Neural Comput & Applic (1999)8:347–367
© 1999 Springer-Verlag London Limited

Neural
Computing
& Applications

Basic Competitive Neural Networks as Adaptive Mechanisms for Non-Stationary Colour Quantisation

A. I. Gonzalez[1], M. Graña[1] and M. Cottrell[2]

[1]Department of CCIA, Universidad Pais Vasco/EHU, San Sebastián, Spain; [2]Institut SAMOS/MATISSE, Université Paris 1, France

In this paper we consider the application of two basic Competitive Neural Networks (CNN) to the adaptive computation of colour representatives on image sequences that show non-stationary distributions of pixel colours. The tested algorithms are the Simple Competitive Learning (SCL) algorithm and the Frequency-Sensitive Competitive Learning (FSCL) algorithm. Both, SCL and FCSL are the simplest adaptive methods based, respectively, on minimising the distortion and on the search for a uniform quantisation. The aim of this paper is to study several computational properties of these methods when applied to non-stationary clustering as adaptive vector quantisation algorithms. Non-stationary colour quantisation is, therefore, representative of the more general class of non-stationary clustering problems. We expect our results to be meaningful for other algorithms that involve either the minimisation of the distortion or the search for uniform quantisers. We study experimentally the effect of the size of the image sample employed in the one-pass adaptation, their robustness to initial conditions, and the effect of local versus global scheduling of the learning rate.

Keywords: Colour quantisation; Frequency-sensitive competitive learning; Non-stationary clustering; Simple competitive learning

1. Introduction

The process of colour quantisation is the codification of a colour image into a finite and small set of colour representatives. Colour quantisation design is the problem of searching for a set of optimal colour representatives in a colour image. The optimality of such a set can be measured perceptually by displaying the inverse of the image codification. It can also be measured quantitatively by computing the distortion of the codification or the Signal-to-Noise Ratio (SNR). In the present paper, we present as experimental results both qualitative and numerical results (distortion). From a numerical point of view, the search for the optimal colour representatives can be put into the general framework of clustering based on representatives or vector quantisation design [1–6]. Colour quantisation is then a vector quantisation defined in the colour space, usually based on the Euclidean distance in this space. One relevant question is that of the colour space and the colour distance employed. It is well known that the Euclidean distance in the RGB space does not preserve the perceptual distance between colours. There are several abstract colour spaces [7], such as the Yuv and the Lab spaces, defined by the CIE in order to obtain better preservation of the perceptual distance, and new colour spaces are being defined to cope with the colour equivalencies needed to support colour processing in complex distributed environments (i.e. the Internet). It can be argued that performing clustering based on minimising the Euclidean Distortion in the RGB colour space is doomed to give perceptually suboptimal results. However, there is a growing body of evidence [7–11] showing that this perceptual suboptimality is of no consequence for most practical applications. We assume this framework to support our experiments in the RGB space.

When an image sequence is considered, the distribution of the pixel colours in the colour space will

Correspondence and offprint requests to: A.I. Gonzalez, Dept. CC1A, Universidad Pais Vasco/EHU, Aptdo 649, 20080 San Sebastián, Spain. E-mail: ccpgrrom@si.ehu.es

be time variant in the general case. The underlying stochastic process is, therefore, non-stationary, and we can't assume any model of the time dependencies. The problem of colour quantisation on image sequences becomes a non-stationary clustering problem. In most of the formulations of the clustering or vector quantisation problems found in the literature, the assumption is that the underlying stochastic process is stationary, and that a given set of sample vectors properly characterises this process. The main line of research that takes into account non-stationary processes is that of Adaptive Vector Quantisation (AVQ) [1], where the dominant approach is that of codebook replenishment [12,13]. We introduce in Section 4 a general formulation of the non-stationary clustering problem, and the application of Competitive Neural Networks as adaptive vector quantisation methods to solve it.

Competitive Neural Networks [14–18] are mathematically derived as stochastic gradient minimisation procedures. Sometimes the objective function is known, as is the case of the Simple Competitive Learning that minimises the Euclidean distortion. Sometimes it is difficult to specify the objective function, and to provide a formal derivation of the learning rule. This is the case of Frequency-Sensitive Competitive Learning, which combines the distortion minimisation of the SCL with the search for a uniform quantisation. Uniform quantisation implies that the probability distribution of the codevectors is uniform (not to be confused with a uniform decomposition of the colour space). To obtain an uniform quantisation, FSCL tries to ensure that the sizes of the clusters found in the sample are equal. However, it does that by penalising the Euclidean distance while computing the nearest codevector to a sample vector. This can't be easily put into the formal framework of stochastic gradient algorithms, but it constitutes a minimal variation of SCL intended to improve its robustness (to avoid empty clusters associated with stuck codevectors). In this paper, we consider both SCL and FSCL as minimal adaptive algorithms whose results can be extrapolated to more sophisticated strategies.

Despite their original definition as stochastic gradient minimisation methods, CNNs have rarely been applied to adaptive vector quantisation, because of their lengthy convergence times and numerical sensitivities [1]. Most work in the literature reports their application to stationary VQ problems, such as the codification of still images. As we have said before, the most successful adaptive vector quantisation strategy is that of codebook replenishment [1, 12,15,19,20] whose optimality, however, has only been proved for stationary sources [21]. Codebook replenishment algorithms also pose serious parameter tuning problems that have not been properly addressed in the literature. In this paper, we try to show the usefulness of the CNN as adaptive vector quantisation algorithms for general non-stationary sources. The case of non-stationary colour quantisation is thus a representative of the general AVQ problem. Our approach is to propose a fast adaptation schedule, based on a one-pass adaptation over a small sample of each image in the sequence. Under this scheme, the computational cost of the adaptation is proportional to the size of the sample, the size of the codebook and the dimension of the search space, and can be calibrated for real time processing. In Section 5, we explore the sensitivity of the SCL and FSCL in this setting to the local/global scheduling of the learning rate, the sample size, the codebook size and the global initial conditions (the starting codebook for the whole sequence). The results reported here will be also of interest when trying to assess the applicability of other CNN architectures to the non-stationary clustering problem. We have found that the sensitivity to the codebook size is shared by the self-organising map of Kohonen, neural gas, soft competition, and other CNNs [22–24].

Section 2 discusses the end applications for non-stationary colour quantisation; Section 3 poses the colour quantisation of general image sequences as a non-stationary clustering problem and the application of CNN to solve it. Section 4 reviews both SCL and FSCL training algorithms and their specific numerical settings. Section 5 presents experimental results. Finally, in Section 6, we present our conclusions and intended lines of work.

2. End Applications of Colour Quantisation

Colour quantisation has applications in visualisation [8,9,25,26], colour image segmentation [18], data compression [19,27] and image retrieval [28]. Early applications of colour quantisation were addressed to visualisation tasks [8,25]. The problem was to render colour images for display in low colour resolution monitors. This can be interest for games that require fast visualisation or that involve network communication. Nowadays, monitors and visualisation devices do not require this reduction of the colour space.

A recent application for CQ is the content-based retrieval of information in multimedia databases that include colour images [28–31]. Usually, the colour space is partitioned at regular intervals and the colour histogram of the image is used as the feature for the search. Colour representatives are sometimes used to index the images in the database [28]. These colour representatives are computed using clustering-based techniques. There are also instances [32] that use the codebooks obtained from adaptive vector quantisation for the search in image databases, where the input vectors are subimage blocks.

One of the recent applications of colour processing is the segmentation of video sequences [32,33]. The variations in the colour histograms are used to identify the units (shots) in the decomposition of the sequence. These units are then used for fast access into the sequence, or to construct an index for the organisation of video databases. The non-stationary clustering approach could be of use for this task, providing that colour representatives are used instead of colour histograms. This is a stationary application because past images can't be quantised at each database increment. However, given a suitable initial sample of the images to be stored in the database, the techniques discussed in this paper can be applied.

The last class of applications of CQ address the segmentation of images. This segmentation is of interest for sensitive interfaces: the colour detection of faces, hands and other human features can be very effective [11,34]. The usual approach is the *a priori* identification in the colour space of the region that corresponds to face colours. This approach is very restricted to the images taken to estimate the face colour region. The adaptive techniques discussed in the paper can be of interest to allow robust detection in the case of illumination changes and other sources of noise. Sensitive interfaces are of increasing interest for personal computers and for control applications, such as enhancing the human interface to robots. The optical flow is a very central issue in many computer vision applications, including robot navigation. The image segmentation obtained applying non-stationary CQ to the preprocessing of image sequences can be used for the robust computation of the optical flow [35], and to other vision tasks, such as stereo matching [36].

3. Non-Stationary Colour Quantisation as a Non-Stationary Clustering Problem

Given a data sample $\aleph = \{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$, with $\mathbf{x}_i \in \Re^d$, the clustering problem is that of finding a partition of the sample $P(\aleph) = \{\aleph_1,\ldots,\aleph_c\}$ into c disjoint clusters, that is optimal in the sense specified by a clustering criterion function C. This partition can be defined in several ways. One of the most widely used definitions is that induced by the cluster representatives: data samples are assigned to the cluster of the nearest Euclidean representative, and the representatives coincide with the cluster centroids. The criterion function is defined over the scattering measures of the clusters. In this case, the clustering problem is equivalent to the design of a vector quantiser based on the sample: the search for a set of representatives $\mathbf{Y} = \{\mathbf{y}_1,\ldots,\mathbf{y}_c\}$ that minimise an error or distortion function E, that is the minimum intracluster scattering. Both the clustering criterion function and the quantisation distortion are based on a given dissimilarity measure.

Colour quantisation is a mapping of a multispectral image $f(x,y) = [f_R(x,y), f_G(x,y), f_B(x,y)] \in [0,1]^3$ into an indexed image $f^c(x,y) \in \{1,\ldots,c\}$, where c is the number of colour representatives, which we will denote $\mathbf{Y} = \{\mathbf{y}_1,\ldots,\mathbf{y}_c\}$ with $\mathbf{y}_i \in [0,1]^3$. The visualisation of the colour quantised image is done through the inversion of the quantisation $\hat{f}(x,y) = \mathbf{y} \Leftrightarrow f^c(x,y) = i$. Visual comparison with the original image gives the perceptual evaluation of the colour quantisation. The numerical evaluation of the quality of the colour quantisation can be done computing the distortion $E = \sum_{x,y} \|\hat{f}(x,y) - f(x,y)\|^2$. From this description of the colour quantisation process, it is obvious that it belongs to the class of clustering and VQ problems, the sample is given by all or some of the image pixels, the vectors are defined in a 3D space, and the criterion function corresponds to the quantisation distortion.

The design of colour quantizers can, therefore, use any of the tools developed for clustering and VQ, including Competitive Neural Networks. The Heckbert algorithm [8] is a clustering algorithm that performs a greedy search for the partition of the image colours. In its original formulation, it performs a recursive splitting of the RGB space based on the histograms of the projections on the colour axis. The axis and the splitting point are selected according to the variance of the histogram and its median. An improvement that ensures minimisation of the distortion (although not globally optimal because of its greedy nature) is to consider the variances of the partitions [37] of the cube. This version of the algorithm will be referred to from now on as the minimum variance Heckbert algorithm. It gives near optimal results, but its complexity is proportional to the dimension of the space and the discretisation of the space axes. Most practical implementations reduce the number of values in each colour axis from 2^8 to 2^5 by a direct truncation, before applying the Heckbert algorithm.

Non-stationary clustering assumes as the source of the data a non-stationary stochastic process $\{\mathbf{X}_t, t = 0,1,..\}$ sampled at selected time instants. Assuming that the stochastic process is a discrete time process (which is appropriate for image sequences), sampling is understood as extracting a sample at each time instant. This gives a sequence of samples, each of them a set of i.i.d. random vectors, at each $t\{\aleph(t) = \{\mathbf{x}_1(t),\ldots,\mathbf{x}_n(t)\}, t = 0,1,\ldots\}$. The non-stationary clustering problem is to find a corresponding sequence of partitions. Each partition is a set of disjoint clusters $P(\aleph(t)) = \{\aleph_1(t),\ldots,\aleph_c(t)\}$ defined over the sample at time t. The sequence of partitions must minimise an infinite time horizon criterion function $C = \sum_{t\geq 0} C(t)$. This minimisation problem can't be solved through techniques related to dynamic programming, because of its infinite time horizon and the lack of any recursive formulation of the cost function.

Given an image sequence $\{f_t(x,y);\ t = 0,1,2,\ldots\}$, the colour quantisation of this sequence is obviously a non-stationary clustering problem, in which the searched partitions are the colour quantisations of the images in the sequence $\{f_t^c(x,y); t = 0,1,2,\ldots\}$ and the infinite horizon criterion function is the accumulative colour quantisation distortion

$$E = \sum_{t\geq 0} E(t) = \sum_{t\geq 0}\sum_{x,y} \|\hat{f}_t(x,y) - f_t(x,y)\|^2 \tag{1}$$

The colour quantisation of the image sequence applies a sequence of colour palettes $\mathbf{Y}(t) = \{\mathbf{y}_1(t),\ldots,\mathbf{y}_c(t)\}$, and it becomes non-stationary colour quantisation when no predictive scheme can be applied in the computation of the colour palettes. The search for these colour representatives varying in time becomes an adaptive vector quantisation problem.

In a rather general statement, the non-stationary clustering problem is thus an infinite time stochastic dynamic programming problem

$$\min_{\{\mathbf{Y}(t)\}} \sum_{t\geq 0} E(t) \tag{2}$$

which is very difficult to solve. To alleviate its complexity, the adaptive assumption states that the problem can be decomposed in time, so the global minimisation can be achieved through instantaneous minimisation:

$$\min_{\{\mathbf{Y}(t)\}} \sum_{t\geq 0} E(t) = \sum_{t\geq 0} \min_{\{\mathbf{Y}(t)\}} E(t) \tag{3}$$

The straightforward approach, under this assumption, is to perform independent optimisations at each time instant. Independent optimisation would then give an optimal result if the optimisation method is globally optimal. We have applied this approach as the benchmark optimal results in our experiments applying the minimum variance Heckbert algorithm to each image independently. However, time independent optimisation discards any use of time dependencies that could improve the accuracy and lower the computational burden. Truly adaptive algorithms must profit from these unknown time dependencies, without trying to uncover them. The adaptive approach assumes that the cluster or colour representatives found at time t–1 can be used as good initial conditions for the optimisation process at time t. That implies the assumption of smooth variation of the optimal cluster representatives. Summarising, the application of competitive neural networks as adaptive vector quantisation algorithms is done as follows: At time t the initial cluster (colour) representatives are those computed from the sample of the process at time t–1. The sample vectors at the present time are presented sequentially as inputs to compute the adaptation equations, and to obtain a new set of cluster representatives. Obviously, for the non-stationary colour quantisation case, the time axis is the image number in the sequence, and the sample data are the image pixel colours. To approach real time performance, we impose a one-pass adaptation at each time step, and small samples. This means that the sample vectors will be presented only once, and that the scheduling of the learning rate and other learning control parameters are adjusted to that time constraint.

4. Basic Competitive Neural Networks

In this section we review the definition of the basic CNN, and discuss the setting of the numerical parameters used when we apply them to the non-stationary colour quantisation problem. In general, CNN algorithms are derived as adaptive algorithms that perform stochastic gradient descent on an energy function [16,17] that corresponds to a clustering criterion function or a vector quantisation distortion function. They look for cluster representatives or codevectors, under the assumption of fixed codebook size. When the energy function is known, the learning rule associated with it can be formally derived. This the case of the Simple Competitive Learning (SCL), in which the minimised function is the Euclidean distortion of the quantisation of a sample $\aleph = \{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$ with a set of codevectors $\mathbf{Y} = \{\mathbf{y}_1,\ldots,\mathbf{y}_c\}$:

$$\min_{\{\mathbf{Y}\}} \quad E = \min_{\{\mathbf{Y}\}} \sum_{j=1}^{n} \sum_{i=1}^{c} \|\mathbf{x}_j - \mathbf{y}_i\|^2 \delta_{ij} \tag{4}$$

$$\delta_{ij} = 1 \ \textit{if} \ i = \operatorname*{argmin}_{k=1,\ldots,c} \left\{ \left\| \mathbf{x}_j - \mathbf{y}_k \right\|^2 \right\}$$

The stochastic gradient is given by

$$\left. \frac{\partial E}{\partial \mathbf{y}_i} \right|_x = -\frac{1}{2}(\mathbf{x} - \mathbf{y}_i), \ i = 1,\ldots,c \tag{5}$$

Then, the Simple Competitive Learning (SCL) is stated as

$$\mathbf{y}_i(\tau+1) = \mathbf{y}_i(\tau) + \alpha(\tau)\delta_i(\mathbf{x}(\tau))[\mathbf{x}(\tau) - \mathbf{y}_i(\tau)], \ i=1,\ldots,c \tag{6}$$

$$\delta_i(\mathbf{x}(\tau)) = \begin{cases} 1 & i = \operatorname*{argmin}_{k=1,\ldots,c} \left\{ \left\| \mathbf{x}(\tau) - \mathbf{y}_k(\tau) \right\|^2 \right\} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where $\mathbf{x}(\tau)$ is randomly chosen in $\aleph$; τ is the adaptation time that counts the number of input vectors presented to the learning rule; $\alpha(\tau)$ is the learning rate which, to guarantee theoretical convergence, must fulfil the conditions [16,24]

$$\lim_{t \longrightarrow \infty} \alpha(\tau) = 0, \quad \sum_{t=0}^{\infty} \alpha(\tau) = \infty \tag{8}$$

$$\text{and} \quad \sum_{t=0}^{\infty} \alpha^2(\tau) < \infty$$

However, these theoretical conditions imply very slow adaptation processes, which in practice are seldom respected. In fact, the sequence of learning rate values proposed below to meet our 'one pass' adaptation constraint overlook the above conditions.

The SCL is therefore a local minimisation algorithm, whose results will be highly dependent on the initial conditions $\{\mathbf{y}_1(0),\ldots,\mathbf{y}_c(0)\}$. One of the more salient features of suboptimal solutions is the occurrence of stuck codevectors, i.e. codevectors whose corresponding cluster is empty. These codevectors never win the competition, and Eq. (6) never applies to them. This situation must not be confused with the search for the natural number of clusters; suboptimal solutions found by SCL must not be taken as indicative of the 'true' number of clusters. The reasoning behind the proposition of frequency sensitive competitive learning is that one sure strategy to avoid bad local suboptimal solutions is to ensure that no representative has an associated empty cluster. This is done in an inverse way, by penalising the bigger clusters. To achieve that, the FSCL [15] keeps a count of how frequently each codevector is the winner, and uses this information to adjust distances from an input to all codevectors. The distance used to determine the codevector to be updated is $d(\mathbf{x},\mathbf{y}_i) = \tau_i\|\mathbf{x}-\mathbf{y}_i\|^2$, where $\tau_i = \sum_{k=1}^{\tau} \delta_i^{FSCL}(\mathbf{x}(\mathrm{k}))$ is the number of times that a codevector has been the winner, and

$$\delta_i^{FSCL}(\mathbf{x}(\tau)) = \begin{cases} 1 & i = \operatorname*{argmin}_{k=1,\ldots,c} \left\{ \tau_i \left\| \mathbf{x}(\tau) - \mathbf{y}_k(\tau) \right\|^2 \right\} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

This new distance penalises the codevector that repeatedly wins, increasing its distance value and giving other codevectors a chance to win the competition. FSCL employs the learning rule (6), like the SCL, but applying (9) instead of (7) to determine the neuron (codevector) to be updated.

The above discussion implies that FSCL cannot be derived from Eq. (4), as its stochastic gradient minimisation. In fact, we haven't found a formal definition of the true function minimised by FSCL. This function may be intuitively seen as a minimisation of the Euclidean distortion conditioned to the uniform quantisation of the space. That is, FSCL tries to ensure that in the limit after training $P[\delta_i(\mathbf{x})=1] \approx \frac{1}{c}$. Once near uniform quantisation is achieved, FSCL will try to minimise the Euclidean distortion. The expected side effect is that the minimisation of the Euclidean distortion will become globally optimal. We consider FSCL as a basic CNN because to-date it is the minimal variation of SCL proposed.

In the experiments we have applied two kinds of scheduling of the learning rate: local and global scheduling. The local scheduling of the learning rate follows the expression $\alpha_i(\tau) = \alpha_0(1 - \tau_i/n)$, where $\tau_i = \sum_{k=1}^{\tau} \delta_i(\mathbf{x}(k))$ and n is the subsample size. This expression implies that the learning rate decreases linearly in the number of times that a codevector 'wins' the competition. It also implies that a local learning rate only reaches the zero value if the codevector 'wins' for all the sample vectors. This local counter is identical to that maintained by the FSCL. Note, however, that here the counter controls the adaptation gain, whereas in FSCL it determines the winning neuron. There is no interference between the two applications of the local counter. The global scheduling of the learning rate, when $\alpha_i(\tau)=\alpha(\tau)\ \forall i$, follows the expression $\alpha(\tau)=\alpha_0(1-\tau/n)$. This expression implies that the learning rate decreases with every sample data item which is presented to the neural network. Zero value is reached when the last sample is presented. While global scheduling conforms to the theoretical formulation, local scheduling of the learning rate is similar to perform as many independent and simultaneous local adaptation processes as units. Part of our work tries to determine the benefits of local scheduling experimentally. Obviously, the sequences of the learning rate parameters given by both local and global strategies do not comply with the conditions (4) imposed by the convergence of the stochastic gradient approach. However, it is the best approximation that works under a 'one-pass' adaptation constraint.

Up to this point, the CNNs are applied to stationary data, represented by a sample $\aleph$. In the case of non-stationary data, we have, according to Section 3, a sequence of data samples $\{\aleph(t)=\{\mathbf{x}_1(t),\ldots,\mathbf{x}_n(t)\},t=0,1,\ldots\}$ over which adaptation rules (6), (7) or (9) will be applied. In the non-stationary case we have two time parameters: that of the reality (t); and that of the internal adaptive computations (τ). At each real time instant, a complete adaptation process will take place. The whole process is as follows:

1. Assume an initial codebook $\mathbf{Y}(0)$, $t=0$.
2. Update the clock $t=t+1$ and take the next sample $\aleph(t)$ of size n.
3. Assume as the initial codebook the result of the adaptation at the previous time instant $\mathbf{Y}(t,0)=\mathbf{Y}(t-1,n)$.
4. Compute the sequence of adaptations of the codebook $\{\mathbf{Y}(t,\tau);\tau=1,..n\}$ applying either SCL or FSCL, with $\mathbf{x}(\tau)$ being extracted from $\aleph(t)$.
5. Resume indefinitely the process from step 2.

5. Experimental Results of Non-Stationary Colour Quantisation

In this section we report some experiments performed on image sequences extracted from video sequences, aimed at evaluating the robustness of SCL and FSCL as adaptive VQ algorithms for non-stationary colour quantisation, and their sensitivity to diverse numerical parameters. The sequence of images used for the experiment is a panning of the laboratory taken with an Apple CCD colour video-camera designed for video-conferencing. Original images have a spatial resolution of 240 × 320 pixels. We have created two sequences. In one of them, each of two consecutive images overlaps roughly 50% of the scene (*sequence1*); in the other, the overlap is of 33% (*sequence2*). Although it must be obvious, we note that our use of finite experimental image sequences does not imply that our approach is only meant for small finite sequences. The main feature of these image sequences is that the distribution of the pixel colours in the RGB space is non-stationary and unpredictable. This is illustrated in Fig. 1, where we present the visualisation of the colour pixels in the RGB unit cube, for some images in *sequence2*. Along the sequence, the cloud of points that correspond to the image when mapped into the RGB unit cube shrinks and expands in several, unpredictable, directions. As these variations are due to the movement of the camera in an unknown visual space, the formulation of any predictive model is not possible. The experiments refer to the computation of sets of colour representatives (colour palettes) of size $c=16$ and $c=256$. These sizes of colour palettes are representative of those that can arise in segmentation and compression tasks, respectively.

The benchmark algorithm used is the minimum variance Heckbert [8] algorithm, as implemented in MATLAB [37]. This algorithm has been applied to all of the images in the sequences in two ways. Figure 2 shows the distortion results of the colour quantisation of the experimental sequences to 16 and 256 colours based on both applications of the Heckbert algorithm. The results consist of the distortion per image curves, and the total distortion over the entire sequence shown in the figure legend. The curves labelled *Time Varying* are produced by applying the algorithm to each image independently; they can be denoted mathematically $E_{TV}(t)$. This corresponds to the optimal strategy for non-stationary clustering discussed in Section 3, assuming the optimality of the Heckbert algorithm. The curves labelled *Time Invariant* ($E_{TI}(t)$) come from the assumption of stationarity of the data: the colour representatives obtained for the first images are used for the colour quantisation of the remaining images in the sequences. Obviously, the distortion is greater in the *Time Invariant* application. The difference between both curves increases as the time evolution of the colour distribution departs from the initial one found in image #1 of the sequence. The gap between those curves gives an indication of the non-stationarity of the data. From our point of view, this gap defines the response space left for truly adaptive algorithms. To accept an algorithm as an adaptive solution its response could not be worse than the *Time Invariant* curve. The *Time Varying* curve defines the best response that we expect,

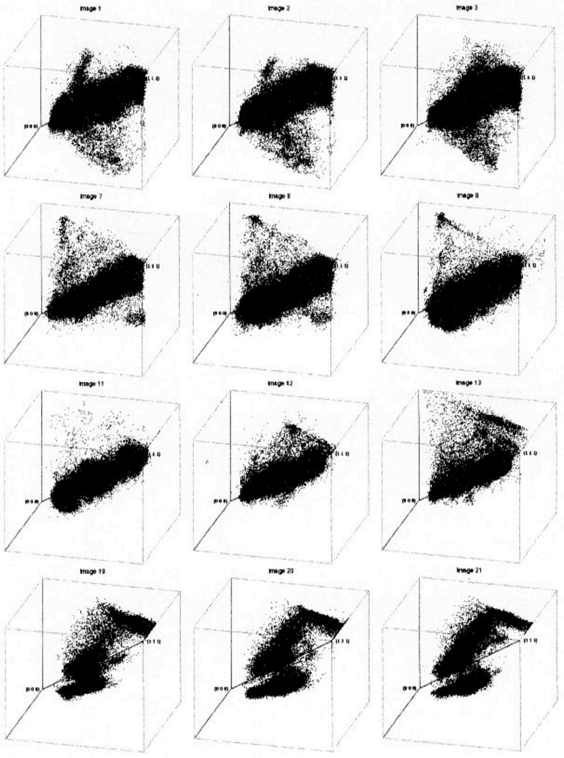
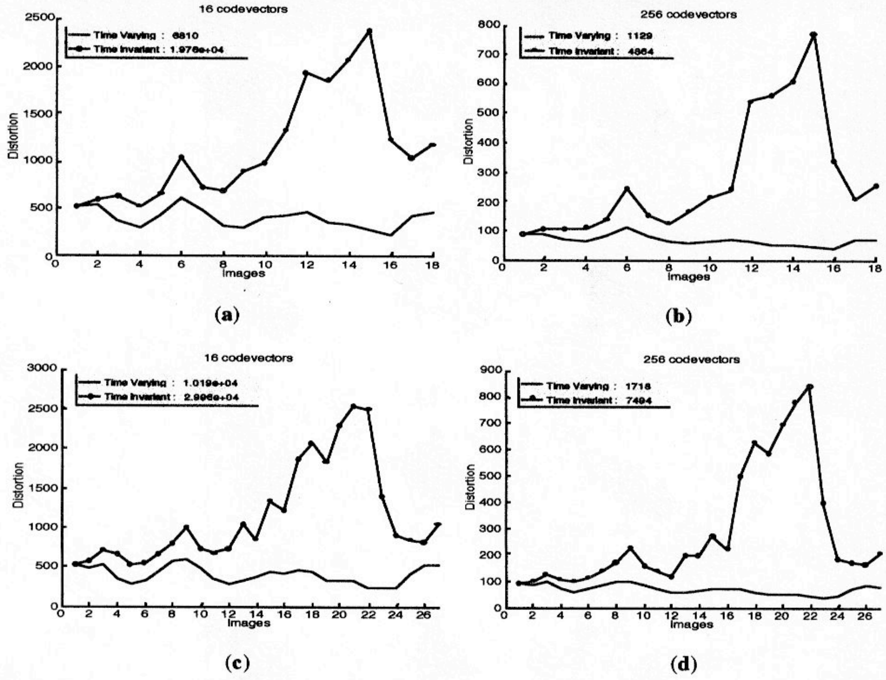


Fig. 1. The visualisation of the pixels in the RGB unit cube of image #1, #2, #3, #7, #8, #9, #11, #12, #13, #19, #20 and #21 of *sequence2*.

although it is not the sequence of global optima. In Fig. 2, it can also be appreciated that *sequence2* curves change more smoothly than *sequence1* curves. The changes in colour distribution are smoother in *sequence2*, therefore it can be expected that the results of adaptive algorithms will be better for *sequence2* than for *sequence1*. Also, it can be appreciated that the *Time Invariant* curve seems to approach the *Time Varying* curve at the end of the sequence in all cases. This behaviour is due to the nature of the image sequences; they are extracted from a closed panning of the scene, so that the final images almost coincide with the initial ones, and their colour distributions come close. This feature is by no means general; it is obviously an artifact of our experimental data. From a qualitative point of view, the closeness serves also to test the ability of the adaptive algorithms to come back to the original optimum.

In the following, the results of the application of the CNN will be given in the relative framework of the *Time Invariant* and *Time Varying* Heckbert results. The relative distortion ($E_R(t)$) shown in the figures is computed as $E_R(t) = \frac{(E_{CNN}(t)-E_{TV}(t))}{(E_{TI}(t)-E_{TV}(t))}$, where $E_{CNN}(t)$ is the per image distortion of the colour quantisation with the colour palette computed by the CNN (either SCL or FSCL). The relative distortion is negative when the CNN improves over the optimal Heckbert *Time Varying*, and it is greater than 1 when the CNN does not behave adaptively (gives results worse than the *Time Invariant*). We have appended to the legend of the curves the accumulated value of the relative distortion along the sequence.



(a)

(b)

(c)

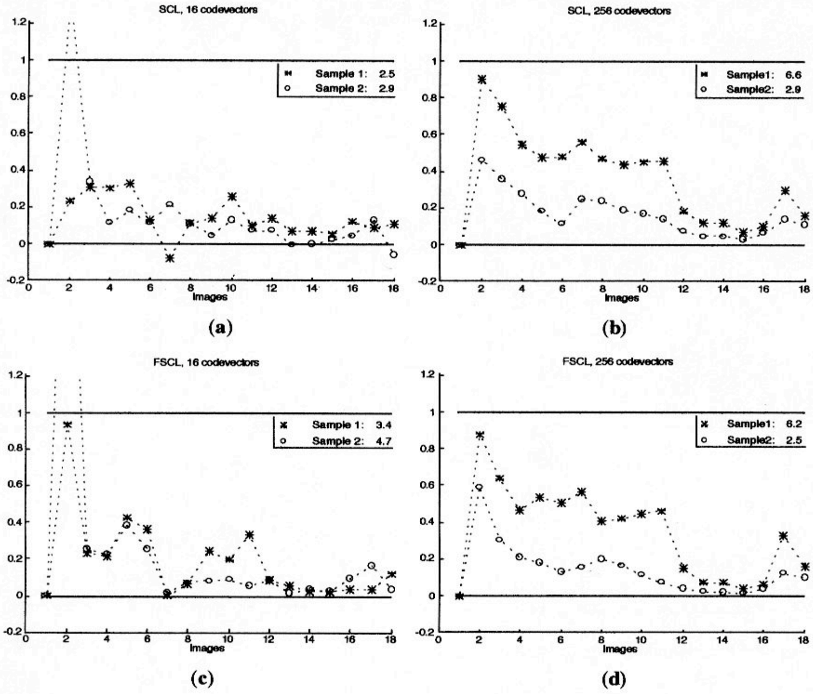
(d)

Fig. 2. Reference distortion values obtained with the application of the *Time Varying* and *Time Invariant* Heckbert algorithm for (a) 16 colours, (b) 256 colours for *sequence1*. For *sequence2* (c) 16 colours, (d) 256 colours. The amounts in the legend display the total distortion along all of the sequences.

From the description in Section 4, it is evident that the computational cost of the CNN applied to find the optimal colour quantisation of each image is of the order $O(d.n.c)$, with n being the size of the sample, $d = 3$ the dimension of the space, and c the number of colours. To evaluate the performance of the algorithm as real time restrictions are imposed, we have employed image samples of size $n = 1200$ ($Sample1 < 2\%$ of image size) and $n = 19{,}200$ ($Sample2 = 25\%$ of image size), respectively. Whereas the distortion results in Fig. 2 are produced by the colour quantisation to a set of colour representatives computed using all the image pixels, both SCL and FSCL will be applied to a subset of the image pixels, extracted randomly. The sequence of image samples *Sample1* represents a stronger real time constraint than *Sample2*, and the application of both SCL and FSCL will be one order of magnitude faster for *Sample1*. The distortion results shown in the figures and tables are the distortion of the entire images when colour quantised with the colour representatives computed by the CNN upon the specified image samples. Besides the real time considerations, the use of image samples also gives some hints about the robustness and extrapolation abilities of both SCL and FSCL.

5.1. Sensitivity to Codebook and Sample Size

The first set of experiments are performed assuming as the initial codebook the colour palette obtained by the Heckbert algorithm for the first image. This is the best initial condition that we can think of to start the adaptation of the remaining sequence (note that all the curves start at zero). The results of this set of experiments are shown in Figs 3 and 4. The experiment includes the computation by SCL and FSCL of the colour representatives under all combinations of image sequence, sample size, colour pal-



(a)

(b)

(c)

(d)

Fig. 3. Relative distortion results for *sequence1* with local learning rates applying SCL to (a) $c = 16$ and (b) $c = 256$, and applying FSCL to (c) $c = 16$ and (d) $c = 256$.

ette size, and local versus global scheduling of the learning rate. In all cases, the codification into 256 colours gives worse relative results. This result is quite important because it indicates the sensitivity of the CNN performance to the size of the colour palette. This result, that we have found very general, implies that the proposition of adaptive algorithms, CNN-like or other, that would search for the 'natural' number of clusters must be taken with great care; moreover, if the distortion is a salient component of the clustering criterion function. The inspection of the figures shows that the algorithms perform adaptively in almost all cases: the relative distortion is less than 1 most of the time. The exception occurs usually at image #2 of the sequence. This can be explained by the narrow gap between the reference curves at this point of the image sequence; however, the algorithms quickly recover.

Each of the plots show the result using both sample sizes. In general, it can be appreciated that the use of the bigger sample improves the results, although the magnitude of this improvement is related to the codebook size. The above-mentioned sensitivity to the number of clusters searched can be appreciated if we compare one-to-one the set of Figs 3(a), (c), (e) and (g), Figs 4(a), (c), (e) and (g) with Figs 3(b), (d), (f) and (h) with Figs 4(b), (d), (f) and (h). This sensitivity is attributable to the ratio between the size of the sample and the number of representatives searched: the sample-to-codebook ratio. We have found that $\frac{n}{c} \cong 100$ is a good ratio in many cases, *sample1* fits this ratio for 16 colours and *sample2* for 256 colours. The significance of this ratio is confirmed by the following observations over the Figs 3 and 4:

- When searching for 16 colours the use of *sample2* does not give an improvement over the results obtained from *sample1*, according to the increase in computational complexity.
- When searching for 256 colours, the use of

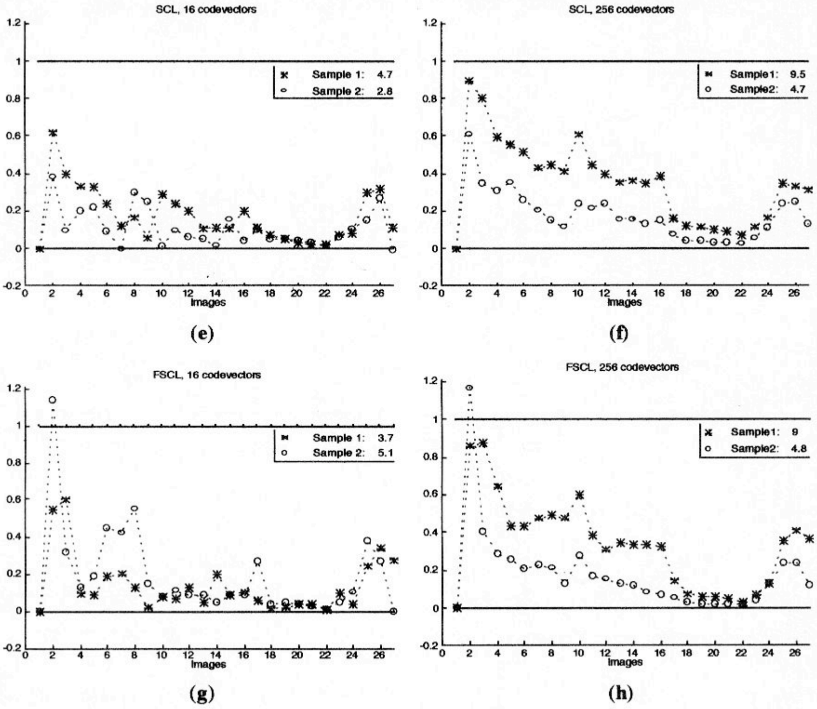


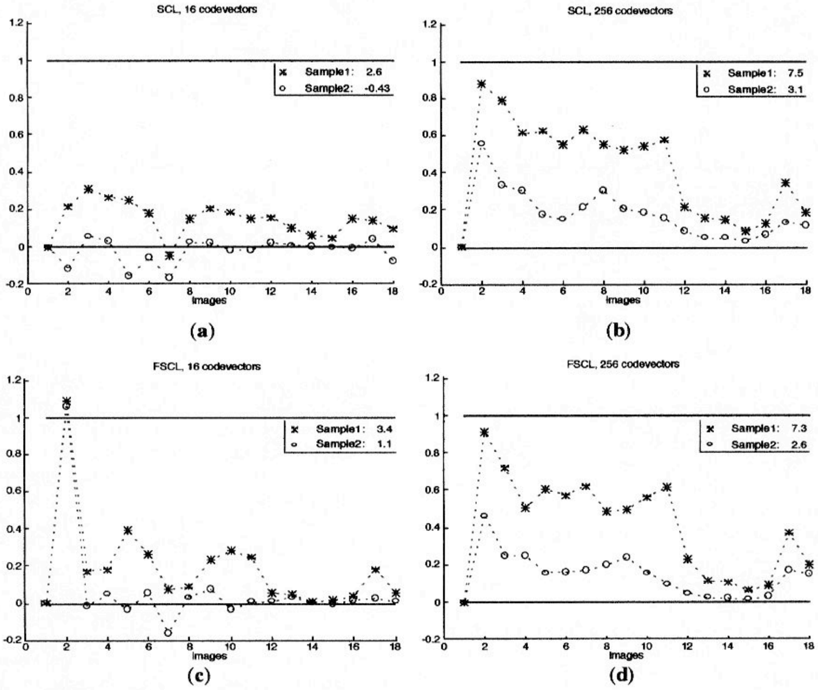
Fig. 3. (cont.) Relative distortion results for *sequence2* with local learning rates applying SCL to (e) c = 16 and (f) c = 256, and applying FSCL to (g) c = 16 and (h) c = 256.

Table 1a. Accumulated relative distortion results of the colour quantisation of experimental sequences with the colour representatives computed adaptively by the SCL and FSCL with local scheduling of the learning rates, for various initial conditions, sample sizes (S1: *sample1*, S2: *sample2*) and number of colour representatives.

		Sequence1				Sequence2			
		c = 16		c = 256		c = 16		c = 256	
		S1	S2	S1	S2	S1	S2	S1	S2
SCL	Heckbert	2.5	2.9	6.6	2.9	4.7	2.8	9.5	4.7
	Threshold	9.1	3.4	15	8.1	11	3.2	23	13
	Sample	3.6	3.2	15	7.2	3.6	3.2	23	10
	RGB box	12	6.4	32	16	15	8.5	52	25
FSCL	Heckbert	3.4	4.7	6.2	2.5	3.7	5.1	9	4.8
	Threshold	6.7	4.6	15	7.2	8	5.7	23	12
	Sample	3.3	4.6	15	5.7	4.7	5.1	23	9.6
	RGB box	3.2	4.8	21	5.4	3.1	5.2	33	8.6

Table 1b. Accumulated relative distortion results of the colour quantisation of experimental sequences with the colour representatives computed adaptively by the SCL and FSCL with global scheduling of the learning rates, for various initial conditions, sample sizes (S1: *sample1*, S2: *sample2*) and number of colour representatives.

		Sequence1				Sequence2			
		$c = 16$		$c = 256$		$c = 16$		$c = 256$	
		S1	S2	S1	S2	S1	S2	S1	S2
SCL	Heckbert	2.6	−0.43	7.5	3.1	3.8	−1	11	4.6
	Threshold	9.8	0.44	18	8.5	12	0.25	28	13
	Sample	4.1	0.05	18	7.4	4.7	−0.39	28	11
	RGB box	13	3.7	36	17	16	7	57	29
FSCL	Heckbert	3.4	1.1	7.3	2.6	3	2.1	10	4
	Threshold	9	1.1	19	7.3	9	2.6	29	13
	Sample	3.5	0.95	19	6	4.4	2.4	29	9.6
	RGB box	4.7	1.4	28	6.8	4.5	2.3	45	11



(a) (b) (c) (d)

Fig. 4. Relative distortion results for *sequence1* with global learning rate applying SCL to (a) c = 16 and (b) c = 256, and applying FSCL to (c) c = 16 and (d) c = 256.

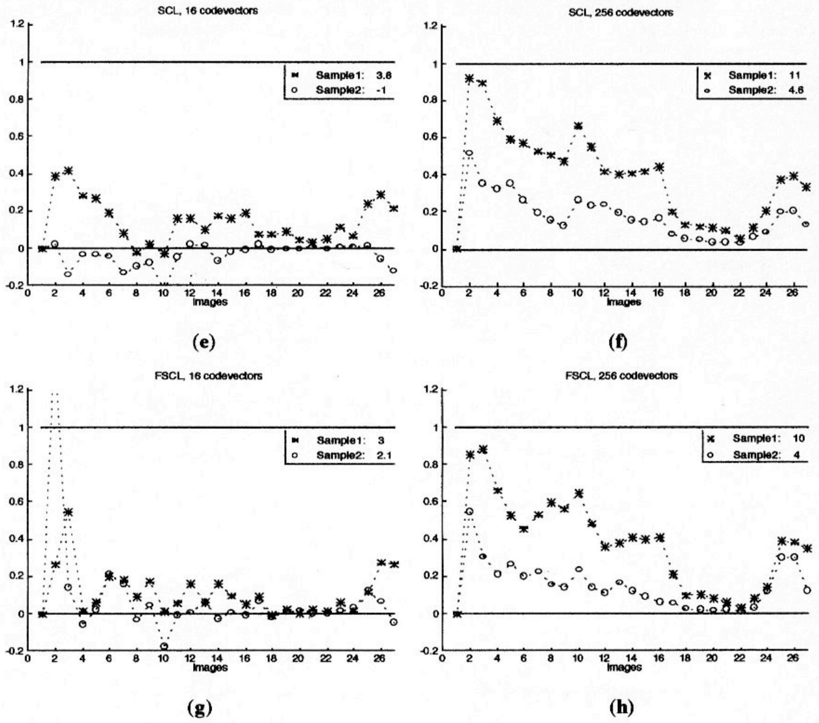


Fig. 4. (cont.) Relative distortion results for *sequence2* with global learning rate applying SCL to (e) $c = 16$ and (f) $c = 256$, and applying FSCL to (g) $c = 16$ and (h) $c = 256$.

sample2 significantly improves the results over those given obtained from *sample1*.

This ratio is of interest to bound the real time applicability of our algorithms, or the suboptimal results that can be expected from the use of small samples imposed by real time constraints. In Table 1 we have gathered the global distortion results that summarise all the experiments, including the sensitivity to initial conditions, which will be discussed later. From Table 1, it can be seen that the impact of the sample-to-codebook ratio also depends upon other elements of the algorithms, such as the initial conditions and the scheduling of the learning rate.

5.2. The Effect of Learning Rate Scheduling

The effect of the scheduling of the learning rate can be appreciated by a comparison of the plots in Fig. 3 with those in Fig. 4. Also in Table 1, we have a separate subtable for each scheduling strategy. Our conclusion is that the global scheduling of the learning rate gives better results than the local scheduling when the amount of available information increases. However, as the information becomes scarce, the local scheduling is more robust and gives better results. To support this conclusion, observe in Table 1 that the results of the global scheduling subtable improve over the corresponding ones of the local scheduling subtable when *sample2* is used to search for 16 colours, and when applying the algorithm to *sequence2* (which is smoother than *sequence1*) starting from relatively good conditions.

5.3. The Effect of Time Subsampling

As we have said, *sequence1* was a more coarse time sampling of the original video sequence. This

Table 2. Mean per image relative distortion, computed averaging the entries in Table 1 and taking into account the different number of images in each sequence.

	Sequence1 (18 images)				Sequence2 (27 images)			
	$c = 16$		$c = 256$		$c = 16$		$c = 256$	
	S1	S2	S1	S2	S1	S2	S1	S2
SCL	0.14	0.07	0.39	0.17	0.16	0.03	0.37	0.17
FSCL	0.19	0.16	0.37	0.14	0.12	0.13	0.35	0.16

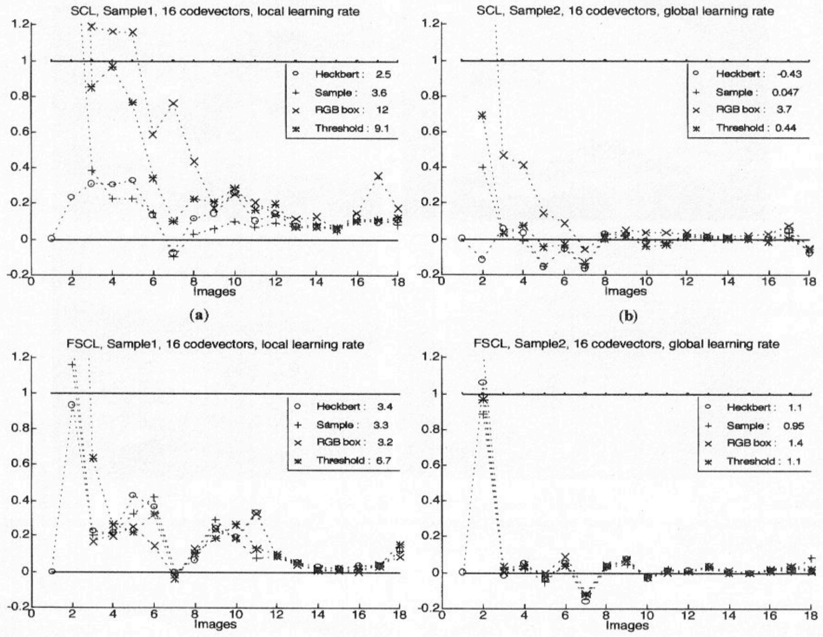
coarseness would lead to more abrupt changes in distribution, that would make sequence 1 less apt for adaptive computation. In Table 2, the mean relative distortion per image has been computed by averaging the entries in Table 1 and taking into account the number of images in each sequence. This table allows to evaluate the impact of the supposed augmented smoothness of *sequence2*. It can be seen from it that other factors have more impact than the increased smoothness produced by a fine time sampling.

5.4. Robustness to Initial Conditions

If we, at this point, try to compare SCL results with FSCL results, the main conclusions are:

1. SCL performs better than FSCL starting from good initial conditions, and high sample-to-codebook ratio is available.
2. FSCL is more robust than SCL, as intended, in the sense that it improves SCL when the algorithm starts from bad initial conditions and the sample data is scarce.

These conclusions are consistent with the proposition of SCL as a local optimisation procedure, and of FSCL as a global optimisation procedure. The global properties of FSCL ensure a good average result, but starting from good initial conditions, the pure local algorithm performs better. These conclusions are made stronger when considering the robustness to initial conditions, evaluated in the second experiment. In Table 1, we present the sum of relative distortion results of the colour quantis-



(a)

(b)

(c)

(d)

Fig. 5. Robustness: relative distortion results of the colour quantisation of experimental *sequence1* with the 16 colour representatives computed adaptively by the SCL and FSCL with optimal learning rates for both sample sizes, and various initial conditions.

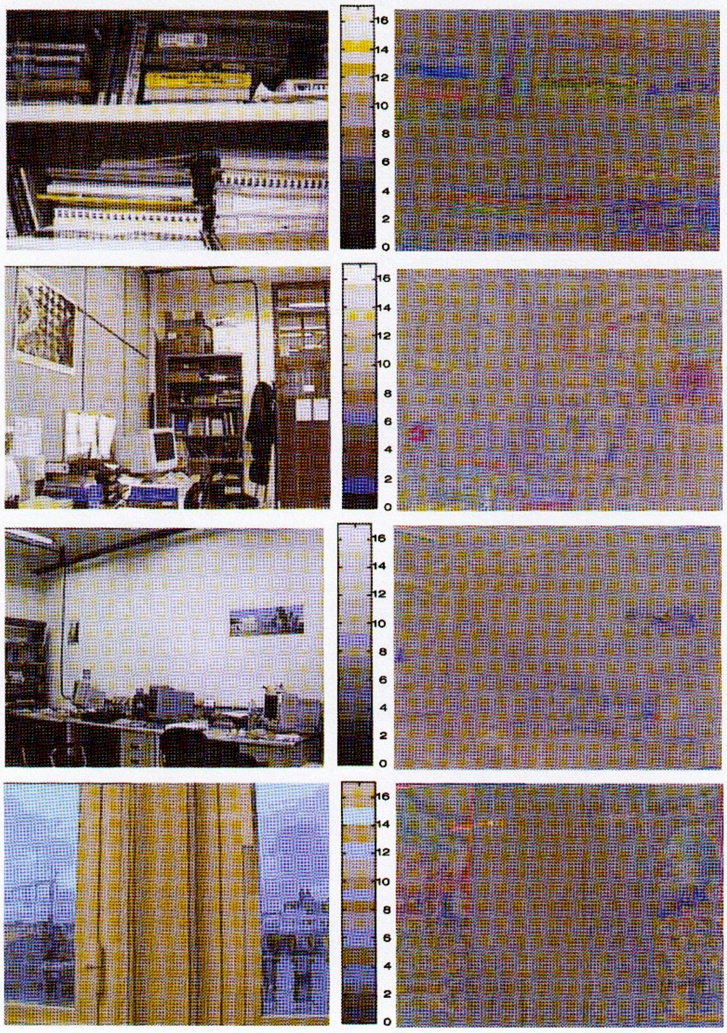


Fig. 6. Results of the colour quantisation of image #2, #8, #12, and #20 of the experimental *sequence2* with the 16 colour representatives (middle images) computed by Heckbert using full size images (Time Varying). On the left the quantised images, and on the right the error images.

ation of entire sequences obtained by applying the SCL and FSCL (with both global and local learning rates), starting from various initial colour representatives: the Heckbert colour representatives for image #1 (*Heckbert*), a threshold-based selection of the sample of image #1 (*Threshold*), random points in the RGB cube (*RGB box*) and a random selection of samples of image #1 (*Sample*). The results given are the sum of relative distortion through the sequence excluding image #1. The results are not

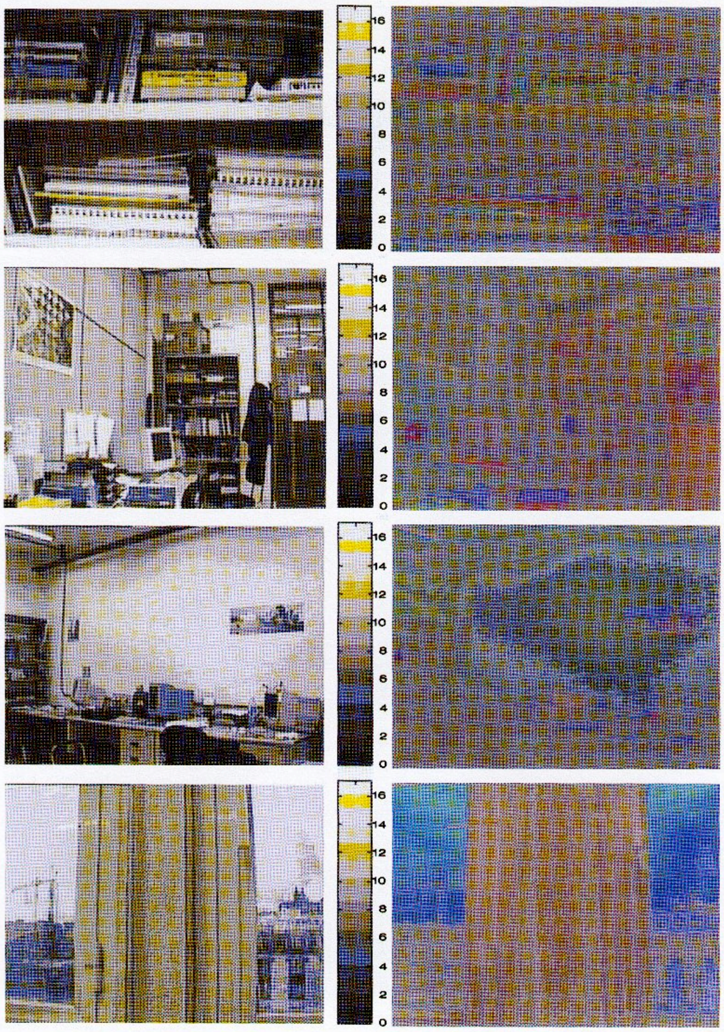


Fig. 7. Results of the colour quantisation of image #2, #8, #12, and #20 of the experimental *sequence2* with the 16 colour representatives (middle images) computed by Heckbert using #1 image (Time Invariant). On the left the quantised images, and on the right the error images.

averaged or normalised anyhow regarding the sequence duration, therefore entries for *sequence2* are bigger than those for *sequence1*. Table 1 shows that there is a remarkable increment of distortion results due to initial conditions. The worst case is the application of the SCL algorithms to search 256 representatives, starting from the *RGB box* initialisation. Also, the table confirms that FSCL is more

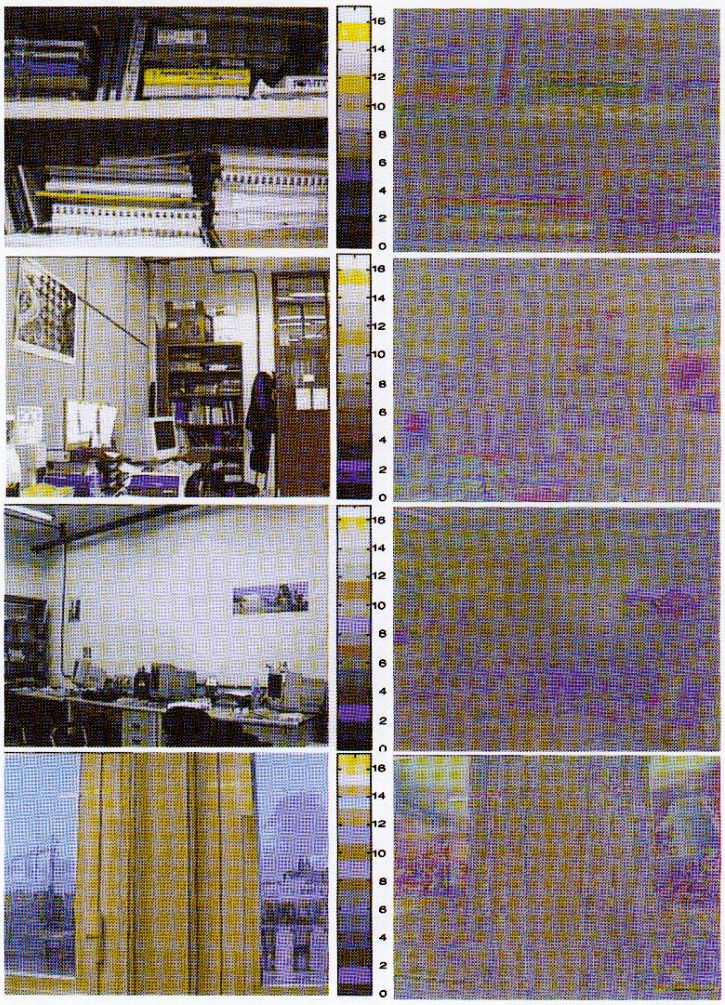


Fig. 8. Results of the colour quantisation of image #2, #8, #12, and #20 of the experimental *sequence2* with the 16 colour representatives (middle images) computed adaptively by the SCL with local learning rates, using *sample1*, and *Heckbert* initial condition. On the left the quantised images, and on the right the error images.

robust than SCL to very bad initial conditions, but that their performance is comparable for good and reasonably good initial conditions. SCL may even improve FSCL in very good initial conditions and numerical circumstances: a large sample and global scheduling of the learning rate.

In Fig. 5 we show some of the responses to diverse initial conditions in detail. The sequence considered is *sequence1*; the number of colour representatives is 16. The worst response is in Fig. 5(a) for SCL using *sample1* and a local scheduling of the learning rate. The algorithm tries to approach

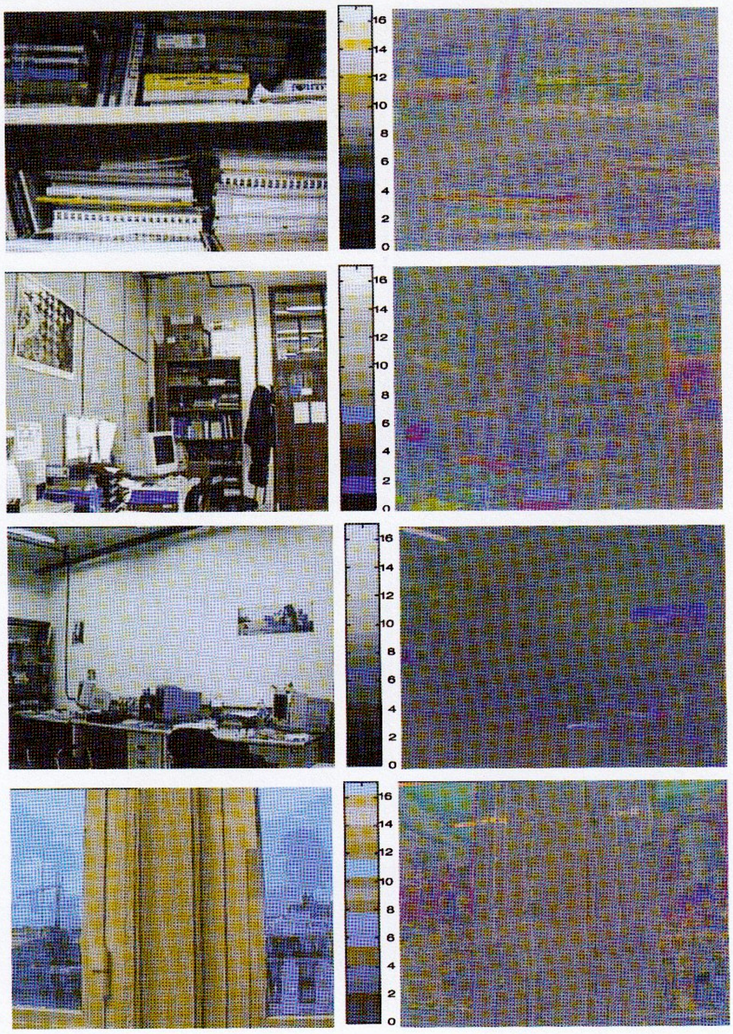


Fig. 9. Results of the colour quantisation of image #2, #8, #12, and #20 of the experimental *sequence2* with the 16 colour representatives (middle images) computed adaptively by the FSCL with local learning rates, using *sample1*, and *Heckbert* initial condition. On the left the quantised images, and on the right the error images.

the response obtained from the Heckbert initial condition, starting from the other initial conditions. Starting from a good initial condition (*Sample*), the SCL gives the same response after five images. Starting from a medium quality initial condition (*Threshold*), the SCL recovers from the bad initialisation after some 14 images. Finally, the worst initial condition (*RGB box*) cannot be recovered in the

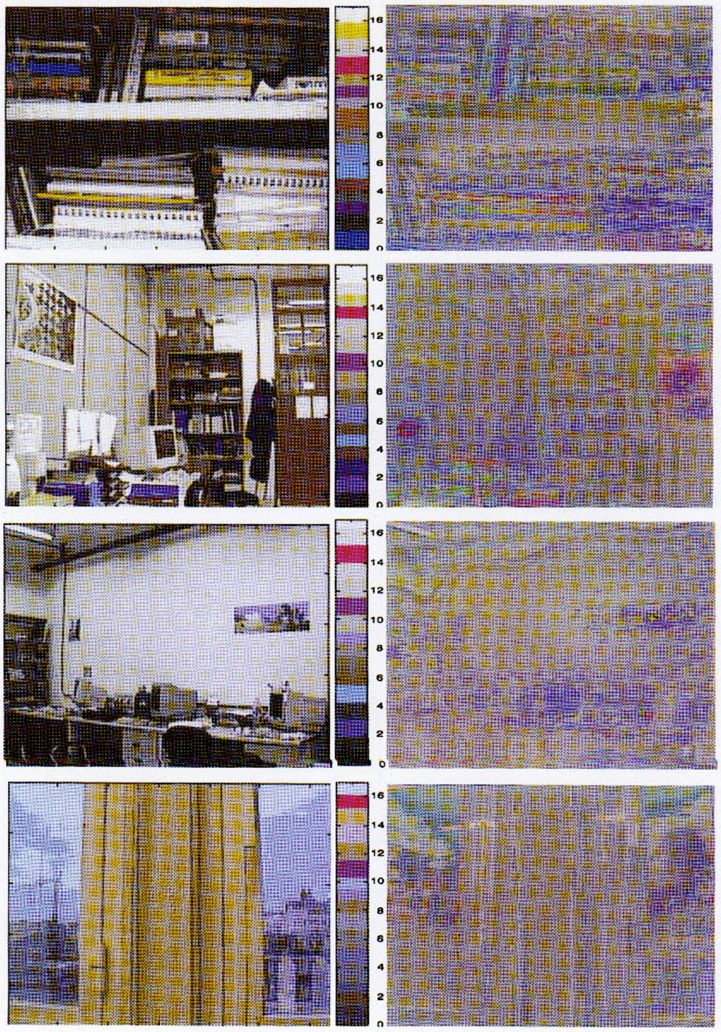


Fig. 10. Results of the colour quantisation of image #2, #8, #12, and #20 of the experimental *sequence2* with the 16 colour representatives (middle images) computed adaptively by the SCL with local learning rates, using *sample1*, and *RGBbox* initial condition. On the left the quantised images, and on the right the error images.

duration of the sequence. On the other hand, the best case is that of Fig. 5(d); with *sample2* and the global learning rate that recovers from the bad initial conditions very fast, its response collapses after two or three images in the one corresponding to the best initial condition. FSCL will be more robust than SCL, in the sense that the effect of bad initial codebooks is recovered faster by FSCL than by

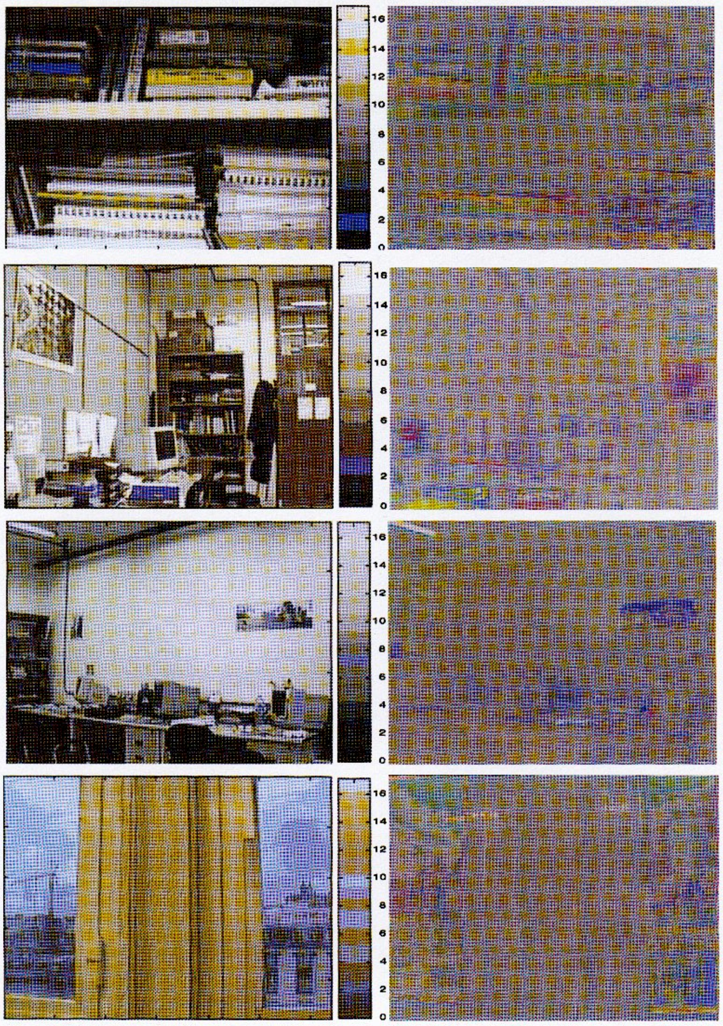


Fig. 11. Results of the colour quantisation of image #2, #8, #12, and #20 of the experimental *sequence2* with the 16 colour representatives (middle images) computed adaptively by the FSCL with local learning rates, using *sample1*, and *RGBbox* initial condition. On the left the quantised images, and on the right the error images.

SCL, under the same numerical settings. This can be appreciated comparing Figs 5(a), (b) with Figs 5(c), (d).

5.5. Visual Results

To give a visual qualitative appreciation of the colour quantisation results, Figs 6–11 show the results of colour quantisation to 16 colour representatives on images #2, #8, #12 and #20 of *sequence2*, under the application of some instances of the algorithms discussed in the paper. These images were selected because they show the sharper transition of colour distribution. For each image we show the colour quantised image (left), the colour representatives found by the algorithm (colour bars in the middle), and colour quantisation error (right) as a colour image obtained from the error in each colour axis. Figures 6 and 7 give the visual results of the *Time Varying* and *Time Invariant* application, respectively, of the Heckbert algorithm over the entire image. The suboptimality of the latter is appreciable from the inspection of the quantised images (left). The inspection of the error images (right) shows the increase in magnitude of the error of the *Time Invariant* relative to the *Time Varying* strategy, as time goes on.

Figures 8 and 9 show, respectively, the results of the application of SCL and FSCL using the optimal initial conditions (*Heckbert*) and the small sample of each image. It can be appreciated that the visual differences between the quantised images (left) obtained from SCL and FSCL are almost negligible. However, looking at the colour representatives shown in the middle colour bars, it can be appreciated that the yellow colour representative is not changed by SCL (Fig. 8, middle), although it is not used in the codification. This yellow colour does correspond to a stuck codevector, that it is not changed by SCL along the whole sequence. The colour representatives of FSCL (Fig. 9, middle) follow more closely those found by the Heckbert *Time Varying* application.

To illustrate the relative increase of robustness to initial conditions of FSCL over SCL, Figs 10 and 11 show, respectively, the results of their application using the worst initial condition (*RGB box*) and the small sample of each image. The existence of stuck codevectors is conspicuous in Fig. 10, while the colour representatives computed by FSCL in Fig. 11 behave similar to those shown in Fig. 9, and again follow better the optimal colour representatives computed by the *Time Varying* Heckbert shown in Fig. 6. The error images in Fig. 11 produced by FSCL are more smooth and of lesser magnitude than those produced by SCL in Fig. 10. The visual difference of the quantised images obtained from SCL and FSCL is not so noticeable.

Regarding the visual evaluation of the results, we can conclude that the adaptive neural algorithms improve significatively over the stationary solution given by the *Time Invariant* Heckbert algorithm. Despite the existence of stuck codevectors, SCL give visual results with a quality similar to the visual results of FSCL.

2. Conclusions

Non-stationary colour quantisation is the problem of finding optimal colour representatives in image sequences. It has been put into the framework of non-stationary clustering problems, and two basic competitive neural network architectures have been proposed as one-pass adaptive vector quantisation methods to solve the problem. The solution provided by the CNN is able for real time implementation, given its $O(d.n.c)$ computational cost, versus the $O(k^d)$ cost of the Heckbert algorithm. Besides real time considerations, these cost figures imply the CNN methods can be applied to higher dimensional instances of non-stationary clustering problems (i.e. hyperspectral images). In this paper, we have concentrated on two basic CNN architectures: SCL and FSCL, trying to assess their performance and numerical sensitivities. This work will be of interest in understanding and anticipating the response of other CNN architectures on this problem.

We have found that both SCL and FSCL show some sensitivity to the number of colour representatives searched (the size of the codebook). This is expected to be a general sensitivity, and has been also found in other CNN architectures. We have studied the response to changes in sample size, and the implicit generalisation ability of SCL and FSCL. A convenient ratio between sample size and codebook size is 100:1. We have studied the effect of local versus global scheduling of the learning rate, finding that the latter is more efficient when there is a surplus of training data, whereas the former provides an additional robustness when the data is scarce. Finally, we have tested the response of both SCL and FSCL to suboptimal initial conditions, finding that the FSCL strategy of looking for uniform quantisers provides greater robustness to bad initial conditions. Our conclusion is that one-pass application of CNN can be a successful family of adaptive vector quantisation, qualified for real-time and high dimensional applications.

References

1. Gersho A, Gray RM. Vector Quantisation and Signal Compression. Kluwer, 1992
2. Hartigan J. Clustering Algorithms. Wiley, 1975
3. Diday E, Simon JC. Clustering analysis. In: Fu KS (ed), Digital Pattern Recognition, Springer-Verlag, 1980, pp 47–94
4. Duda RD, Hart PE. Pattern Classification and Scene Analysis. Wiley, 1973
5. Jain AK, Dubes RC. Algorithms for Clustering Data. Prentice Hall, 1988
6. Fukunaga K. Statistical Pattern Recognition. Academic Press, 1990
7. Pratt WK. Digital Image Processing. Wiley, 1991
8. Heckbert P. Color image quantisatión for frame-buffer display. Computer Graphics 1980; 16(3): 297–307
9. Joy G, Xiang Z. Center cut for color image quantization. The Visual Computer 1993; 10: 62–66
10. Xiang Z. Color image quantisation by minimizing the maximum intercluster distance. ACM Trans Graphics 1997; 16(3): 260–276
11. Litmann E, Ritter H. Adaptive color segmentation – A comparison of neural and statistical methods. IEEE Trans Neural Networks 1997; 8(1): 175–185
12. Chen OT, Chen BJ, Zhang Z. An adaptive vector quantisation based on the gold-washing method for image compression. IEEE Trans Circuits & Systems for Video Tech 1994; 4(2): 143–156
13. Fowler JE. Adaptive Vector Quantisation for the coding of nonstationary sources. PhD Thesis, Ohio University, 1995
14. Mao J, Jain AK. A self-organizing network for hyperellipsoidal clustering. IEEE Trans Neural Networks 1996; 7(1): 16–29
15. Ahalt SC, Krishnamurthy AK, Chen P, Melton DE. Competitive learning algorithms for vector quantisation. Neural Networks 1990; 3: 277–290
16. Hertz JA, Krogh AS, Palmer RG. Introduction to the Theory of Neural Computation. Addison Wesly, 1991
17. Kosko B. Stochastic competitive learning. IEEE Trans Neural Networks 1991; 2: 522–529
18. Uchiyama T, Arbib MA. Color image segmentation using competitive learning. IEEE Trans Pattern Analysis and Machine Intelligence 1994; 16(12): 1197–1206
19. Gong Y, Zen H, Ohsawa Y, Sakauchi M. A color video image quantisation method with stable and efficient color selection capability. Int Conf Pattern Recognition 1992; 3: 33–36
20. Furlani JL, McMillan L, Westover L. Adaptive colormap selection algorithm for motion sequences. Proc Multimedia'94, San Francisco, CA, 1994, pp 341–347
21. Zhang Z, Wei VK. An on-line universal lossy data compression algorithm via continuous codebook refinement – Part I: Basic results. IEEE Trans Inf Theory 1996; 42(3): 803
22. Gonzalez AI, Graña M, D'Anjou A, Cottrell M. On the application of Competitive Neural Networks to time-varying clustering problems. In: Silva FL, Principe J, Almeida LB (eds), Spatiotemporal Models in Biological and Artificial Systems. 105 Press, 1996, pp. 49–55
23. Gonzalez AI, Graña M, D'Anjou A, Albizuri FX, Cottrell M. A sensitivity analysis of the self organizing map as an adaptive one-pass non-stationary clustering algorithm: the case of color quantisation of image sequences. Neural Processing Lett 1997; 6: 77–89
24. Gonzalez AI, Graña M. Competitive neural networks as adaptive algorithms for non-stationary clustering: experimental results on the color quantisation of image sequences. Int Conf Neural Networks, Houston, TX, 1997; 3: 1602–1607
25. Orchard MT, Bouman CA. Color quantisation of images. IEEE Trans Signal Process 1991; 39(12): 2677–2690
26. Lin TS, Chang LW. Fast color image quantisation with error diffusion and morphological operations. Signal Process 1995; 43: 293–303
27. Chen RJ, Chien BC. Three dimensional morphological pyramid and its application to color image sequence coding. Signal Process 1995; 44: 163–180
28. Kankanhalli MS, Mehtre BM, Wu JK. Cluster based color matching for image retrieval. Pattern Recognition 1996; 29(4): 701–708
29. Swain MJ, Ballard DH. Color indexing. Int J Computer Vision 1991; 7(1): 11–32
30. Androutsos D, Plataionitis KN, Venetsanopoulos AN. A perceptually motivated technique to query by example using color cardinality. Proc Multimedia Storage and Archiving Systems IV, Proc SPIE 1999; 3846: 137–145
31. Jain AK, Vailaya A. Image retrieval using color and shape. Patt Recog 1996; 29(8): 1233–1244
32. Zhong D, Chang S, Smith JR. Differential compression and optimal caching methods for content-based image search systems. Proc Multimedia Storage and Archiving Systems IV, Proc SPIE 1999; 3846: 413–421
33. Chen JY, Taskiran C, Albiol A, Bouman CA, Delp EJ. VIBE: A video indexing and browsing environment. Proc Multimedia Storage and Archiving Systems IV, Proc SPIE 1999; 3846: 148–164
34. Lee CH, Kim JS, Park KH. Automatic human face location in a complex background using motion and color information. Patt Recog 1996; 29(11): 1877–1889
35. Graña M, Echave Y. Real-time optical flow computation based on adaptive color quantization by competitive neural networks. Intelligent Robots and Computer Vision XVIII, Proc SPIE 1999; 3837: 165–174
36. Xie M. Automatic feature matching in uncalibrated stereo vision through the use of color. Rob Aut Sys 1997; 21: 355–365
37. Wu X. Efficient statistical computations for optimal color quantization. In: Arvo J (ed), Graphics Gems II. Academic Press, 1991, pp 126–133