
Semantic Segmentation Tree for image content representation

Ruba Al-Haj and Zaher Al Aghbari*

Department of Computer Science,
University of Sharjah, UAE
E-mail: ruba@sharjah.ac.ae
E-mail: zaher@sharjah.ac.ae
*Corresponding author

Abstract: In this paper, we present a novel algorithm for representing image content by constructing a hierarchy of semantic image regions, called a Semantic Segmentation Tree (SSeg-tree). First, the hill-manipulation algorithm divides an image into several visually coherent segments (small regions), which form the leaves of the SSeg-tree. Then, the method groups these segments based on well-defined spatio-visual grouping rules to produce bigger and more semantic regions, which form the intermediate nodes of the SSeg-tree. The SSeg-tree is a region-based description of the image semantic content that could be useful in many applications such as CBIR and filtering unwanted objects.

Keywords: image representation; semantic segmentation tree; spatio-visual grouping rules.

Reference to this paper should be made as follows: Al-Haj, R. and Al Aghbari, Z. (2008) 'Semantic Segmentation Tree for image content representation', *Int. J. Virtual Technology and Multimedia*, Vol. 1, No. 1, pp.23–38.

Biographical notes: Ruba Al-Haj received her BSc from United Arab Emirates University, UAE, in 1996, and the MSc in Computer Science from University of Sharjah, UAE, in 2005. Her research interests include multimedia databases, data mining, and image semantic representation and classification.

Zaher Al Aghbari received his BSc from the Florida Institute of Technology, Melbourne, USA, in 1987, and MSc and PhD in Computer Science from the Kyushu University, Fukuoka, Japan, in 1998 and 2001, respectively. He was with the Department of Intelligent Systems, Kyushu University, Japan, from 2001 to 2003. Since 2003, he has been with the Computer Science Department, University of Sharjah, UAE, where he is now an Associate Professor. His research interests include multimedia databases, data mining, multi-dimensional indexing, image/video semantic representation and classification, and arabic handwritten text retrieval.

1 Introduction

Many researches have proposed different algorithms to represent and retrieve images based on global features, such as the colour histogram of the whole image. However, retrieving images by global features is too crude an approach to produce satisfactory results (Pauwels and Frederix, 1999). Thus, it is essential to find semantic regions in an image and then represent the image by the local features of its semantic regions. As a consequence, Content-Based Image Retrieval (CBIR) will be more effective. The process of identifying the important regions in the image is called ‘image segmentation’. Image segmentation algorithms can be classified into two broad categories (Leung and Malik, 1988):

- region-based
- contour-based approaches.

Region-based approaches (Garrido et al., 1998; Comaniciu and Meer, 2002) try to partition the image pixels into regions. The pixels of each region are visually coherent in terms of one, or more, features such as colour, texture and brightness. On the other hand, contour-based approaches (Malik et al., 2001; Schluter et al., 2000) try to detect edges of regions and then exploit a curvilinear continuity technique to link the detected edges; as a result, regions are identified.

The segmentation results of both the region-based and contour-based approaches are regions that are visually coherent; however, they may not be meaningful to users, or the underlying application. On the contrary, a semantic region (object) is a two-dimensional entity that is meaningful, such as a car, building and human, where this semantic region may be formed by grouping several spatially close, but not necessarily visually coherent, smaller regions. However, the unsupervised formulation of semantic regions for general applications is a very difficult problem, as different applications may have different interpretations of the semantics of the same image.

A straightforward solution is to define the semantic regions and design a segmentation algorithm accordingly for each application. The disadvantage of this solution is that it is difficult to reuse the segmentation data of one application for a different application and it can only work for domain-specific applications. Another solution is to find a compromise between the application’s semantic needs and an unsupervised general semantic segmentation, such as the region-based image representation by a binary partition tree (Green, 1995). This method (Green, 1995) proposes to first segment an image into a set of regions, and second estimate the similarities between these regions. These regions are represented by a binary partition tree, where the leaves are the initial regions and a higher-level node is generated by grouping two children. Here, the grouping is based on the visual similarities, which can be misleading and may not result in the formulation of semantic regions at the higher levels of the tree.

In this paper, we present a novel algorithm for representing image content by constructing a hierarchy of semantic (meaningful) image regions. First, we propose a novel method for segmenting an image into visually coherent regions in the colour feature space by utilising the ‘Hill-Manipulation’ algorithm. Next, we group these regions based on well-defined spatio-visual grouping rules that produce semantic regions in the intermediate nodes of the SSeg-tree.

The rest of this paper is organised as follows. Section 2 discusses our hill-manipulation segmentation. In Section 3, we define the spatio-visual grouping rules and discuss the construction of an SSeg-tree. The results of our experiments are presented in Section 4. Finally, we conclude this paper in Section 5.

2 Image segmentation

Image segmentation is a process of decomposing an image into coherent regions. Coherence is usually defined as similarity of pixel values, i.e., a piecewise constant model is enforced over the image (Malik et al., 2001). The essential design goals of most segmentation methods are:

- the extracted regions are visually coherent
- segmentation should give satisfactory results on general image data, i.e., no a priori knowledge is assumed
- segmentation process should be unsupervised.

Our colour-based segmentation method achieves these goals. Before presenting our hill-manipulation-based segmentation algorithm, we discuss the feature space in which segmentation takes place.

2.1 Feature space

A good colour space for segmentation is one in which the perceived colour differences should correspond to their Euclidean distances in this chosen colour space. The HSV colour model satisfies this property (Green, 1995). Furthermore, the HSV model simulates the human perception of colours, where the H , S and V components represent hue, saturation and value (brightness), respectively. Thus, we map the image into the HSV colour space.

For each image, we find the minimum and maximum values of each of the three colour components to set up the coordinate axes of the HSV space. Each axis runs from minimum to maximum values. These values are normalised so that the minimum value equals zero and the maximum value is one. Then, the H , S and V components are quantised to 36, 10, and 10, respectively, within the minimum–maximum range of each component. These quantisation values can be changed by the user, if necessary, to suite a specific image collection.

2.2 Segmentation background (hill climbing)

Our proposed segmentation algorithm is based on non-parametric, cluster-based hill-climbing technique demonstrated in Ohashi et al. (2003). In general, hill climbing means searching all the neighbours of the current position, then making an uphill move towards the bin with larger number of pixels (majority colour). The algorithm in Ohashi et al. (2003) uses this technique to cluster the image contents into group of regions according to their colours. First, it creates a three-dimensional histogram using the three colour components of the image pixels. Then, it detects the peaks of clusters in that histogram. It utilises the histogram bins rather than the pixels themselves to find the

peaks of clusters; thus, it can find the peaks efficiently. Then, the algorithm associates the pixels with the detected clusters based on the local structures of the clusters.

Having computed the global three-dimensional HSV colour histogram of the image, the hill-climbing algorithm is outlined as follows:

- 1 *Start at a non-zero bin of the colour histogram and make uphill moves until reaching a peak (local maximum) as follows:*
 - 1.1 *Compare the number of pixels of the current histogram bin with the number of pixels of neighbouring bins.*
 - 1.2 *If the neighbouring bins have different numbers of pixels, the algorithm makes an uphill move towards the neighbouring bin with larger number of pixels.*
 - 1.3 *The uphill climbing is continued (repeat Steps 1.1–1.3) until reaching a bin from where there is no possible uphill movement. Hence, the current bin is identified as a peak.*
 - 1.4 *Neighbouring pixels that lead to the found peak are grouped together as one cluster, to form an initial segment.*
- 2 *Select another unclimbed bin as a starting bin and perform Step 2 to find another peak. This step is continued until all non-zero bins are associated with a peak.*

2.3 Hill-manipulation algorithm

Many existing segmentation techniques result in either under-segmented, or over-segmented, images because they cluster non-similar colours together, or they divide similar colours into several clusters, respectively. In this work, after a comprehensive study to a large collection of colour histograms that represent images in the different colour spaces, we found two main sources for incorrect segmentation that have not been addressed by the previous segmentation techniques.

First source of incorrect segmentation is having wide histogram hills. By studying the RGB and HSV colour spaces, we found that the eight corners, which represent eight distinguishable colours of the famous RGB cube, form eight distinguishable colours in HSV colour space. The distance between any two distinguishable colours is 60° . Thus, a histogram hill (cluster) should not exceed this distance to have coherent colours. Within this hill width, Λ_w , the hue changes gradually from one colour to the other. Therefore, it is obvious that grouping adjacent colours (within 60°) is visually accepted. But if the hill width Λ_w exceeds 60° , we will end up with non-accurate segmentation that groups non-similar colours in one cluster. To avoid having such clusters, we introduced a sliming factor $S_f = 60^\circ$ to restrict the clustering process. Each cluster must have colours within distance $< S_f$ from each other.

Second source of incorrect segmentation is having a slim but dominant histogram hills. That is having a dominant region that contains important details within it, where these details are regions of different degrees of the same colour as that of the dominant. A dominant hill is detected if $\Lambda_{size_l} > \Lambda_{av_size}$ and $\Lambda_w \leq S_f$, where Λ_{size_l} , Λ_{av_size} and Λ_w are the size of the current hill, average size of a hill in the image, and the width of the hill, respectively. In our algorithm, we handle this problem by dividing the dominant hill into several sub-hills where each sub-hill is of a different colour.

The hill-manipulation algorithm has two basic rules to detect these two sources of incorrect segmentation. These rules are applicable for many segmentation techniques that utilise a colour histogram. The rules are:

- if the distance between any two colours in hill (cluster) X is greater than S_f , then X is a wide hill
- if number of pixels in hill X is greater than the average number of pixels of all hills and the distance between any two colours is less than S_f , then X is a dominant hill.

Our algorithm starts by applying the hill-climbing algorithm (Ohashi et al., 2003). When clusters' peaks have been found and all image pixels have been associated with their corresponding peaks, a verification step takes place to ensure that all clusters obey the above two rules.

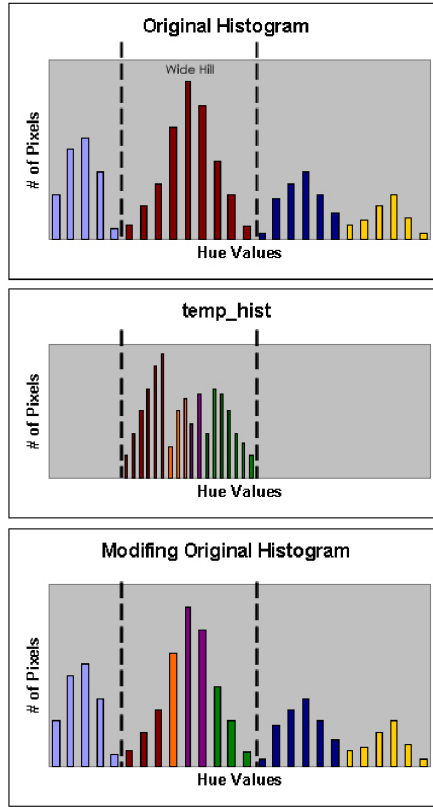
The hill-manipulation algorithm test each of the created clusters to determine whether it is a wide hill or a dominant hill, which are the sources of incorrect segmentation as explained earlier. If a cluster fails to comply with any of the two rules, then the slimming process takes place to divide the wide cluster into sub-clusters if the hill was detected to be wide, or dismantling process of the dominant hill so that each bin becomes a separate sub-hill if the hill was detected to be dominant.

Figure 1 illustrates the solution of both cases. The hill-manipulation algorithm is outlined as follows:

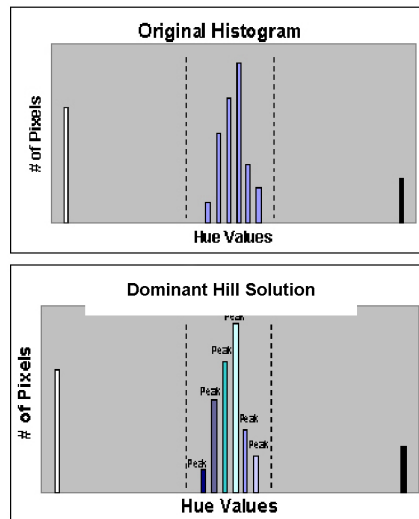
For cluster $i = 0$ to maximum number of clusters found using quantisation values 36, 10, 10 for H , S and V , respectively:

- 1 *If the hill of region i $\Lambda_w > S_f$, then i is a wide hill*
 - 1.1 *Create a new histogram $temp_hist$ with double the number of quantisation levels, where H , S and V are 72, 20 and 20, respectively.*
 - 1.2 *Map hill i into $temp_hist$. As a result, the pixels of hill i are redistributed and the wide hill is divided into several sub-hills.*
 - 1.3 *Call the hill-climbing algorithm using $temp_hist$ to find the peaks of the newly created sub-hills.*
 - 1.4 *Map these sub-hills back into the original histogram.*
- 2 *Else if $\Lambda_{size_l} > \Lambda_{av_size}$ and $\Lambda_w \leq S_f$, then i is a dominant hill*
 - 2.1 *Represent each of its bins as a separate sub-hill (cluster).*

Figure 1 Solutions to sources of incorrect segmentation: (a) wide hill solution and (b) dominant hill solution



(a)



(b)

3 Semantic grouping of segments

Our goal of grouping segments is to provide a compact, structured and meaningful representation of the image content. However, it is difficult to design a general purpose grouping structure that would satisfy general semantic representation of contents, owing to the diversity of both the image classes and the applications' needs. Hence, we propose a compromise between the applications' needs and the unsupervised general semantic segmentation, by grouping the segments produced by the hill-manipulation algorithm in a tree-like hierarchy based on some *spatio-visual grouping rules*.

In this paper, such a tree-like hierarchy is called a Semantic Segmentation Tree (SSeg-tree), whose nodes are obtained by grouping two or more children regions. The grouping is based on well-defined spatio-visual grouping rules that guarantee the formulation of semantic regions at the intermediate nodes of SSeg-tree. Therefore, before we introduce the SSeg-tree, we define the spatio-visual rules below.

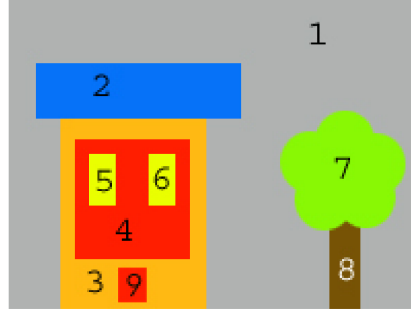
We aim at defining spatio-visual grouping rules that are suitable for a large number of applications. By investigating a large number of images, we observed the following:

- smaller regions that are included within a larger region are most likely part of a semantic region represented by the larger region
- regions are almost of an arbitrary shape
- objects most likely consist of coherent colour regions, which are adjacent or included in each other
- a background intersects with most regions in the image.

3.1 Spatio-visual grouping rules

Based on these observations, we defined the following spatio-visual grouping rules to construct semantic objects that satisfy the needs of a wide range of applications. Before introducing the rules, we need to identify some terms, let R_1, R_2, \dots, R_N be the set of segments (initial regions) that resulted from the segmentation process. Also, in the definition of the spatio-visual grouping rules let \subseteq, \cap and \oplus mean *included_in*, *intersect* and *group_with*, respectively. R_m is the relationship matrix that represents the *include* and *intersect* relationships between the segments. The *intersect* relationship will be represented by a positive value of one in R_m , and the *include* relationship is represented by negative values. The different values of *include* relationship represent the different levels of *include*. Figure 2 illustrates this, for example, we have segment 5 included in segment 4, so $R_m[5][4] = -1$, then segment 4 is included in segment 3, in a higher level of *include*; therefore, $R_m[4][3] = -3$. Notice that $R_m[9][3] = -3$, because segments 4 and 9 are in the same level of *include* to 3. So they must be represented by the same value. The odd negative numbers denote the *included_in* relationship, where the even negative numbers denote the *includes* relationship.

Figure 2 Constructing the relationship matrix R_m of an image with nine regions



		Region Number								
		1	2	3	4	5	6	7	8	9
Region Number	1	0	1	1	0	0	0	1	1	0
	2	1	0	1	0	0	0	0	0	0
	3	1	1	0	-4	0	0	0	0	-4
	4	0	0	-3	0	-2	-2	0	0	0
	5	0	0	0	-1	0	0	0	0	0
	6	0	0	0	-1	0	0	0	0	0
	7	1	0	0	0	0	0	0	1	0
	8	1	0	0	0	0	0	1	0	0
	9	0	0	-3	0	0	0	0	0	0

Rule 1: If $R_i \subseteq R_j$ and $-R_k \subseteq R_j \forall k \in [1..max_regions]$ and $k \neq i$, then $R_j = R_i \oplus R_j$.

As stated by Rule 1, if R_j includes only R_i , then they are grouped together to form R_j . Figure 3(a) illustrates this case, where R_1 includes R_2 , thus they are grouped in R_1 .

Rule 2: If $[R_i \dots R_k] \subseteq R_j$, && $[R_i \dots R_k]$ are in same level of include, then

1 Find the R_y and $R_z \in [R_i \dots R_k]$ with the minimum colour distance and group them $R_y = R_y \oplus R_z$

2 Repeat step 1 until one region is left, then apply Rule 1.

This rule states that if more than one region is included in R_j , then merge the *included_in* regions starting with the most similar in colour hierarchically, until there is only one region included in R_j . Then, apply Rule 1, see Figure 3(b).

Rule 3: If $R_i \subseteq R_j$, $R_j \subseteq R_k \dots R_l \subseteq R_n$, then first $R_j = R_i \oplus R_j$, then $R_k = R_k \oplus R_j, \dots$, until $R_n = R_n \oplus R_l$.

This rule states that if we have different levels of include, we should start grouping deepest level-of-include regions i and j , where $R_m[i][j] = -1$, then we move to the next deepest level of include, and so on until there is only one region included in another, then we apply Rule 1, see Figure 3(c).

Rule 4: If $R_i \cap R_j$ and $D(i, j) < D(k, l) \mid \forall k, l \in [1..max_regions]$, then $R_j = R_i \oplus R_j$.

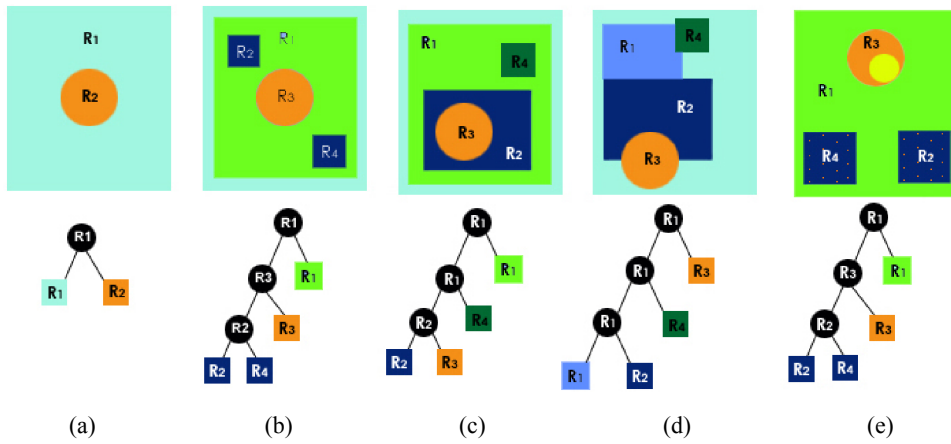
As illustrated in Figure 3(d), if there is a set of regions intersecting each other, the two most similar regions in terms of colour are grouped first, then the next two most similar regions, and so on until one region is left.

Rule 5: If $[R_j, \dots, R_k] \cap R_i$, && $\neg R_y \cap R_z$ where $R_y, R_z \in [R_j, \dots, R_k]$, then start by grouping:

- 1 Regions with the most similar colour-texture.
- 2 Repeat until no more grouping is available.

This rule deals with the background problem, it assumes that the regions with an adjacency relationship to all other regions is most probably the background. This rule is satisfied if the regions inside the background have no further adjacency between them. Since this is the last merging process, large objects have been identified. Thus, merging objects according to colour is inappropriate. Hence, we choose to merge according to colour-texture to group, see Figure 3(e).

Figure 3 Five cases to illustrate the spatio-visual grouping rules: (a) Rule 1; (b) Rule 2; (c) Rule 3; (d) Rule 4 and (e) Rule 5



4 Semantic Segmentation Tree (SSeg-tree)

The SSeg-tree is a hierarchical semantic representation of image content. The leaves of the SSeg-tree represent the initial regions (segments) produced by the hill-manipulation algorithm. The intermediate nodes of the tree represent regions that are obtained by grouping two or more children regions. The root, which is formed by grouping all the children regions, represents the whole image. The spatio-visual grouping rules lead to the formulation of semantic regions, or part of the larger semantic region, in the intermediate nodes.

Unlike the binary partition tree (Salembier and Marques, 1999), which groups regions based only on the visual homogeneity criterion, the SSeg-tree leads to more semantic grouping of regions since the grouping is based on intuitive spatio-visual

grouping rules (see Figure 2 for illustration). A semantic region (object) in an image is usually represented by one region, which may consist of several contiguous smaller regions; thus, we argue that our spatio-visual grouping rules are more reliable to lead to the formation of semantic regions.

The process of constructing the SSeg-tree requires that rules be applied in a certain order that reflects the priorities of these rules. For example, Rule 3 has the highest priority because it is most likely that regions included in another region is most likely part of the semantic object represented by the bigger object. The merging algorithm orders the execution of rules as follows:

- 1 *Construct the relationship matrix R_m*
- 2 *For every pair of regions, R_i and R_j , that satisfy Rule 3*
 - a *Merge regions R_i and R_j*
 - b *Add a node to the SSeg-tree to represent combined regions, R_i and R_j*
 - c *Update R_m to reflect the new change.*
- 3 *Repeat Step 2 using Rule 2, then Rule 1*
- 4 *For every pair of regions, R_i and R_j , that satisfy Rule 4*
 - a *Merge regions R_i and R_j*
 - b *If a new include relationship is formed as a result of the merge in step a*
 - i *Go to Step 2.*
- 5 *Apply Rule 5 until all segmented regions are inserted in the SSeg-tree.*

5 Experiments

We evaluated the performance of the hill-manipulation segmentation algorithm and the construction of the SSeg-tree using the spatio-visual grouping rules on 150 real images from seven different categories: people, animals, nature, buildings, flowers, transportation, and general. During the experiments, hill-manipulation algorithm performance was evaluated against well-known segmentation algorithms, namely the SRG and hill-climbing algorithms, to show the strength of our algorithm against other segmentation algorithms, as shown here.

5.1 Segmentation results

Figure 4 shows examples of the results of the Hill-Manipulation segmentation algorithm, where the original images are on the top row and the segmentation results are on the bottom images. The left image of Figure 4 illustrates the ‘accuracy’ of the segmentation method, the face initially is segmented as one part, forming a wide hill, then it is split by the hill-manipulation algorithm to form the eyes, nose, mouth and face capturing the details, which have different colours. Further, the other images show the accuracy of the hill-manipulation algorithm in capturing all the important details automatically. Even for a difficult image such as the image of a sea fish (the fourth from the left), where

the whole image is bluish, the hill-manipulation algorithm was able to capture all the necessary details of the image.

Figure 4 Segmentation in the HSV colour space. Top: original images. Bottom: images showing the generated segments

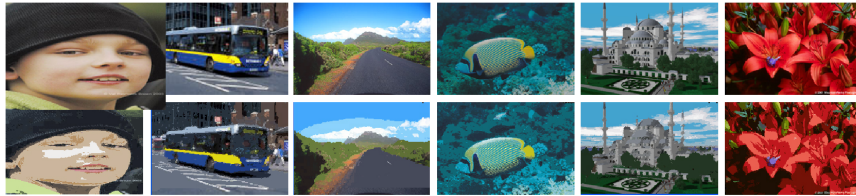
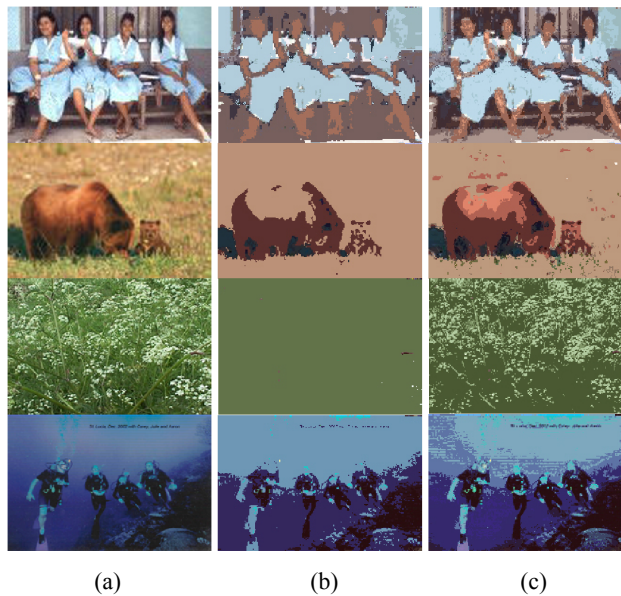


Figure 5 shows the result of comparing hill-manipulation algorithm with hill climbing (Ohashi et al., 2003). The hill-manipulation algorithm was able to extract more details because some regions are detected to wide hill or dominant hill (see Section 2.2) has been divided to new smaller regions, showing the details as outlined by the hill-manipulation algorithm. This can be seen in the bear image. The hill-climbing algorithm merged part of the bear body with the brown weeds forming one big dominant region. In the next image, this region was slimmed, and as a result, it separated the bear body from the background.

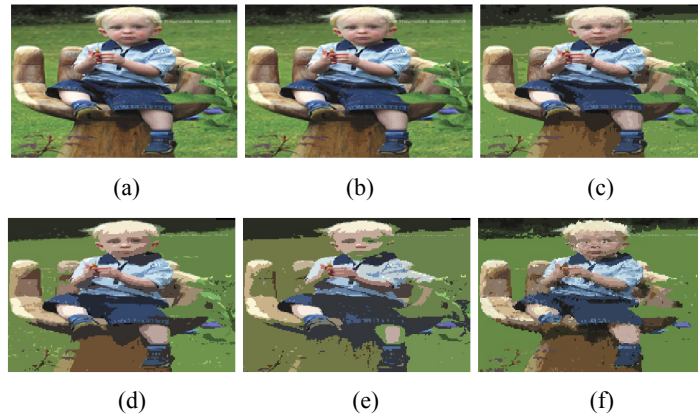
Figure 5 Comparison between hill climbing and hill manipulation: (a) original image; (b) segmented image using hill climbing and (c) segmented image using hill manipulation



Another comparison is shown in Figure 6, it is between the proposed method and SRG. We decided to choose SRG because it is a very popular segmentation method, and many algorithms have used it. In the used SRG algorithm, the seeds are chosen randomly and different threshold values have been used as shown in Figure 6. In Figure 6(b), the problem is the huge number of regions produced, this is because the threshold value is

very small (threshold = 10). Since the image has 30,024 pixels and the produced regions are 11,448, each region consists of, on the average, three pixels, which is considered as an over-segmentation of the image. When we increased the threshold to 30, the number of regions reduced to 2634, but still the number is big. When we used threshold to be 50, distinct regions started to merge like the left ear of the boy. Starting from threshold > 50, the results were unsatisfactory. On the other hand, hill manipulation produced better results without the need of any parameter or threshold values defined by the user and gave only 160 regions.

Figure 6 Comparison between SRG and improved hill climbing: (a) original image; (b) result of SRG using threshold = 10; (c) result of SRG using threshold = 30; (d) result of SRG using threshold = 50; (e) result of SRG using threshold = 70 and (f) result of hill manipulation



5.2 Construction SSeg-tree results

To evaluate the construction of a semantic SSeg-tree, which contains semantic regions in its intermediate nodes, we tested the spatio-visual grouping rules on real images that contain semantic regions with complicated structures. We show that our spatio-visual grouping rules naturally lead to the formulation of larger semantic regions in the intermediate nodes of the SSeg-tree. An example (panda image) of an SSeg-tree is shown in Figure 7. The image consists of 11 segments that are produced by the hill-manipulation segmentation algorithm, where these segments become the leaves of the SSeg-tree. Grouping starts by merging the small yellow and black parts of the nose because they have the deepest level of *include* relationship (Rule 3). Next, both eyes and the nose will be at the same level of *include*, so the algorithm groups the most similar pairs in terms of colour, which are the eyes. Then, the new node will be grouped with the nose (Rule 2). Next, the recently formed node is grouped with the panda's body since it is *included_in* it (Rule 1). No more *include* relationship for this image as all regions are intersecting each other. Thus, regions are grouped according to the most similar colour along with the adjacency relationship (Rule 4), until we reach the case where all the three remaining regions are adjacent to each other. These three regions are the panda, the left part of the background and the right part of the background. The two background regions will be grouped according to the most consistent colour-texture (Rule 5). Then, we end up having the root that represents the whole image. Notice that the longest branch in the tree

is the one that starts with merging the *include* relationship regions since it is the starting of the grouping process. Creation of the larger semantic regions in the intermediate nodes of the SSeg-tree is due to the use of the spatio-visual grouping rules.

Figure 7 An example of an SSeg-tree: grouping the 11 initial segments is based on the spatio-visual grouping rules

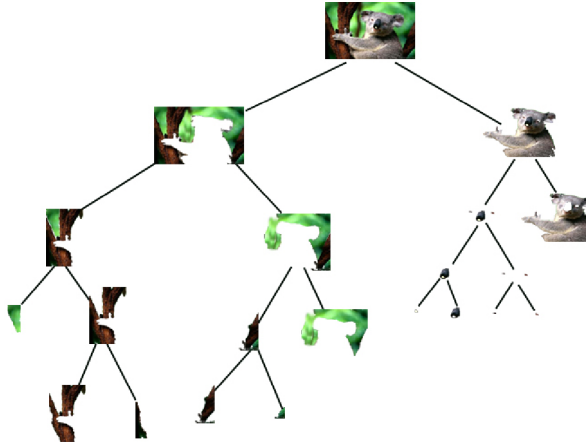


Figure 8 shows more examples of grouping regions using the well-defined spatio-visual grouping rules. On the top is a SSeg-tree of a car image, in which the segments of a car object were grouped into one node as indicated in Figure 8 (top). At the bottom is an SSeg-tree of a building image, in which the segments of the building object were grouped in one node as indicated in Figure 8 (bottom). These examples show the effectiveness of our spatio-visual grouping rules. Further, Table 1 shows the average effectiveness of the spatio-visual grouping rules. The effectiveness was measured as the percentage of images of which all distinct objects are fully recognised and assigned separate nodes in the SSeg-tree. Note that the average effectiveness of the spatio-visual grouping rules is quite high for all categories, which shows the feasibility of the spatio-visual grouping rules.

Figure 8 SSeg-tree construction example: (Left) original images, and (right) the corresponding SSeg-trees, where the car node and building node are indicated in both trees

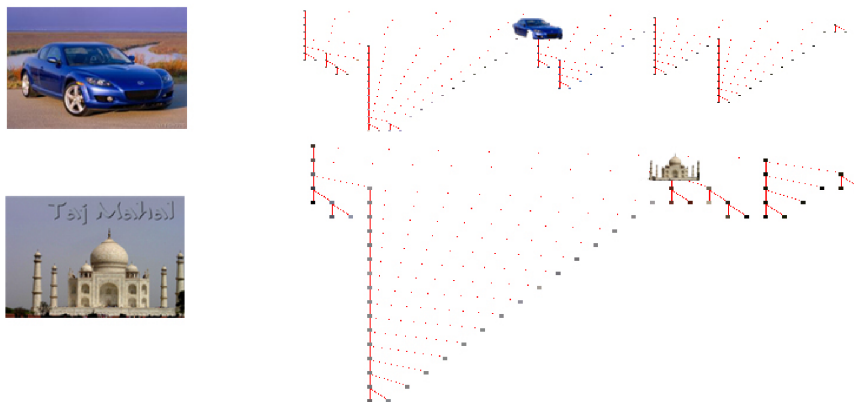


Table 1 Average effectiveness of the SSeg-tree construction algorithm

<i>Image category</i>	<i>Percentage of images with correctly recognised objects</i>
People	75
Animals	83
Nature	80
Buildings	92
Flowers	88
Transportation	86
General	80

The constructed SSeg-tree can be used very effectively in the CBIR systems, where the user provides the system with a sample query that represents the target object. The constructed SSeg-tree of all images in the DB is checked one by one to examine whether they contain the queried object or not

This is accomplished by comparing the queried object with the root, left, and right nodes of the SSeg-tree, and then the search advances towards the node that holds the maximum similarity with the query object. This similar child node becomes the new root. The search continues until an exact match is reached, the current root node is most similar to the query object, or a leaf node is reached, where in this case the leaf node will be the best match. Furthermore, SSeg-tree can be used to enhance the effectiveness of various multimedia indexing applications, such as the systems in Waluyo et al. (2005), Yang and Hurson (2005), Taniar and Rahayu (2003, 2004) and So (2005).

Another application that can use the SSeg-tree is the filtering systems, whose aim is to remove a specific unwanted part of an image, or group of images. The removed object is replaced by the background, or any specific filling colour according to the application needs. In our implementation of the filtering system, the user will browse all the available objects in the image, then he selects the object to be filtered. The system then searches for that object in the SSeg-tree, and the pixels associated with the found object node are changed to one of the available choices such as the background, white and black colours.

6 Conclusion and future work

We have presented a novel hill-manipulation image segmentation algorithm that generates visually coherent segments and a new SSeg-tree to represent the content of an image. The segmentation algorithm is non-parametric and unsupervised. As demonstrated by our experiments, we can conclude that our segmentation method has the following properties:

- it produces visually coherent segments
- it works on general images, i.e., no a priori knowledge is assumed
- it does not need any hand-tuning of parameters during segmentation.

The proposed segmentation method proves to work very well on all image categories, even with low contrast, blurred and noisy images. Furthermore, it shows a great improvement when compared with the SRG and hill-climbing methods. The complexity

here is not a comparison point, because the whole segmentation process and building the SSeg-tree is done once offline for every image, and the result is stored in a database.

Moreover, we have introduced the SSeg-tree that contains semantic regions in its intermediate nodes. We have argued that the spatio-visual grouping rules are intuitive and thus naturally lead to the formulation of semantic regions in the intermediate nodes of the SSeg-tree, unlike the approach of grouping regions solely based on visual coherence. The experiment shows that the system was able to construct meaningful objects from a wide range of examined images, and present them as separate nodes in the SSeg-tree. The generated SSeg-tree is useful in a number of applications, such as information retrieval, where the SSeg-tree helps in identifying region(s) of interest corresponding to a given query. Another application that can use the SSeg-tree is the filtering technique, where the tree structure helps to identify undesired regions in the image, remove their content (simplify the regions), and merge them with neighbouring ones.

As a future work, we will try to segment images according to features other than colours, like texture, shape, etc. For example, the shape could be useful to identify the region's boundaries. Another interesting subject for future research is to consider more features when building the SSeg-tree to improve its quality. Some of the features that can be added are shapes and textures. Thus, users can specify the searching criteria, i.e., which features to base the search on. Moreover, a combination of features like colour and shape will be very useful in improving the search results.

One more interesting subject to be researched in the future is the use of SSeg-tree in tracking video objects. For example, the SSeg-tree will be constructed for a given frame. Then all the objects in that frame will be represented. After having the SSeg-tree, objects will be identified, therefore, they can be tracked in the following frames.

References

- Comaniciu, D. and Meer, P. (2002) 'Mean shift, a robust approach toward feature space analysis', *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 5, May, pp.1–18.
- Garrido, L., Salembier, P. and Garcia, D. (1998) 'Extensive operators in partition lattices for image sequence analysis', *EURASIP Signal Processing*, Vol. 66, No. 2, April, pp.157–180.
- Green, P. (1995) *Understanding Digital Colour*, Blueprint, an imprint of Chapman and Hall, USA.
- Leung, T. and Malik, J. (1988) 'Contour continuity in region based image segmentation', *5th Euro. Conference on Computer Vision*, June, Germany.
- Malik, J., Belongie, S., Leung, T.K. and Shi, J (2001) 'Contour and texture analysis for image segmentation', *Int'l Journal of Computer Vision*, Vol. 43, No. 1, pp.7–27.
- Ohashi, T., Aghbari, Z. and Makinouchi, A. (2003) 'Hill-climbing algorithm for efficient colour-based image segmentation', *IASTED International Conference on Signal Processing, Pattern Recognition and Applications*, SPPRA, Rhodes, Greece, June.
- Pauwels, E.J. and Frederix, G. (1999) 'Finding salient regions in images: non-parametric clustering for image segmentation and grouping', *Journal of Computer Vision and Understanding*, Vol. 75, Nos. 1–2, pp.73–85.
- Salembier, P. and Marques, F. (1999) 'Region-based representation of image and video: segmentation tools for multimedia services', *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 9, No. 8, December, pp.1147–1167.
- Schluter, D., Kummert, F., Sagerer, G. and Posch, S. (2000) 'Integration of regions and contours for object recognition', *IEEE 19th Int'l Conference on Pattern Recognition*, September, Barcelona, Spain.

- So, S. (2005) 'Efficient image indexing and retrieval over mobile devices', *Journal of Mobile Multimedia*, Rinton Press, Vol. 1, No. 2, pp.149–160.
- Taniar, D. and Rahayu, J.W. (2003) 'Global B+ tree indexing in parallel database systems', *Intelligent Data Engineering and Automated Learning, Lecture Notes in Computer Science*, Springer-Verlag, Hong Kong, Vol. 2690, pp.701–708.
- Taniar, D. and Rahayu, J.W. (2004) 'Global parallel indexing for multi-processors database systems', *Information Sciences*, Elsevier, Vol. 165, Nos. 1–2, pp.103–127.
- Waluyo, A.B., Srinivasan, B., Taniar, D. and Rahayu, J.W. (2005) 'Incorporating global index with data placement scheme for multi channels mobile broadcast environment', *Embedded and Ubiquitous Computing, Lecture Notes in Computer Science*, Springer-Verlag, Nagasaki, Japan, Vol. 3824, pp.755–764.
- Yang, B. and Hurson, A.R. (2005) 'Hierarchical semantic-based index for ad hoc image retrieval', *Journal of Mobile Multimedia*, Rinton Press, Vol. 1, No. 3, pp.235–254.