TEXTURE SEGMENTATION FOR DEFINING DRIVEABLE REGIONS

A Grunes and J F Sherlock

Royal Signals and Radar Establishment
St Andrews Road
Malvern, Worcs WR14 3PS UK

Segmentation of forward looking images of suburban roads using some simple texture measures is described. Driveable regions are labelled using a Markov random field model. High resolution is achieved by a double pass method.

Considerable interest is being shown in developing the capability to drive automated vehicles through road systems and more complex environments typical of industrial installations. Recent work has tended to concentrate on the techniques necessary to process, in real time, TV images obtained from cameras mounted on such a vehicle. Suitable processing of the images provides an input to, principally, the heading control system. The approach has the advantage of flexibility and maybe cheapness when compared with systems using markers, painted lines or buried cables.

Because of the potentially large amounts of data which would need to be processed in real time, the reported work has proposed solutions to the relatively simple problem of driving along well defined, uncluttered roads [1, 2]. For several reasons, it is not evident that the methods which have been described would be applicable to complex or ill defined road systems and open terrain.

In the work described in this note, an attempt has been made to segment scenes on the basis of texture. Texture features are not necessarily easy or quick to calculate. However they do have the advantage of allowing simple, closed regions to be extracted. The procedure is essentially simple. A set of features is defined and statistically significant numbers of these are measured for the types of texture that it is required to identify. Feature values for a sample of the unknown texture are then compared with those of the reference set. A label is attached to the sample according to the closeness of the two sets of features. The labelling process should take into account the probability that several textures may produce similar statistics and that the matching may not be exact. No more than three features have been used in the current work, a Markov process being used to produce final, labelled images of suburban and country scenes obtained from a forward looking camera.

THE HIDDEN MARKOV CHAIN MODEL

The hidden Markov model represents a process by a sequence of states which are probabilistically related. In the model used in this work the transition probability is determined only by an observable random variable which is related to the state by a probability function.

The model is described by two matrices[3]. The first, $A = a[i,j];\ i=1,..,n,\ j=1,..,n,$ contains the position independent state-to-state transition probabilities where, denoted by t the length of the sequence and by $q(x) = q<i>$ that at position x in the chain the process is in state $q<i>$,

$$a[i,j] = P(q(x+1) = q<j> \mid q(x) = q<i>).$$

The second matrix, $B = b[j,k];\ j=1,..,n,\ k=1,..,m,$ relates the probabilities of observing certain outcomes given each of the states where, denoting by $o(x) = o<k>$ that at position x the observation has the value $o<k>$,

$$b[j,k] = p(o(x) = o<k> \mid q(x) = q<j>).$$

A texture label is assigned to a region of the image according to the rule that $q(x) = q<i>$ where

$$i = \text{argmax}\ \{P(q(x) = q<i> \mid O)\},\ O=o(1),..,o(t).$$

As a result of the simplifying assumptions which have been made, this problem is equivalent to

$$i = \text{argmax}\{a(x,i)b(x,i)\},$$

where

$$a(x,i) = P(o(1)...o(x) \cap q(x) = q<i> \mid A,B),$$
$$b(x,i) = P(o(x+1)...o(t) \mid q(x) = q<i> \cap A,B.$$

The probabilities 'a' and 'b' are often referred to as the 'forward' and 'backward' probabilities.

For low resolution analysis, the image is segmented into 30 rows of 32 non-overlapping windows each containing 16x16 pixels. Each window is scaled to a common mean brightness. Using, as

BMVC 1990 doi:10.5244/C.4.42

features, the number of local extrema, which is a count made of local maxima and minima having grey-level differences above a set of threshold, and the variance of the grey-level distribution, statistics are collected for each window. Earlier work also included an edge element feature, which used a count of the number of edgels above a set threshold generated by a 3x3 Sobel operator. This feature has since been removed as it was found to add little to the final segmentation. The vector of statistics for each outcome is then used to model both a 32-state horizontal sequence and a 30-state vertical sequence using the rows and columns of windows. Four labels are used, foliage and brick defining regions to be avoided, whilst road and grass labels represent potentially deriveable regions.

The first window of a row (column) may be any state with equal probability, so there is no need for a vector of initial state probabilities.

The A matrix elements can be chosen in order to avoid rapid state changes. The greater the diagonal elements the less likely are sudden changes. The A matrix is a 4x4 array and is estimated beforehand. It is possible to build into A various rules to make, for example, the grass-to-grass transition probability low, but the road-to-road probability high. However, in this work a symmetric matrix has been used, generally with high valued diagonal elements.

The B matrix gives the probability of the outcome conditional on each state. Since a window can be put into one of four classes for the extrema feature and into one of five for the variance feature, there are 20 possible class combinations for each state and the B matrix is a 4x20 array. It is assumed that the statistics are independent.

After having formulated the A and B matrices, it is possible to evaluate the 'forward' and 'backward' probabilities. They are computed recursively as follows:

$$a(0,i) = i/n \quad i=1,.,n; \quad b(t,j) = 1 \quad j=1,.,n,$$

$$a(x+1,j) = \{\Sigma_i a(x,i)A(i,j)\}B(j,o(x+1)) \quad x=0,\dots,t-1 \quad j=1,.,n,$$

$$b(x,i) = \Sigma_j A(j,i)B(j,o(x+1))b(x+1,j) \quad x=t-1,\dots,1 \quad i=1.,.n.$$

A window is then given a label which is associated with the most likely state, $q(i)$, to have resulted in the observation, O.

THE UNDERFLOW PROBLEM

The implementation of the forward-backward technique is marred by underflow problems on finite word length computers for all but the most trivial applications, due to the use of recurrence. However, converting the computation of joint likelihoods into the computation of a posteriori probabilities results in essentially the same forward-backward algorithm except for the presence of a scaling factor. Devijver[3] suggests that both $a(x,i)$ and $b(x,i)$ are multiplied by a scaling coefficient, $N(x)$, where

$$\begin{aligned} N(x) &= [\Sigma_j A(1,j)B(j,o(x))]^{-1} && x=1, \\ &= [\Sigma_i \Sigma_j \tilde{a}(x-1,i)A(i,j)B(j,o(x))]^{-1} && x=2,.,t. \end{aligned}$$

The forward probabilities are then computed inductively by the recurrence

$$\begin{aligned} \tilde{a}(x,j) &= N(x)A(1,j)B(j,o(1)) && x=1 \\ &= N(x)\Sigma_i \tilde{a}(x-1,i)A(i,j)B(j,o(x)) && x=2,\dots,t, \end{aligned}$$

and the backward recurrence by

$$\begin{aligned} \tilde{b}(x,j) &= 1 && x=t, \\ &= N(x+1)\Sigma_k A(j,k)B(k,o(x+1))\tilde{b}(x+1,k) && x=t-1,\dots,1. \end{aligned}$$

However, using the forward recurrence as stated above does not lead to an efficient implementation. A more efficient though equivalent method is as follows:

$$\begin{aligned} G(x,j) &= \Sigma_i \tilde{a}(x-1,i)A(i,j) && x=2,\dots,t, \\ a'(x,j) &= A(1,j)B(j,o(1)) && x=1, \\ &= G(x,j)B(j,o(x)) && x=2,\dots,t, \\ N(x) &= [\Sigma_j a'(x,j)]^{-1} && x=1,\dots,t, \\ \tilde{a}(x,j) &= N(x)a'(x,j) && x=1,\dots,t. \end{aligned}$$

With this formulation, $G(x,j)$ is the a priori probability of the process being in state j at time x, given only the sequence of previous observations.

2D LABELLING TECHNIQUES

The Markov property can be extended to two dimensions for the case of regular arrays of pixels. In a system described by such a model, the probability of the label 'h' at position (m,n) can be expressed as

$$P(h(m,n)) = P\left\{ h(m,m) \;\middle|\; \begin{matrix} h(m-1,n-1) & h(m-1,m) & h(m-1,n+1) \\ h(m,n-1) & & h(m,n+1) \\ h(m+1,n-1) & h(m+1,n) & h(m+1,n+1) \end{matrix} \right\}$$

This specifies that the label attached to any point or region of the image is determined only by the labels assigned to the eight nearest neighbours. In practice, this is complicated to calculate and in the present work, a hidden Pickard random field image model has been used, as a simplification.

It capitalises on the observation that rows (columns) in each set of k consecutive columns (rows) of the random field form a k-dimensional vector Markov chain. It is also assumed that the dependence of h(m,n) upon any o(k,l) which is not a nearest neighbour to the left or above is zero.

By then applying the conditional independence property to the equation i = argmax{a(x,i)b(x,i)}, the resulting factors have the same probabilistic structure as G(x,j)and $\tilde{b}$(x,j) above. Therefore they can be computed using one-dimensional recurrences along the mth row and nth column respectively.

Denoting by mGn(q)the values of Galong row n for each state q, nGm(q)the values of Galong column n for each state q, and similarly for the b's, the labelling criterion then becomes

i = argmax{Lm,n(q)},

where

$$Lm,n(q) \propto mGn(q)*nGm(q)*B(q,o(m,n)) \\ *mbn(q)*nbm(q).$$

The application of this approach requires four sweeps over the image to calculate the G's and $\tilde{b}$'s. In the course of the last sweep the labelling criterion can be computed.

RESULTS

A programme implementing the approach described above has been written in 'FORTRAN 77' to run under 'UNIX'. Sequences of up to 250 frames of video have been captured by a forward-looking camera mounted on the roof of the RSRE autonomous vehicle demonstrator. These have been processed using various values of the transition matrix 'A'.

Figures 1-3 show some results which are typical of those obtained, The original image has been chosen to illustrate a reasonable variety of textures. The resolution of the processed images is to 4x4 pixel patches. This has been done by a two level analysis. At the first level, segmentation is done using 16x16 pixel non-overlapping patches. At positions where there is an indicated change of texture, feature statistics are re-calculated using 16x16 pixel patches overlapped on 4 pixel centres. The Markov process is then used again to label the 16 pixels at the centre of each large patch. The processed image, figure 2, has been generated with high value diagonal elements in the A-matrix. It shows the road ahead of the vehicle clearly defined while also demarcating the regions of grass and foliage. The segmentation of brickwork is less reliable. This could probably be improved by including a measure of both edgel density and orientation. Based on this and similar results, it appears that useful segmentation of forward looking images can be obtained with a very small texture feature set measured over a few ranges of values.

The usefulness of the Markov model is illustrated by comparing figure 2 with figure 3 which shows the same section of road processed using 'A' matrix elements of equal value. This approximates to processing without a Markov model as all transitions are equally probable. It is clear that the image is more fragmented than when the model is biassed against rapid texture transitions.

Though the Markov model produces less fragmented images it is not evident that such images are closer to reality than those produced without it. However, to dispense with the Markov model itself implies that a model has been used in which all transitions are equally probable and all observations are unaffected by noise. Evidently, this is unrealistic though it is not generally easy to decide exactly what the final image should look like given the initial image as opposed to the real world. A good rule of thumb may be to assess the results on the basis of what one would expect to see within each region. This implies that the 'A' matrix values are chosen to give results which are intuitively correct.

The results which have been obtained take no account of the texture gradient though this has not proved to be significant problem. Roads are represented by the smoothest texture values. As a result, roads continue to be correctly labelled into the far distance. Grass-like regions do, however, tend to be labelled as road in the distance. Such a result is not contrary to common experience. To a viewer, all surfaces will tend to look alike in the distance as the texture differences become less marked. The problem can be overcome by having different reference sets which are derived from either foreground or background texture samples. An independent range measurement could then be used to make the decision when to change from one set to the other. Alternatively, the problem can be avoided by increasing the look-down angle of the camera. For moderate speeds up to, for example, 20 mph quite large look-down angles can be used while still providing safe look ahead distances.

CONCLUSIONS

The results which have been obtained suggest that roads and other driveable regions can be reliably defined by using only two simple texture measures. The use of a Markov model allows fragmented regions to be coalesced into intuitively better representations of a scene in a consistent manner.

REFERENCES

1. **Dickmanns, E. D. and Zapp, A.** "Guiding Land Vehicles Along Roadways by Computer Vision" *Proc. AFCET, Congress Automatique,* (1985) Toulouse.

2. **Morgan, A D., Dagless, E. L., Milford, D. J. and Thomas, B. T.** "Road Edge Tracking for Robot Road Following" *Proc Alvey Vision Conference,* Manchester, (1988) pp 179-184.

3 **Devijver, P. A. and Dekesel, M. M.,** "Learning the Parameters of a Hidden Markov Random Field Image Model : A Simple Example" *Pattern Recognition, Theory and Applications* (1987) pp 141-163.

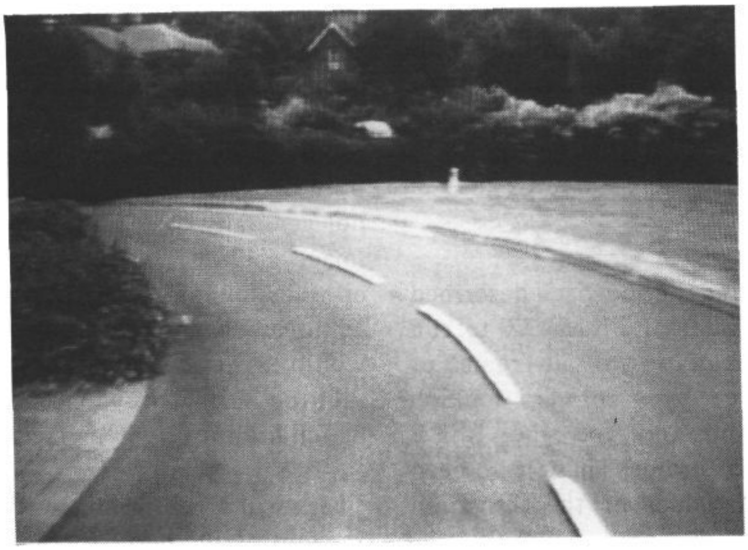

Figure 1. Original grey scale image

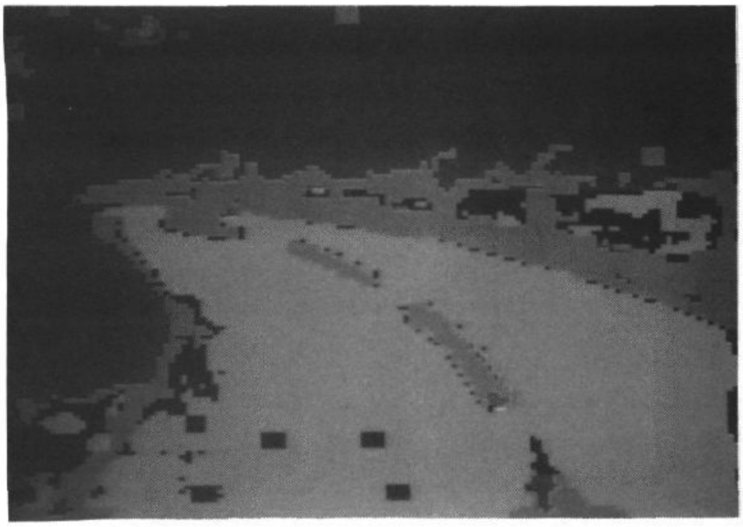

Figure 2. Texture based segmentation

$$A = \begin{vmatrix} 0.55 & 0.15 & 0.15 & 0.15 \\ 0.15 & 0.55 & 0.15 & 0.15 \\ 0.15 & 0.15 & 0.55 & 0.15 \\ 0.15 & 0.15 & 0.15 & 0.55 \end{vmatrix}$$

Key

Yellow = Road-like
Green = Grass-like
Blue = Foliage-like
Red = Brick-like

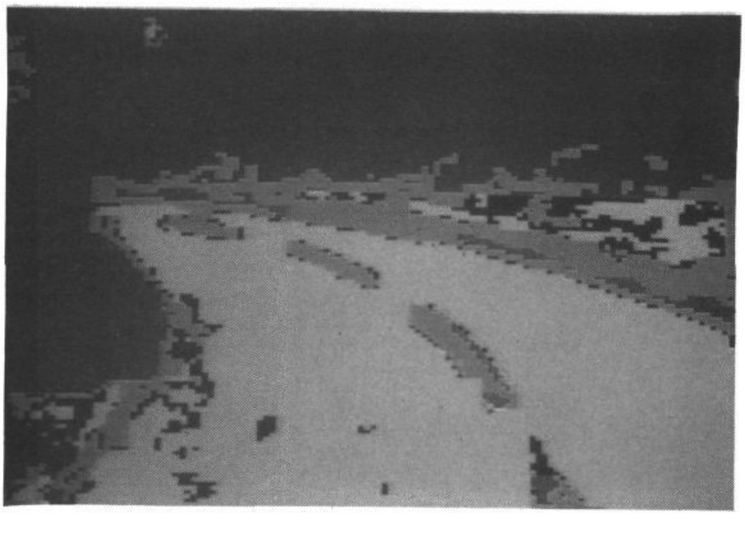

Figure 3. Segmentation with equal transition probabilities

$$A = \begin{vmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{vmatrix}$$