An Unsupervised Clustering Approach to Location Classification

Mathew Price and Gerhard de Jager

Department of Electrical Engineering
University of Cape Town
Rondebosch 7701, South Africa
mathew@dip.ee.uct.ac.za

Abstract

Conventionally, tracking people through an environment has been achieved by monitoring a series of fixed cameras. With the advent of wireless technologies, the option of inverting the paradigm and monitoring instead, from each person's point-of-view, has become more accessible. By taking video sequences from a person moving through various environments, this paper explores a process for classifying the different locations encountered using chromatic information gathered from images. This involves extracting a set of simple features from each frame, applying an unsupervised clustering algorithm and classifying new images with a *nearest neighbour* method.

1. Introduction

The general paradigm for tracking people has been by observing their movements from fixed-position cameras. As the people move through different areas of the environment, the system can be switched to the most appropriate camera view. However, in certain scenarios, this method can be uneconomical and awkward.

An example could be monitoring the positions of 20 to 30 people in a large, automated industrial plant. Depending on the size of the plant, the video network could require anything up to 200 cameras in order to cover all relevant locations. An alternative system could be instead, to equip each person with an on-body wireless camera and use the images from those cameras to detect their general location in the environment. This system is obviously not a total replacement for most CCTV systems, however there are numerous advantages in employing mobile cameras for monitoring:

- The problem of multi-view tracking is reduced, since a continuous feed is available from the camera and no view switching is necessary.
- Although the exact positioning is not as consistent as ActiveBadges (radio positioning tags), the video information can provide detailed information of each person's activities and is not dependent on the person being occluded by others in a cluttered environment (where CCTV systems sometimes fail).
- As wireless technology progresses, a network of mobile cameras can also be more economical (no cables/video switches and less cameras).

This paper addresses the problem of taking the video feed from a moving person and training a data set to recognise distinctive areas so that the general location can be classified. The process followed involves collecting features from recorded image sequences, analysing and visualising clusters using Self-Organising Maps and implementing a clustering algorithm which then passes a cluster map to the classifier.

2. Pre-Processing and Feature Extraction

Since a major requirement of the system is to provide up-to-date information, the classification process needs to run as close to real-time as possible. In order to facilitate this process, a very simple feature set was created using averaged Hue and Saturation values, extracted from masked areas of the input frame (similar to the method used in [2]). A rectangular-block mask was chosen since the grid size can be easily changed thus allowing control over the length of the feature vector (see figure 1).

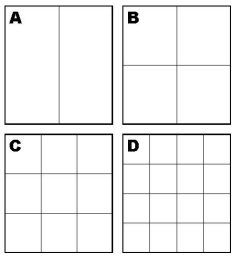

Figure 1: *Various feature extraction masks (produces 2-,4-,9- or 16-dimension features).*

Feature extraction proceeds firstly by pre-processing each image. This involves converting the image from RGB to HSV colour space. The Hue and Saturation channels provide a useful measure of chromaticity in the image (found to provide good class separation), while being more robust against lighting variations than the RGB model [1]. A full analysis of all possible features was not conducted as, the emphasis was placed on investigating whether clusters did exist and how to approach the classification problem. Fine-tuning of the actual feature composition was left for future work.

Following the colour-space transform, a median filter is applied for reduction of noisy pixel values. Finally, the mask is applied and values of mean Hue (H_m) and Saturation (S_m) are calculated for each masked area. The final feature vector $p(i)$ is a $2n$-dimension row vector where n is the number of blocks in the mask of the i_{th} frame.

$$p(i) = [H_{m_1}\ S_{m_1}\ H_{m_2}\ S_{m_2}\ \ldots\ H_{m_n}\ S_{m_n}] \qquad (1)$$

For the training phase, the output of the feature extractor is a matrix of i feature vectors which can then be applied to the clustering algorithm.

For development purposes, a simplified set of features consisting of only H_m values for a 2 x 1 mask was used. This limited the feature vectors to a 2-D set and allowed initial two-dimensional visualisation during testing of the clustering algorithm. Once the operation of the algorithm was verified, experimentation continued with higher dimension features.

3. Clustering Algorithm

Since initial observations of the 2-D features showed that the features of similar images tended to form small regions, a clustering approach was favoured.

A general problem with many clustering methods is that they require a user to specify the number of patterns or classes. This is not a problem if this is a known fact, e.g. sorting red apples from green (2 patterns). Unfortunately, this is not practical when the number of patterns is harder to specify, e.g. splitting an environment into separate locations for tracking. Naturally in the latter case, we would like the network to discern the most distinguishable locations and determine however many separate patterns exist.

Another issue, was the problem of visualising clusters past a 2-D feature space. One solution to both these problems, was to use a Kohonen Network. After initial tests, though, the Kohonen network was solely used as a visualisation tool, while a less computationally-intense clustering algorithm was implemented.

3.1. Kohonen Networks

A Kohonen Network (or Self-Organising Map), is a two-layer network where the input layer is interconnected to the output layer (like a conventional network), however, the output layer (competitive layer) is also structured to form a two-dimensional grid (see figure 2). During training, each output node is moved so as to be closer to an input vector. In addition, neighbouring output nodes are also moved towards each other. This has the effect of quantitising the input vectors, by folding the grid of neurons around the presented data. Eventually, the output grid becomes an ordered map with similar prototypes close together. Effectively, the network's weights are trained while at the same time, the topological information is preserved.

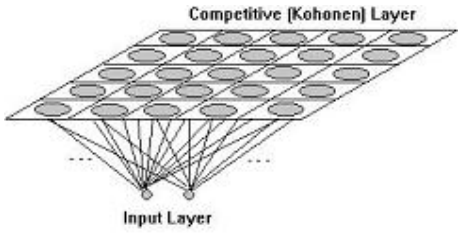

Figure 2: *Kohonen Network Structure.*

3.2. Training

Initially, a Supervised Learning scheme was applied in which each location was defined by a labelled training set. This information was then integrated during training in order to maximise the exclusion of erroneous noisy samples. Although the performance of the system was fair, not all location data could always be consistently separated based on the labels. For this reason, a new implementation, governed by an Unsupervised Learning method was developed. This allowed the training process to decide which locations were best for classification, based on the Euclidean distance between each cluster.

The basic training algorithm is as follows:

- A training set is applied in the form described previously (Feature Extraction phase).
- As with training the probability weights of a PNN, a hyper-surface is created by summing small Gaussian kernels to each training sample. The result is a surface having peaks where the kernels overlap thus forming a cluster. If $d_{(i,j)}$ is the Euclidean distance between two features, then $q(i)$, below, is an estimate of the probability of the i_{th} feature falling into a distinguishable class (i.e. the higher and sharper the peak, the more unique the cluster).

$$q(i) = e^{-d^2_{(i,1)}/\sigma} + e^{-d^2_{(i,2)}/\sigma} + \ldots + e^{-d^2_{(i,n)}/\sigma} \quad (2)$$

Figure 3 below shows an example surface generated for a training sequence with 2-D features. Here, two distinct clusters dominate, while some smaller regions also exist.

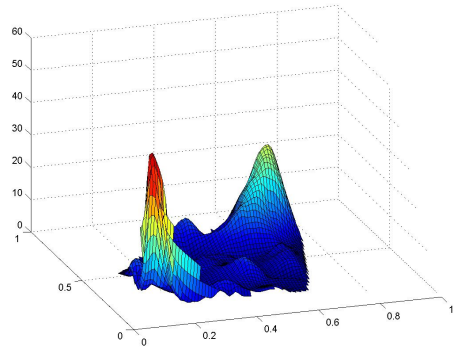

Figure 3: *Summed Gaussian kernel map.*

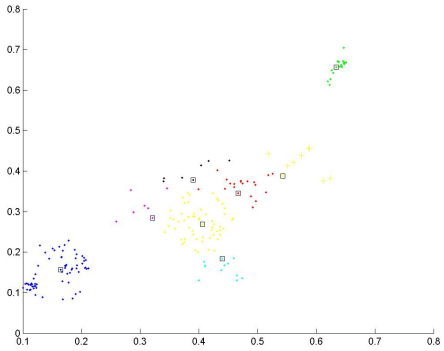

Figure 4: *Example of a trained cluster map (2 features).*

- The list of weighted points is then sorted in order of probability $q(i)$ and the highest value is selected as a starting point. Neighbouring points are grouped into the cluster until a distance threshold is reached. At this

point, the cluster splits and a new highest point is elected to be the centre of the next cluster. The process repeats until the list of points is depleted. Points that are not allocated due to distance thresholding or being part of an excessively small cluster, are discarded. Figure 4 shows an example of a generated cluster map.

3.3. Classification

Conventionally, the classification process is based on a desire to provide a clean-cut 'yes' or 'no' (class 'A' or 'B') answer. However, with the location classification system, since similar-looking images could sometimes be clustered together, a different approach was needed. Since the input provides a large amount of redundant data recurring at a high rate (many similar frames), the necessity of classifying each frame is reduced. In fact, most of the time, the primary goal is to detect a location change and only if possible refine the sub-location. Thus the classifier is designed to ignore any frames whose features are not extremely close to a cluster centre. In this case a no-class (unclassified) result is returned. This is achieved by relying on good shaping and filtering from the clustering algorithm during the training phase and performing the actual classification using the nearest neighbour method.

4. Results and Discussion

Preliminary tests of the system revealed some interesting facts, however, more extensive experiments (with more diverse data) and fine-tuning are needed to fully quantify its exact limitations.

The tests conducted on the current implementation, consisted of 3 different location data sets: indoor-house; indoor-office and outdoor-garden. Each of these sets are composed of a training and a testing video sequence and are formatted as follows: 24-bit colour, 176 x 144 pixel, 15 fps.

A setting of σ = 0.1 x 10^{-4} was used for the Gaussian kernel size. Experimentation with this value allowed control over the amount of classes generated by each training sequence. The KNN Classifier was set to use 3, 5 and 10 neighbours located close to the cluster centre. In practice, this had little effect since the clusters were quite compact in most cases (requirement of the system) and therefore even a setting of 1 neighbour seemed to provide adequate classification.

Since the implementation is geared towards an Unsupervised Learning system and hence labelled image sets are not available, measurement of the performance of the classification process is awkward. Instead, human observation was used as a means to compare whether test frames were indeed correctly classified by each class. This was accomplished by comparing a list of trained class prototypes with the classified frames for each cluster. Naturally, it is not feasible to show the matches of each class for each test sequence, therefore, the demonstration figures only show a few examples.

Figure 5, shows the cluster map generated for the indoor-house data set, using 2 features. Each coloured set of dots or crosses denotes a trained cluster (class). A black square marks the detected centre of each cluster (based on the peaks of the summed Gaussian kernel map), while the black stars show a plot of the classified test points. An interesting revelation from the cluster map is that separation of the major location types is achieved with only 2 features. In fact, after further observations, it was concluded that most environments could be separated into a general location index in this manner. For further separation of each location into smaller sub-locations, more features are required.

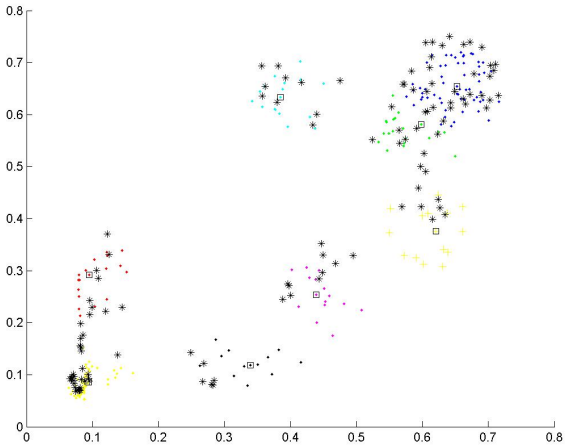

Figure 5: *Cluster map for indoor-home data set overlaying classified test points.*

Figure 6 shows a prototype image set with example images associated with each class (extracted from the marked centre point). In this case, 8 classes were detected, however some of these classes actually overlap. It is suggested that future implementations merge overlapping clusters into one class for neatness, however this is not a primary concern as it does not really affect the matching process.

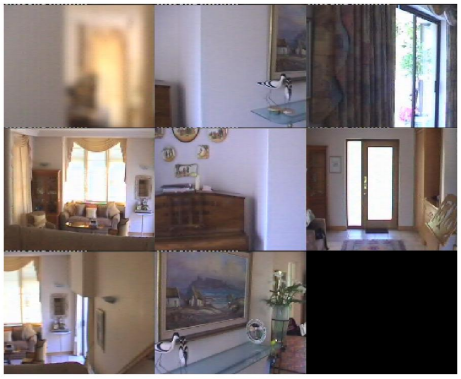

Figure 6: *Example class prototypes for the indoor-home data set. Class 1 is the top-left image and the sequence follows a left-to-right, top-to-bottom order.*

Finally, figures 7,8 and 9 are the set of classified frames associated with classes 4,5 and 6 (middle 3 images from figure 6) respectively. The montages show some inaccuracies with class 5, however, the mostly the classification for positioning purposes is quite accurate. One important factor which was found to affect performance quite extensively was the camera's Automatic Gain Control. During transitions between different rooms, the camera attempted to compensate for the lighting differences. This adjustment can cause a sequence of frames to appear tinted by an unnatural colour and therefore cause an off-

set between the trained cluster and future test frames. For this reason it is recommended that the AGC of the camera be disabled (if possible).

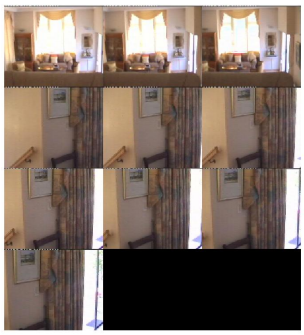

Figure 7: *Classified indoor-home test frames for class 4.*

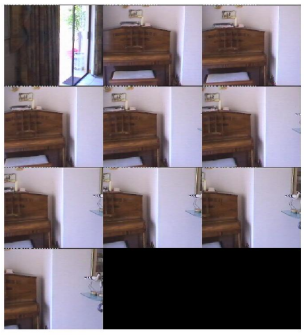

Figure 8: *Classified indoor-home test frames for class 5.*

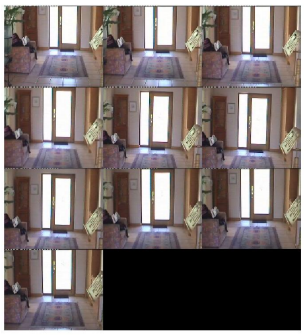

Figure 9: *Classified indoor-home test frames for class 6.*

Some examples of tests conducted on the other two data sets are shown in figures 10, 11 (indoor-office), 12 and 13 (outdoor-garden). Figures 10 and 12 are montages of the trained images associated with an arbitrary cluster from each data set, while figures 11 and 13 are the set of test images classified as belonging to those classes. Since the office and outdoor images were more similar, complete separation with just 2 features was not possible. Therefore, both these sequences were trained with a 2 x 2 mask - totalling 8 features (Hue and Saturation channels). As is seen, the matching process was fairly successful.

Visualisation of the clusters formed by training the outdoor scene is provided by the SOM map in figure 14. The dark blue areas show regions of closeness between the data, while the brighter colours (red and yellow) show areas where the data is more sparsely located, therefore outlining cluster borders. Thirteen clusters were detected, however only about 6 unique clusters exist. As previously stated, merging of similar classes would improve the compactness and thus the accuracy of the detected cluster value.

5. Conclusions

Using simple measures of pixel chromaticity as features, it is possible to extract information about the location of a camera in an environment. This was applied to a location-classification scheme for tracking the movements of a person equipped with a wearable camera. The extracted information does form clusters in the feature space and matching of test sequences to trained cluster maps was accomplished.

The use of an RGB to HSV transform allows greater tolerance in the system against lighting variances (nomalised RGB components are also a viable options), however large rotations and jerky movements of the camera were found to cause instability during classification.

Classification using a simple *nearest neighbour* system while using a highly selective training procedure provides better separation of pattern clusters. Ensuring that the borders are maximised and that the clusters are as compact as possible simplifies classification and ensures faster execution.

Care should be used when using feature masks with too many divisions. As shown, classes of images are separable without the use of high dimension features. In fact, increasing the mask size past a 4 x 4 grid significantly reduces the tolerance of the matching process.

Self-Organising Maps are highly useful for visualising clusters in data with high dimensions and can also be used for classification in systems where the number of patterns is variable.

While this system was suggested for the application of monitoring a person's view and therefore position (unobstructed) in a large complex environment, a range of other possibilities exist for exploration (e.g. personal visual locator).

6. Acknowledgements

Thanks to DebTech, a division of De Beers Consolidated Mines Ltd., for support of ongoing research.

7. References

[1] Aoki H., Schiele B. and Pentland A., "Realtime personal positioning system for wearable computers.", Technical

report, MIT Media Laboratory, 1999.

[2] Clarkson B. and Pentland A., "Unsupervised clustering of ambulatory audio and video.", Technical Report 471, MIT Media Laboratory, Perceptual Computing Group, 1998.

[3] Dudo R. O. and Hart P.E., "Pattern Classification and Scene Analysis", Wiley Interscience, 1973.

[4] Vesanto J., Himberg J., Alhoniemi E. and Parhankangas J., "SOM Toolbox for Matlab 5.", Technical report, Helsinki University of Technology, April 2000.

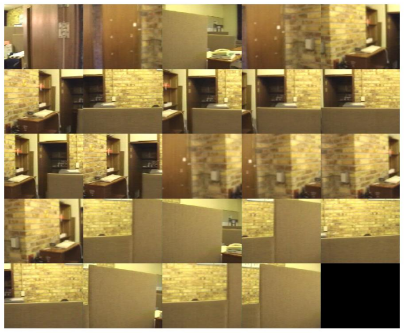

Figure 10: *Example from indoor-office data set. Class 1 trained cluster.*

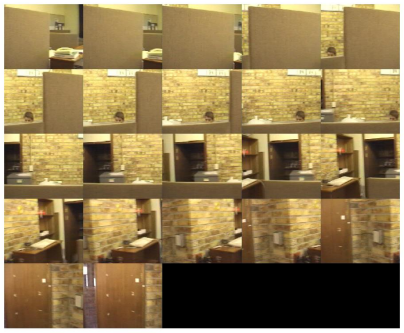

Figure 11: *Matched frames for Class 1 (indoor-office) f set.*

Figure 12: *Example from outdoor-garden data set. Class 3 trained cluster.*

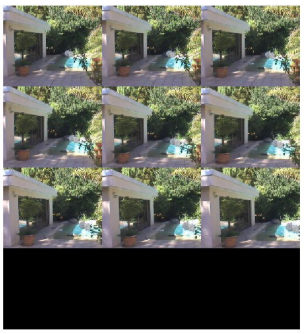

Figure 13: *Matched frames for Class 3 (outdoor-garden) from test set.*

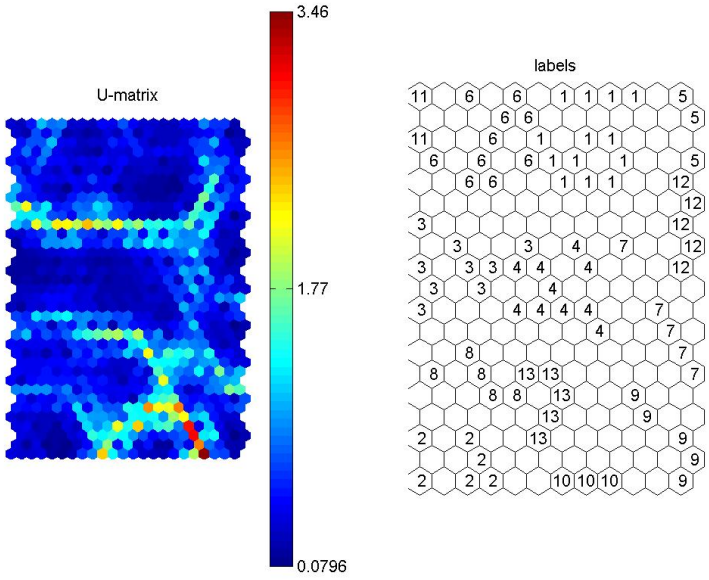

Figure 14: *SOM map generated for outdoor-garden data set. Dark blue areas show compact clusters while yellow and red areas are cluster boundaries.*