PCA-BASED SEEDING FOR IMPROVED VECTOR QUANTIZATION

G. Knittel and R. Parys
WSI/GRIS, University of Tübingen, 72076 Tübingen, Germany

Keywords: Vector quantization, Image compression, Principal component analysis, Clustering.

Abstract: We propose a new method for finding initial codevectors for vector quantization. It is based on Principal Component Analysis and uses error-directed subdivision of the eigenspace in reduced dimensionality. Additionally, however, we include shape-directed split decisions based on eigenvalue ratios to improve the visual appearance. The method achieves about the same image quality as the well-known k-means++ method, while providing some global control over compression priorities.

1 INTRODUCTION

Vector quantization (Gray, 1984; Gersho and Gray, 1992) has become one of the standard methods for lossy image compression. Vectors are formed by non-overlapping blocks of n*m pixels, in case of RGB images the vector dimension is n*m*3. In VQ, the potentially large set of image vectors is replaced by a small set of representative vectors (here called codevectors), while trying to minimize the overall error. Often, clustering methods are used to find a proper set of codevectors (collectively called a codebook). A frequently used method is k-means (Lloyd, 1982). Starting from an initial set of random codevectors (seeds), each vector is assigned to its nearest codevector, thereby forming clusters. Once clustering is finished, the codevectors are moved to the center of their respective cluster, and then clustering is started anew. This process is repeated until the system reaches a stable state. Each vector will now be replaced by the index of its codevector in the codebook. Decompression merely consists of a table look-up.

As a number of authors have pointed out, the accuracy of the clustering algorithm depends to a large degree on the selection of seeds, since clustering typically converges to only locally optimal solutions (Barbakh and Fyfe, 2008; Fritzke, 1997; Ostrovsky et al., 2006). Accordingly, much research effort has been spent on improved seeding methods (Bradley and Fayyad, 1998; Pena et al., 1999). Central to this work, however, is the recently proposed k-means++ algorithm (Arthur and Vassilvitskii, 2007).

In this short note we demonstrate one of the weaknesses of the k-means++ selection method and propose a method for alleviating these effects. We compare random selection, k-means++ and the proposed PCA-based seeding method.

2 PCA-BASED SEEDING

Basically, the method generates a potentially unbalanced binary subdivision tree. As opposed to other trees such as kd-trees, our subdivision algorithm is error-guided and uses image properties reflected in the eigenvalues. As for all trees, we have to make decisions about which node to split, and where the cut should be made. A detailed description follows.

All image vectors are subjected to a PCA. The split is made using the principal component of each pixel block. Several locations to place the cut are possible, such as a median cut, but best results are achieved by using the center of gravity. For each of the two groups the image error contribution is computed, i.e., the sum of squared errors relative to their respective centers of gravity. The one with the larger error is split, using the same procedure as for the parent. For the split decisions, all subvolumes generated so far are taken into account. Processing is finished when there are as many leaf nodes as there are codevectors to assign.

Knittel G. and Parys R. (2009).
PCA-BASED SEEDING FOR IMPROVED VECTOR QUANTIZATION.
In *Proceedings of the First International Conference on Computer Imaging Theory and Applications*, pages 96-99
DOI: 10.5220/0001808100960099

This method achieves the same or even slightly better image quality (in terms of PSNR, see Table 1) compared to k-means++.

Looking at the decompressed images from random seeding (Figure 1d), k-means++ seeding (Figure 2a) and PCA-based seeding (Figure 2f), it is striking to see how well the smooth color transition of the sky is reproduced by random seeding. However, this is not a hidden power of the k-means algorithm, but simply due to the random choice if selection is uniformly distributed. This is reflected in the large number of seeds from sky in the codebook (see Figure 1e).

In contrast, both the k-means++ and the PCA-based seeding produce noticeable banding artefacts. In both cases, this is a direct consequence of the design intentions, since the involved image vectors are close to each other and won’t cause significant image errors. While it is non-obvious how a remedy could be integrated into the k-means++ algorithm, we will present a method for reducing these artifacts in the PCA-based seeding. It takes advantage of the fact that processing is done in eigenspace.

At some point in the subdivision process the sky (or similar areas) will have been separated into a distinct cluster, and will be subjected to a PCA. It’s quite obvious that the eigenvalues will exhibit a certain property: they will drop sharply in size since all image vectors are more or less aligned from dark to light blue. This property is less pronounced or absent in more noisy or diverse image areas.

Thus we can use the eigenvalue statistics (cluster shape) as a further split criterion. As a simple example, we have used the ratio of the largest (ev1) and the second largest (ev2) eigenvalues to select the group to be split next. That is, if ev1/ev2 > T the subvolume is split regardless of the image error.

For Figure 3a, we have set the threshold T to 2.5, whereas for Figure 3f T was set to 1.5. As can be seen in the initial codebooks (see Figure 3b, g), the allocation of codevectors to sky can be controlled quite well. Other than reserving more codevectors to these specially shaped clusters, operation is not affected and thus most details are still preserved as in Figure 2f. “Codevector stealing” begins to become visible in Figure 3f, (see the letter “S” in Figure 3i). However, T=2.5 seems to be a good compromise, while T=1.5 appears to be overdone.

Further parameters to include in the split decision are the population count of a cluster, and the absolute spatial extent along the principal component.

It should be mentioned that the method is neutral in case there are no such areas in the image, since then a corresponding eigenvalue ratio will not occur. The opposite doesn’t hold, though. A high ratio doesn’t mean that there is a smooth transition, as in the case of the brown building, which nevertheless gets assigned more codevectors. Excluding these areas for true unsupervised compression is subject of future research.

3 RESULTS

To demonstrate the differences in image quality, we have chosen a set of images from http://www.imagecompression.info. Each image was compressed into its own codebook. We have used a block size of 8x8 pixels for a vector dimension of 192. The number of codevectors was chosen such that the compression rate was roughly the same for each image. The results are given in Table 1, in terms of PSNR [dB].

Table 1: Seeding comparison.

Img. Name	Rand	KM++	PCA	bpp
Artificial	31.8	35.7	37.9	2.2
Bridge	29	29.5	29.5	2.1
Cathedral	32.1	32.4	32.6	2.3
Deer	30.1	30.5	30.7	2.2
Fireworks	29.8	37.3	42.9	2.3
Hdr	35.7	38.5	39	2.2
Leaves	26.3	27	27.1	2.3

As test case for showing the potential of eigenvalue-based subdivision we have selected an image with the following properties:

- a large area with a smooth color transition to make quantization artefacts (banding) visible,
- a high amount of image detail with known shape such as traffic lights or street signs.

Original image size is 1024x768 pixels, or 12,288 blocks. Since the algorithms perform roughly the same on small codebooks, we have chosen a codebook of 1k codevectors to expose the differences. Processing times and image quality are summarized in Table 2.

Table 2: Compression time and image quality.

Method	Seeding [s]	Clustering [s]	PSNR [dB]
Random	0.016	37.5	24.17
KM++	10	35.6	24.99
PCA	39.3	36.5	25.07
PCA, T=2.5	67.6	36.8	25.03
PCA, T=1.5	69	36.8	24.85

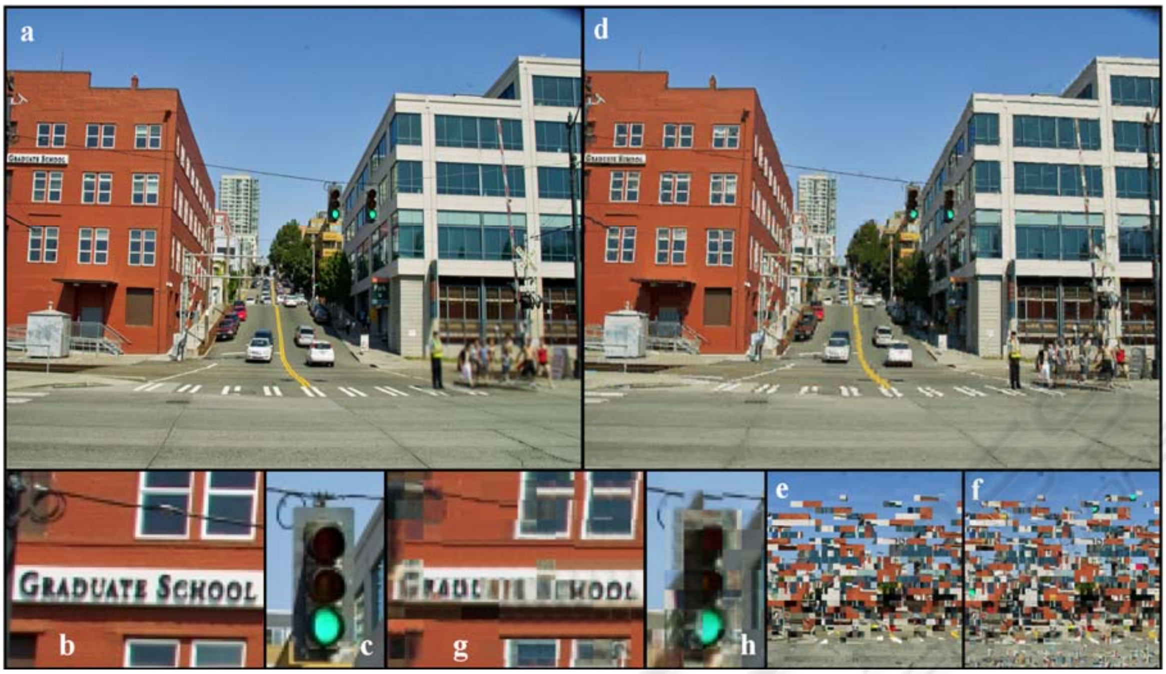

Figure 1: a) original photograph, resolution 1024x768; b) cut-out of size 110x80; c) cut-out 50x80; d) k-means clustering using random seeds; e) initial codebook (1k codevectors); f) final codebook; g) and h) cut-outs of decompressed image.

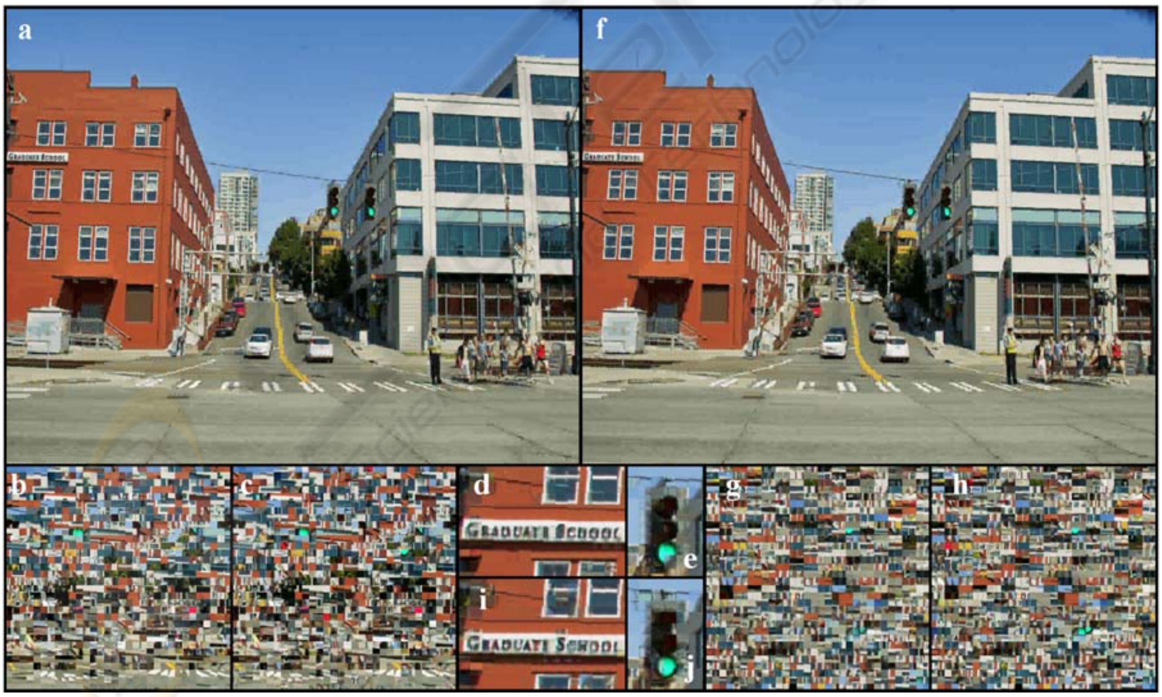

Figure 2: a) k-means++ seeding; b) initial codebook; c) final codebook; d) and e) cut-outs from a); f) PCA-based seeding; g) initial codebook; h) final codebook; i) and j) cut-outs from f).

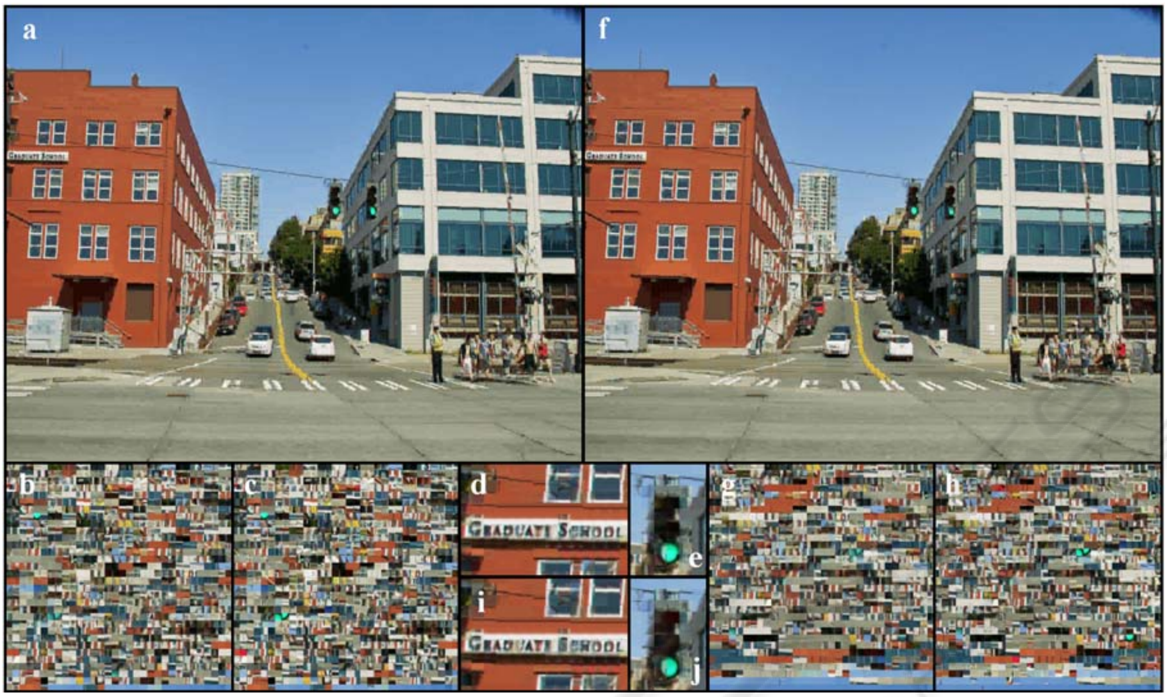

Figure 3: a) PCA-based seeding, T=2.5; b) initial codebook; c) final codebook; d) and e) cut-outs from a); f) PCA-based seeding, T=1.5; g) initial codebook; h) final codebook; i) and j) cut-outs from f).

4 CONCLUSIONS

We have presented an alternative to k-means++ seeding which performs equally well in terms of PSNR. It is based on Principal Component Analysis, and performs error-directed subdivision in eigenspace. Most notably, the method offers some global parameters to adjust compression priorities based on local image properties. These can be used to reduce quantization artefacts on smooth color transitions.

It might appear to be somewhat irrelevant to try to improve the appearance of such image areas, especially in the view of the image errors everywhere else. However, for other image material with less recognizable features like rocks or bushes, banding on such areas might become the dominant source of visual image degradation. Also, with careful use of the compression parameters the achievable image improvement is free, i.e., it does not increase the error significantly in other parts of the image.

REFERENCES

Arthur, D., Vassilvitskii, S., 2007. k-means++: the advantages of careful seeding. In *Proc. 18th annual ACM-SIAM symposium on discrete algorithms*, pages 1027-1035.

Barbakh, W., Fyfe, C., 2008. Clustering with alternative similarity functions. In *Proc. 7th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases*, pages 238-244.

Bradley, P. S., Fayyad, U., 1998. Refining initial points for K-means clustering. In *Proc. 15th Int. Conf. on Machine Learning*, pages 91–99.

Fritzke, B., 1997. The LBG-U method for vector quantization - an improvement over LBG inspired from neural networks. In *Neural Processing Letters*, Vol. 5, No. 1, pages 35-45.

Gray, R. M., 1984. Vector quantization. In *IEEE ASSP Magazine*, Vol. 1, No. 2, (1984), pages 4-29.

Gersho, A., Gray, R. M., 1992. *Vector quantization and signal compression*, Kluwer Academic Publishers.

Lloyd, S. P., 1982. Least squares quantization in PCM. In *IEEE Trans. on Information Theory*, Vol. 28, 1982, pages 129-137.

Ostrovsky, R., Rabani, Y., Schulman, L., Swamy, C., 2006. The Effectiveness of Lloyd-Type Methods for the k-Means Problem. In *Proc. 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 165-176.

Pena, J. M., Lozano, J. A., Larranaga, P., 1999. An empirical comparison of four initialization methods for the k-means algorithm. *Pattern Recognition Lett.*, Vol. 20, No. 10, pages 1027–1040.