Scale-space trees and applications as filters, for stereo vision and image retrieval

J. Andrew Bangham, Kimberly Moravec, Richard Harvey and Mark Fisher
School of Information Systems, University of East Anglia,
Norwich, NR4 7TJ, UK.
[ab|klm|rwh|mhf]@sys.uea.ac.uk

Abstract

Images are re-mapped as scale-space trees. The minimal data structure is then augmented by "complement nodes" to increase the practical value of the representation. It is then shown how the resulting *ctree* can be used to remove noise from images, provide a hierarchical way to estimate a dense disparity map from a stereo pair and to provide a basic segmentation of images for image retrieval purposes.

1 Introduction

There have been a number of attempts to represent images in a hierarchical fashion. Tree data structures, widely adopted in computer graphics, are natural candidates for describing decompositions of scenes, since they provide mechanisms for preserving object hierarchies. There is now renewed interest in object-based scene descriptions which is being driven by the proposed MPEG-4 and MPEG-7 standard [1].

Early work attempted to interpret images in terms of a semantic hierarchy [10] and relate these to quad-trees [17]. Proposals, for analyzing images into tree-like hierarchies, from the signal and image processing community have included Laplacian pyramids [7,8] and wavelets [21, 22, 24]. Both are computationally efficient but they have some fundamental drawbacks in the context of producing trees: they have the potential to introduce spurious features at large scale (and therefore branches); large scale features are blurred and hence the resulting, large scale, fuzzy blobs do not accurately represent the shape of objects. The first problem, spurious extrema, has been nicely solved. It had long been recognized that objects tend to be associated with extrema in intensity images. This formed the starting point for attempts to isolate meaningful edges [23, 26, 27] and subsequently for the theoretically tidy representation of images known as scale-space [2, 12, 18, 35]. The, now classical, method uses a set of Gaussian, or near Gaussian, filters [9, 19, 20]. The problem remains that they produce blurred images (inevitably for linear systems) and this is, perhaps, sufficient reason not to use them as a starting point for creating trees with semantic meaning.

Several non-linear alternatives have subsequently emerged including: anisotropic diffusion in which the conductivity is allowed to vary with image gradient [30]; erosion and dilation with elliptic paraboloids [15, 34]; and nested sets of connected-set alternating sequential filters [3, 5, 14, 31, 33]. The performance of these systems as scale-space processors has been reviewed [11] and the latter, in particular, appear to have the potential

BMVC 1999 doi:10.5244/C.13.12

to produce meaningful regions. They have provided a starting point for attempts to build object trees [13, 32].

Here, we attempt to resolve some problems associated with these approaches and show how the resulting scale-tree might be applied to three very different problems namely those of simplifying images (denoising), obtaining dense stereo maps and of image retrieval.

2 Methods

2.1 Building the tree from a grayscale image

We assume the underlying theory of the scale-space decomposition [3, 4, 14]. The image is decomposed by identifying local max and local min (extremal) level-sets according to their scale, namely area. The process starts at the smallest scale, slicing off small-scale, local, extrema irrespective of shape, and proceeds to the largest scale, namely the whole image. Adapting the notation in [14] the complete decomposition is denoted by

$$S^C(f) = S_m S_{m-1} \cdots S_2 S_1 \tag{1}$$

where m is the maximum scale associated with image, namely the total number of pixels in the image. Differences between successive stages, namely the regions sliced off, are defined as granule functions $G_s = S_{s-1}f - S_s f$ which may contain more than one granule, g_{s_i}. Granules are connected regions in G with non-zero value. To build the tree then, at every stage of the scale decomposition, (1), every granule, g_{s_i}, is labeled as a node, N_{si}. As a new node is created a link is also created between it and previous nodes that have been merged with that node. Each link represents a parent relation, P,

$$P\left(N_{s_i'}\right) = N_{s_j''} \text{iff} \begin{cases} s' < s'' \\ \tilde{g}_{s_i'} \subset \tilde{g}_{s_j''} \\ \forall, \min |g_{s_i'} - g_{s_j''}| = |g_{s_i'} - g_{s_j''}| \end{cases}$$

where $\tilde{g}_{s_i}$ represents the support region, or footprint, of granule g_{s_i}. An example image and scale-tree is shown in Figure 1.

The order complexity, C of computing the tree appears to be bounded: $n < C < n \log n$ and this is consistent with the fast execution times illustrated in Figure 1B and C which shows the cpu time and memory usage of our current implementation when applied to real images of areas from 500 to 500,000 pixels.

To be of value, a tree representation of an image should be a mapping of the image and so have the perfect reconstruction property familiar from quadrature mirror filter-banks and certain wavelets: it should be possible to switch between the image and tree representations and choose to use whichever is the more convenient. The need for this property, "Invertibility" [14], is one of the reasons for preferring this scale-space. The theory establishes that if the value associated with each node is the granule amplitude, then the image $f = \sum_{s=1}^{m} g_s$ where m is the scale of the root node (area of image). It also shows that node values can be set to zero (so deleting the node) and the resulting tree can still be interchanged with its new associated image. This does however lead to a problem for applications envisioned in this paper. For example, it might be desirable to select and render or match a branch with branch-root of scale $m_b < m$ in isolation from the rest of the

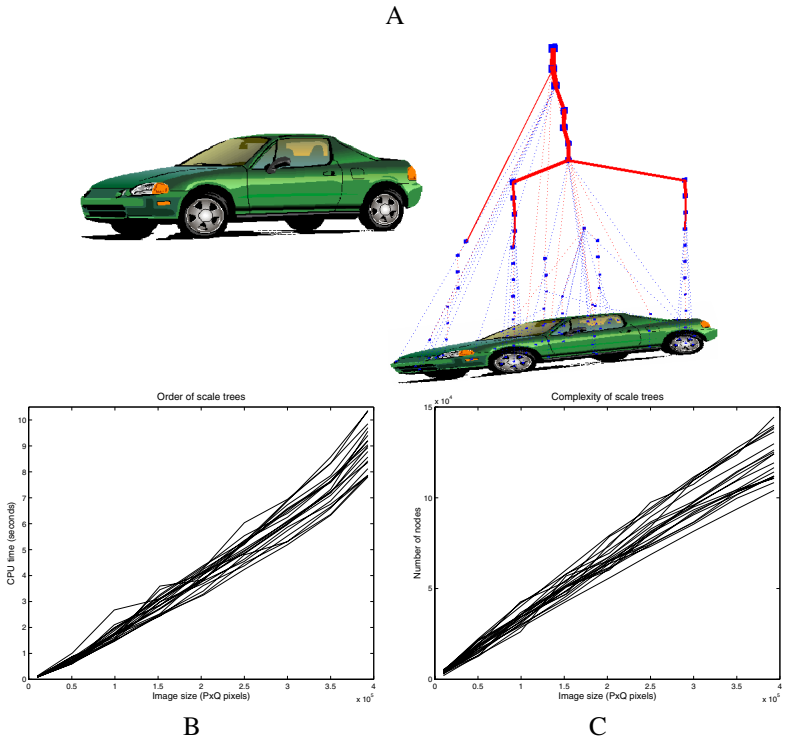


Figure 1: A) An example image and associated tree. B) Computation time (ordinate 0 to 10 s) on a 300 Mhz Pentium II. C) Memory usage (for a number of real images)

image. As it stands this could be awkward because the rebuilding operation would require the values of nodes lying between the branch-root and the image root to be known. This can be avoided by associating the grayscale-image-value with each granule (and therefore node) as it is removed. This is equivalent to $f_s = \sum_{p=s}^{m} g_p$. An image is then rebuilt by starting at the branch root and *replacing* the pixel values in the regions of support for each child node by $f_s = \sum_{p=s}^{m} g_p$ where s is the scale of the child. Let the replacement operation be ρ. Since the pixel value at any point in the image is the sum of granule amplitudes we have

Theorem 1 *Suppose that* $(g_s)_{s=1}^{m}$ *is the* M*-granule decomposition of an image,* $f = \mathbf{Z}^2$*, with finite support then the original image can be rebuilt from the nodes* $f = \rho_{s=1}^{m} f_s$.

Such a representation is analagous to the Visual Object Planes of MPEG4.

2.2 Complement trees or *ctrees*

Although the scale-tree is a mapping of the image, and so completely represents it, the tree is not necessarily in the most convenient form. It is implicit, in the reasoning behind scale-space, that interesting objects are represented by local extrema. This is often true but not always. For example, the scale-tree in Figure 2B,C of Figure 2A, has no node representing the castle-roof even though it is clearly an object. We overcome this by

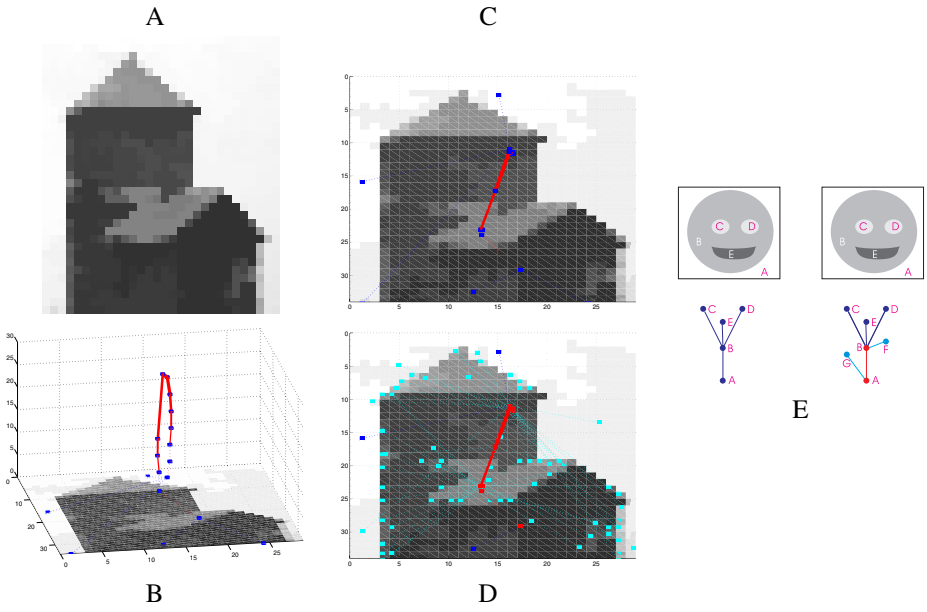


Figure 2: A) A simplified image with features (the castle roof) that should be nodes in the tree. B) The associated scale-tree. C) Viewed from the top. Nodes are at the centre of mass of each associated segment. There is no node associated with the castle roof. D) Complement tree showing nodes associated with every flat zone in the image including the roof. E) Left is a simple scale tree with $A \subset B \subset \{C,D,E\}$. Right is the ctree with additional nodes $G = A \bigcap \bar{B}$, $F = B \bigcap \overline{E \bigcup C \bigcup D}$

augmenting the scale-tree to explicitly represent the regions implicitly associated with non-leaf nodes. The new segments are defined by c''_{j_n} where $\bigcup c''_{j_n} = C''_j$ where if $P(N_{s'_i}) = N_{s''_j}$ then $C''_j = \overline{\bigcup \tilde{g}'_{s_i}} \bigcap \tilde{g}''_{s_j}$. These complement regions are the *non-extremal* level-sets and the new tree is called a *ctree* (Figure 2E). On the left of the panel is an image and its associated scale tree. On the right is the augmented tree with two additional nodes. The cyan links and patches represent the complement nodes. Leaves are always associated directly with pixels and non-leaf nodes, shown in red, represent the topological structure of the image. Although the tree is now complicated, the conjecture is that the structure nodes introduce a way to generate a simpler, object based, representation of the image. Furthermore ctrees can be simplified easily.

2.3 Simplifying the trees

In Figure 1A the tree has been built from a grayscale image, derived from the lefthand graphic, quantised to 16 levels and with all extremal nodes (leaves in the scale-tree) up to scale 30 deleted. The resulting tree, right panel, still provides a hierarchical access to the object. This approach is practical but ugly and it does not exploit properties of the tree. Alternatives include scale-space tracking in which unbranched chains are collapsed to the peak in a scale-selection measure [13].

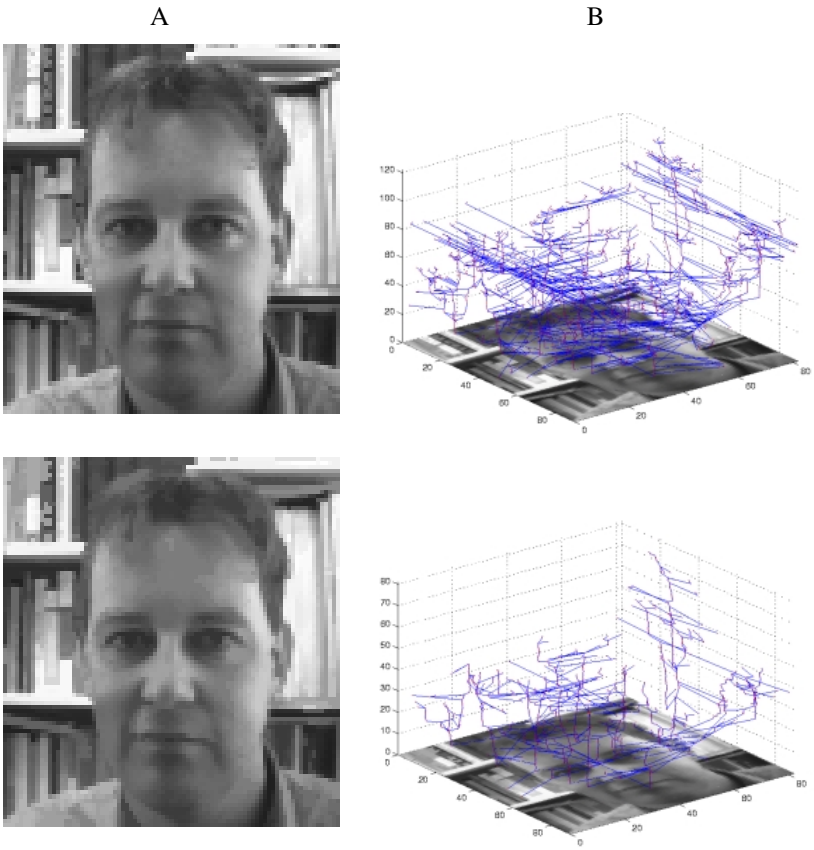


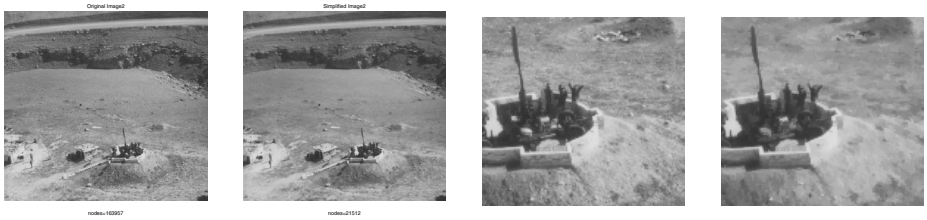
Figure 3: (A) A real image represented by a scale-tree with 3123 nodes (B). This has been simplified to 1100 nodes (C) and the rebuilt image is (D). The semantic meaning in the image has changed little despite the reduction in complexity of the tree.

The overall problem is to find criterea for merging segments specified by ctree nodes. Two major sources of nodes are noise-like textures that create small-scale nodes that add little to the understanding of the image and blurred objects that lead to extended branches in the tree. Both may be addressed by associating a significance score with each edge in the tree and removing edges with a low significance. Since each node represents a parent-child relationship the significance of the difference between the parent region and the child region is tested. Two tests are immediately available: the conventional t-test with a Student's-t confidence distribution [16] and a parametric likelihood ratio with a χ^2 confidence [6, 25] distribution.

If the image data are assumed to be distributed as a multivariate normal in the regions $R_p(G)$ and $R_c(G)$ where, for level-set trees formed from the intensity data, $R_p(G) \subset R_c(G)$ then it is simple to define a likelihood ratio, λ for the data given that the two regions have separate means and variances versus the hypothesis that the two regions

are homogeneous and so have a common mean and variance. In [6] it is shown that $-2\log\lambda = N\log|S^{\alpha}| - N_p\log|S_p| - N_c\log|S_c|$ where $N_{p,c}$ are the number of points in $R_{p,c}(G)$, $S_{p,c}$ are the sample covariance matrices computed in those regions and S^{α} is the sample covariance computed over the combined region. The statistic $2\log\lambda$ is distributed as approximately $\chi^2(2d)$ where d is the dimensionality of the image features. For one-dimensional features, such as grey-scale intensity, by associating a variance of $h^2/12$ with each level-set (h is the intensity separation between adjacent level sets) it is easy to show that, where $A = \left(N_p + N_c + \frac{N_p N_c}{N}(I_p - I_c)^2\right)$ then, $\log\lambda = \frac{N}{2}\log\left(\frac{A}{N}\right)$ with an associated confidence, $\delta = 1 - \left(A^2/N^2\right)^{-2}$. An alternative is the conventional t-statistic: $t = (12N_pN_c/N)^{1/2}(I_p - I_c)$ which is distributed as Student's t distribution with $\nu = N^2\left(N_c^2/(N_p-1) + N_p^2/(N_c-1)\right)^{-1}$. Either gives acceptable results but here we use the likelihood ratio, λ, since it extends to multidimensional features and, for scalar features, the confidence δ is closed form.

Figure 3 shows how this can work for a real image. Edges, where the confidence score computed between the parent's complement node and a child node was less than 0.99 have been removed in panel C. The resulting simplified image, shown in panel D, is very similar to the original, panel A. Likewise a different image, Figure 4. Although



A B C D

Figure 4: A,C) Original image (*lzw.tif size 37KB) has a tree with 163957 nodes. B,D) Simplified image (21KB) has a tree with 21512 nodes.

the new image is considerably simplified, it looks similar to the original and preserves important details, such as the gun barrel (and arms up in surrender). In these examples the ctree is being used as a denoising filter.

We now provide two further illustrations of how the scale-tree can be used in computer vision.

2.4 Stereo disparity

For stereo vision with parallel calibrated cameras, features of two or more images are matched and the three dimensional geometry of the scene recovered from the shift, *disparity*, along the epipolar line of each image feature. To obtain a dense disparity map from a pair of images, the ctree obtained from the left image and then simplified (see above) is traversed in *preorder*. For each node the disparity is computed from the position and error of the best match. This disparity is then assigned to the node. If the error of the disparity for this node is lower than that of its parent, then its disparity is accepted in preference to that of its parent. This process is similar in principle to that used above to simplify a tree.

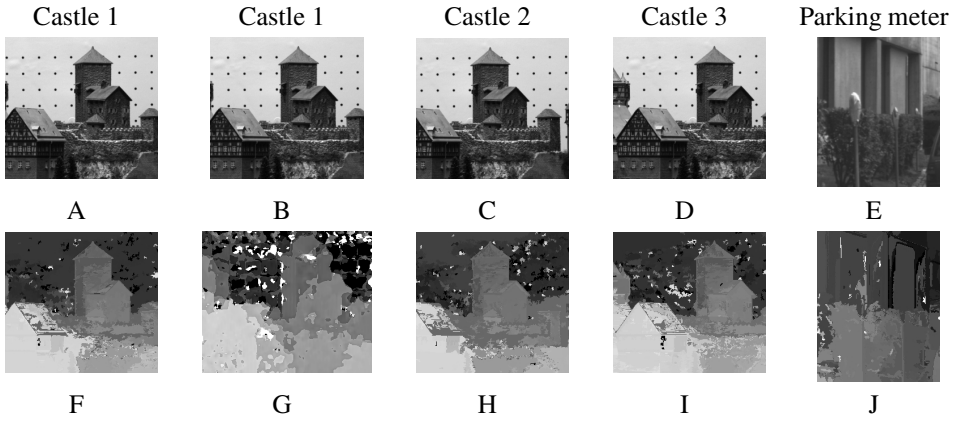


Figure 5: (A-E) Left images of stereo pairs and (F-J) their dense disparity maps. The algorithm was developed using A and tested on the remainder. (F,H,I,J) show disparities obtained from the ctree and (G) is the disparity from an SSD sliding window method.

Figure 5A (B is the same as A), C and D shows three stereo pairs of a calibrated castle scene [28]. The resulting dense disparity maps are shown in Figure 5F,G,H,I. A comparison of Panels (F) and (G) allows the new method (F) to be compared with a sum of square differences associated with a fixed window (G). A comparison of estimated disparities for regions with known ground truth show that the new method has promise: new method Panel (F) has mean disparity error of 0.6 with standard deviation of 0.5 compared to 1.2 and 7.8 respectively for the SSD method. A more rigorous comparison using random dot stereograms and a variant on this algorithm is reported elsewhere and demonstrates it is not much worse then conventional SSD in highly textured regions but is much better in low textured ones [29]. In Figure 5 the new method fails when there is no reasonable match to be found but the resulting disparity map is sharp-edged.

2.5 Image retrieval

Often objects within an image are of more interest than the image itself and so both the objects and the whole image should be used to generate features. The scale-tree appears to be a candidate for generating a suitable segmentation of the objects. As with disparity estimates the absence of a convolution window should mean that the segments are sharply delineated. Moreover, the tree structure should allow a top-down approach.

A way is needed to automatically select the *significant* very large scale, nodes (near to the root) that approximately segment entire objects. The numbered nodes in Figure 6 are examples, of which sixty-three is the root. It is clear that the choice of node has to be made on evidence drawn from entire branches rather than adjacent nodes.

The intention is to get an intuition into how an image retrieval system might operate rather than attempting to elaborate a method of reliably identifying the key nodes. We associate with each node a feature that provides a some measure of not only itself and all of its children. Here we take a very simple measure of this *importance*, I, of the node which is defined as $I = D/D_{\text{root}}$ where D is the depth. Leaves have depth 1 and the depth of any other node is defined as $D = 1 + \max D_c$ where D_c are the depths of all the children

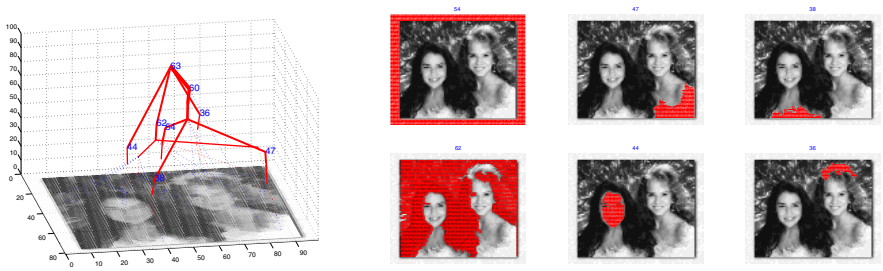


Figure 6: Left: scale-tree showing *significant* nodes. Right panel: associated segments.

of that node. I is reflected in the thickness of the edges plotted in Figures 1A and 6A. To select those nodes that have segments that (roughly) include objects that might be used in the image retrieval process we search for nodes with important brothers. To build a database of important nodes we constrain the number of such nodes to lie between 2 and 10 so, for example, six segments are identified as shown in red on the right of Figure 6.

To illustrate how an image retrieval system based on object color might operate, each segment in every image is characterized by a feature vector: $\mathbf{F}_s = \{\overline{r}, \overline{g}, \overline{b}, \sigma_r, \sigma_g, \sigma_b\}$ where r, g, b are red, blue and green colors. Having added segments for all the images to the database the feature space is "squared-up" so that each feature has zero mean and unit standard deviation. On recall the images are ranked in feature space using the distance between the ith and jth image defined in terms of the distances between the pth segment in the ith image and the qth segment in the jth image. Denote by $\mathbf{F}_i(p)$ the feature associated with the pth segment in the ith image. The distance between the ith and jth image is $d_{i,j} = \frac{1}{P_i} \sum_{P=1}^{P_i} \min_q \left| \mathbf{F}_i(p) - \mathbf{F}_j(q) \right|$ where the ith image has P_i segments. Note this allows multiple matches between one segment in the exemplar (ith) image and segments in the jth image [1]

Results from a simple test database are shown in Figure 7. The top-left image in Figure 7A is the exemplar. The closest match is shown to the right (namely itself) followed by less successful matches. The database has 720 images of which a random 10 are shown in the bottom panel. The database has 19 images with white borders (like the exemplar) namely 2.5% and 29% of the 19 are in the set of top 24 matches. The database has 190 portraits (head and torso) namely 26% and the top 24 contained 62%. Meaningful tests results await a standardized database with an accepted ground-truth and it hoped that this will emerge from, among other initiatives, MPEG. However, the experiment does show how the tree might be used for image retrieval.

3 Discussion

We propose the ctree as a structure for representing bit-images. Leaves represent connected level-sets and the remaining nodes provide a hierarchical structure. The structure is independent of object rotation and other distortions that do not change the topology. In images of densely packed objects this means less than for the simple cases presented

[1] If the exemplar shows one cup (object) it ought to match an image with two cups at least as well. Conversely, if the the exemplar has two cups then it should match an image with one cup but not as well as one with two.

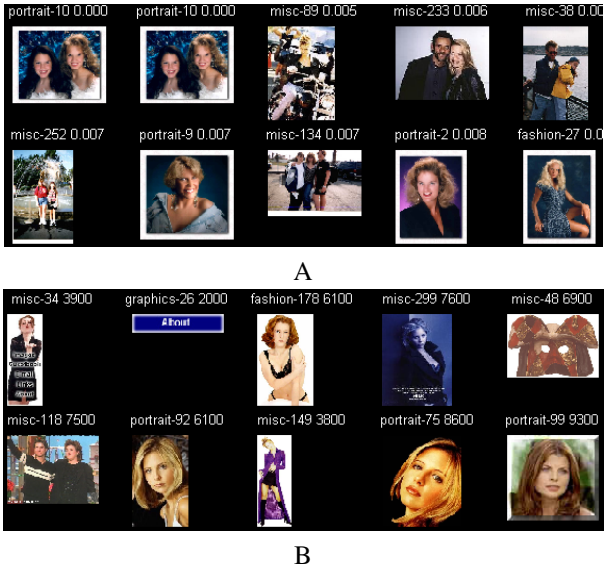


Figure 7: A) Example image (top left) and the images that matched it. B) Random images from the same database.

here, but it is still an advantage. In many instances the data structure provides access to entire objects and this opens the way to a new attack on the old problem of segementing images in a meaningful way. We have shown how nodes can be merged using statistical criteria but the methods need to be developed further. Future work will explore how, for example, both more node features and more nodes can be included in each decision to merge. Perhaps it will be possible to work up through the tree collecting evidence: a Bayesian decision, to merge or not, would then be based on probabilities obtained from a training set, the evidence propogated from the children of the nodes under consideration and features from the nodes themselves.

References

[1] Overview of MPEG-4 functionalities supported in MPEG-4 version 2. In *ISO/IEC JTC1/SC29/WG11 N2079, MPEG98/March*, March 1998.

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

[3] J. A. Bangham, P. Chardaire, C. J. Pye, and P. D. Ling. Multiscale nonlinear decomposition: The sieve decomposition theorem. In *IEEE Transactions On Pattern Analysis and Machine Intelligence*, volume 18:5, pages 520–528, 1996.

[4] J. A. Bangham, J. Ruiz-Hidalgo, R. W. Harvey, and G. C. Cawley. The segmentation of images using scale-space trees. In *Proc. British Machine Vision Conference (BMVC-98)*, Southampton, UK, September 1998.

[5] 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.

[6] A. Basman, J. Lasenby, and R. Cipolla. The creep-and-merge segmentation systems. Technical Report CUEDDIF-INFENG/TR295, Cambridge University, July 1997.

[7] P. Burt. Smart sensing with a pyramid vision machine. *Proc. IEEE*, 76:1006:1015, 1988.

[8] P. J. Burt and E. H. Adelson. The Laplacian pyramid as a compact image coding. *IEEE Trans. Comm. Com*, Vol. 31:pp 532–540, 1983.

[9] James L. Crowley and C. Sanderson. Multiple resolution representation and probabilistic matching of 2- gray-scale images. *IEEE Trans. Patt. Anal. and Mach. Intell.*, 9(1):113–121, January 1987.

[10] K.S. Fu. *Syntactic Pattern Recognition and Applications*. Prentice Hall Advances in Computing Science and Technology Series. Prentice Hall, New Jersey, 1982.

[11] 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.

[12] T. Iijima. Basic theory of pattern normalization (for the case of a typical one-dimensional pattern. *Bulletin of the Electrotechnical Laboratory*, 26:368–388, 1962.

[13] J.A.Bangham, J.R.Hidalgo, G.C.Cawley, and R.W.Harvey. Analysing images via scale-trees. In *British Machine Vision Conference*, 1998.

[14] 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.

[15] Paul T. Jackway and Mohamed Deriche. Scale-space properties of the multiscale morphological dilation-erosion. In *Proceedings of the 11th IAPR Conference on Pattern Recognition*, 1992.

[16] M.G. Kendall, A. Stuart, and J.K. Ord. *Advanced Theory of Statistics*. Griffin, 1987.

[17] Allen Klinger and Charles R. Dyer. Experiments on picture representation using regular decomposition. In *Computer Graphics and Image Processing*, volume 5, pages 68–105, 1976.

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

[19] T. Lindeberg. Scale-space from discrete signals. *IEEE Trans. Patt. Anal. Mach. Intell.*, 12:234–254, 1990.

[20] T. Lindeberg. *Scale-space theory in computer vision*. Kluwer, 1994.

[21] S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet transform. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 11:pp 674–693, 1989.

[22] S. G. Mallat. Zero-crossings of a wavelet transform. *IEEE Trans. on Information Theory*, Vol. 11:pp 1019–1033, 1991.

[23] Stephane Mallet and Sifen Zhong. Characterisation of signals from multiscale edges. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14:710:732, 1992.

[24] Stephane G. Mallet. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11:674:693, 1989.

[25] K.V. Mardia, J.T. Kent, and J.M. Bibby. *Multivariate analysis*. Academic Press, 1979.

[26] D. Marr. *Vision*. W. H. Freeman and Company, New York, 1982.

[27] D. Marr and E. Hildreth. Theory of edge detection. In *Proc. R. Soc. Lond. B*, volume 207, pages 187–217, 1980.

[28] M.Maimone and S.Shafer. The CMU calibrated imaging stereo datasets. http://www.cs.cmu.edu/People/cil/cil.html.

[29] Kimberly Morovec, Richard Harvey, and J. Andrew Bangham. Improving stereo performance in regions of low texture. In *Proc. British Machine Vision Conference (BMVC-98)*, pages 822–831, Southampton, UK, September 1998.

[30] 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.

[31] P. Salembier, A. Olivieras, and L. Garrido. Motion connected operators for image sequences. In *Signal Processing VIII Theories and applications*, volume II, pages 1083–1086, 1996.

[32] Phillippe Salembier and Luis Garrido. Binary partition tree as an efficient representation for filtering, segmentation and information retrieval. In *IEEE Int. Conference on Image Processing, ICIP'98*, Chicago (IL), USA, October 4-7 1998.

[33] J. Serra and P. Salembier. Connected operators and pyramids. In *Proceedings of SPIE Conference on Image Algebra and Mathematical Morphology, Volume 2030*, pages 65–67, 1993.

[34] R. van den Boomgaard and A. Smeulders. The morphological structure of images: the differential equations of morphological scale-space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(11):1101–1113, November 1994.

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