Detecting Scale-Sensitivity in Image Hierarchies for Coding and Compression

J. Alejandro Butrón Guillén[1] and Richard Harvey[2]

[1]Centro Nacional de Actualización Docente CNAD
Estanislao Ramírez s/n, col. Selene. México 13420, D.F.
[2]School of Computing Sciences, University of East Anglia
Norwich, NR4 7TJ, UK.
Contact: abutron@cnad.edu.mx, rwh@cmp.uea.ac.uk

Abstract. This paper examines human sensitivity to errors introduced by two types of lossy image coders. Our interest is a special type of morphological scale-space tree called a sieve because such trees are thought to be useful for image understanding and other purposes, but we also examine a conventional lossy coder: JPEG. The paper introduces a new way of measuring image quality, a type of Turing test and we show how the method can be normalized to compare different images and coders. We conclude that content of the image can have a significant effect on the perception of image quality.

1 Introduction

Motivated by MPEG-7 and MPEG-4, image and video analysis and compression based on connected-set mathematical morphology ([1] for example) have become topics of some interest. Lossy compression based on such techniques typically involves deleting small-scale regions and/or coding regions using an approximation to their true shape. A key property of many connected set methods is their hierarchical structure [1], [2], in which the image is represented as a tree with small-scale regions as the leaves and root representing the whole image. Such trees can become large, it is therefore of some interest to know how many of the leaves may be deleted from a tree without affect the quality perceived by a human observer. The sensitivity of observers to the small scale contained in images is thus investigated and reported. Images are filtered with the sieve, a type of hierarchical connected-set representation, and a subjective test devised that allows us to measure the effect of deleting regions from the hierarchy.

The rest of the paper is organized as follows. In Section 2, we review the sieve algorithm and describe the hierarchical structure for decomposing images into hierarchies of contours. Section 3 presents the experimental setup for detecting the sensitivity to contours, together with a discussion of the results, and in Section 4 we make some initial conclusions for the development of a compressor based on the sieve.

© E. V. Cuevas, M. A. Perez, D. Zaldivar, H. Sossa, R. Rojas (Eds.)
Special Issue in Electronics and Biomedical Informatics,
Computer Science and Informatics
Research in Computing Science 35, 2008, pp. 87-96

2 The sieve algorithm

The sieve performs a non-linear decomposition of an image by removing extrema of increasing scale, and thus making a simplification of the input image. It is a filter based on graph-morphology [3] that removes features of the image without introducing new extrema. The theory is well established [4], [5], [6], [2], and what follows is a summary of this theory and a description of the hierarchical representation.

An example of a progressive simplification performed by the sieve is shown in Figure 1. This simplification is obtained by using the sieve at different increasing scales. The sieve treats the input image as a graph G = (V,E), where V represents the set of vertices for each pixel contained in the image, and E the set of edges that describe the connectivity between those vertices.

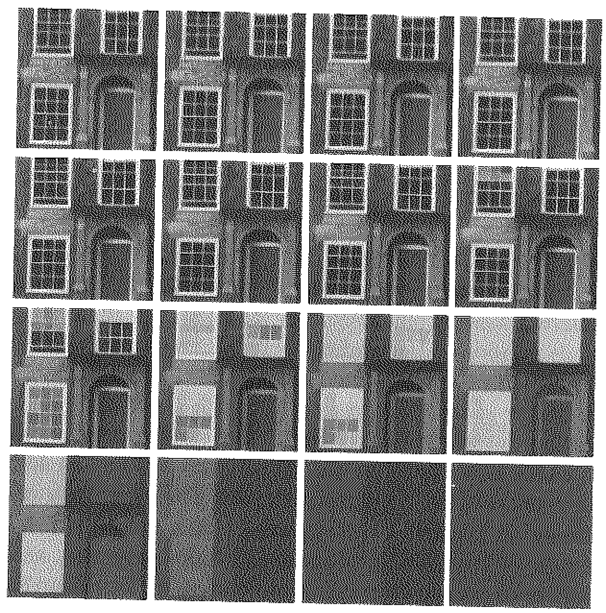
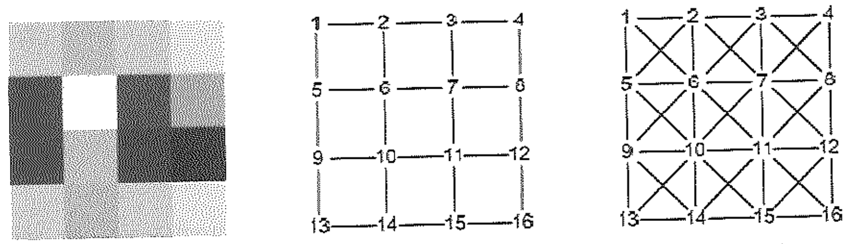


Fig. 1. Sieve decomposition of an original image (top left) processed at scales 2^n, with n = 0 ... 15 (shown left-to-right). Note that, unlike wavelets, sieves can be applied at arbitrary scales.

An example of this representation is given in Figure 2 for a greyscale image of 4 x 4 pixels, where the vertices (V) are:

$$V = \{1,2,3,\ldots,14,15,16\} \tag{1}$$



(a) (b) (c)

Fig. 2. Graph Representation: (a) Greyscale image. (b) Vertex labelling and the four-connectivity edges. (c) Vertex labelling and the eight-connectivity edges.

and for four-connectivity the set of edges E_4,

$$E_4 = \{\{1,2\},\{1,5\},...,\{12,16\},\{15,16\}\} \tag{2}$$

or using eight-connectivity the set of edges E_8,

$$E_8 = \{\{1,2\},\{1,5\},\{1,6\},...,\{11,16\},\{12,16\},\{15,16\}\} \tag{3}$$

where the image intensities define a function f, of the vertex number $f(v)$ with $v \in V$. Based on the connectivity between pixels of a given image, $C_r(G)$ denotes the set of connected subsets C of size r, within G,

$$C_r(G,v) = \{\xi \in C_r(G) \mid v \in \xi\} \tag{4}$$

ξ are the set of connected regions of size r that contain the vertex v. For example, for the image in Figure 2 the set of four-connected regions of size 2 for the vertex $v = 10$ are

$$C_2(G,10) = \{\{6,10\},\{9,10\},\{10,11\},\{10,14\}\} \tag{5}$$

The sieve removes extrema (maxima or minima) by applying a morphological operator φ_s over $C_r(G)$. φ may be a morphological opening γ_r, closing ψ_r, or the alternating sequential filters $\mathcal{M}$ and $\mathcal{N}$, defined on connected sets as an operator $\varphi_r : \mathbf{Z}^V \rightarrow \mathbf{Z}^V$ for each integer $r \geq 1$, as

$$\gamma_r f(x) = \max_{\xi \in C_r(G,x)} \min_{u \in \xi} f(u) \tag{6}$$

$$\psi_r f(x) = \min_{\xi \in C_r(G,x)} \max_{u \in \xi} f(u) \tag{7}$$

$$\mathcal{m}_r f(x) = \gamma_r(\psi_r(f(x))) \tag{8}$$

$$\mathcal{n}_r f(x) = \psi_r(\gamma_r(f(x))) \tag{9}$$

The sieve operates on an image by applying the operator φ from scale one and increasing sequentially until no new maxima or minima are found. The differences between successive stages of a sieve are called granule functions, and contain all the connected sets that have been removed from the image for each scale s.

2.1 Sieve-trees

The complete sequence of granules that are obtained at the different stages of the sieve can be represented in a hierarchical structure, called the sieve tree [2], [7]. An example decomposition is shown in Figure 3. The original image is shown on the top right and successive simplifications are shown down the right-hand side. The granules are shown on the left and the containment of one region within another is indicated by an edge of the tree. The head region is seen to contain the face region and four children are linked from this node -- two eyes, a mouth and a nose. Each eye has two children. This tree has a depth of five and so is sometimes described as having five *levels*. The highest level nodes are nodes 1 and 2 which are at level 5, nodes 3, 4, 5 and 6 are all at level 4, node 7 is at level 3, node 8 is at level 2 and the root, node 9, is at level 1.

The sieve tree represents a transformation of the image to the granule domain, and it is possible to recover the original image by parsing the nodes of the tree and merging all the regions, taking in consideration the intensity profile for each node.

3 Visual comparisons between images

A graphical user interface (GUI) displays a group of images in a controlled environment with constant lighting and CRT displaying, and it is the interface in which the observer answers a subjective question about the originality of each image, as shown in Figure 4. The test set of sixteen images were captured at a resolution of 2,464 x 1,648 pixels, 24 bits per pixel, under natural conditions, using a Canon EOS 1D camera and recorded in uncompressed form. These were then resized to 640 x 428 pixels, 24 bits per pixel using the bicubic downsampling in Matlab. Figure 5 shows this test set. The images were processed at scales 0, 10 and 100 with the sieve algorithm, and also saved at quality levels of 20, 60 and 100, using the JPEG compressor. A total of 96 images were obtained from the original set, each image was displayed twice in a random order, having a total of 192 displayed images to 20 observers, students with good correct vision, same images with a disagreement were reshown a third time for a later vote. The viewing distance was variable depending on the observer and the time for the experiment was not limited.

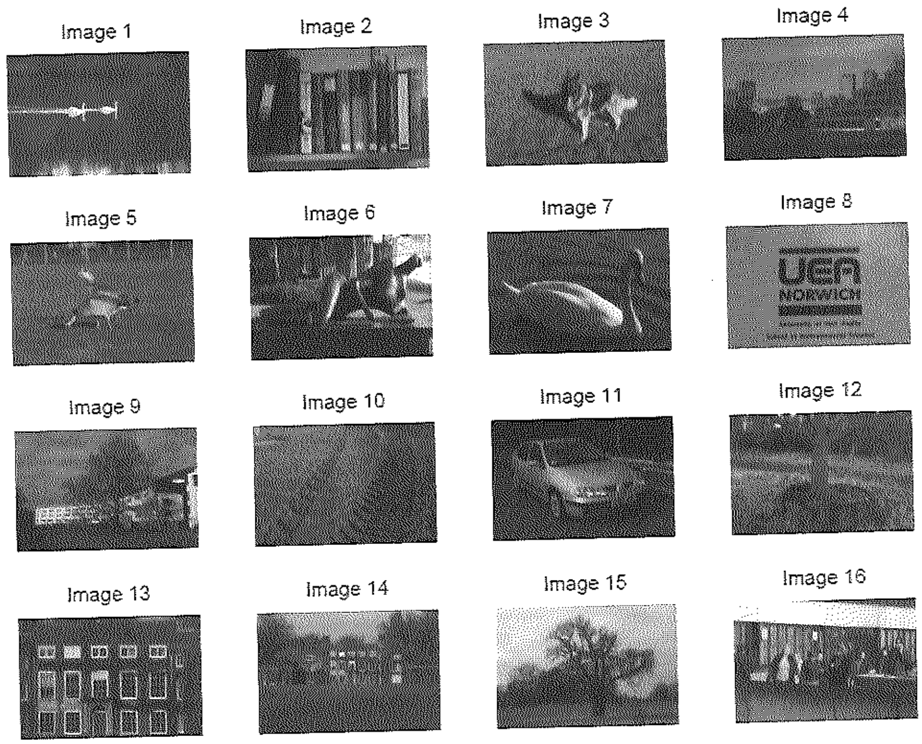


Fig. 5. Test set, taken under natural light conditions using a Canon EOS 1D digital camera, and recorded in uncompressed form. These were then resized to 640 x 428, 24 bits per pixel, using the downsampling in Matlab.

A decreasing score was obtained for all the images when sieved at scales 10 and 100, as shown in Figures 6(e, f), as the regions of small scale (and to which the eye is susceptible) were removed. As mentioned before, not all the images were accepted even when compressed at 100% with JPEG or sieved at area-scale of 0. This because, for comparing the acceptance of each image depending on their original score, the images compressed with JPEG and processed with the sieve algorithm were normalized with the value they obtained at this compressing value of 100% and area-scale of 0, for creating a "ratio of ratios" that considers the original score.

The results for the JPEG compressor are presented in Figure 7(a). The graph for JPEG 60% quality shows that some images improve against the originals at 100%. All the images decrease their acceptance when compressed with JPEG at 20% quality. Considering the same original scoring, Figure 7(b) shows the behaviour for the images when sieved at different scales. Only three images, 2, 8 and 7, have a scoring ratio of 0.8 or more at scale 10, the rest of the images are below 0.6. The graph for scale 100 shows the decreasing score for the images, arranged as in area-scale of 10.

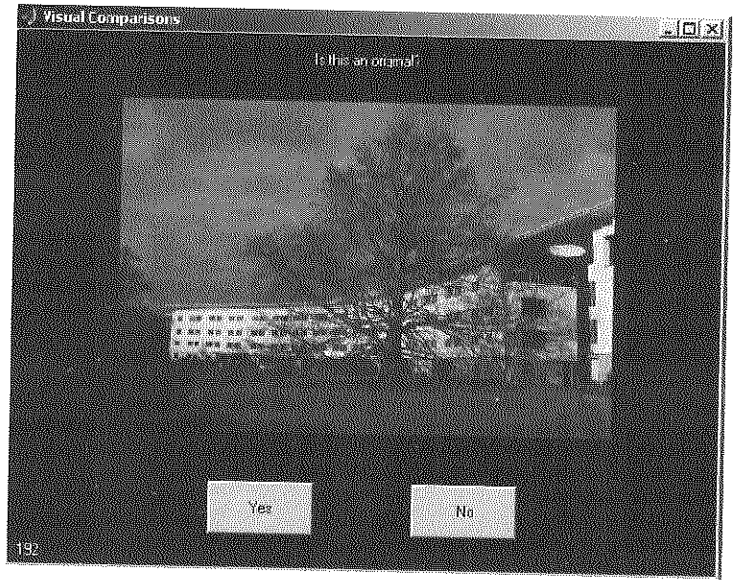


Fig. 4. Interface for a subjective test.

The acceptance of an image as an original depends not only in the compression quality or the scale at which they were sieved, but also in the contents of each image. Even when compressed at 100% quality with JPEG, not all the observers could identify them as originals. Figure 6(a) shows the fraction of observers that answered "yes" to the question "is this an original?" For image 1, 40% of the observers answered "yes" whereas for image 10, 95% answered "yes", illustrating the well known effect that the content of the image affects the visibility of compression errors. Only images 10 and 13 got a 95% acceptance. A similar behaviour is plotted in Figure 6(d), where the images were sieved at scale 0, and thus have the same quality level as when compressed at 100% with JPEG. The difference between Figure 6(a) and (d) is an indication of the variation in the experiment, since they are both acceptance ratios measured on identical sets.

Figure 6(b) (JPEG 60% quality) improves some scores and degrades others. Moving to Figure 6(c), all scores degrade with images 3, 10 and 13 degrading the least. A significant factor seems the presence of visual texture - JPEG artifacts are hard to spot in textured regions. Image 13 is slightly surprising, although we note it contains quite a lot of texture and the object boundaries align with the pixel grid, so are well represented by JPEG coding blocks.

The acceptance for all the images decreases in Figure 6(c), as the quality of the compressed file was of 20% only.

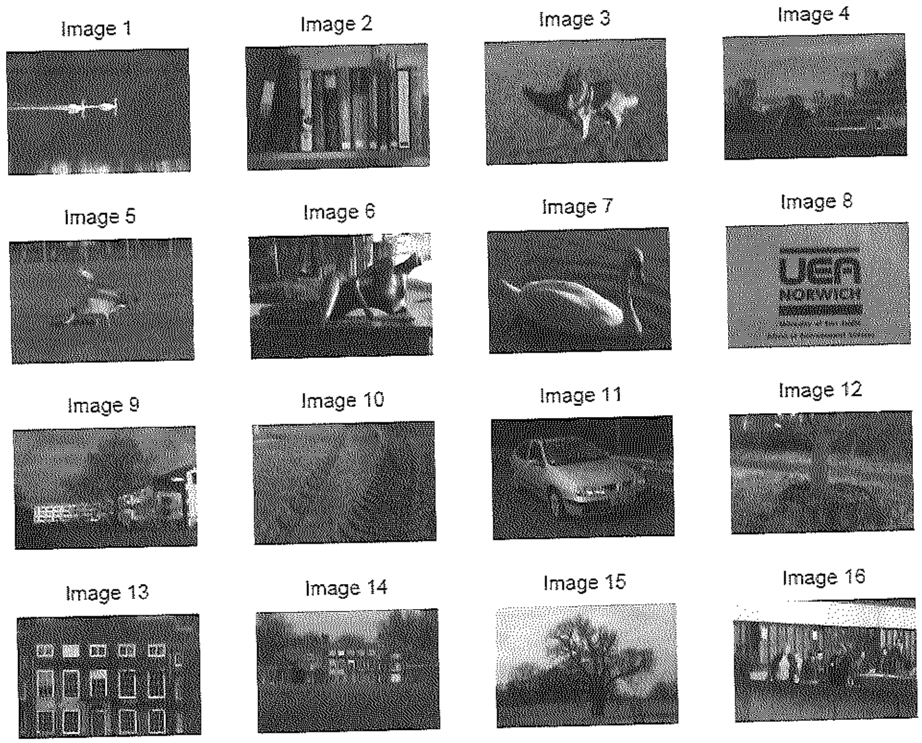
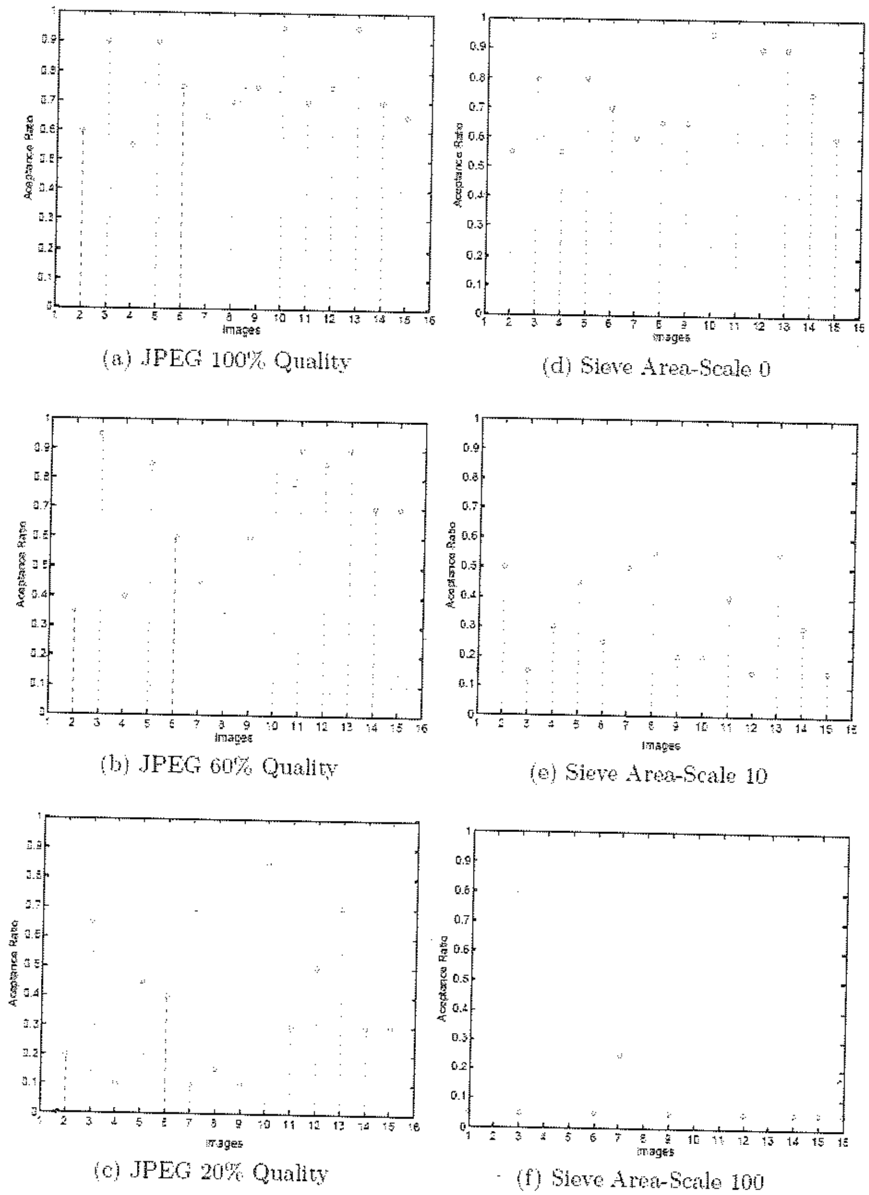


Fig. 5. Test set, taken under natural light conditions using a Canon EOS 1D digital camera, and recorded in uncompressed form. These were then resized to 640 x 428, 24 bits per pixel, using the downsampling in Matlab.

A decreasing score was obtained for all the images when sieved at scales 10 and 100, as shown in Figures 6(e, f), as the regions of small scale (and to which the eye is susceptible) were removed. As mentioned before, not all the images were accepted even when compressed at 100% with JPEG or sieved at area-scale of 0. This because, for comparing the acceptance of each image depending on their original score, the images compressed with JPEG and processed with the sieve algorithm were normalized with the value they obtained at this compressing value of 100% and area-scale of 0, for creating a "ratio of ratios" that considers the original score.

The results for the JPEG compressor are presented in Figure 7(a). The graph for JPEG 60% quality shows that some images improve against the originals at 100%. All the images decrease their acceptance when compressed with JPEG at 20% quality. Considering the same original scoring, Figure 7(b) shows the behaviour for the images when sieved at different scales. Only three images, 2, 8 and 7, have a scoring ratio of 0.8 or more at scale 10, the rest of the images are below 0.6. The graph for scale 100 shows the decreasing score for the images, arranged as in area-scale of 10.



(a) JPEG 100% Quality

(d) Sieve Area-Scale 0

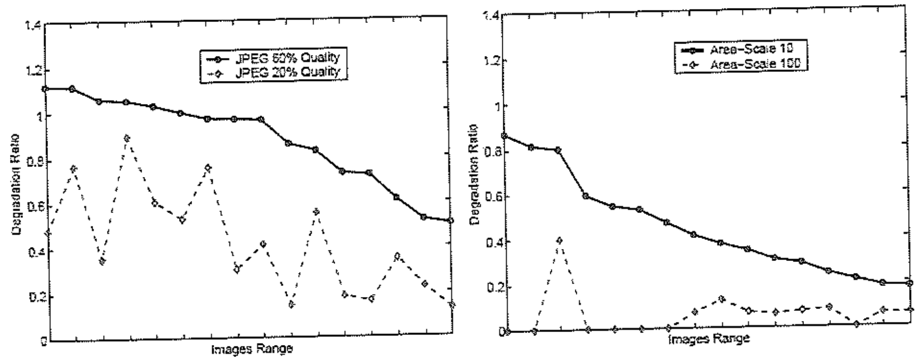
(b) JPEG 60% Quality

(e) Sieve Area-Scale 10

(c) JPEG 20% Quality

(f) Sieve Area-Scale 100

Fig. 6. Results of the subjective experiment. The x-axis is a number that identifies each of the images in the test set. The y-axis is the acceptance scores for each image when compressed at different quality levels and sieved at different area-scales.



(a) JPEG compressed images (b) Sieved images

Fig. 7. Acceptance ratio of ratios for each normalized image with JPEG 100% quality and sieved at 0 area-scale. The images are arranged from higher score to lower. The JPEG range is 15,3,11,10,12,5,13,16,14,9,6,4,7,2,8,1 considering the 60% rate, and the sieve range is 2,8,7,13,4,5,11,14,1,6,16,9,15,10,12,3 considering the area-scale of 10.

4 Conclusion and directions for further research

The results of this experiment represent an indication of the susceptibility of the human eye to the small scale contained in an image. The images that were sieved at scale ten were easily detected and labeled as non-original by the observers. We can conclude that the observers were highly likely to detect the loss of even quite small scale regions. So, even if lossy compression is used, there may be a need to code small-scale regions, when using the sieve as the core of a compressor.

Future work will compare additional categories, like the CCIR/ITU-R 500 5-Point Scale, for improving the quality assessments. Also, it will be investigated how to integrate the "ratio of ratios" with other normalized scores, like Z-scores.

References

1. P. Salembier, A. Oliveras, and L. Garrido (1998). Anti-extensive connected operators for image and sequence processing. IEEE Transactions on Image Processing, 7(4):555-570.
2. J.A. Bangham, J. R. Hidalgo, and R. Harvey (1998). Robust morphological scale-space trees. In Proceedings of Noblesse Workshop on Non-linear Model-Based Image Analysis, pp. 133-139.

3. J. A. Bangham, R. Harvey, Paul D. Ling, and Aldridge R. V. (1996). Morphological scale-space preserving transforms in many dimensions. Journal of Electronic Imaging, 5(3):283-299.
4. J. A. Bangham, P. D. Ling, and R. Harvey (1996). Scale-space from nonlinear filters. IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(5):520-528.
5. J. A. Bangham, P. Chardaire, C. J. Pye, and P. D. Ling (1996). Multiscale nonlinear decomposition: the sieve decomposition theorem. IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(5):529-539.
6. R. Harvey, J. A. Bangham, and A. Bosson (1997). Scale-space filters and their robustness. In Springer, editor, Scale-Space Theory in Computer Vision, Lecture Notes in Computer Science, pp. 341-344.
7. J. A. Bangham, J. Ruiz Hidalgo, and G. Cawley (1998). The segmentation of images via scale-space trees. In Proceedings of the 9th British Machine Vision Conference, pp. 33-43.