The segmentation of images via scale-space trees

J. Andrew Bangham, Javier Ruiz Hidalgo,
Richard Harvey and Gavin Cawley
School of Information Systems, Norwich, NR4 7TJ, UK.
Email: {ab,jrh,rwh,gcc}@sys.uea.ac.uk

Abstract

A useful representation of an image would be an object tree in which nodes represent objects, or parts of objects, and which includes at least one node that, together with its children, represents each object: a *grandmother* node. It is shown that scale-trees, obtained from greyscale images, approximate such a tree. It is then shown how they may be modified using other attributes to more closely become object trees. The result is a data structure that provides "handles" for every element of the image that can be used for manipulating the image. This segmentation has potential for object recognition.

1 Introduction

One goal for a computer vision system is to extract, from an image, a simple description in terms of meaningful objects of interest, their positions and geometric relations to one another. Segmentation is the grouping of those picture elements (pels [1]) associated with each object into a cluster. Once objects have been segmented from the background and each other and after occluding and occluded objects have been distinguished, we are in a powerful position to edit and understand the image in terms of semantically meaningful segments, shapes and other clues. The result is also an apposite starting point for implementing a coder, for the proposed MPEG 4 standard, in which an audio visual scene may be understood as a composition of primary audio visual objects, according to a script that describes their spatial and temporal relationships [1].

Many schemes for clustering pels into segments that are likely to represent objects or parts of objects have emerged including: split and merge, sematics and region growing [2,3]. More recently [4], the image has been represented as a fully connected graph, $G = (V,E)$, where the set of vertices, V, represent pels with attributes, A, that might include inter-pel spatial distance, greyscale value, hue and others. E is the set of edges and represents the connectivity between pels. Each connection edge has a strength W. In this framework the segmentation task is one of finding a partition of the graph such that similarities within subgraphs are high and similarities between subgraphs are low. The computational load in this method lies mainly in finding the eigenvectors of an $n \times n$ matrix, an $O(n^3)$ process. However, as only a few eigenvectors are needed with a low precision this may be significantly reduced [4]. With the appropriate choice of W it is possible to cluster pels representing certain objects and so segment them from the rest.

In this paper a graph approach is also used but the idea is to try and represent the image using an acyclic object-tree. In an object-tree, T_O, objects, or parts of objects, are

[1] We use the term pel to denote pixels with additional features such as motion, depth and so on.

represented by nodes and at least one node will, with its children, completely represent each object. That node would be a "grandmother node". For example, one grandmother node might represent a face and it would have children that represent smaller scale objects that lie within it. Some of these nodes may themselves be grandmother nodes, representing the mouth, eyes, etc. Here we address the problem of creating an object tree. The next step of snapping off the branches at the grandmother nodes, to allow objects to be precisely segmented, will be addressed elsewhere.

The new method starts with an order complexity, $O(n \log n)$, process that re-maps G to an acyclic graph, or scale-tree, T. Nodes represent sets of pels that are clustered using a nearest neighbours in a combination of spatial distance and greyscale value. The clusters are seeded on regional extrema for it happens that these are often associated objects, as postulated by Witkin [5] (see section 3). The result is a scale-tree with branches that tend to be associated with objects. However, this is only a first approximation to the final goal, T_O and so T is refined using further evidence from A, the attributes.

Consider a heuristic method for achieving this by generating a graph where the vertices represent the granules and the edges represent an approximation to the probability that granules form part of a larger object. We might start with a uniform prior that assumes that that the probability that any pair of granules comprising the image are component parts of a larger object, i.e. the probability that granules s_a and s_b belong to the same object is a small, non-zero, constant. $P(O_{s_a} = O_{s_b}) = C$ Where O_s is a cardinal number denoting the object of which granule s is a component part. Our degree of belief that the granules belong to the same object is then be updated, using Bayes' rule, according to some attribute A of granules s_a and s_b, such as the colour, saturation, greyscale value, motion vector, disparity, measures of shape, etc.

$$P(O_{s_a} = O_{s_b} | A_{s_a}, A_{s_b}) = \frac{P(A_{s_a}, A_{s_b} | O_{s_a} = O_{s_b}) P(O_{s_a} = O_{s_b})}{P(A_{s_a}, A_{s_b})} \tag{1}$$

Note that the quantities on the right hand side of this equation can all be estimated relatively easily. However, this operation then needs to be performed for every pair of granules in the hierarchy and over all attributes of interest yielding a computationally very expensive process, even for images of a modest size. The enormous search space needs to be shattered by attempting to pick pairs of granules where the application of Bayes' rule is likely to result in the largest modification of the *a priori* probability. A heuristic for achieving this is now discussed.

Consider just three sub-segment attributes, x, y (position) and v (greyscale value). Simple thresholding, widely exploited in machine vision applications, effectively sets $P(A_{s_a}, A_{s_b} | O_{s_a} = O_{s_b}) = f(D_{a,b})$ where $D_{a,b}$ is a distance measure such as $D_{a,b} = h(k((x_a - x_b)^2 + (y_a - y_b)^2) + (v_a - v_b)^2)$, k is a weighting such that nearest neighbour sub-segments of any greyscale value will always be closer than sub-segments that are separated by more than one pel, and h is a step function. The position of the step, in h, has to be separately determined, often manually. The result of thresholding is a set of disjoint clusters where each is expected to represent an object and the algorithm is a nearest neighbour clustering algorithm. A more common view of this clustering follows.

2 Scale-space clustering

The method used to obtain the clusters in this work is referred to as a connected set sieve. Like the well known diffusion based filters [5–8] connected set sieves, or alternating sequential filters, perform a decomposition by scale whilst preserving scale-space causality [9,10]: an essential property since we demand that nodes of the tree represent features in the original image. These processors transform the image to the granularity domain. Each node of the tree represents a granule. The transformation is invertible [11] which means that the image may be rebuilt from the tree nodes and, if nodes are deleted, that the resultant, simpler, tree will be identical to the tree obtained from an image in which the corresponding objects have been deleted. There are, therefore, three characteristics of the connected set (area) sieve that make it suitable for segmentation: (i) it has no structuring element or window and so does not change the shape of features and preserves edges, (ii) the shapes it preserves, represented by the nodes, are robust to the effects of noise and occlusion [12], (iii) images can be rebuilt from edited trees.

The connected set sieve is also defined as operating over a graph [13] so, in principle, operates on images defined in any finite number of dimensions. As in the earlier description the graph is denoted $G = (V,E)$. Defining $C_r(G)$ as the set of connected subsets of G with r elements allows the definition of $C_r(G,x)$ as those elements of $C_r(G)$ that contain x, $C_r(G,x) = \{\xi \in C_r(G) | x \in \xi\}$. Morphological openings and closings, over a graph, may be defined as

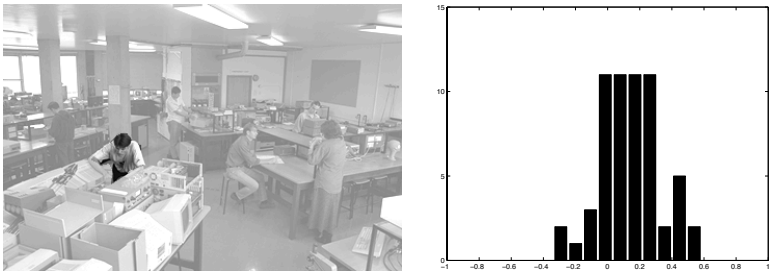
$$\psi_r f(x) = \max_{\xi \in C_r(G,x)} \min_{u \in \xi} f(u) \qquad \gamma_r f(x) = \min_{\xi \in C_r(G,x)} \max_{u \in \xi} f(u) \tag{2}$$

The effect of an opening of size one, ψ_2, is to remove all *maxima* of area one when working in two dimensions. Applying ψ_3 to $\psi_2 f(x)$ will now remove all maxima of area two and so on. The M and N operators are defined as $M^r = \gamma_r \psi_r$ and $N^r = \psi_r \gamma_r$ and hence remove extrema. Sieves, and filters in their class such as alternating sequential filters with flat structuring elements, depend on repeated application of such operators at increasing scale. The output at scale r is denoted by $f_r(x)$ with $f_1 = Q^1 f = f$ and $f_{r+1} = Q^{r+1} f_r$ where Q is one of the γ, ψ, M or N operators. Here, the M operator is used. The differences between successive stages of a sieve, called *granule functions*, $d_r = f_r - f_{r+1}$, contain non-zero regions, called *granules*, g_r's of only that scale. Each g_r is a connected set of r pels and forms a node in the scale-tree. Illustrations of sieves and formal proofs of their properties appear elsewhere [10]. In practice a tree may be generated in less than a second on a Pentium PC.

A scale-tree, T, is formed from the small scale leaves down. Small scale granules that merge become the children of the larger parent of which they are subsets. Each node has attributes, A. Further attributes such as motion and stereo disparity can be obtained by analysing multiple images. However, since the scale-space processors identify extrema it is appropriate to obtain evidence that clusters seeded from extrema are associated with objects in a variety of images.

3 Should clusters be seeded using regional extrema?

Volunteers were asked to draw around, what they chose to identify as, objects. Figure 1A shows an example. If this region includes regional extrema it will be associated with



A B

Figure 1: A: a photograph with a segment that has been segmented manually (highlighted for the illustration). B: a histogram that represents the proportion of the manually selected object that is represented by scale-tree branches. The abscissa, centred on zero, is a difference of two ratios where positive values reflect objects that form branches. The ordinate is the number of observations.

whole branches of the scale-tree. One way to quantify this is to find the fraction of its area that is locally more extreme than the region outside, r_o. If the region is entirely represented by complete branches, then this fraction would be 1. The fraction is then compared with a control segment obtained by randomly translating the region shape to another position in the image and again finding the ratio, r_c. Figure 1B shows the distribution of $r_o - r_c$ obtained from 60 objects selected by 5 people from 6 images. The majority of differences are positive, showing that the manually segmented objects are more often associated with extrema than random segments. This supports the view that scale-trees, obtained from sieves, are likely to be useful for representing objects in a wide variety of images.

4 Towards scale-trees

Consider the stylised image (drawn using a 3D drawing package) shown in Figure 2. It can be represented using a hierarchy of granules as shown in the associated tree, D. The root node represents the image as a whole. Within this are a number of objects that include two pieces of paper and a table, on which are further pieces of paper. Each object is associated with at least one node and therefore each node of the trees shown in Figure 2 represents an object, or part of an object. We note that, unlike a scale-space tree obtained by linear filtering [5], there is little movement of the apparent position of objects as scale increases and it is easy to find the boundary of each node (granule).

Such a scale-tree represents a considerable abstraction of the original image for it is invariant under simple geometric transformations that preserve the topology of the original image. This is illustrated in the remaining panels of Figures 2. In each case distortion of the original image (either by altering the viewing angle, B, or moving a piece of paper, C causes the (x,y) co-ordinates of the nodes to change but the tree topology is invariant. This is an important characteristic of this approach which could be exploited for tracking and recognition purposes.

It can be seen here that the scale-tree T is identical to the object tree T_O. For example, the grandmother nodes associated with the objects in Figure 2A, B and C are easily identified in Figure 2D, E and F. The rest of the paper is concerned with the process of

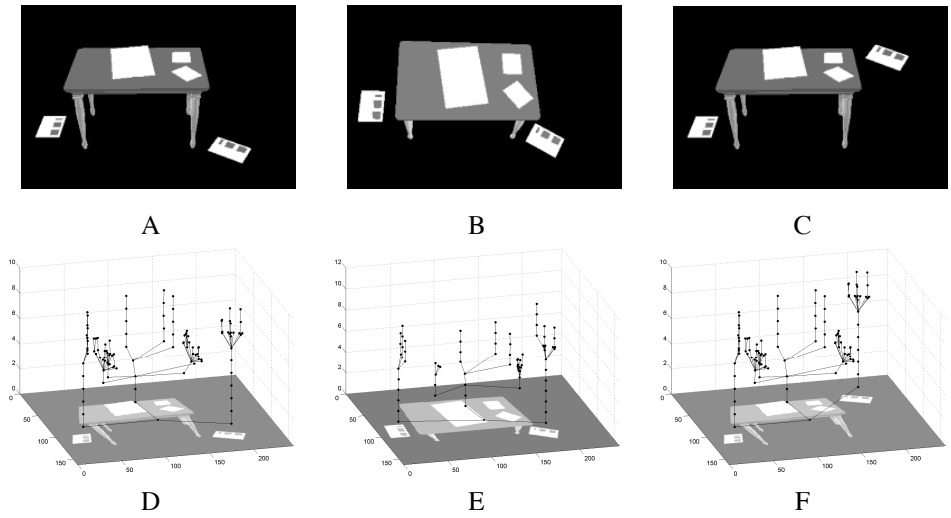


Figure 2: Images with associated scale-trees underneath. A: the original image. B: viewed from a different angle. C: one piece of paper moved. The scale-trees, D, E, F, have nodes with the co-ordinates given by the centroid of each granule and tree depth.

converting scale to object trees. To start with, the utility of some simple rules are examined although, in the future, these will have to be based on probability. There are two ways of reducing the number of nodes: collapsing long unbranched chains and pruning irrelevant children. Figure 3A, shows the effect of blurring the image (using a Gaussian filter). It produces long, unbranched, chains Figure 3D, and leaves the overall structure intact, since the filter preserves scale-space causality. A simple rule that collapses such chains by selecting the mid-node and deleting the remainder is all that is needed in this case. Figure 3B, E shows the result. It sharpens the image and simplifies the tree. It may be noted that the original image is itself slightly blurred, due to dithering during the rendering process, and this too can be simplified, Figure 3C and F and Figure 3B and C are very similar.

The number of branches of the scale-tree increases markedly when objects are textured. Figure 4A shows the result of adding "wood-grain" texture to the table top. However, most of the new nodes contrast little with their parents and so, again, a simple rule that deletes all low contrast nodes serves to prune the tree effectively. The result, shown in Figure 4B, E may be compared with a similarly pruned version of the original, Figure 4C,F. Of course, the original texture may be retrieved, if needed, by referring to the original image and a probability based decision one which nodes to prune will eventually be required.

5 Parsing real trees

So far, only synthetic images have been presented. Figure 5A shows a small, simple, image – a pair of snooker balls. Nevertheless it is associated with a complicated tree that should, if possible, be simplified. The first step is to concatenate long chains of, non-

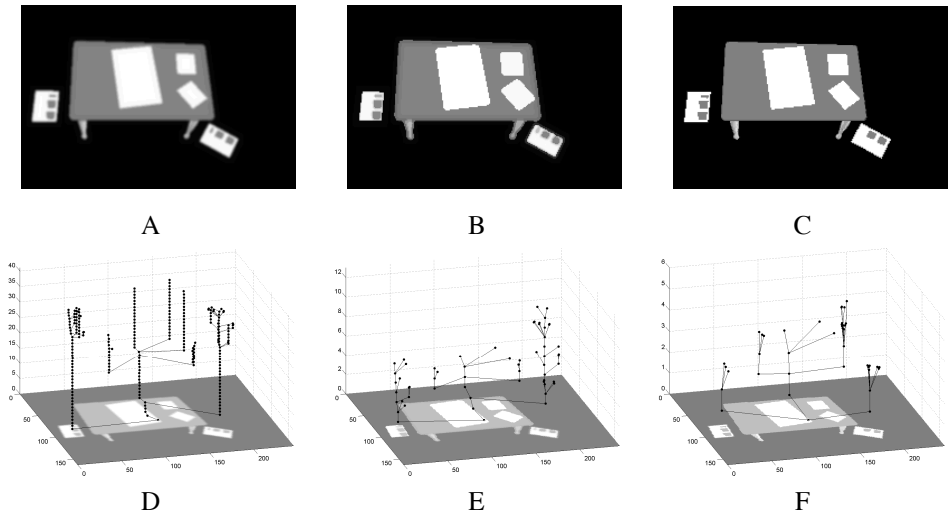


Figure 3: A: the image after Gaussian smoothing and, D, associated scale-tree. B and E after collapsing the tree. C and F the original image after a similar operation.

branching, low contrast nodes into a single node, and the second is to prune low contrast detail.

First the collapsing process. For each branch the sequence $g_{s_i}, i = 1 \ldots N$ represents the granule amplitude at each node where N is the number of nodes in that branch. Each node has scale s_i. Figure 5D plots the rate of change of granule intensity $\Delta_i = g_{s_i}/(s_i - s_{i-1}), i > 2$ versus the index, i. i is the tree depth measured down the branch. The peak in Δ_i is, for an object, the node at which its rate of change of intensity with respect to scale is maximized. For a sharp shape of area s the sequence of Δ_i is all zero except for one value at its true scale, s. For blurred images, such as shown in Figure 5A, the chains are collapsed onto the node with the maximum Δ_i. Figure 5B shows the image and its tree after this operation. Granules have been removed, the tree is simplified and the objects have sharp edges. The next step is to prune the tree. Figure 5C shows the image after the tree has been pruned by removing all nodes with an amplitude that differs from their parent by less than two units. This heuristic uses a hard decision but it would be desirable to replace it with a more principled step. The two rules for simplifying trees are now applied to a whole images.

Figure 6 top, and middle rows shows a sequence and its associated simplified trees. The scale-tree, which has been collapsed and pruned, does not change by much as its associated object changes scale, rotation, etc. This illustrates that the scale-tree is, in practice, somewhat invariant to scale, rotation and minor shape changes and that the rules developed so far can usefully simply images. In this case the sequence size is reduced from 120 Kb to 12 Kb.

In another application it might be necessary to segment the image. For example, Figure 7A shows an image of two faces. (B) Shows a node that has been identified manually as approximating a grandmother node, with the help of an image editor. (Methods for automatically recognising grandmother nodes, and therefore objects, will be considered

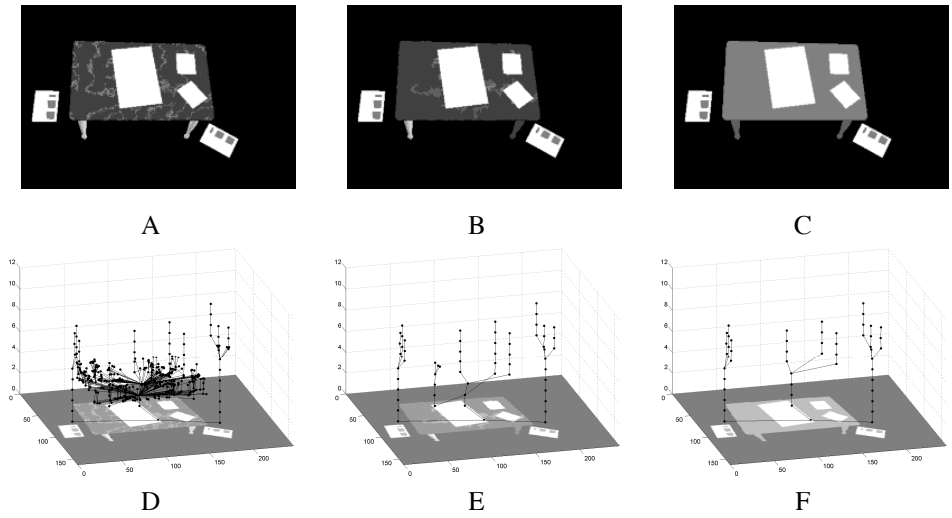


Figure 4: A and D textured image. B and E after pruning. C and F original image after pruning.

in another paper.) Having segmented the face it is an easy matter to change the spatial coordinate attributes of the grandmother node and so move the whole face. In this case the scale-tree itself is a good approximation to the desired object tree, however, it will often be necessary to bring in other evidence and modify the T to make it better approximate T_O. In the next example, new information is used to prune the tree.

Figure 8 (top row) shows frames from a game of tennis. The second row shows the result of pruning the scale-tree of stationary objects. In this motion filtering example the full tree is used without the collapsing or pruning step previously described. The moving objects have been extracted from the sequence as follows. Initially each node is visited, starting at the root, and the flat-zone associated with each node is translated around the equivalent region of the next image to find the minimum absolute difference. This reflects the motion vector. The motion vector for a parent node tends to be a good estimate for the motion vector for its children and so the search is straightforward. The results illustrate how information, beyond greyscale value and position, might be used to modify the tree. The problem is that it has only simplified the scale-tree, it would be better if it converted it to a full object-tree based description of the original image. In other words, rather than deleting nodes, they must be re-linked to form objects.

The real goal is to modify the tree without removing nodes and so change the nature of the scale-tree and make it closer to an object-tree. New branches will have to be formed by re-allocating nodes from the scale-tree. Figure 9A shows a red teapot against a beige background. Although it forms a local extremum, thresholding does not satisfactorily segment the image either when applied to the greyscale value or saturation, Figure 9B. (Beige contains enough red to make hue an unsatisfactory feature.) The hole in the handle node is part of the teapot tree, Figure 9C. The problem is to re-assign that node (and its children) to the background so as to make the node that currently generates the silhouette in Figure 9C solely represent the teapot, i.e. make it a grandmother node. This requires

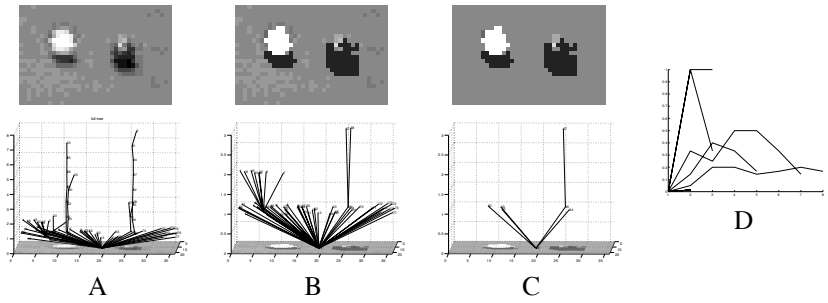


Figure 5: A: 4-bit greyscale excerpt from a snooker sequence and its associated scale tree. B: after collapsing long unbranched chains using a search over scale-space. C: after pruning low contrast nodes. D: granule amplitude as a function of scale.

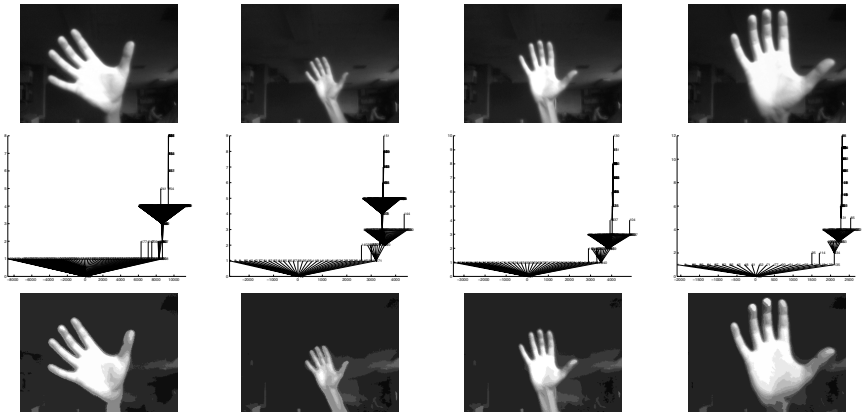


Figure 6: Top row, some frames from a 8 bit greyscale movie sequence. The second row shows collapsed and pruned scale trees side on. The third row shows the images corresponding to the reduced trees.

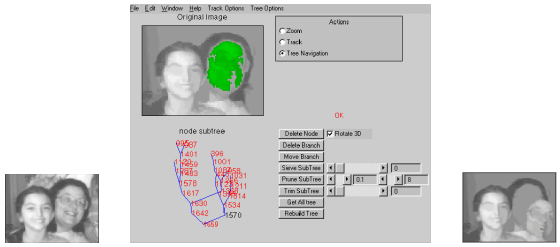


Figure 7: Editing using a grandmother node. A: the original image. B: a screenshot of our tree based image editor, T has been pruned and a node that approximates the grandmother node, N_g, and associated segment of the righthand face object are highlighted. C: the effect of changing the spatial position elements of N_g.

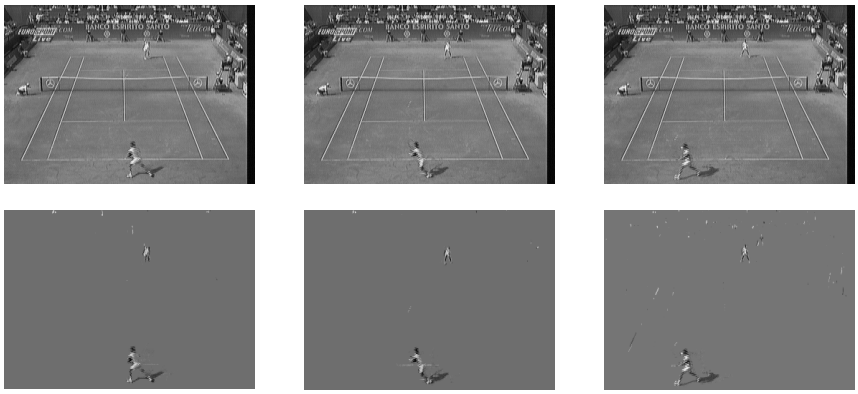


Figure 8: Top row: frames from a 8 bit greyscale movie. Second row: the same frames from which the moving players have been extracted with the help of a scale tree.

A B C D

Figure 9: A: the original teapot (it is coloured red). B: the saturation (of hsv). C: a silhouette obtained from the scale-tree representing the teapot. D: the silhouette obtained from the grandmother node that represents the teapot.

extra evidence from, for example, the saturation.

A list of quantised saturation values is built from the scale-tree nodes. Two-way pointers, from the scale-tree to the appropriate point in the saturation list, are created. The saturation values are then clustered and any node in the scale-tree that differs significantly from its parent is re-linked to a more similar node. In this case the difference is measured as the distance from the nearest cluster in the saturation domain, however, in principle the distance could be measured in any feature space. The result is an overall tree structure that now contains a grandmother node, that only represents the teapot. The outline it represents is shown in Figure 9D.

6 Discussion

In this tree representation of an image each node is a granule and has, at some stage, been associated with an extremum and so is bounded by an edge. Each node has a set of attributes including granule amplitude, shape (coded using a combination of a bit map and pointers) and position. The x, y and scale attributes are likely to be robustly estimated in the presence of noise and clutter [12]. The branching structure of the tree is determined by the way features lie within each other. This scale tree can represent an image in a manner that is consistent with an object tree, where the objects fit the definition required

of a primary visual object sitting on a plane as envisaged in MPEG 4.

The unaltered tree can be used for motion filtering of an image. Furthermore the concept of objects residing at a particular scale, in scale-space, can be used to reduce the extent of the tree and the tree may be automatically pruned to leave just those features that would, in a pattern recognition sense, be considered objects of interest.

Although nodes in this new scale-tree do represent some, perhaps many, of the objects in the image it is not expected that it accurately identifies all of them. However we think the tree is a good first approximation to the correct object tree that can be obtained from a single image. Evidence has been presented showing that the structure can be modified in the light of further evidence. For example, having separated out moving objects it might be appropriate to form a single scale-space branch from all nodes that are moving together. Likewise a stereo pair of images would allow nodes to be assigned to the same object using disparity information: one would expect nodes representing a single object to lie on the same visual object plane. In other words, further images provide more information that allows the tree to be re-arranged and better approximate an object tree.

Once the scale-tree is as close to an object tree as possible, and in the examples used here the two are already almost identical, then the tree will be a very useful representation of the image. Not only can it be used for filtering, but it can also be used for object recognition. This can be done at two levels. (1) The tree structure itself codes object topology that is, to a large extent, independent of geometrical scaling, rotations and distortions and (2) a more detailed matching can be performed by also using attributes of the nodes.

References

[1] F. Pereira. MPEG 4: a new challenge for the representation of audio-visual information. In *Proc. Picture Coding Symposium 1996*, 1996.

[2] S.C. Zhu, A. Yuille, and T.S. Lee. Region competition: Unifying snakes, region growing, and bayes/mdl for multi-band image segmentation. In *ICCV95*, pages 416–423, 1995.

[3] S.C. Zhu and A. Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *PAMI*, 18(9):884–900, September 1996.

[4] J. Shi and J. Malik. Self inducing relational distance and its application to image segmentation. In *Proceedings of the 5th European conference on computer vision*, pages 528–543, June 1998.

[5] A. P. Witkin. Scale-space filtering. In *8th Int. Joint Conf. Artificial Intelligence*, pages 1019–1022. IEEE, 1983.

[6] J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.

[7] J. Babaud, A. P. Witkin, M. Baudin, and R. O. Duda. Uniquenss of the gaussian kernal for scale-space filtering. *IEEE Transactions on Patern Analysis and Machine Intelligence*, 8:26–33, 1986.

[8] P.Perona and J.Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Patt. Anal. Mach. Intell.*, 12(7):629–639, July 1990.

[9] J.A. Bangham, P.D. Ling, and R. Harvey. Nonlinear scale-space causality preserving filters. *IEEE Trans. Patt. Anal. Mach. Intell.*, 18:520–528, 1996.

[10] J.A.Bangham, R.Harvey, and P.D.Ling. Morphological scale-space preserving transforms in many dimensions. *J. Electronic Imaging*, 5(3):283–299, July 1996.

[11] J.A. Bangham, P. Chardaire, C.J. Pye, and P.D. Ling. Multiscale nonlinear decomposition: the sieve decomposition theorem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5):529–539, 1996.

[12] R. Harvey, J.A. Bangham, and A. Bosson. Scale-space filters and their robustness. In *Proc. First Int. Conf. on Scale-space theory*, pages 341–344. Springer, 1997.

[13] H.J.A.M. Heijmans, P. Nacken, A. Toet, and L. Vincent. Graph morphology. *Journal of Visual Computing and Image Representation*, 3(1):24–38, March 1992.