1.0 denotes maximum confidence that a given shape is 'human'; a value of -1.0 denotes maximum confidence that a given shape is 'non-human'. A value of 0 denotes ambiguity in a given object's shape. Classification is maximally inaccurate at 90° and 270°.

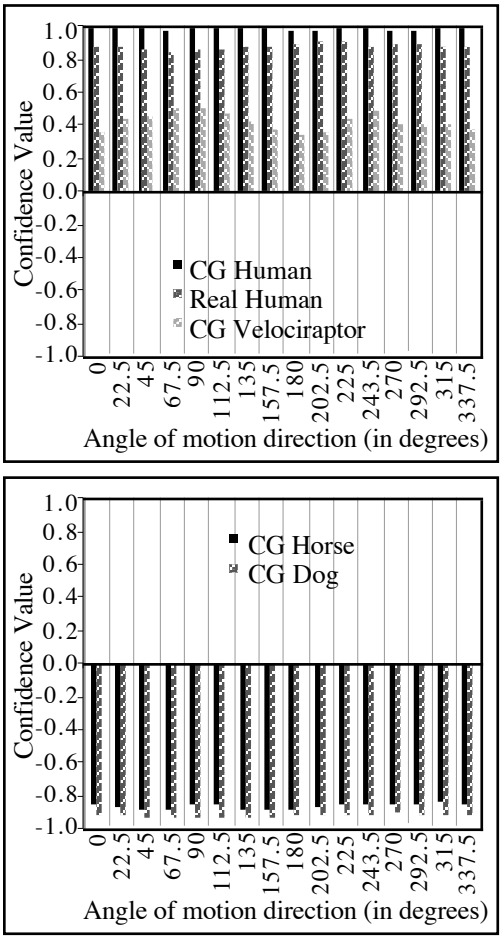

Figure 7. Neural network classification of objects moving in fixed arbitrary directions. When the object is moving at 0° it is moving from the left side of the image to the right (as in Fig. 3); when the object is moving at 90° it is facing the camera. Each value on the graph is the mean classification of 20 different unseen examples of the given object class at the given angle. A confidence of +1.0 denotes maximum confidence that a given shape is 'human'; a value of -1.0 denotes maximum confidence that a given shape is 'non-human'. A value of 0 denotes ambiguity in a given object's shape. The network is able to correctly classify unseen objects also moving in arbitrary directions.

It was therefore necessary to amend the training set to include humans and non-humans moving in fixed

arbitrary directions on the ground plane. A neural network containing the same architecture was trained, again using only CG images of humans and non-humans. In this experiment, however, the objects in the training set were moving in all directions along the ground plane. The training set contained 600 CG humans, 300 CG dogs, 300 CG horses, and 200 CG inanimate outdoor objects, such as trees and cars, and the network was trained to within 5% error. Again, the velociraptor class was omitted from the training set in order to test the network's generalisation skills. The test set contained 20 previously unseen examples of each of the animate object classes, including velociraptors, in each of the 16 directions measured (0°, 22.5°, 45°, ..., 337.5°).

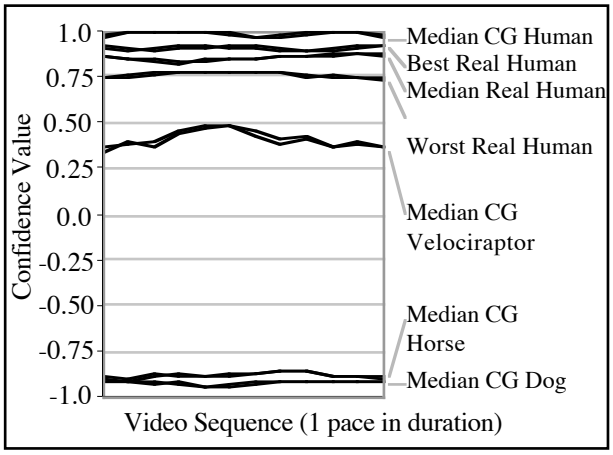

Figure 8. Neural network classification of objects moving in fixed arbitrary directions over a time period of one pace of their motion. The confidence value is measured by subtracting one output unit's value from the other's. A confidence of +1.0 denotes maximum confidence that a given shape is 'human'; a value of -1.0 denotes maximum confidence that a given shape is 'non-human'. A value of 0 denotes ambiguity in a given object's shape. For each object shown, two consecutive paces have been overlaid to illustrate the cyclical nature of the network's classification during repetitive motion.

As can be seen from Fig. 7, this more representative training set results in a more robust classification of unseen shapes by the neural network, irrespective of the angle of viewing / object motion. Both the CG and real human shapes obtained a strong 'human' classification, whereas the CG dog and CG horse shapes obtained a strong 'non-human' classification. As with the previous experiments the 'near miss' bipedal CG velociraptor object class obtained a 'more human than non-human' classification. Nevertheless it is still consistently weaker than the CG and real human classifications, in all 16 directions.

Fig. 8 shows that, as with previous experiments [7], the new network classifies a given object in a cyclical fashion when the object travels in a straight line along the ground plane. The cycle repeats once with each pace taken, and is independent of the object's direction of motion along the ground plane, provided it is in a straight line. The graph shows the network's classifications of 2 consecutive paces overlaid for a number of objects, each object walking in a fixed arbitrary direction. The two paces are overlaid on the graph, to show their similarity. As with Fig. 7, the network can be seen to classify moving human objects differently to moving non-human bipeds and quadrupeds.

5. Discussion

This paper details an experimental evaluation on the use of active contour models in a video sequence. The resulting series of contours can be translated into shape-, location-, resolution- and control point rotation-invariant vectors. These vectors can in turn be used to train a neural network to classify different classes of object shapes.

The neural network architecture from previous experiments has been trained using an extended training set, containing shapes of each object class viewed from arbitrary angles. Consequently, the neural network can correctly classify previously unseen objects moving in arbitrary directions around the ground plane.

Despite being trained using only computer generated shapes, the neural network is able to accurately classify real human shapes, as well as unseen CG examples of humans, horses and dogs. In addition, when presented with CG examples of an entirely new object class (velociraptor), the network is able to classify these shapes as being neither very 'human' nor 'non-human', without affecting its ability to classify the previously experienced object classes.

By combining an active contour model with a neural network, the only processing that needs to happen in real time is the active contour relaxing its energy, and the resulting contour being passed through the (previously trained) neural network. Potential applications of this

technique include target-based tracking scenarios, and other vision-based surveillance domains where processing overhead demands that only part of the video image be processed rather than the entire video image.

References

[1] Cootes T.F., Taylor C.J., Cooper D.H. & Graham J., Training models of shape from sets of examples. In *Proceedings of the British Machine Vision Conference* 1992, pp. 9-18.

[2] Sonka M., Hlavac V. & Boyle R., *Image Processing, Analysis and Machine Vision,* Chapman & Hall, 1994.

[3] Watt A. & Policarpo F., *The Computer Image*. Addison-Wesley, 1999.

[4] Marr D., *Vision*. Freeman, 1982.

[5] Kass M., Witkin A. & Terzopoulos D., Snakes: active contour models, In *International Journal of Computer Vision* 1988, pp. 321-331.

[6] Tabb K., Davey N., George S. & Adams R., Detecting partial occlusion of humans using snakes and neural networks. In *Proceedings of the 5th International Conference on Engineering Applications of Neural Networks*, Warsaw, 13-15 September 1999, pp.34-39.

[7] Tabb K., Davey N., Adams R. & George S., Analysis of human motion using snakes and neural networks. In *Lecture Notes in Computer Science: Articulated Motion and Deformable Objects (LNCS 1899)*, ed. Francisco J. Perales Lopez, Springer Verlag.

[8] Williams D.J. & Shah M., A fast algorithm for active contours and curvature estimation. In *CVGIP - Image Understanding* 55, 1992, pp. 14-26.

[9] Tabb K. & George S., Snakes and their influence on visual processing. *Dept. of Computer Science Technical Report*, University of Hertfordshire, 1998.