AN EVOLVING LOCALISED LEARNING MODEL FOR ON-LINE IMAGE COLOUR QUANTISATION

Da Deng and Nikola Kasabov

Department of Information Science
University of Otago
PO Box 56, Dunedin, New Zealand
{ddeng, nkasabov}@infoscience.otago.ac.nz

ABSTRACT

Although widely studied for many years, colour quantisation remains a practical problem in image processing. Unlike previous works where the image can only be quantised after the whole set of image data is acquired, we propose to use an evolving localised learning model for on-line colour quantisation. This approach is compared with some conventional algorithms.

1. INTRODUCTION

Image colour quantisation is a process for reducing the number of colours of a digital colour image. It is one of the most frequently used operations in computer graphics and image processing and is closely related to vector quantisation and image compression. Despite of the popularity of 24-bit graphics hardware, the practical value of colour quantisation remains and is also found in image analysis applications. On the other hand, it remains a time consuming task although being widely studied for many years.

Typically a two-phase process is needed to reduce a given true colour image into one with less colour resolution. The first phase, is to select a palette with best representative colours. The second, is to map each colour in the image to a corresponding colour in the palette. There are two general classes of quantisation methods: fixed and adaptive. In fixed quantisation, a pre-defined set of display colours and a fixed mapping from the image colours to display colours are used. Fixed quantisation is very fast, but sacrifices the quantisation quality. In adaptive quantisation, colour space of the image is partitioned into clusters of a target number and the centroids of these clusters define the resulting colour map. Some popular implementations include median-cut [5] and octree[6] etc. These methods are mainly based on colour histogram, requiring the whole set of image data to be acquired so that partitioning can proceed. They give much better quantisation results, but take much more time than fixed quantisation. Another branch of the adaptive solution is connected with clustering algorithms [9][8]. Although not popularly applied yet, this approach is promising with good adaptability and on-line learning ability.

In the context of data clustering and vector quantization (VQ), assume we have a data manifold χ of dimension D, i.e., $\chi \subseteq R^D$. We aim at finding a set of prototypes $\mathcal{W} = \{\mathbf{w}_1, ..., \mathbf{w}_N\}$, which encodes the data manifold with small quantization error. VQ usually utilises a competitive rule, i.e., the input vector x is represented by the best matching unit $\mathbf{w}_{i(\mathbf{x})}$, which satisfies

$$\|\mathbf{x} - \mathbf{w}_i\| \leq \|\mathbf{x} - \mathbf{w}_j\|, \forall j \neq i, i, j \in [1, N] \quad (1)$$

The goal is to minimise the reconstruction error

$$E = \int d^D x P(\mathbf{x}) \|\mathbf{x} - \mathbf{w}_{i(\mathbf{x})}\|^2 \quad (2)$$

Here $P(\mathbf{x})$ is the probability distribution of data vectors over the manifold χ.

A straightforward approach for clustering and VQ is the well known K-means algorithm. Its on-line version is [7]:

$$\Delta \mathbf{w}_i = \begin{cases} \gamma(\mathbf{x} - \mathbf{w}_i), & \text{if } i = i(\mathbf{x}) \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

with γ as the learning rate. Such an on-line learning rule is likely to be trapped in local minima. A solution to this is to adopt some "soft" computing schemes in which not only the "winner" prototype is modified, but all reference vectors are adjusted depending on their proximity to the input vector.

Kohonen's self-organizing feature map (SOM) [4] is another VQ-related algorithm in the neural network category. SOM features abilities of topology preserving and approximation of data distribution. The topology of the low dimensional feature map is pre-determined. This is good for visualisation purpose, but it also limits its data modelling ability, as the data manifold can be rather complicated. The size of SOM is also fixed and therefore it is not an ideal choice for on-line tasks.

The constraint of a low dimensional map topology is removed in the neural-gas model [7], with a learning rule similar to SOM, but the prototype vectors are organised in the original manifold of the input space. The weight updating rule is similar to that of SOM, but requiring the calculation of neighbourhood rank of the prototypes related to the current input, since no topology order is defined. This brings up the time complexity for each step of weight adapting. In [3] Fritzke introduced a growing neural gas (GNG) model which originates from the neural gas, but the network is allowed to adaptively grow. GNG needs to calculate local resources for prototypes, which introduces extra computational effort and reduces their efficiency.

We proposed an evolving self-organising map algorithm (ESOM) [2] as an evolving variation of SOM for on-line VQ and classification tasks. It features fast one-pass on-line learning, evolvable network structure and good topology preserving ability. In this paper, we will further explore the plausibility of using this algorithm to tackle the problem of image colour quantisation.

2. THE ESOM ALGORITHM

Following the context of vector quantisation, our approach is to allow the feature map to be evolved quickly and acquire topological representation in the meantime. The neighbourhood of neurons are not pre-defined, but are dealt 'as-it-is', i.e., according to their mutual distances in the original space. Thus ESOM avoids the time complexity of searching for neighbourhood ranking as in the neural gas algorithm, and is free of the constraint of a low dimensional map topology.

Activations on prototype nodes are first defined. Given an input vector x, the activation on the i-th node is defined as a matching score:

$$a_i(\mathbf{x}, \mathbf{w}_i) = e^{-2\|\mathbf{x}-\mathbf{w}_i\|^2/\epsilon^2} \tag{4}$$

where ϵ is a radial. Nodes may be connected. If a connection between node i and node k is to be set, the connection strength is defined as

$$c_{i,k} = \frac{\epsilon}{\epsilon + \|\mathbf{w}_i - \mathbf{w}_k\|} \tag{5}$$

ESOM adopts a localised, soft-winning competitive mechanism, in which the best winning unit and its neighbours are updated. We define a cost function as weighted quantisation error for the input vector:

$$E = \int d^D x \sum_{i=1}^{N} P(\mathbf{x}) a_i(\mathbf{x}, \mathbf{w}_i) \|\mathbf{x} - \mathbf{w}_i\|^2 \tag{6}$$

Here $a_i(\mathbf{x}, \mathbf{w}_i)$ acts as a weighting factor assigned to the i-th prototype vector. So the prototype which matches better should also contribute more to the matching error. The on-line stochastic approximation of Eq.(6) gives:

$$E_{app} = \Sigma a_i \|\mathbf{x} - \mathbf{w}_i\|^2 \tag{7}$$

By gradient descent we have the following weight updating rule in a simplified form:

$$\Delta\mathbf{w}_i = \gamma a_i (\mathbf{x} - \mathbf{w}_i) \qquad i \in [1, N] \tag{8}$$

Here γ is a learning rate held as a small constant.

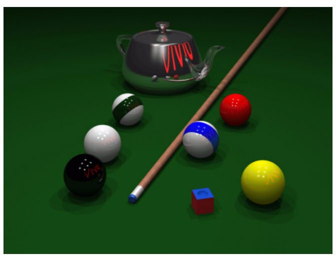
ESOM starts with a null network, and gradually allocates new prototypes when new data samples can not be matched well onto existing prototypes. Node insertion is straight-forward. The new node is inserted using exactly the poorly matched input vector. When handling clustered data, this simple approach shows advantage over the mid-point interpolation heuristics used in GNG. Although direct allocation is sensitive to noise and may introduce some artifacts in clustering, this can be mitigated by automatic deletion of obsolete nodes.

The ESOM algorithm is summarised as follows:

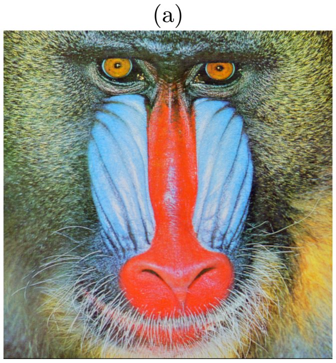
1. Input a new data vector $\mathbf{x}$;
2. If there are no prototype nodes, go to Step 4;
3. Matching. Look for a prototype subset $\mathcal{S}$ consisting of the best matching unit and its neighbours.
$$\mathcal{S} = \{\mathbf{w}_b\} \cup \{\mathbf{w}_i | c_{b,i} > 0, i = 1..N\} \tag{9}$$
Here $\mathbf{w}_b$ is the best matching prototype. Calculate activations on nodes in $\mathcal{S}$ using Eq.(4), go to Step 5;
4. Insertion. Create a new node in the network representing the input:
$$\mathbf{w}_{N+1} = \mathbf{x}, N \leftarrow N + 1 \tag{10}$$
Connections are then made from the newly inserted node to the first two nearest neighbours.
5. Updating. Modify all prototypes in $\mathcal{S}$ with Eq.(8), update the strength of connection to the winner.
6. Deletion. After every T_p steps of learning time, prune the weakest connection. If isolated nodes appear, prune them as well;
7. Go back to Step 1 (until no more data are available).

3. ESOM FOR ON-LINE COLOUR QUANTISATION

We apply the ESOM algorithm in the problem of on-line colour quantisation and compare the results with those achieved by other methods including median-cut, octree, and Wu's method [10]. Three test images are chosen: Pool Balls, Mandrill and Lenna, as shown in Fig.1. The Pool Balls image is artificial and contains smooth colour tones and shades. The Mandrill image is of 262144 (512×512) pixels but has a very large number of colours (230427). The Lenna image is widely used image processing literature and contains both smooth areas and fine details. With all these images very good quantisation results are obtained with ESOM, shown in Fig.2.



(a)

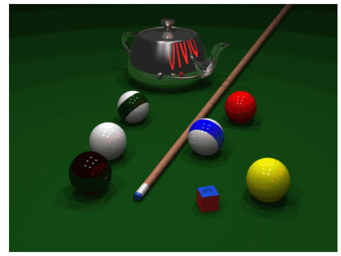


(b)

(c)

Fig. 1. The source images. (a) Pool Balls, (b) Mandrill. (c) Lenna.

The RGB colour space is used directly in on-line clustering. Here we denote the image as I, with a pixel



(a)

(b)

(c)

Fig. 2. Test images quantised to 256 colours. (a)Pool Balls with $\epsilon = 18.6$, (b)Mandrill, $\epsilon = 20.4$, (c)Lenna, $\epsilon = 31.9$. $T_p = 2000$, $\gamma = 0.05$ in all three cases.

number of N. The input vector to the ESOM algorithm is now a 3-dimensional one: $I_i = (R_i, G_i, B_i)$. The on-line clustering process of ESOM will construct a colour map $\mathcal{C} = \{c_j | j = 1...256\}$. Each image pixel is then quantised to the best-matching palette colour c_m, a process denoted as $Q : I_i \rightarrow c_m$. To speed up the calculation process, the L_α norm [1] is adopted as an approximation of the Euclidean metric used in ESOM. It is defined as

$$\|\mathbf{x}\|_\alpha = (1-\alpha)\sum_{j=1}^{n} |x_j| + \alpha \mathtt{max}_{j=1}^{n} |x_j| \qquad (11)$$

where the vector $\mathbf{x} \in R^n$. We use $\alpha = 1/2$.

The quantisation root mean square error (QRMSE) is defined as

$$\epsilon_{I,Q} = \left[\frac{1}{N} \sum_{i=1}^{N} d(I_i, c_m)^2 \right]^{\frac{1}{2}} \quad (12)$$

Apart from the quantisation error, quantisation error variance is another factor which influences the visual quality of the quantised image. The standard deviation of error is defined as

$$\sigma = \left[\frac{\sum_i (\|I_i - c_m\| - \epsilon_{I,Q})^2}{N} \right]^{\frac{1}{2}} \quad (13)$$

Quantisation performance of different methods is compared in Table 1.

Table 1. Quantisation performances: quantisation error / error deviation

Methods	Pool Balls	Mandrill	Lenna
Median-cut	2.58 / 8.28	11.32 / 5.59	6.03 / 3.50
Octree	4.15 / 3.55	13.17 / 4.98	7.56 / 3.83
Wu's	2.22 / 2.19	9.89 / 4.56	5.52 / 2.94
ESOM	2.43 / 2.56	9.47 / 3.86	5.28 / 2.36

Generally ESOM not only achieves a very small value of average quantisation error, its error variance is also the smallest. This explains why images quantised by ESOM have better visual quality than those done by other methods. On the other hand, ESOM works much slower than the compared methods. With the 512×480 sized Lenna image, it takes 2 seconds to construct the quantisation palette on a Pentium-II system running Linux 2.2. Although by using an evolving model the time searching for best matching colours is much less than using a model with fixed number of prototypes, the time consumed in searching needs to be further reduced. We intend to utilise tree structures to organise the colour palette so that searching for best matching colours can be more efficiently done. Besides, the merit of using ESOM is that there is a potential of hardware parallelism implementation, which will boost up the speed greatly and we believe this will enable real-time quantisation of video streams.

4. CONCLUSION

We propose to use an evolving localised learning model ESOM for the problem of on-line colour quantisation. Good results have been obtained with some benchmark colour images. Compared with other conventional methods we achieved smaller quantisation error and better visual quality in the quantised images. With further improvement of the computational efficiency, we will investigate in applying this approach to real world applications such as progressive image display and video display over the Internet.

5. ACKNOWLEDGEMENTS

This work is supported by the Foundation for Research, Science and Technology (FRST), New Zealand, under grant UOOX0016. The ESOM algorithm is implemented as part of the project repository RICBIS, URL http://divcom.otago.ac.nz/infosci/kel/RICBIS.

6. REFERENCES

[1] D. Chaudhuri, C.A. Murthy and B.B. Chaudhuri:", A modified metric to compute distance. Pattern Recognition 7(25) (1992), 667-677.

[2] D. Deng and N. Kasabov: ESOM: An algorithm to evolve self-organizing maps from on-line data streams. Proc. of IJCNN'2000 VI, Como, Italy (2000), 3-8.

[3] B. Fritzke: A growing neural gas network learns topologies, in Advances in neural information processing Systems, D. Touretzky and T.K. Keen eds., Cambridge MA: MIT Press (1995), 625-632.

[4] T. Kohonen: Self-Organizing Maps, second edition, Springer (1997).

[5] P. Heckbert: Color image quantization for frame buffer display. Computer Graphics (SIGGRAPH) 16(3) (1982) 297-307.

[6] M. Gervautz and W. Purgathofer: A simple method for color quantization: octree quantization. in Graphics Gems, A. Glassner, ed., Academic Press, New York (1990), 287-293.

[7] T.M. Martinetz, S.G. Berkovich and K.J. Schulten: "Neural-Gas" network for vector quantization and its application to time-series prediction. IEEE Trans. on Neural Networks 4 (1993), 558-569.

[8] J. Puzicha, M. Held, J. Ketterer, et al.: On Spatial Quantization of Color Images. Technical Report IAI-TR-98-1, University of Bonn (1998).

[9] O. Verevka and J.W. Buchanan: Local K-means Algorithm for color image quantization. Proc. of GI'95 Quebec, Canada (1995).

[10] X. Wu: Color quantization by dynamic programming and principal analysis. ACM Trans. on Graphics 11(4) (1992) 348-372.