Experiments and conclusions: The described algorithm has been tested on a great number of images of 8 bits depth. For each reconstructed image the relation between the peak signal/noise ratio (PSNR) and the bits-per-pixel ratio R has been derived. The PSNR has been evaluated as

$$\mathrm{PSNR} = 10\log_{10}\frac{255^2}{\mathrm{MSE}} \qquad (4)$$

where the MSE denotes the mean square error over the entity. All images are 256 × 256 and are divided into two sets. The first set includes a great number of synthetic images, all constructed as autoregressive Gaussian models with correlation factor ρ in the range (0.95, 0.99) In Fig. 2 a typical resultant PSNR/R relation curve is given together with the corresponding curve derived from

Table 1: Performance comparison between the presented and other known image encoding methods

Image 'Lena'					
Rate	QT-DPCM	QT-OBA	SQ	DPCM	ADPCM
0.67	31.9	30.4	26.0	29.4	30.9
1.00	33.0	32.6	27.1	31.4	32.5
2.00	37.0	37.2	30.7	35.4	36.6
Image 'Baboon'					
Rate	QT-DPCM	QT-OBA	SQ	DPCM	ADPCM
0.67	27.0	27.1	23.1	27.5	27.9
1.00	30.0	29.6	26.0	29.6	30.0
2.00	32.1	32.1	34.0	33.6	33.4

QT-DPCM: presented Algorithm; QT-OBA: for QT with 'optimal bit allocation coding technique'; SQ: scalar quantiser, DPCM: classical DPCM; ADPCM: adaptive DPCM

the application of the 'optimal bit allocation' method [3] on the same image. The second image set contains a series of well known standard images. The best results where found when the algorithm was applied on 'Lena' and the worst on 'Baboon'. In Table 1 results from these two cases together with results derived from several popular image encoding techniques described in [5] are given.

19 December 1995
Electronics Letters Online No: 19960352

S.A. Vergopoulos, M.S. Sangriotis and C. Metaxaki-Kossionides (*University of Athens, Department of Informatics, Panepistimiopolis, TYPA Buildings, 15784 Athens, Greece*)

References

1 SAMET, H.: 'The quadtree and related hierarchical data structures', *ACM Comput. Surv.*, 1984, **16**, pp. 187–260
2 DECOULON, F., and JOHNSEN, U.: 'Adaptive block schemes for source coding of black-and-white facsimile', *Electron. Lett.*, 1976, **12**, pp. 61–62
3 SHUSTERMAN, E., and FEDER, M.: 'Image compression via improved quadtree decomposition algorithms', *IEEE Trans. Image Process.*, 1994, **3**, pp. 207–215
4 JAIN, A.K.: 'Image data compression: A review', *Proc. IEEE*, 1981, **3**, pp. 349–389
5 BLAIN, E.M., and FISCHER, R.T.: 'A comparison of VQ techniques', *EURASIP Image Commun.*, 1991, **3**, (1)

Iterative inter-lattice interpolation

R. Manduchi

Indexing terms: Image coding, Image processing

The author introduces an iterative algorithm for interpolating an image when the sampling lattice is not uniform, but differs from block to block. This technique may find applications in adaptive sub-sampling for image coding.

Introduction: Adaptive sub-sampling for image coding has been discussed in [1]. The image is divided into blocks, and each block is assigned a sub-sampling lattice for encoding. The sub-sampling density in each block is chosen according to the spectral content of the image in the block. Only separable lattices have been considered in [1]. However, when coding image segments characterisd by oriented patterns and lines, it seems profitable to also use non-separable sub-sampling lattices [2]. An important problem arising from such an approach is the interpolation (by the decoder) to the original lattice of definition of the image. Indeed, standard techniques based on linear interpolation (filtering) assume that the sub-sampling grid is uniform (a lattice). We present a simple iterative algorithm for interpolating the image when the blocks are characterised by different sub-sampling lattices.

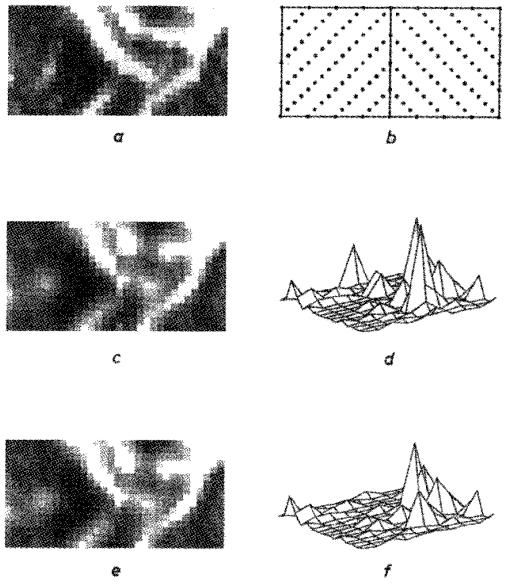

Fig. 1 *Sampled image*

a Original image (rows 395:410, columns 108:139 from 'Lena')
b Sampling lattices on the two adjacent sub-blocks
c Reconstruction using independent interpolation
d Corresponding reconstruction error surface on stripe (10 pixels wide) at junction of blocks
e Iterative algorithm
f Corresponding reconstruction error surface on stripe (10 pixels wide) at junction of blocks

Theory: Let β_1, β_2 be adjacent regions (blocks) of an image, which is sampled on the points of lattice Λ_1 in β_1, and of lattice Λ_2 in β_2 (see Fig. 1*b*). We would like to interpolate the image to a lattice Γ (with $\Gamma \supset \Lambda_1$, $\Gamma \supset \Lambda_2$). The standard procedure for interpolating from e.g. Λ_1 to Γ consists of adding null-valued samples on $\Gamma \backslash \Lambda_1$ (the set of points of Γ that do not belong to Λ_1), and then smoothing the resulting signal with a suitable filter $H_1(\mathbf{f})$ [3]. In the given case Λ_1 and Λ_2 may not coincide, and we need a different strategy. Natural requiremeuLs for a feasible interpolation procedure are: *i* the Nyquist property [3] (the original samples are preserved) and *ii* consistency: if all the blocks of the image are sub-sampled on e.g. Λ_1, the procedure should give the same result as the standard interpolation over the entire image with filter $H_1(\mathbf{f})$.

Our algorithm computes, at each iteration, an estimate of the interpolated values on Γ using standard interpolation within each block. Clearly, to interpolate the samples of e.g. $\Gamma \backslash \Lambda_1$ at the boundary of β_1, we would need a number of samples of the signal at points of Λ_1 belonging to β_2 (and to other adjacent blocks). But some or all of such values are missing if $\Lambda_1 \neq \Lambda_2$. In their place, we use the estimates computed in the previous iteration. The new interpolation thus refines the estimates on the points of Γ within β_1. The procedure is then iterated inverting the roles of β_1 and β_2.

We now state the algorithm assuming, for the sake of simplicity, that the image is composed of only contiguous blocks β_1 and β_2. Let $l(\mathbf{x})$ be the sub-sampled signal, and let $H_1(\mathbf{f})$ ($H_2(\mathbf{f})$) be suitable filters for the interpolation from Λ_1 (Λ_2) to Γ in the standard way. Finally, let $\hat{l}^{(0)}(\mathbf{x}) = 0$ for $\mathbf{x} \in \Gamma$, and let l_0 be the average brightness level. The algorithm starts with $i = 1$, $j = 2$, $n = 1$, and iterates through the following steps:

1. Set

$$q(\mathbf{x}) = \begin{cases} l(\mathbf{x}) & \mathbf{x} \in \Lambda_i \cap \mathbf{B}_i \\ \hat{l}^{(n-1)}(\mathbf{x}) & \mathbf{x} \in \Lambda_i \cap \mathbf{B}_j \\ l_0 & \text{everywhere else in } \Lambda_i \end{cases}$$

2. Interpolate $q(\mathbf{x})$ to Γ using filter $H_i(\mathbf{f})$, obtaining $\hat{l}^{(n)}(\mathbf{x})$. Our estimated interpolated signal at the n-th iteration is thus

$$\bar{l}^{(n)}(\mathbf{x}) = \begin{cases} \hat{l}^{(n)}(\mathbf{x}) & \mathbf{x} \in \Gamma \cap \mathbf{B}_i \\ \hat{l}^{(n-1)}(\mathbf{x}) & \mathbf{x} \in \Gamma \cap \mathbf{B}_j \end{cases}$$

3. Switch the values of i and j, and set $n = n + 1$.

The algorithm is terminated after a fixed number of iterations. It is easily seen that, if the Nyquist condition is satisfied by $H_1(\mathbf{f})$ and $H_2(\mathbf{f})$ (for Λ_1 and Λ_2, respectively), then both the Nyquist and the consistency properties are satisfied by our procedure. Note also that the algorithm can be put to work for any choice of the basic interpolation technique (e.g. non-linear filtering), and for any shape of the blocks. Hence, two adjacent blocks characterised by the same sub-sampling lattice may be merged into one super-block. The algorithm can be readily generalised for a case with more blocks. A scanning order of the blocks is chosen, and interpolation within each block is carried out using the estimates computed in the neighbouring blocks in the previous iteration.

Experiments: We have tested the interpolation algorithm on two contiguous blocks of the image 'Lena'. Large enough blocks were chosen to guarantee that the two 16×16 sub-blocks shown in Fig. 1 are not affected by boundary effects, but in the region joining them. The two blocks were sub-sampled on the oriented lattices shown in Fig. 1*b* (sub-sampling ratio = 4), and then interpolated using the iterative algorithm (terminated after two iterations per block). Oriented Nyquist filters have been used for interpolation, and the average level l_0 was set to 128. The reconstruction SNR, computed on the two sub-blocks, was 18.1 dB. The reconstructed image and the squared reconstruction error are shown in Fig. 1 *e* and *f*. For comparison, we have also considered a simpler technique: each block is interpolated separately, padding the outside region with samples set to l_0 In this case, we obtained SNR=17.6. The border effect at the junction of the blocks is more pronounced in this case as Fig. 1 *c* and *d* clearly show.

Conclusions: We have presented an algorithm to interpolate an image when the sub-sampling geometry differs from block to block. Research is in progress toward a complete coding scheme based on adaptive sub-sampling.

5 February 1996

Electronics Letters Online No: 19960387

R. Manduchi (*Stanford University, Computer Science Department, Robotics, Room 254, Gates Building 1A, Stanford, CA 94305-9010, USA*)

References

1 BELFOR, R.A.F., HESP, M.P.A., LAGENDIJK, R.L., and BIEMOND, J.: 'Spatially adaptive subsampling of image sequences', *IEEE Trans. Image Process.*, 1994, **3**, pp. 492–500

2 TAUBMAN, D., and ZAKHOR, A.: 'Orientation adaptive subband coding of images', *IEEE Trans. Image Process.*, 1994, **3**, pp. 421–437

3 DUBOIS, E.: 'The sampling and reconstruction of time-varying imagery with applications in video systems', *Proc. IEEE, Apr.*, 1985, **73**, pp. 502–522

Scale invariant texture classification by iterative morphological decomposition

W.K. Lam and C.K. Li

Indexing terms: Image processing, Texture, Segmentation

A new algorithm for scale invariant texture analysis is introduced. The method makes use of the iterative morphological decomposition to find a set of scale invariant parameters. Experiments show that very accurate result can be obtained for classifying eight natural textures.

Introduction: Texture analysis is desirable for incorporating scale invariance so that many applications such as remote sensing and robot navigation, as well as biomedical and industrial applications, do not need to strictly control the scale so that it is the same for both training and testing samples. A few techniques possessing scale invariant properties have been investigated e.g. moment invariants and Gaussian Markov random field. However, these methods do not in general give good results. Recently, Dougherty *et al.* and Wang *et al.* have analysed the textured image by mathematical morphology. Wang's iterative morphological decomposition (IMD) [2] was shown to be better than Dougherty's granulometric moments in texture classification [3]. Furthermore, IMD has an additional advantage over granulometric moments in terms of easy adaptation for scale invariance. In the following, we derive a scale invariant algorithm from IMD and carry out some experiments to evaluate its performance.

Scale invariant IMD: Many textures can be described as an ordered collection of similar primitive elements. Assume that such a uniform texture $f(x, y)$ is made up of repeatable elements $t(x, y)$ and can be modelled as

$$f(x,y) = \sum_{m,n}^{M,N} t(x - m\Delta x, y - n\Delta y) \qquad (1)$$

where Δx and Δy are the periods in x and y, respectively. The assumption ensures that at any location $(x/r, y/r)$ of scale r, there is always a corresponding $f(x, y)$ that can be mapped to. However, when applying IMD to $f(x, y)$, the texture is decomposed into a scale-dependent set of component images. Therefore $f(x/r, y/r)$ may not have the same component images as $f(x, y)$ if they are decomposed by the same set of structuring elements. To circumvent this deficiency, T-operators [1] may be chosen for IMD, i.e. invariant in horizontal and vertical translations. If $f(x/r, y/r)$ is under horizontal scaling only, i.e. no scaling in the gray scale, the scale of $f(x/r, y/r)$ can be estimated in accordance with a (T, H)-Euclidean granulometry whose properties have been discussed in [1].

A granulometry at λ can be found by scaling the function first then applying a standard structuring element to the scaled function. Henceforth, if there are two scaled textures, $f(x/a, y/a)$ and $f(x/b, y/b)$, their granulometries at λ are equal to

$$\begin{aligned} \Psi_\lambda(f)(x/a, y/a) &= \Psi_1(1/\lambda * f)(x/a, y/a) \\ &= \Psi_a(a/\lambda * f)(x, y) \end{aligned} \qquad (2)$$

$$\begin{aligned} \Psi_\lambda(f)(x/b, y/b) &= \Psi_1(1/\lambda * f)(x/b, y/b) \\ &= \Psi_b(b/\lambda * f)(x, y) \end{aligned} \qquad (3)$$

For every ψ_λ applying to $f(x/a, y/a)$, there is a corresponding ψ_λ applying to $f(x/b, y/b)$ that may lead eqn. 2 equal to eqn. 3. Therefore, we can choose a reference ψ_λ for any scaled $f(x/r, y/r)$.

Scale invariant IMD algorithm: Let $G = g_i$, $i = 1, 2, ..., L$ be an increasing sequence of structuring elements, where g_L is the largest element, applied to a textured image for finding its morphological granulometry. To obtain granulometric features invariant to scale change, the sequence should be scaled according to the scale of that image. It is estimated by finding a scaled g_L such that gray level mean of the Lth component image is equal to a predefined mean. The following lists the procedures of the algorithm.

Step 1: Choose a proper positive threshold value T, a maximum element size g_M, and set counter $c = 0$. Step 2: Choose a desirable