

Computation of Vehicle Trajectories Using a Neural Network

Xu Zhang
Centre for Transport Studies
University College London
Gower Street, London WC1E 6BT, UK

Abstract

In this paper, a method will be described to calculate vehicle trajectories using a neural network for TV imagery of vehicular traffic. Constraints on the smoothness both in time and space are successfully combined in a unified form. Other constraints, for example geometrical structures of corners, can readily be included. The neural network handles inconsistent corner data in a robust manner, and the resulting trajectory plots are, visually at least, near-optimal.

1 Introduction

This paper is based on work which is being carried out, using TV imagery of vehicular traffic, to compute vehicle trajectories as part of a motion analysis system. Many methods for measuring object motion have been developed and these, in general, are based on three sources of information. The first is the similarity of motion over a small neighbourhood; the second uses the temporal continuity of motion; the third uses invariant properties of the moving object, for example image features such as lines or corners. These sources of information intrinsically define the properties of motion. By exploring these properties of motion, three main types of motion analysis methods have been developed: feature-based matching, differential methods and spatial temporal filtering. Model based tracking, recently explored by Worrall et al (1991) and Koller et al (1992), can be broadly classified into the first category. However, model-based methods are still in a relatively early stage of development. Since they require the recognition of individual vehicles, they may face difficulties when dealing with very congested traffic situations, which are very important for traffic surveillance and control and, it is in these situations that many other existing image based systems are likely to produce unreliable result. Differential methods, which use the motion constraint equation together with other constraints, typically the spatial smoothness of motion, run into problems when frame-to-frame displacements exceed more than a small number of pixels: such displacements are relatively common in traffic surveillance and control imagery. Spatio-temporal modelling considers motion in an (x,y,t) -space: motion calculation at any point then becomes the calculation of orientation of a trajectory of this point in the three dimensional space. This method faces much the same problems as the previous one when applied to traffic imagery.

In this paper, we introduce a corner-based approach which calculates vehicle movements by using a neural network to find the correct matching of corner points.

Constraints based on the smoothness both in time and space are successfully combined to solve this matching problem. Other constraints, based for example on geometrical structures of corners, can readily be included. Any feature-based method has to combat the inconsistency of features detected in real world scenes. The neural network handles inconsistent data in a robust manner, and the resulting trajectory plots are, visually at least, near-optimal.

In the following sections, we briefly review an existing trajectory-finding method in section 2. In section 3, the Hopfield neural network and a new method of weight adaptation to improve its behaviour will be briefly discussed. Section 4 introduces the architecture of the neural network for solving the trajectory finding problem. Section 5 describes the design of the energy function and section 6 discusses the weights of neuron links. Section 7 discusses the initial values for neurons and some other network parameters. Section 8 shows some results using this method. Finally, in section 9, we conclude with suggestions for further improvements.

2 Sethi and Jain's Trajectory Finding Method

In 1987, Sethi and Jain proposed a method to find feature trajectories by utilizing the continuity of motion (Sethi and Jain, 1987). The problem can be stated as follows. Given n frames taken at different time instants and m points in each frame, to establish motion correspondence is to map a point in one frame to another point in the next frame such that no two points map onto the same point. There are m^n possible combinations, each forming a trajectory: m trajectories should be found altogether. Since motion cannot change abruptly, the trajectories should be smooth and uniform. In their paper, Sethi and Jain introduced a measure of smoothness, which they called path coherence. The smoothness is affected by changes in direction and the distances moved. Rangarajan and Shah (1991) introduced a similar measure of motion smoothness, which they called the proximal uniformity cost function. By exploring all possible combinations of the points in different frames and minimizing their path coherence, m trajectories can be found. Further details can be found in the paper by Sethi and Jain.

Since the performance of this method depends on the consistency and robustness of the feature detector, Sethi, Salari and Vemuri (1988) proposed another relaxation scheme to solve the same problem. However, our experience with this new method is that it still does not give satisfactory results with our traffic scenes. In this relaxation approach, the local neighbourhood condition is taken into account in a heuristic manner, as well as the continuity condition. This encouraged us to try to find a near-optimum solution based on the use of neural networks.

3 Hopfield Neural Networks and Their Weights

We assume that the reader is familiar with the theory of the Hopfield neural network (Hopfield, 84, 85). The neuron updating rule is:

$$\frac{du_i}{dt} = -\frac{u_i}{\tau} + \sum_j W_{ij}v_j - T_i \quad (1)$$

This can be rewritten in the more familiar form ($\tau=1$):

$$u_i(t+\Delta) = u_i(t) + \Delta \frac{du_i(t)}{dt} = (1-\Delta)u_i(t) + \Delta \left(\sum_j W_{ij}v_j - T_i \right) \quad (2)$$

The particular output function we use is:

$$v_i(t) = 1 / (1 + e^{-u_i(t)/T}) \quad (3)$$

where T is called the temperature. The inverse function of (3) is

$$u_i = -T \ln\left(\frac{1}{v_i} - 1\right) \quad (4)$$

This network minimises the following so-called energy function or object function:

$$E = -\frac{1}{2} \sum_{i \neq j} W_{ij}v_i v_j + \sum_i T_i v_i \quad (5)$$

Hopfield showed how this type of neural network can be used to solve optimization and minimization problems. We use a similar approach to model the trajectory finding problem.

To use this neural network in optimization, the first step is to construct the energy function. When the energy function has been constructed then the weights can be fixed as explained below. For the energy function shown by equation (5), the weight connecting neuron i to j will be W_{ij} . Zhang and Forshaw (1993) showed that if the weights are defined by the following equation, the neural network so constructed minimizes the same energy function.

$$\bar{W}_{ij} = W_{ij} - A = W_{ij} - (1-\alpha) \sum_{s \neq i} W_{is} v_s v_i / \sum_{s \neq i} v_s v_i, \quad \bar{T}_i = T_i - B = T_i - (1-\alpha) \sum_s T_s v_s / \sum_s v_s \quad (6)$$

During neural network evolution, the parameter T is gradually reduced from a large initial value to a small final value and the α parameter is gradually increased from 0 to 1. This can be shown to improve the speed of convergence.

4 A Neural Network for Feature Trajectory Finding

To find the velocity of a vehicle, it is necessary to identify the vehicle. However, it is quite difficult to do so using image processing techniques, particularly when there is congestion. Nevertheless, such congested situations are of great interest for road traffic control. Instead of finding the velocities of individual vehicles, we compute the velocities of vehicle corners by finding their trajectories and then analyze them directly to produce descriptions of traffic flow. Corners are local features and there are usually many corners for one vehicle. With reasonably large numbers of corners, it does not matter if some corners are lost, nor does it matter if some are mismatched. By measuring the velocities of many corners, we expect to get a generally sensible

representation of traffic scenes.

As indicated above, we view the feature trajectory finding problem as an optimization problem, considering not only the property of temporal motion continuity but also spatial smoothness. We found the Hopfield network to be a suitable tool for doing this.

For a sequence of images, a corner point has two indices: the i -th corner in the k -th image. The (x,y) coordinate of the corner is denoted as X_i^k . For a pair of corners in two successive images, X_i^k and X_j^{k+1} , the vector connecting them is $\bar{X}_{ij}^k = X_j^{k+1} - X_i^k$. We constructed a neural network such that the state v_{ij}^k of a neuron represents a pair of corner points (X_i^k, X_j^{k+1}) in successive images. If $v_{ij}^k = 1$, then these two points are inferred to be corresponding points, otherwise they are not. Suppose that for images $k, k+1, k+2$, there are n_k, n_{k+1}, n_{k+2} points respectively, then there will be $n_k n_{k+1}$ neurons linking corners from image k to image $k+1$, and $n_{k+1} n_{k+2}$ neurons linking corners from image $k+1$ to $k+2$. The weight of the link connecting neuron v_{ij}^k and v_{st}^{k+1} is $W_{(ijk)(st)}$. Even though each neuron is connected to every other neuron in the Hopfield network, for this particular problem the network can be thought of as having two layers of neurons. Within each layer neurons are all connected and each neuron in one layer is connected to every neuron at the other layer. The first layer corresponds to the first set of corner pairs which have to be matched, whilst the second layer corresponds to the second set of corner pairs.

5 Design of the Energy Function

As mentioned above, the state v_{ij}^k of a neuron represents a possible match between a corner point X_i^k in the k -th image and a corner X_j^{k+1} in the next image. This pair forms a vector, which is the displacement of the corner from image k to image $k+1$. For any other neuron with state v_{st}^k , that represents a match between another pair of corners, if the distance between the two pairs, i.e. vectors, represented by the two neurons is small, then it is likely that the displacements of the two corners will be similar. To reflect this spatial continuity of motion we designed the neighbourhood support term:

$$E_1 = \frac{1}{2} \sum_{\substack{k,j,j',s,t \\ i \neq j, j' \neq t}} \frac{\|\bar{X}_{ij}^k - \bar{X}_{s't'}^k\|}{\|\bar{X}_{ij}^k\| \|\bar{X}_{s't'}^k\|} v_{ij}^k v_{s't'}^k \quad (7)$$

where

$$\|\bar{X}_{ij}^k - \bar{X}_{s't'}^k\| = \text{minimum} \{ \|X_i^k - X_s^k\|, \|X_j^k - X_t^k\| \}$$

From this, it can be seen that, given a certain number of active neurons, then, if two neurons fire, to minimize the energy, the active neurons will tend to support similar displacements or velocities, with the degree of support decreasing as the distance between them becomes larger.

A second term E_2 favours non-clashing trajectories. If two neurons sharing one corner fire, the energy will increase instead of decreasing:

$$E_2 = \frac{1}{2} \sum_{\substack{k,j,s,t \\ i=s/j=t}} V_{ij}^k V_{st}^k \quad (8)$$

For a neuron v_{ij}^k representing a possible match between a corner point X_i^k in the k -th image and a corner X_j^{k+1} in the next image, $k+1$, and another neuron v_{st}^{k+1} , which represents a match between another pair of corners from image $k+1$ to $k+2$, then if $s=j$, the second trajectory represented by the second neuron is a continuation of the first trajectory represented by the first neuron. These two neurons share the same corner in the image $k+1$. If these two trajectories form one continuous line, then we may expect that the two vectors do not change very much in terms of both magnitude and direction, since they depict the movement of the same corner at two successive instants of time. The term E_3 describes this type of temporal smoothness of motion.

$$E_3 = \frac{1}{2} \sum_{\substack{k,j,s,t \\ j=s}} \|\bar{X}_j^k - \bar{X}_s^{k+1}\| V_{ij}^k V_{st}^{k+1} \quad (9)$$

The E_4 term in equation (10) states that if the trajectories of the two firing neurons are joined, then the energy will decrease. This term favours unbroken trajectories.

$$E_4 = -\frac{1}{2} \sum_{\substack{k,j,s,t \\ j=s}} V_{ij}^k V_{st}^{k+1} \quad (10)$$

These four terms E_1 , E_2 , E_3 , and E_4 together express the characteristics of both spatial and temporal smoothness of motion. When they are put together with suitably chosen weighting factors α_i as: $\alpha_1 E_1 + \alpha_2 E_2 + \alpha_3 E_3 + \alpha_4 E_4$, then the two smoothness characteristics are successfully combined.

To force the network to produce approximately the expected number of trajectories, an extra term E_5 is required:

$$E_5 = \frac{1}{2} \sum_k (\sum_{ij} V_{ij}^k - N)^2 \quad (11)$$

where N is the expected number of trajectories. This term can be expanded as

$$\begin{aligned} \frac{1}{2} \sum_k (\sum_{ij} V_{ij}^k - N)^2 &= \frac{1}{2} \sum_k \left(\sum_{\substack{ij,s,t \\ i=s/j=t}} V_{ij}^k V_{st}^k + \sum_{ij} (V_{ij}^k)^2 - 2N \sum_{ij} V_{ij}^k + N^2 \right) \\ &\approx \frac{1}{2} \left[\sum_{\substack{k,j,s,t \\ i=s/j=t}} V_{ij}^k V_{st}^k - (2N-1) \sum_{ij} V_{ij}^k + N^2 \right] \end{aligned} \quad (12)$$

when V_{ij}^k approaches 0 or 1, $(V_{ij}^k)^2$ approximates to 0 or 1 respectively because $(V_{ij}^k)^2$ approximately equals V_{ij}^k .

Rangarajan and Shah showed in their paper (1991) that it is necessary to apply more weight to shorter trajectories. E_6 in equation (13) supports this argument:

$$E_6 = \sum_{i,j,k} \|\bar{X}_y^k\| V_y^k \tag{13}$$

These energy terms, suitably weighted, compose the final energy function:

$$E = \alpha_1 E_1 + \alpha_2 E_2 + \alpha_3 E_3 + \alpha_4 E_4 + \alpha_5 E_5 + \alpha_6 E_6 \tag{14}$$

6 Determining the Weight

When we inspect the energy function, we see that the energy comes from two parts, the link and the threshold. The links between the neurons can be divided to two types. One comes from the interconnection between neurons in the same layer. The other comes from the interconnection between neurons from different layers:

$$W_{(i,j,k)(s,t,l)} = W_{(i,j,k)(s,t,k)} \delta_{kl} + W_{(i,j,k)(s,t,k+1)} \delta_{l,k+1} \tag{15}$$

where

$$W_{(i,j,k)(s,t,k)} = -\alpha_1 \frac{\|\bar{X}_y^k - \bar{X}_{st}^k\|}{\|\bar{X}_{st}^k - \bar{X}_{st}^k\|} (1 - \delta_{is})(1 - \delta_{jt}) - \alpha_2 (\delta_{is} + \delta_{jt}) - \alpha_5 \tag{16}$$

$$W_{(i,j,k)(s,t,k+1)} = -\alpha_3 \|\bar{X}_y^k - \bar{X}_{st}^{k+1}\| \delta_{js} + \alpha_4 \delta_{jt} \tag{17}$$

and

$$T_{(i,j,k)} = \alpha_5 \frac{2N-1}{2} - \alpha_6 \|\bar{X}_y^k\| \tag{18}$$

Then

$$\bar{W}_{(i,j,k)(s,t,l)} = \bar{W}_{(i,j,k)(s,t,l)} \delta_{lk} + \bar{W}_{(i,j,k)(s,t,l)} \delta_{l,k+1} \tag{19}$$

$$\bar{W}_{(i,j,k)(s,t,k)} = W_{(i,j,k)(s,t,k)} - (1 - \alpha) \sum_{\substack{p,q,m,n \\ p+m/q+n}} W_{(p,q,k)(m,n,k)} V_{pq}^k V_{mn}^k / \sum_{\substack{p,q,m,n \\ p+m/q+n}} V_{pq}^k V_{mn}^k \tag{20}$$

$$\bar{W}_{(i,j,k)(s,t,k+1)} = W_{(i,j,k)(s,t,k+1)} - (1 - \alpha) \sum_{p,q,m,n} W_{(p,q,k)(m,n,k+1)} V_{pq}^k V_{mn}^{k+1} / \sum_{p,q,m,n} V_{pq}^k V_{mn}^{k+1} \tag{21}$$

$$\bar{T}_{(i,j,k)} = T_{(i,j,k)} - (1 - \alpha) \sum_{p,q} T_{(p,q,k)} V_{pq}^k / \sum_{p,q} V_{pq}^k \tag{22}$$

For each cycle, the updating rule is

$$u_y^k(t + \Delta) = (1 - \Delta) u_y^k(t) + \Delta \left(\sum_{\substack{l,s,t \\ l \neq k / l \neq s / j \neq t}} \bar{W}_{(y,k)(st)} V_{st}^l - \bar{T}_y^k \right) \tag{23}$$

7 Initial Value Setting

As the neuron statuses are iteratively adjusted, we want the status of each neuron to change smoothly and to be independent of the updating order (the neurons are sequentially updated). In order to achieve this, we have to limit the change in value of each neuron status. From equation (3), the derivative of v with respect to t can be obtained by:

$$\frac{dv}{dt} = \frac{dv}{du} \frac{du}{dt} = \frac{e^{-uT}}{T(1+e^{-uT})^2} \frac{du}{dt} = \frac{1}{T}(1-v)v \frac{du}{dt} \quad (24)$$

du/dt can be calculated, through equation (1), ($\tau=1$), and (4), by

$$\frac{du_i}{dt} = T \ln \left(\frac{1-v_i}{v_i} \right) + \sum_j W_{ij} v_j - T_i \quad (25)$$

given some initial values for the v 's. Then, we have

$$\frac{dv_i}{dt} / v_i = (1-v_i) \ln \left(\frac{1-v_i}{v_i} \right) + \frac{1}{T} (1-v_i) (\sum_j W_{ij} v_j - T_i) \quad (26)$$

Let $|dv/dt| \leq v/10$ (for example). We then have

$$-(1-v_i) \ln \left(\frac{1-v_i}{v_i} \right) - \frac{1}{10} \leq \frac{1}{T} (1-v_i) (\sum_j W_{ij} v_j - T_i) \leq -(1-v_i) \ln \left(\frac{1-v_i}{v_i} \right) + \frac{1}{10} \quad (27)$$

To avoid initial discrimination against any one trajectory, we chose to set all v 's equal to 0.5 and, therefore, u 's equal to 0. Then the inequality (29) becomes:

$$-\frac{1}{10} \leq \frac{1}{T} (1-v_i) (\sum_j W_{ij} v_j - T_i) \leq \frac{1}{10} \quad (28)$$

T should satisfy the inequality (29):

$$T \geq 10(1-v_i) |\sum_j W_{ij} v_j - T_i| \quad (29)$$

As mentioned above, the neuron weights we use are adjusted over time. They are expressed by equation (6). Replacing W_{ij} by \bar{W}_{ij} and T_i by \bar{T}_i , and letting α equal zero, we have

$$T \geq 10(1-v_i) |\sum_j \bar{W}_{ij} v_j - \bar{T}_i| = 5 \left| \sum_j (W_{ij} - A) - T_i + B \right| \quad (30)$$

Let

$$T_i = 5 \left| \sum_j (W_{ij} - A) - T_i + B \right| \quad (31)$$

The temperature T is then selected as the largest of the T_i . Although, these equations appear to be complicated, the interested reader will find that they are quite straightforward in practice.

Assuming that we expect the number of trajectories to be N , and the number of neurons for one pair of images to be n . Another commonly used way to set the initial v values for each neuron is to set them to N/n with a small perturbation. This number will be very small. By equation (4), $-u_i$ will be quite large. From equation (1) and (26), we can see that it is then impossible to control the value of neuron status change.

α is determined in our system by:

$$\alpha = e^{-10 \frac{E(t) - E(t+1)}{E(0)}}$$

where we set $E(-1) = 2 * E(0)$. As E , the instantaneous energy, decreases monotonically and stabilizes, α changes from a very small value to 1. $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6$ are set to 100, 150, 1, 180, 20, 0.05 respectively. Δ is set to 0.05, and the temperature $T(t+1) = 9 * T(t) / 8$. These values were determined empirically.

8 Experimental Results

Figure 1-1 shows three sets of corners overlaid together detected from an isolated vehicle. The empty squares represent corner points in the first image in time; the solid squares represents corners in the second image; the cross, the third. It can be seen that the corners detected are quite inconsistent. However, it is a common problem for many corner detectors. Figure 1-2 shows the results that the neural network is applied to match these corners. Figure 2-1 shows a whole image of a traffic scene. The result of applying the network is shown in figure 2-2. The corners are detected by a moving corner detector (Zhang, 1993).

It can be noted that even though any individual matching can be wrong, when given a reasonable number of corners that are from the same object, the overall match gives a meaningful result. Thus the inconsistency in corner detection can to some extent be successfully ignored.

9 Conclusion and Further Remarks

In this paper, we have described a method using a neural network to find corner feature trajectories. Some promising results are shown. The neural network is of a Hopfield type, but a new way to adaptively modify the neuron weights is used. This helps in improving the network convergence.

One problem is that it is difficult to find a systematic way to determine the relative weights α_i for each separate energy term, and the speed of calculation still needs to be improved. We are currently trying to implement this algorithm in a transputer

array. There are also many other possible ways to improve the results. For example, the network could be set up differently. Instead of assigning a neuron to a pair of corners to be matched, one might assign a neuron to a triplet of corners. Feature structures, such as brightness and orientation, have not yet been included in the network implementation. They could be included by adding new energy terms. However, the use of more terms creates extra difficulty in apportioning the relative weights among the different components, and if they are higher-order constraints they would also give a dramatic increase in computing time. However, a possible useful addition would be to use multi-scale optimisation scheme(Xu et al, 1988, 1991; Mjolsness et al 1991).

10 Acknowledgement

This research is supported by SERC GR/H/14564. Thanks are given to my supervisors Dr. M. R. B. Forshaw and Professor Richard E. Allsop for their help and encouragement. Thanks are also due to the reviewers for their helpful comments and suggestions.

11 References

- Hopfield, J. J. (1984) Neurons with Graded Response Have Collective Computational Properties like Those of Two-State Neurons, *Proc. Natl. Acad. Sci. USA*, Vol. 81, 3088-3092.
- Hopfield, J. J. and D. W. Tank (1985) "Neural" Computation of Decisions in Optimization Problems, *Biol. Cybern.*, Vol. 52, 141-152.
- Koller, D., K. Daniilidis, T. Thórhallsson and H. -H. Nagel (1992) Model-Based Object Tracking in Traffic Scenes, *Proceedings Second European Conference on Computer Vision, Italy*, May, 1992.
- Mjolsness, E, C. D. Garrett, and W. L. Miranker (1991) Multiscale Optimization in Neural Nets, *IEEE Transactions on Neural Networks*, Vol.2, No. 2.
- Rangarajan, K. and M. Shah (1991) Establishing Motion Correspondence, *CVGIP:Image Understanding*, Vol.54, No. 1, 56-73.
- Sethi, I. K., V. Salari and S. Vemuri (1988) Feature Point Matching in Image Sequences, *Pattern Recognition Letters* 7, 113-121.
- Sethi, I. K. and R. Jain (1987) Finding Trajectories of Feature Points in a Monocular Image Sequence, *IEEE Trans. PAMI* 9, 56-73.
- Worrall, A. D., G. D. Sullivan and K. D. Baker (1991) Model-based Tracking, *Proc. British Machine Vision Conference*.
- Xu, X., W. T. Tsai and N. K. Huang (1988) A Generalized Neural Network Model, *Abstracts of the First Annual INNS Meeting*, 1 (Supplement 1), 150.
- Xu, X. and W. T. Tsai (1991) Effective Neural Algorithms for the Travelling Salesman Problem, Vol.4, 193-205.
- Zhang X. and M. R. B. Forshaw (1993) Optimization Using a Modified Hopfield Neural Network, to appear in the *First European Congress on Fuzzy and Intelligent Technologies*, Aachen, Germany, September 7-10, 1993.
- Zhang X. (1993) A Moving Corner Detector for Dynamic Images, in preparation.

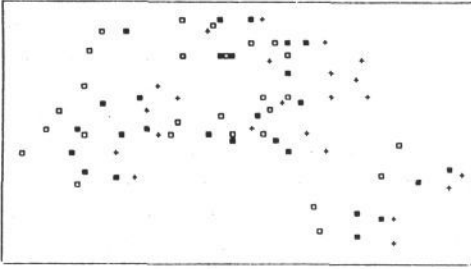


Figure 1-1 Corner data from an isolated car in three sequential real images

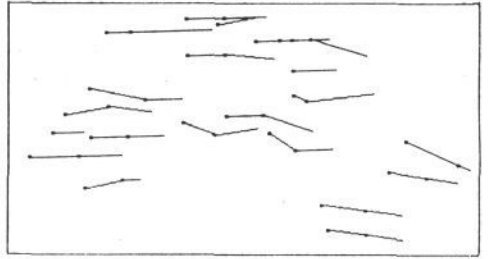


Figure 1-2 Results of matching

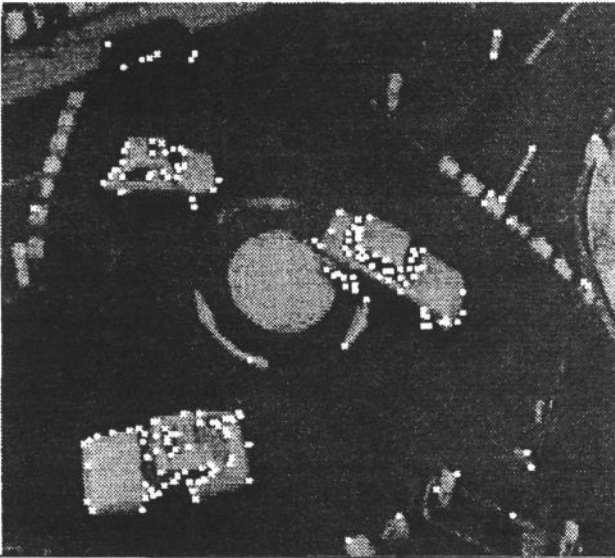


Figure 2-1 Image overlaid with one set of corners



Figure 2-2 Results of matching