Link Shifting Based Pyramid Segmentation for Elongated Regions

Milos Stojmenovic, Andres Solis Montero, and Amiya Nayak

SITE, University of Ottawa, Ottawa, Ontario, Canada K1N 6N5
{mstoj075,amon,anayak}@site.uottawa.ca

Abstract. The goal of image segmentation is to partition an image into regions that are internally homogeneous and heterogeneous with respect to other neighbouring regions. An improvement to the regular pyramid image segmentation scheme is presented here which results in the correct segmentation of elongated and large regions. The improvement is in the way child nodes choose parents. Each child node considers the neighbours of its candidate parent, and the candidate parents of its neighbouring nodes in the same level alongside the standard candidate nodes in the layer above. We also modified a tie-breaking rule for selecting the parent node. It concentrates around single parent nodes when alternatives do not differ significantly. Images were traversed top to bottom, left to right in odd iterations, and bottom to top, right to left in even iterations, which improved the speed at which accurate segmentation was achieved. The new algorithm is tested on a set of images.

Keywords: image segmentation, pattern recognition.

1 Introduction

Image segmentation is an important step in many computer vision applications. It decomposes an image into homogeneous regions, according to some criteria [6]. It is rarely achieved comprehensively for any single application, and algorithms that do perform well in one application are not suited for others.

Pyramid segmentation was proposed in [7, 10], and further elaborated on in [3]. Pyramids are hierarchical structures, which offer segmentation algorithms with multiple representations with decreasing resolution. Each representation or level is built by computing a set of local operations over the level below (with the original image being at base level or level 0 in the hierarchy). Level L consists of a mesh of vertices. Each vertex at level L is linked to at most one parent node at level $L+1$. The value of the node is derived from the values of all its children nodes at the level below. When this is applied to children nodes transitively down to the base level, the value of each node at a given level is decided by the set of its descendent pixels at the base level (its receptive field). Each node at level L also represents an image component of level L segmentation, consisting of pixels belonging to its receptive field (if nonempty). Thus each level has its predefined maximum number of components. However its minimum number of components is left open and depends on a concrete algorithm.

Pyramids were roughly classified into regular and irregular. Regular pyramids have a well-defined neighbourhood intra-level structure, where only natural neighbours in

D. Ślęzak et al. (Eds.): SIP 2009, CCIS 61, pp. 141–152, 2009.

the mesh that defines a level of a pyramid are considered. Inter-level edges are the only relationships that can be changed to adapt the pyramid to the image layout. Literature also places a condition of a constant reduction factor for regular pyramids [6]. Regular pyramids can suffer several problems [1, 5, 6]: non-connectivity of the obtained receptive field, shift variance, or incapability to segment elongated objects.

To address these problems, irregular pyramids were proposed which vary in structure, increase the complexity and run time of the algorithms, and/or are dependant on prior knowledge of the image domain to be designed successfully. In the irregular pyramid framework, the spatial relationships and reduction factors are non-constant. Existing irregular pyramid based segmentation algorithms were surveyed in [6, 5]. The described methods all appear very complex, have higher time complexities compared to regular pyramids and all consider the connectivity of the receptive field (base layer) as a design goal.

In this research, we first observe that the connectivity of the receptive field (region in segmentation) is not always a desirable characteristic. For instance, in forestry applications, the same type of vegetation could be present in several parts of the image, and treating them as single segment may in fact be preferred for further processing. In other applications an object that is partially obscured and 'divided' by other objects may also be more naturally identified as a single object instead of processing it in multiple pieces. Our proposed segmentation algorithm allows for non-connectivity of receptive fields.

Our algorithm is inspired by the regular pyramid linked approach (PLA) originally proposed by Burt et al. [3]. Their approach differs from previous pyramid structures in that nodes from one level may alter their selection of parent at each iteration of the segmentation algorithm. Starting from level 0, links between level L and level $L+1$ are decided. Each vertex at level L has four fixed candidate parents, and chooses the one which is the most similar from the higher level. The local image property value of each parent is recalculated by averaging the local image property of its current sons. The process continues until the son-parent edges do not vary, or a certain number T of iterations is reached. The process then continues at the next higher level. The main advantage of this method is that it does not need any threshold to compute similarity between nodes. In the original method [3], each node must be linked to one parent node. Antonisse [1] removed the need by introducing 'unforced linking' which allows the exclusion of some vertices from the linking scheme, and the presence of small components in the segmented image. This method has two parameters, one for the degree of similarity for decision linking, and one for the minimal size of a component (that is, from which level unforced linking is permitted). Antonisse [1] also proposed path fixing (some arbitrary path from the image level to the top level is fixed). This was indirectly applied here by the initial selection of random numbers for each pyramid vertex and the tie breaking rule is applied on them. Antonisse [1] also proposed randomized tie-breaking: when potential parent nodes have the same value, the choice of the parent is randomized. We have changed this rule in this paper.

Burt et al [3] however limit the choice of parent to 4 fixed nodes directly above the child. This approach has contributed to more successful segmentation, but due to the limited selection of parents, elongated and generally large segments that cover a significant portion of the image are not considered 'joined'. Figure 1 below demonstrates this weakness in their algorithm, even at the top level of the segmentation process.

Since each node can only chose among the parents in its immediate neighbourhood, nodes at opposite ends of an image, (or an elongated segment) can never point to the same parent node.

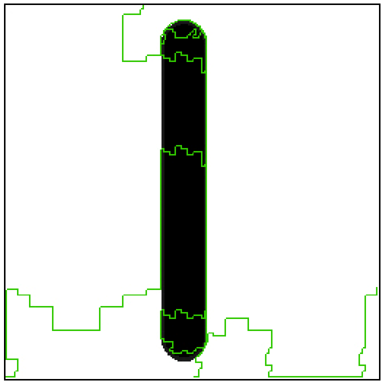

Fig. 1. Elongated regions pose challenges for regular pyramid based segmentation

We strive to design an algorithm that would properly handle elongated objects, while not enforcing connectivity of the receptive field and preserving shift invariance (the stability when minor shifts occur), having favourable time complexity (receiving an answer within seconds, depending on the image size), and overall simplicity, so that the algorithm can be easily understood, implemented, and used in practice.

To achieve our goals, we have made some simple changes in the way parent nodes are selected in the regular pyramid framework, which resulted in major improvements in their performance, including reduced shift variability and handling elongated objects. Instead of always comparing and selecting among the same four candidate parent nodes, each vertex at the current level will compare and select the best among the current parent, its neighbours, and current parents of its neighbouring vertices at the same level. Either of 4- or 8-connectivity neighbours can be used.

Experiments were conducted on both simple shapes, to validate the proposed methods, and on several real images. Evaluating segmentation quality in imagery is a subjective affair, and not easily done. "The ill-defined nature of the segmentation problem" [6] makes subjective judgment the generally adopted method (existing quantitative measurements are based on subjective formulas). In general, it is not clear what a 'good' segmentation is [6]. For this reason, no quantitative evaluation measure is applied to verify the results. The quality of the obtained segmentations appears visually satisfying for at least one level in each image.

2 Literature Review

2.1 Overview

There exists a vast literature on the segmentation problem, and hundreds of solutions were proposed. A list of references and descriptions of basic approaches (thresholding, edge-based, region-based, matching, mean-shift, active contour, fuzzy, 3D graph-based, graph cut and surface segmentation) are given in [8]. Marfil et al. [6] present a concise overview of existing pyramid segmentation algorithms. Our discussion is concentrated to such schemes which, compared to others, offer segmentations with low or no parameters, and flexibility with respect to the output (one answer is given at each level of pyramid).

We found one existing publicly available image segmentation software solution by [2], which is used here for comparing our results. It provides segmentation at various levels and appears to generally work well. The algorithm itself is sophisticated and based on linear algebra and statistics tools. In our opinion it is currently the most complete non supervised image segmentation system available.

2.2 Segmentation Algorithms

A plethora of segmentation algorithms exists [8]. However, their practicality is often not clear. For example, Shi and Malik [9] propose a 'normalized cut' that calculates both similarities and differences between regions. The normalized cut in an image revolves around calculating eigenvalues of matrices of the size n x n, where n is the number of pixels in the image. Their results appear to work relatively well only for very small images, and when segment connectivity is enforced, but are impractical for processing images larger than roughly 100 x 100 pixels, which is the size of their test set.

Pyramid based segmentation algorithms are described in [6], together with the data structures and decimation procedures which encode and manage the information in the pyramid. Our new algorithm is based on the one originally proposed by Burt et al [3]. They further the idea that the original image should be down sampled in successive steps and that pixels from one level should point to parents in the next. Segmentation can be done on the whole pyramid, and a given segmentation result is achieved on each level. Details are given in the next section, together with the changes made by our new algorithm.

2.3 Region Similarity and Unforced Linking

Antonisse [1] introduced the 'unforced linking' concept of holes in the segmentation process. This means that a pixel in one level does not necessarily need to select any parent in the above level if all of the proposed parents are sufficiently different from it. Antonisse uses a 2 step segmentation process, one of which is a clustering step on each layer which we do not employ in our work. Let include(u) be m times the standard deviation of a 3x3 neighbourhood around the node u (m is a parameter, =2 for 95% confidence and =3 for 99% confidence assuming normal distribution of pixel intensities). Find the most similar parent q. If it has the value which differs by at most include(u) of from u's value, u links to q. Otherwise, u fails to link to q and becomes the root of a new sub pyramid [1]. The similarity between two regions u and q can be measured also directly using their pixel distributions. For example, [4] propose a statistical test that analyzes two regions for homogeneity. Their test involved comparing the grey level distribution of pixel intensities of the two regions and evaluating the level of overlap of the 2 distributions. The test measures the similarity of variances of image intensities of two regions. Thus even two regions with the same mean values can be declared as dissimilar only due to different variances in their regions. In our implementation, we use a simple threshold for both region similarity and unforced linking.

3 Pyramid Image Segmentation Algorithm

Here we describe the proposed pyramid segmentation algorithm. It is nearly identical to that of [3], except in the way that children choose their parent nodes. This change resulted in major improvements in its performance, while preserving its simplicity. Another enhancement includes a tie breaking rule for parent selection.

The input to the algorithm is a greyscale, single channel image of dimensions 2^Nx2^N pixels, for $N \geq 2$. Each pixel (at level 0) u has integer value $I(u)$ in interval [0,255]. The output is the original image overlaid with the resultant segmentations at each level. In the pseudo code and discussion below, L is the level of the pyramid, L = 0, 1…, N. The bottom level (L = 0) is a matrix 2^N x 2^N pixels, representing the original image. Level L is matrix with 2^{N-L} rows and columns, i, j = 0, 1, 2 … 2^{N-L}-1. Top level $L=N$ has one element.

3.1 Creating the Initial Image Pyramid

We describe and use the overlapping image pyramid structure from [3]. Initially, the children of node [i, j, L] are: $[i', j', L-1]=[2i+e, 2j+f, L-1]$, for $e,f \in \{-1,0,1,2\}$. There are a maximum of 16 children. That is, each {2i-1, 2i, 2i+1, 2i+2} can be paired with each {2j-1, 2j, 2j+1, 2j+2}. For $i=j=0$ there are 9 children, and for i=0 and j>0 there are 12 children. There are also maximum index values 2^{N-L+1}-1 for any child at level L-1 which also restricts the number of children close to the maximum row and column values. For $L=N$ there are four children of single node [0, 0, N] on the top: [0, 0, N-1], [0, 1, N-1], [1, 0, N-1], [1, 1, N-1] since minimum index is 0 and maximum is 1 at level N-1. Two neighbouring parents have overlapping initial children allocations. Conversely, each child [i, j, L] for $L<N$ has 4 candidate parent nodes (if they exist) [i'', j'', L+1]=[($i+e$)/2, ($j+f$)/2, L+1], for $e,f \in \{-1,1\}$, where integer division is used. Pixels at the edges of the image would have fewer parents to choose from. The average intensity of all possible children is set as the initial intensity value $I(v)$ of the parent v, for nodes at levels $L>0$. This calculation is relatively slow since almost every parent has 16 table lookups to perform; therefore, an improvement was built into the system. Integral images were built for every layer in the pyramid, and for each set of points that needed to be summed, the number of table lookups was reduced from 16 to 4. This resulted in a 4x improvement in the construction of the image pyramid. The integral image technique first calculates cumulative sums of intensity values of all pixels with a first index $\leq i$ and second index $\leq j$. The summary value inside a rectangle is then found easily from these cumulative sums at the four rectangle corners.

3.2 Testing Similarity of Two Regions, Unforced Linking and Tie-Breaking Rule

Our algorithm makes use of a procedure for comparing the similarity of two regions, namely receptive fields of a vertex u and its candidate parent v. These two vertices are similar (*similar(u,v)=true*) if their intensities are roughly the same. In the current implementation of the algorithm, we use a simple threshold based (the threshold value used in algorithm is $S=15$) algorithm for testing similarity. We also use a test for dissimilarity of two regions, to decide whether of not the best parent is an acceptable link, for the unforced linking option. We use simple threshold based comparison against threshold value $D=100$.

Each pixel chooses the parent with the closest intensity to itself out of all candidate parents for its initial choice. The tie-breaking rule in [BHM] is as follows. When two or more 'distances' are equally minimal, one is chosen at random (randomization is done when needed). However, if one of them was selected parent in the previous iteration, it is not then changed. We observed a slow merging process when this rule is employed. For example, two neighbouring pixels would not change and merge two neighbouring parents when all of them have the same intensities. We have modified this rule as follows. Each vertex v is initially assigned a random number $r(v)$ in [0,1] which was never changed later on. Also, each vertex v at level L has a counter $c(v)$ which contains the number of children that selected that parent in the last iteration. Before the first iteration, $c(v)=0$ for each vertex. Let w be a child node that compares parent candidates u and v, and let better(w,u,v) be one of u or v according to the comparison. The function is as follows.

If $|I(w)-I(u)| < |I(w)-I(v)|$ **then** *better=u* **else** *better=v*;
If similar(w,u) and similar(w,v) **then** {
If $c(u)>c(v)$ **then** *better=u*; **If** $c(u)<c(v)$ **then** *better=v*;
If $c(u)=c(v)$ **then** {IF $r(u)>r(v)$ **then** *better=u* **else** *better=v*}}.

3.3 Candidate Parents

Among candidate parents, each vertex selects the one which the closest to it, using a tie-breaking rule to decide among them. For the first iteration, each vertex has up to four candidate parents, as per initial setup [3] described above. This fixed set of candidate parents has been changed (for further iterations) in our algorithm by a dynamic flexible set of candidate nodes that revolves around the current parent selection and the parent selection of neighbouring vertices at the same level. Suppose node $w = [i, j, L]$ is currently linked to parent $u = [i'', j'', L+1]$ in iteration t, which we will denote simply by $p(w)=u$. The full notation would lead to $p[i, j, L][t]=[i'', j'', L+1][t]$, and is convenient for easy listing of candidate parents. One set of candidate parents consists of the current parent and its 8 neighbours at the same level. Thus, in our notation, the candidate parents for the next iteration are: $[i''+e, j''+f, L+1]$, where $e, f \in \{-1,0,1\}$. This produces a maximum of 9 candidate parents, which is a 3x3 grid centered at currently linked parent. We also added four additional parent candidates, by considering the current selection (from the previous iteration) of neighbouring vertices at the same level. For $w=[i, j, L]$ and iteration $t+1$, we also consider $p[i+1, j, L][t]$, $p[i-1, j, L][t]$, $p[i, j+1, L][t]$, and $p[i, j-1, L][t]$ as parent candidates, if they exist. Both sets allow us to shift the parent further away in the next iteration, and possibly link the current child to a remote parent after the iterative process stabilizes (with no more changes in selected parents). This change of parent selection nodes is directly responsible for the ability of our new algorithm to handle elongated objects. Note that, alternatively, we could consider 5 candidate parents, or 8 parents of the neighbouring nodes. The current selection is an initial option, as a compromise between time complexity, and the quality of segmentation.

3.4 Pyramid Segmentation

Once the image pyramid and pointer pyramid structures have been initialized, the segmentation procedure may begin. The decisions are attained at a given level (starting at layer 0, and working toward the top of the pyramid) after several iterations, before the process advances to the next level, where each pixel modifies its selection of parent in the next layer until equilibrium is reached. At layer L, each pixel points to the parent which has the closest intensity to itself in layer $L+1$, among the 9+4 candidate parents. The intensities of the parents in layer L are recalculated based on the average intensity of the pixels in its current receptive field at the end of each iteration. Since these averages are calculated from the averages of its children, they must be appropriately weighted (by the number of pixels in the receptive fields of the children). This cycle of choosing parents, recalculating intensities, and reassigning parents continues for T=5 iterations per pair of layers, or until there is no change in parent selection between layers. Each node u also has an associated random number $r(u)$, counter $c(u)$ of the number of children that selected it as unique parent, and $n(u)$, number of pixels in its receptive field. Note that $n(u)$ is used in calculating $I(u)$ from intensity values of children, e.g. if w_1, w_2 and w_3 are children of u then $I(u) = (I(w_1)n(w_1)+I(w_2)n(w_2)+ I(w_3)n(w_3))/n(u)$, $n(u) = n(w_1) + n(w_2) + {}_{\mathrm{n}}(w_3)$. The algorithm can be, at the top level, described as follows.

Create initial image pyramid (section 3.1)

For levels $L = 1$ **to** N **Do**
$t = 0$ (iteration counter)
Repeat $t = t+1$
each vertex u at level L selects parent $p(u)$ among the 9+4 candidate ones
apply unforced linking (however 'broken' $p(u)$ is still needed in the next iteration)
each vertex v at level $L+1$ recalculates $I(v)$ based on its current receptive field
Until t=T or none of links changed in iteration t.
Endfor; Display segmentation for each level in pyramid.

We also mention a simple modification in the general parent selection algorithm that has secured declaration of elongated shapes as single component at a very early stage. Initially, the traversal of pixels was always top to bottom, left to right. Elongated shapes along the NW-SE direction were immediately discovered as single components; however shapes in the NE-SW direction remained multi-componential for several levels. To avoid this problem, pixels were traversed top to bottom, left to right in odd iterations, and bottom to top, right to left in even iterations. The elongated components were then immediately recognized, at the first level of the process, as illustrated in Figure 2. Without this correction in traversal, all of the levels seen below (b-h) would have been obtained. However, with this change in traversal pattern, the whole shape, as seen in Fig. 2 -h was obtained in the first level of the segmentation.

This greatly improved the accuracy of the segmentation algorithm for more complex shapes as well.

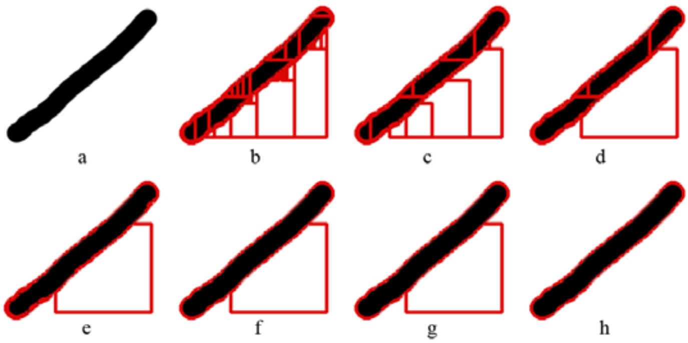

Fig. 2. [SNH] segmentation results per layer

4 Experimental Results

The algorithm presented here was designed to solve the problem of correctly segmenting elongated shapes in the framework of regular pyramid segmentation. As such, we demonstrate the results of the algorithm on a set of images, containing simple black shapes on a white background, of size 256x256 pixels. The processing time per image is 16 seconds on a single core of a Pentium 1.66 GHz dual core machine, implemented in C# on the Windows XP operating system. Our algorithm was also applied to several realistic images of size 256x256 pixels and were reduced to single channel greyscale representation as input. Although it is relatively easy for us to judge whether or not a simple shape laid against a high contrast background is segmented properly, it is far more difficult to evaluate the precision of a segmentation of real imagery. The judgement of the quality of the segmentation is subjective.

4.1 Segmentation of Basic Shapes

The shapes below are listed in 3 columns. The first column shows the original image, the second column shows the results of our algorithm proposed here, with segmentation result outlined in red, and the third column shows the results of the original pyramid segmentation algorithm proposed by [3], outlined in green. Since the shapes below have few segments, only the highest layer in the resultant segmentation pyramids is displayed, and chosen to represent the segmentation of each image.

We notice that our algorithm segmented the shapes correctly all of the time when considering basic shapes. The [3] algorithm does not work well with elongated shapes in images. The shapes listed above all fit into this category, so it is not surprising that this algorithm performs poorly on them. The [3] algorithm also has difficulties even with some non-elongated shapes. For example, it did not detect a hole in example 8, while the other hole, just a bit larger, is divided into several segments.

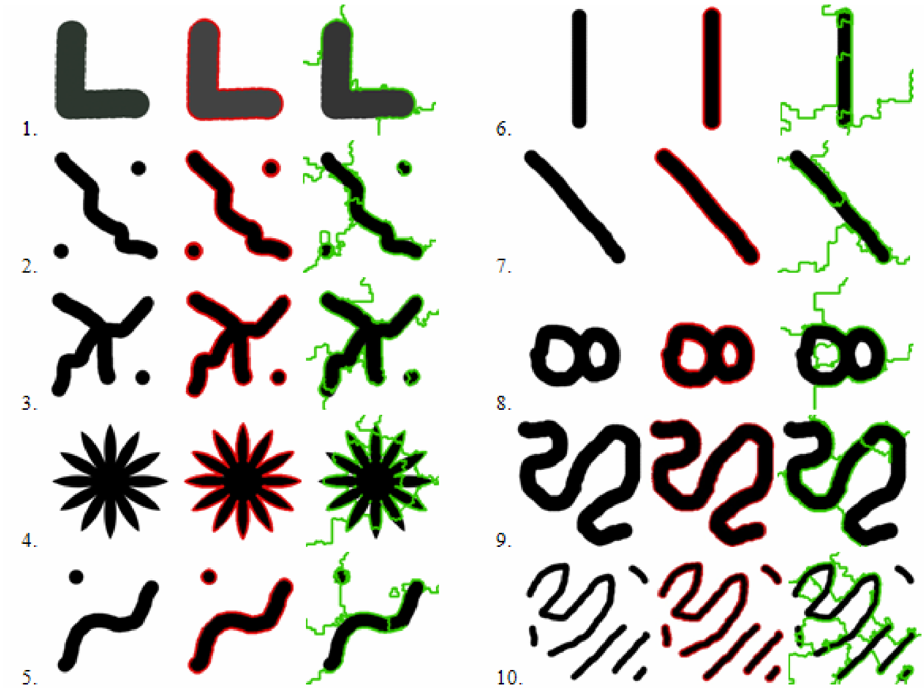

Fig. 3. Comparison of new algorithm with [3] algorithm on simple examples

4.2 Segmentation of Other Imagery

The elongated segment finding procedure was applied to some examples of everyday imagery. We compared our results with those of a [2], who also employ a type of hierarchical segmentation structure, but take into considerations texture as well as colour. Both approaches are relatively parameterless, and form a good comparison basis. The images in Figure 4 are all 256x256 pixels, their segmentation pyramids contain 8 levels in our representation and 13 in that of [2].

We show only the best level of segmentation of each pyramid since they best reflect the desired segmentation results for these images. We see that both methods segment relatively well. They segment out the wolf, high grass in the foreground and the fence posts. Our algorithm had more success at finding the sheep, the middle fencepost, and the sky in the top right corner. Both the teddy bear and the cat are equally well isolated by the algorithms, yet [2] does not segment out the cat's tail, and adds 3 other unnecessary segments around the bed sheets. This is due to superior texture matching property of [2], which in this example hinders more than helps its segmentation ability. The small dog was equally well found by both algorithms, yet they equally failed to segment the yellow container the dog sits in. This is because the container is round, and as such produces a gradient colour effect: from light yellow to dark yellow. Our algorithm fails to pick up on this change in tones, and cuts the segment into two pieces, while the [2] algorithm fails because there is no texture there to

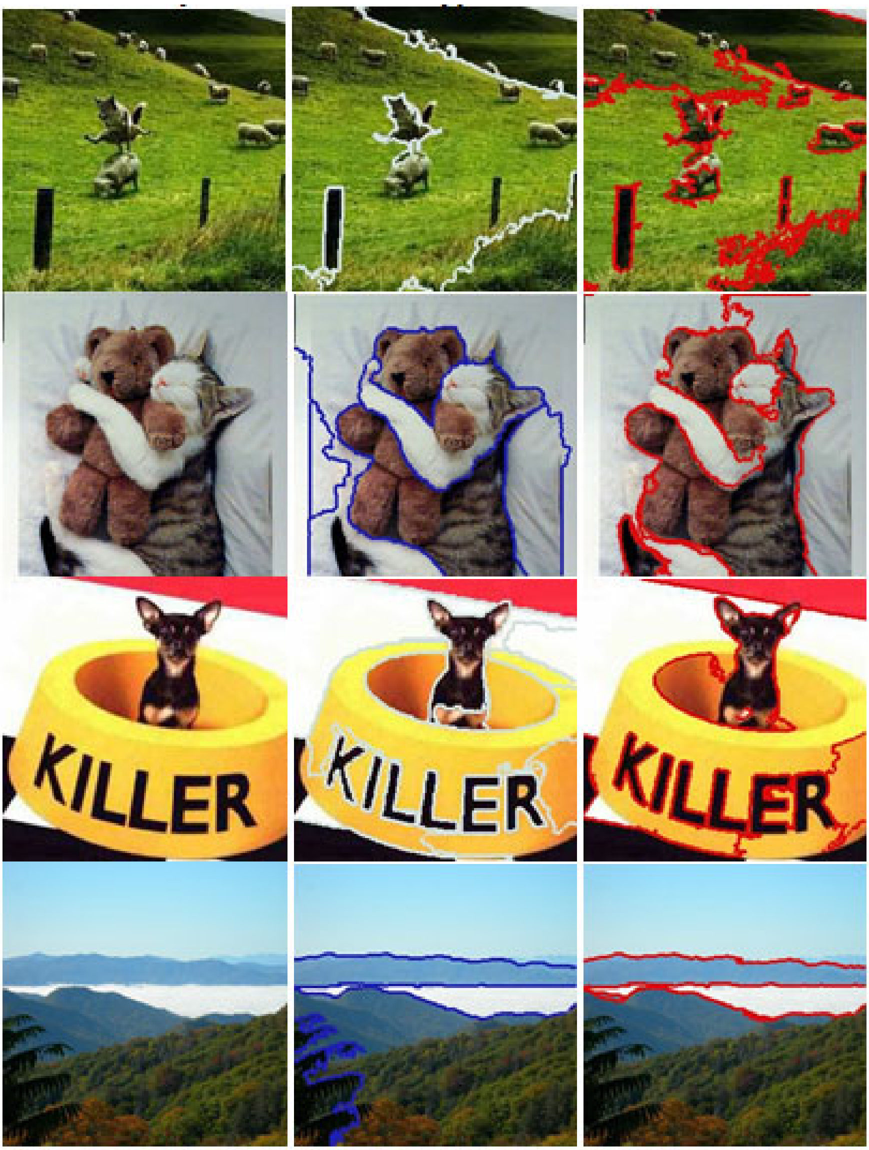

Fig. 4. Sample images and segmentation of [2] and our method

match, and primarily relies on its colour matching ability, which is no better than that of our algorithm. The last example was nearly identically segmented by the algorithms except for the large tree in the bottom left corner. This tree is found by the [2] algorithm, but not by the one we propose. This once again illustrates the superior texture matching ability of their solution. However, based on this test set, the algorithms are fairly competitive.

5 Future Work

The algorithm shows a promising way forward in the quest for successful image segmentation. Since this method relies on colour intensities, it has a tendency of over-segmenting covered regions that appear homogenous to human observers, yet distinguishably heterogeneous to the program, at (the lower layers of the segmentation pyramid). It also has the opposite tendency at the higher levels. The selection of the proper level appears possible, but is not the same for every picture, and eventually a human observer may be needed to select the best outcome for each picture, according to its further processing needs.

To improve the outcome of this segmentation algorithm, one would have to have at least some prior knowledge of the scene that is to be segmented. Such knowledge includes the minimum possible segment size, and possibly a range of pixel intensities within a region that could be considered homogenous. Other solutions may include considering more than just greyscale intensities of input data. In the current implementation, just the RGB layers are considered, and they are combined into just a single layer greyscale representation of the original image. By considering the Euclidean distance between two 3D points in an RGB space instead of simply considering greyscale differences, more accurate parent selection could be achieved at the expense of increased computation time. Another point to consider is to include some way of grouping similar textures together along with colours, such as the algorithm proposed by [2]. We are currently considering such an addition of texture to color in region similarity tests.

We are planning to enhance the similarity and dissimilarity of two regions (to avoid using an arbitrary threshold) by using a statistical testing, which is more intuitively acceptable than the one described in [4]. It was not done for this version because this new algorithm will slow down the execution of the program, which at this testing stage is also an important consideration.

We have used and experimented with the overlapping image pyramid structure as originally proposed in [3]. This refers to the fact that parent vertices at level 1 have overlapping receptive fields. Antonisse [1] already argued that perhaps a non-overlapping structure could perform better. We left this modification for further study, so that we can first investigate the impact of single major change proposed here, the use of flexible parent links. We may further experiment with the versions with different candidate parent sets, such as 5 candidate parents instead of 9 around current parent, or 8 parents of neighbouring nodes instead of 4.

References

1. Antonisse, H.: Image Segmentation in Pyramids. Computer Graphics & Image Processing 19, 367–383 (1982)
2. Alpert, S., Galun, M., Basri, R., Brandt, A.: Image segmentation by probabilistic bottom-up aggregation and cue integration. In: IEEE Conf. on Computer Vision and Pattern Recognition, CVPR 2007 (2007)

3. Burt, P., Hong, T., Rosenfeld, A.: Segmentation and estimation of image region properties through cooperative hierarchical computation. IEEE Trans. Systems, Man, and Cybernetics 11(12) (1981)
4. Chen, S., Lin, W., Chen, C.: Split-and-Merge Image Segmentation Based on localized feature analysis and statistical tests. Graphical models and Image Processing 53(5), 457–475 (1991)
5. Fu, K., Mui, J.: A survey on image segmentation. Pattern Recognition 13, 3–16 (1980)
6. Marfil, R., Molina-Tanco, L., Bandera, A., Rodriguez, J., Sandoval, F.: Pyramid segmentation algorithms revisited. Pattern Recognition 39, 1430–1451 (2006)
7. Riseman, E., Hanson, A.: Design of a semantically-directed vision processor, Tech. Rep. 74C-1, Department of Computer Information Science, University of Massachusetts, Amherst, MA (1974)
8. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis, and Machine Vision. Thompson (2008)
9. Shi, J., Malik, J.: Normalized Cuts and Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8) (2000)
10. Tanimoto, S., Pavlidis, T.: A hierarchical data structure for picture processing. Computer Graphics and Image Processing 4, 104–119 (1975)