Hierarchical Colour Image Segmentation by Leveraging RGB Channels Independently

Sheikh Tania, Manzur Murshed, Shyh Wei Teng, and Gour Karmakar

Federation University Australia, Gippsland Campus, Churchill Vic 3842, Australia
{ttania,manzur.murshed,shyh.wei.teng,gour.karmakar}@federation.edu.au

Abstract. In this paper, we introduce a hierarchical colour image segmentation based on cuboid partitioning using simple statistical features of the pixel intensities in the RGB channels. Estimating the difference between any two colours is a challenging task. As most of the colour models are not perceptually uniform, investigation of an alternative strategy is highly demanding. To address this issue, for our proposed technique, we present a new concept for colour distance measure based on the inconsistency of pixel intensities of an image which is more compliant to human perception. Constructing a reliable set of superpixels from an image is fundamental for further merging. As cuboid partitioning is a superior candidate to produce superpixels, we use the agglomerative merging to yield the final segmentation results exploiting the outcome of our proposed cuboid partitioning. The proposed cuboid segmentation based algorithm significantly outperforms not only the quadtree-based segmentation but also existing state-of-the-art segmentation algorithms in terms of quality of segmentation for the benchmark datasets used in image segmentation.

Keywords: Cuboid segmentation · Agglomerative merging · Colour Image segmentation.

1 Introduction

Image segmentation is a significant pre-processing step for most image processing techniques such as object recognition, object-based image classification, and content-based image retrieval. These computer vision techniques are being used in many cutting edge applications, including the latest medical imaging, traffic control system, remote sensing, and video surveillance.

There are mainly two different approaches to image segmentation [2]: (i) edge detection-based and (ii) region-based segmentation. Edge detection aims to detect edges between different group of pixels and does not guarantee to detect the closed object contours. Region-based segmentation divides an image into disjoint regions. Image derivative was the basic concept of early edge detection algorithms [5]. Recently, gPb-OWT-UCM [2] put forth a contour detector and combined it with a hierarchical image segmentation. Again, there exists a wide spectrum of region based techniques, including normalized cut [24], mean shift

[7], random walk [11], region grow [30], graph-based [9], watershed [32], fuzzy c-means clustering [17], JSEG [8], and hierarchical segmentation [27], [33], [18].

Even with the overabundance of segmentation techniques, there are demands for new techniques because all the methods could not be suitable for all types of applications. For example, segmentation techniques for medical imaging cannot be applied to natural images. A recent work [21] proposes an unconventional way of segmentation, where the output of the algorithm comes in the form of cuboids. The motivation for the task is to facilitate efficient and effective content-based image retrieval and video coding. In this era of multimedia, most video contents are transmitted and also stored in the compressed form, thus the post-processing (e.g., video object recognition [12], action recognition [6], and video summarization [3]) needs to decompress them. Embedding relevant coding metadata can facilitate the post-processing exempting the decompression. The only available data are in the form of rectangular blocks in the compression format. The arbitrarily shaped partitions of video segmentation techniques are not appropriate to serve the purpose. The rectangular-shaped output of cuboid segmentation can serve it best.

Cuboid segmentation [21] uses simple statistical features (e.g., mean and variance), derived from the distribution of pixel values. The main idea is to use a greedy heuristic to recursively split a rectangular image into two rectangular halves with an optimal split-line, orthogonal to one of the axes so that the value of the distance metric is maximized. The recursion is terminated when the distance is below a threshold or when the targeting granularity is reached. Cuboid segmentation has some outstanding benefits. It is very simple to implement and preserves spatial relationships among neighbours. The complexity is bounded to linear form by utilizing the integral imaging approach [29] to generate the colour moments. Again, the cuboid segmentation is very fast as the depth of recursion is bounded logarithmically. Moreover, it can also take advantages of matrix processing and the specialized hardware e.g., the graphics processing unit. Again, the hierarchical approach is amenable to parallel processing. Lastly, the output of segmentation (i.e., the boundary of cuboid segments) can be stored using only the four corner points of each cuboid.

The cuboid segmentation can easily be confused with the quadtree (QT) decomposition [26] as both of them produce segments of rectangular shape. QT decomposition is widely used for coding purposes. The main idea is to segment the image into homogeneous and heterogeneous parts to allocate fewer bits to homogeneous regions and more bits to those regions that contain additional detail and sharper features. Although both the algorithms partition an image recursively, they differ from each other greatly. First of all, the original quadtree decomposition requires images not only with equal height and width but also the value of height and width must be of the power of two but cuboid segmentation does not care about the image size. Again, the QT decomposition divides the image into four equal-sized blocks in each iteration until a specified homogeneity criterion is achieved. In contrast, the cuboid segmentation does not use any arbitrary division. Rather it uses the local optimum distance to divide the image into two regions based on a specific feature space (e.g., colour, texture, or both).

However, the very first cuboid segmentation algorithm suffers from over-segmentation of regions with a similar colour. This is because it quantizes the HCL colour model into a fixed number of colours in cubic spaces and uses the Euclidean distance measure in that cubic space. In CSeg'18 [28], the HCL colour space and quantization is ignored and l_∞-norm is used while selecting the optimal split s^* among all possible splits. This means for each possible split, a dominant RGB channel was selected for which the distance between the halves in that channel is the maximum of all three channels. As an image (or sub-image thereafter) is static while considering the optimal split, the distance-dependent dominant channel selection leads to inconsistency in the decision process as different channels may become dominant for different splits of the same image. Again, in the CSeg'18, contrast measure has been used to determine whether a cuboid needs to be partitioned further. The contrast value used in CSeg'18 can be influenced by the local contrast, which limits the efficacy of accurate decision making for cuboid partitioning.

In this paper, the first shortcoming of CSeg'18 has been overcome in two stages. A dominant channel is selected first among the three RGB channels where the variance is the maximum. Then the best split based on the maximum distance between the cuboid halves is chosen in that dominant channel. The variance represents the variation among the pixels of a cuboid more accurately. Hence, the variation of contrasts within a cuboid indicates the use of variance in decision making for partitioning will improve the performance of segmentation. Therefore, we use the variance of any cuboid instead of its contrast to decide further partitioning of it, thus addressing the second shortcoming of CSeg'18.

We use the statistical property of the pixel values to find out the maximum contrast between the foreground and the background of an image to eventually perform the cuboid segmentation algorithm to distinguish regions. Besides, we include the variance as a threshold to determine the homogeneity of image regions. Because of perceptually distinctive differences in the contrast between foreground and background objects and the more intuitive distance value, the proposed cuboid segmentation algorithm can detect and separate the background and foreground objects more accurately.

The idea of superpixels is being used in many segmentation techniques [22], [15], [16], [33] as they are more convenient to compute region-based image features by reducing the number of image primitives significantly. Although the output of cuboid segmentation is a set of cuboids, these cuboids can admittedly be considered as superpixels. Among the existing superpixel segmentation techniques, Turbopixels [14] and SLIC [1] have linear time complexity with the ability to control the number of segments. However, in both the algorithms the initial seeds evolve over a smaller spatial extent resulting in a collection superpixels of almost similar size. On the contrary, cuboid segmentation enforces strong spatial relationships among neighbouring pixels and the homogeneity criteria used to terminate the split process results in a combination of cuboids of homogeneous image regions with different sizes.

In this paper, we use cuboids as superpixels and then exploit the hierarchical agglomerative merging approach to group them into meaningful regions describing objects in an image. By controlling the number of cuboids, the complexity of further merging process is also bounded to linear form that ensures the linear complexity of total segmentation process. Comprehensive experiments are performed using four standard measurements for quantitative evaluation namely the Probabilistic Rand Index (PRI) [31], Global Consistency Error (GCE) [19], Variation of Information (VoI) [20] and Boundary Displacement Error (BDE) [10] on benchmark datasets (BSDS500 [19] and MSRC [25]). These are widely used to evaluate segmentation performance. In comparing the results, we have implemented the same merging algorithm using the output of CSeg'18 and quadtree. The experimental results exhibit better performance for the CSeg'19 based approach than both the CSeg'18 based and quadtree-based approaches. In addition, the proposed approach gains in the evaluation criteria against some state-of-the-art techniques.

The rest of the paper is organized as follows. Section 2 depicts the theory behind our proposed technique and the proposed algorithm. The performance studies are presented in Section 3. Section 4 provides the conclusion.

2 Proposed Technique

In this section, we present our partitioning technique that uses simple statistical features of pixel intensities to partition an image into a collection of cuboids. Next, we describe the hierarchical merging process of the cuboids to form meaningful image regions.

2.1 Cuboid Segmentation

Variance and its impact on pixel data: Variance is a measurement that depicts the dispersion of a dataset relative to its mean, or expected value. For a dataset $x_1, x_2, ..., x_N$ with mean $\overline{x}$, the variance σ^2 can be denoted as

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2. \tag{1}$$

A low variance indicates that the data points tend to be close to the mean, while a high variance indicates that the data points are spread out over a wider range of values. We used this insight to select the dominating channel in each iteration of our proposed algorithm. As RGB colour space ($0 \leq R, G, B \leq 255$) is orthogonal in nature, the dominating channel among R, G or B is the one that has more pixel values far from their mean. As natural images may contain different objects of different colours, the dominant channel for the different parts of an image might not be the same (Figure 1). Therefore, in each iteration, we

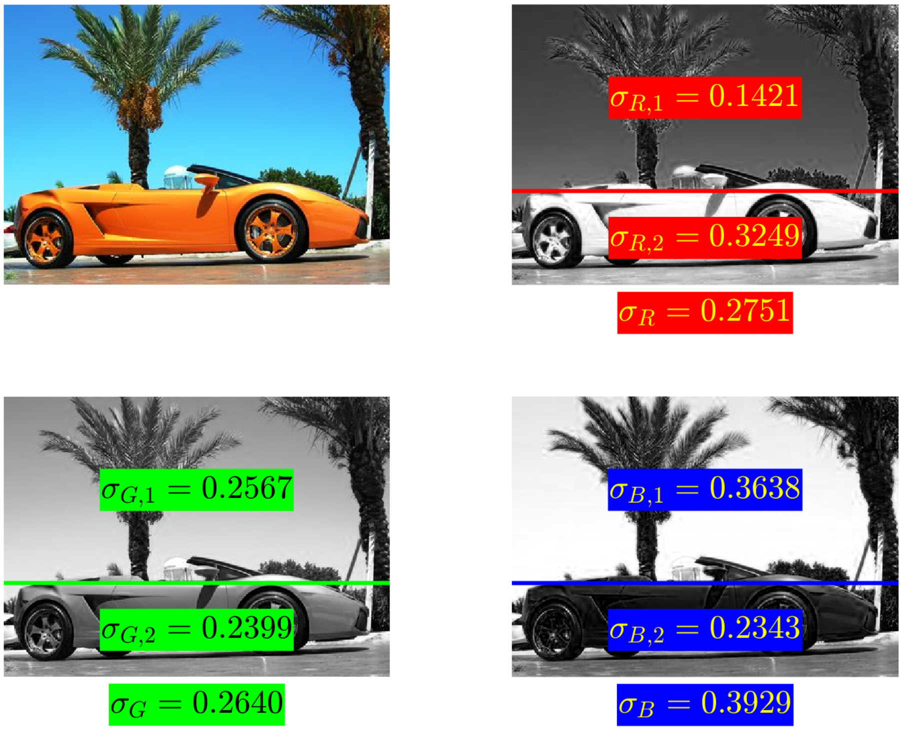

Fig. 1. In clockwise order from top-left: original image, channel R, channel B, and channel G. While the image and Cuboid 1 (top) have the dominant B channel, Cuboid 2 (bottom) has dominant R channel.

first select the dominant channel and then perform the cuboid segmentation only on that channel. As a consequence, the more we separate different objects or parts of an image, the dominating channel turns more explicit. As RGB colour space is not perceptually uniform, the difference of the centroid (RGB mean) between any two distinct part of the image is not perceptually consistent. Since each channel of the RGB colour model is more intuitive separately than the whole colour model, the distance of the centroid in a single channel will be more perceptually consistent. This paves the way of manipulating the distance of their mean colour in a single channel which conforms with human perception closely.

Proposed algorithm: Let $I_{X,Y} = (R^I_{X,Y}, G^I_{X,Y}, B^I_{X,Y})$ denote an RGB image of size $X \times Y = n$. If $\sigma^2_{\max}(I_{X,Y})$ defines the maximum variance among the three colour channels, the dominating colour channel, Ch of an image can be defined as

$$Ch = \begin{cases} R, & \text{if } \sigma^2_{\max}(I_{X,Y}) = \sigma^2(R^I_{X,Y}); \\ G, & \text{else if } \sigma^2_{\max}(I_{X,Y}) = \sigma^2(G^I_{X,Y}); \\ B, & \text{otherwise.} \end{cases} \tag{2}$$

If $\mu_I^{R,G,B}$ is the first order raw RGB colour moment of image I, then the colour-contrast distance between two images I and J can be estimated as,

$$D^*_{I,J} = |\mu_I^{Ch} - \mu_J^{Ch}|. \tag{3}$$

To determine the homogeneity property of regions we use a threshold called variance threshold, V_{X,Y,n_s} and the variance of a region is the average of the variances in each R, G and B channel

$$\sigma^2(I_{X,Y}) = \frac{1}{3}\left(\sigma^2(R^I_{X,Y}) + \sigma^2(G^I_{X,Y}) + \sigma^2(B^I_{X,Y})\right). \tag{4}$$

Image I can be split into two sub-cuboids I_i^1 and I_i^2 of sizes $i \times Y$ and $(X-i) \times Y$ pixels respectively, using a vertical line $x = i + 0.5$ in $X - 1$ ways with $i \in 1, 2, ..., X - 1$. Similarly, it can be split into two sub-cuboids I^1_{X-1+j} and I^2_{X-1+j} of sizes $X \times j$ and $X \times (Y - j)$ pixels respectively, using a horizontal line $y = j + 0.5$ in $Y - 1$ ways with $j \in 1, 2, ..., Y - 1$.

For a user-defined number of cuboid, n_s, a split is considered valid only if the variance of the cuboid meets the variance threshold and the area of the cuboid meets the area threshold

$$v(s|I_{X,Y}) = XY \geq A \wedge \sigma^2(I_{X,Y}) \geq V_{X,Y,n_s}. \tag{5}$$

The colour contrast distance of the half-cuboids is the objective function

$$f(s|I_{X,Y}) = D^*_{I_s^1, I_s^2}. \tag{6}$$

Then the greedy optimization heuristic to find the best split of I from the possible $X + Y - 2$ ways as

$$\begin{gathered} \max_{1 \leq s \leq X+Y-2} f(s|I_{X,Y}) \\ \text{subject to } v(s|I_{X,Y}). \end{gathered} \tag{7}$$

A hierarchical partitioning algorithm may be designed by recursively splitting the two half-cuboids using the optimal split s^*. The algorithm terminates when all possible ways of splitting are found to be obsolete that means if one or both of the variance threshold and area threshold is not satisfied. The algorithm is now formally presented below as Algorithm 1.

2.2 Merging

The output of the previous stage is a full binary tree where the original image represents the root and the segmented cuboids represent the leaf nodes. In

Algorithm 1 CSeg'19 $(I_{X,Y})$

```
$d_{\max} = 0$;
$s^* = 0$;
for $s = 1, 2, ..., X + Y - 2$ do
    if $v(s|I_{X,Y,n_s}) \wedge f(s|I_{X,Y}) > d_{\max}$ then
        $d_{\max} = f(s|I_{X,Y})$;
        $s^* = s$;
    end if
end for
if $d_{\max} > 0$ then
    CSeg'19$(I^1_{s^*})$
    CSeg'19$(I^2_{s^*})$
end if
```

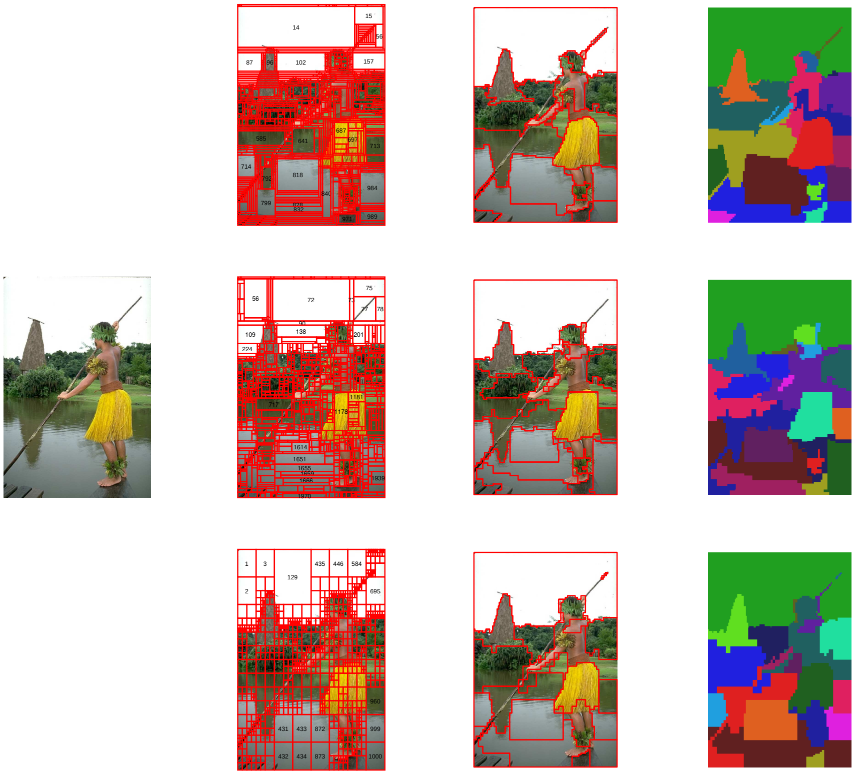

Fig. 2. Visual comparison of CSeg'19 (top), CSeg'18 (middle) and quadtree (bottom) based segmentation outcome.

this stage, we perform merging of the leaf cuboids using hierarchical agglomerative clustering of Ward's minimum variance method [13] where the criterion for choosing the pair of clusters to merge at each step is based on the optimal value of an objective function. Ward's minimum variance criterion minimizes the total within-cluster variance. The merging is accomplished in two steps. In the first step, we merge the pair of sibling leaf nodes that contribute minimum impact in the division process of their parent node.

Let C_1 be an internal node of size n_1 and $C_{1.1}$ and $C_{1.2}$ be two child nodes of it with size $n_{1.1}$ and $n_{1.2}$. If σ_1^2, $\sigma_{1.1}^2$ and $\sigma_{1.2}^2$ be the variance of the parent (i.e. the variance if they are merged) and two leaf nodes correspondingly, the within cluster variance of the C_1 and C_2 can be defined as,

$$\sigma_{C_{12}}^2 = n_1\sigma_1^2 - (n_{1.1}\sigma_{1.1}^2 + n_{1.2}\sigma_{1.2}^2). \tag{8}$$

If $C \in C_1, C_2, ..., C_N$ be the set of internal cuboids having two leaf nodes as children, then in each iteration the internal cuboid having minimum value of impact factor will be considered to be merged. In the next step, we consider merging of those pair of nodes that are a neighbour of each other using the same criteria. The first step of merging eliminates some trivial splitting while the second step emphasizes merging of the image regions partitioned into two different subtrees as well as into the same subtree.

3 Results and Discussion

In this section, we present both the quantitative and the visual results of our proposed technique and compare them with others.

3.1 Evaluation Metric and Dataset

In our experiment, we use four quantitative metrics widely used in evaluating performance of image segmentation techniques, namely PRI [31], GCE [19], VoI [20] and BDE [10]. We conducted our experiments using the two widely used benchmark image datasets: Berkeley Segmentation Data Set 500 (BSDS500) [19] and Microsoft Research Cambridge (MSRC) dataset [25]. The BSDB500 dataset contains 500 natural images with 300 train, and 200 test images and each image has multiple ground truths. The MSRC dataset comprises 591 natural images of 21 object classes. Among the 21 classes, we used 7 classes having 203 images. To compare our proposed technique, we have implemented another algorithm by exploiting the output of quadtree decomposition and then using the same hierarchical merging algorithm we used in our technique.

3.2 Performance Evaluation

To preserve the size of images, we implemented the quadtree algorithm depending on the aspect ratio of images. For both techniques, we used the same value of area threshold $A = 4$ for all the images. The value of the variance threshold varies according to the contrast of an image. As an image is divided into two blocks in cuboid segmentation, while it is divided into four blocks in quadtree decomposition, we set the variance threshold in such a way that in the split step, the number of cuboids, n_s remain at least 2000 in both the techniques. It is evident that if the number of cuboids is higher in merging phase, the more accurate segmentation results are expected to have in the final segmentation output. We set the number of cuboids, n_s to be equal for both QT and Cseg'19 to ensure none of them is penalized. According to the resolution of images, we set $n_s = 1000$ for BSDS500 and $n_s = 550$ for MSRC dataset, which ensures the complexity of the whole process to be linear. We also implement the merging algorithm using the output of CSeg'18 on both of the datasets.

Table 1. Performance comparison of CSeg'19, CSeg'18 and quadtree (QT) based segmentation for BSDS500 and MSRC dataset

	BSDS500				MSRC			
	PRI↑	VoI↓	GCE↓	BDE↓	PRI↑	VoI↓	GCE↓	BDE↓
CSeg'19	**0.813**	**1.785**	**0.089**	**11.135**	**0.804**	**1.188**	**0.116**	**9.121**
CSeg'18	0.808	1.817	0.097	11.470	0.787	1.195	0.121	9.125
QT	0.809	1.836	0.098	11.224	0.799	1.196	0.122	9.413

Table 2. Performance comparison of the proposed technique with state-of-the-art methods for BSDS500 dataset using 300 training images used in the reported result

	BSDS500			
	PRI↑	VoI↓	GCE↓	BDE↓
NCut [24]	0.79	1.84	-	-
SWA [23]	0.80	**1.75**	-	-
ICM [27] (only colour)	0.79	1.79	-	-
CSeg'19	**0.82**	**1.75**	**0.09**	**11.34**

As both techniques yield hierarchical region trees, there can be many possible segmentation outcomes. Selecting a single outcome from them involves personal choice because every person has their perception of segmentation. We select the criteria OIS (Optimum Image Scale) [2] where the optimum number of segments or regions, n_r is selected by an oracle on a per-image basis. The quantitative evaluation of both techniques is shown in Table 1. The CSeg'19 based technique outperforms both the CSeg'18 based and quadtree based techniques for all the metrics for both BSDS500 and MSRC datasets.

In Table 2, we also compare the results of our proposed cuboid based technique with those of several state-of-the-art techniques reported in their papers where 300 training images of BSDS500 data set were used.

Among those, Iterative Contraction and Merging (ICM) [27] is another hierarchical image segmentation algorithm. In that work, they provided their per-

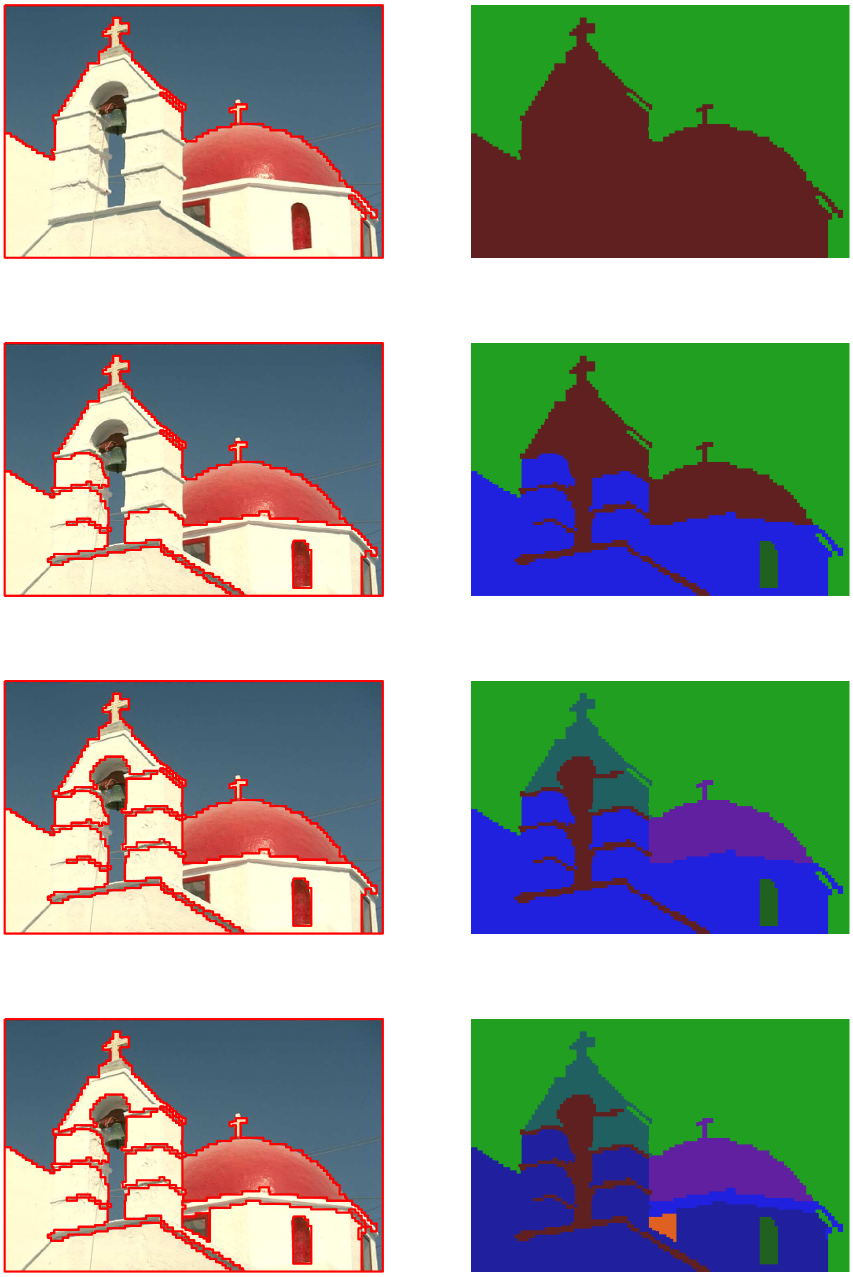

Fig. 3. (Top to Bottom) Segmentation result of CSeg'19 based hierarchical image segmentation for n_r = 2, 4, 6, and 8.

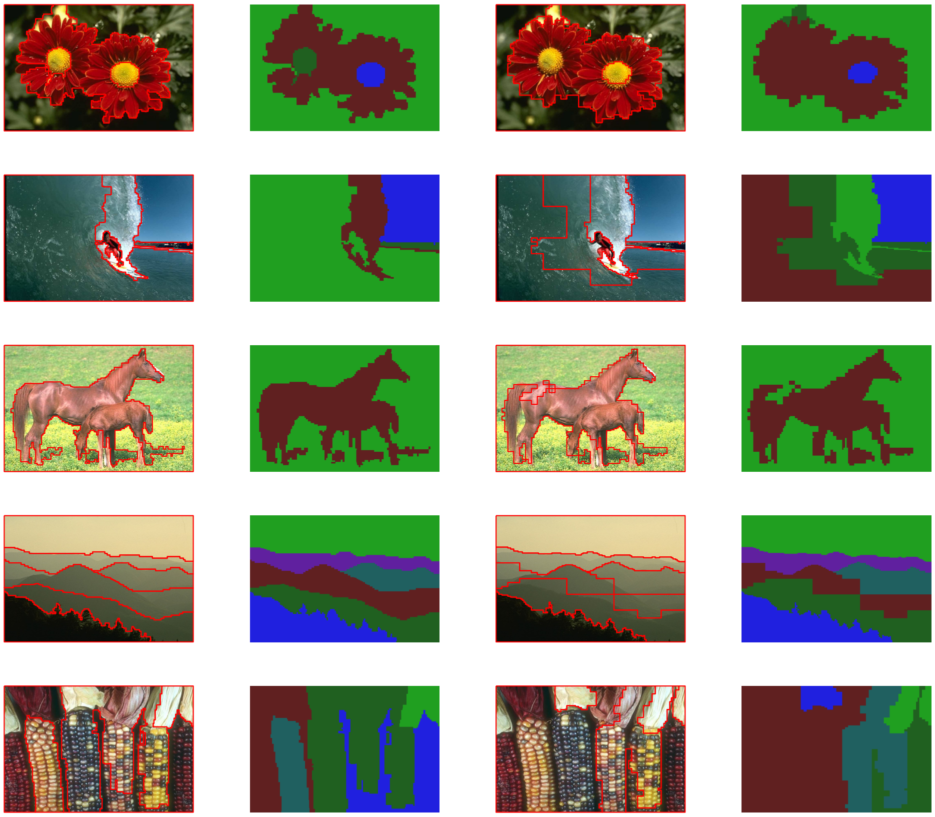

Fig. 4. Visual comparison of CSeg'19 based (left 2 columns) and quadtree based (right 2 columns) hierarchical segmentation techniques (top to bottom) $n_r = 4$, 4, 2, 5, and 5. First and third column present segment boundaries, and the second and fourth column present corresponding region maps.

formance evaluation over a different combination of colour, size, texture, border, and spatial intertwining factors. As our proposed technique uses only the colour feature, we compare our performance with their performance that is based on colour. All the benchmark scores rather than the ICM are collected from [2] where they choose only PRI and VoI as evaluation metric and reported values are presented by two digits followed by the decimal point. Our proposed CSeg'19 based technique performs better for both PRI and VoI than all other techniques.

Figure 2 imparts the qualitative performance of CSeg'19, CSeg'18, and QT based segmentation outcomes for $n_r = 20$. The outcome of CSeg'19 indicates larger cuboids for large homogeneous regions (sky and lake water) and smaller cuboids at curved region boundaries. In contrast, QT decomposition produces relatively smaller blocks for both large and small homogenous regions. As a consequence, we can observe some visually prominent regions like the stick and the hula skirt in CSeg'19 based method than the quadtree based method. Besides, CSeg'18 based approach cannot distinguish the hand and the upper portion of the stick properly. Figure 3 exemplifies the hierarchical segmentation showing the different number of image regions. We can see that the image contents become more explicit as the number of segment increases. Again Figure 4 illustrates some output of both the techniques for the same number of segments. Quadtree based results present more distortions than CSeg'19 based results in separating the regions.

3.3 Complexity Analysis

According to master theorem of divide and conquer recurrences [4], if $T(n)$ denotes the total time for the algorithm on an input of size n, and $f(n)$ denotes the amount of time taken at the top level of the recurrence then the time can be expressed by a recurrence relation of the form

$$T(n) = aT(\frac{n}{b}) + f(n) \tag{9}$$

where a is the number of subproblems in the recursion and b is the factor by which the subproblem size is reduced in each recursive call. By comparing the asymptotic behaviour of $f(n)$ with $n^{log_b a}$, there are three possible cases,

$$T(n) = \begin{cases} \Theta(n^{\log_b a}), & \text{if } f(n) = O(n^{\log_b a}); \\ \Theta(n^{\log_b n} \log^{k+1} n), & \text{if } \Theta(n^{\log_b a} \log^k n), k \geq 0; \\ \Theta(f(n)), & \text{if } f(n) = \Omega(n^{\log_b a}). \end{cases} \tag{10}$$

As CSeg'19 recursively divides an image $I_{X,Y}$ of $n = X \times Y$ pixels at the optimal split into two cuboids, we may assume $a = 2$. For the sake of simplicity, we may also assume $b = 2$, as each cuboid will have roughly $O(n/2)$ pixels, and $f(n) = O(\sqrt{n}) = \Omega(n^{\log_b a})$, as the optimal split is selected from $X + Y - 2$ possible splits, and $X = O(\sqrt{n})$ and $Y = O(\sqrt{n})$. Hence, by (10), $T(n) = \Theta(n)$.

In the merge step, the complexity of hierarchical clustering is $O(N^2)$ for input of size N. We used the number of cuboids in the merging step in such a way that $N = O(\sqrt{n})$. Thus the overall complexity of the proposed algorithm is $\Theta(n) + O(\sqrt{n}^2) = O(n)$, i.e., the complexity order of CSeg'19 is *linear*.

4 Conclusion

In this paper, we have introduced an innovative cuboid partitioning method to split an image into homogeneous regions. By using the initial cuboids, we have gradually merged adjacent regions into greater ones and finally formed a hierarchical tree. In each step of merging, we have adopted the minimum variance criteria to detect the most similar region pairs among all the neighbouring region pairs. We have used the colour feature only. The computational complexity of our proposed segmentation technique is linear. Results of quantitative evaluation admit the performance of the proposed technique is superior over the existing state-of-the-art methods, including quadtree based segmentation technique.

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: Slic superpixels compared to state-of-the-art superpixel methods. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(11), 2274–2282 (Nov 2012)
2. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(5) (May 2011)
3. Bagheri, S., Zheng, J.Y., Sinha, S.: Temporal mapping of surveillance video for indexing and summarization. Computer Vision and Image Understanding **144**, 237 – 257 (2016), http://www.sciencedirect.com/science/article/pii/S1077314215002581, individual and Group Activities in Video Event Analysis
4. Bentley, J.L., Haken, D., Saxe, J.B.: A general method for solving divide-and-conquer recurrences. SIGACT News **12**(3), 36–44 (Sep 1980), http://doi.acm.org/10.1145/1008861.1008865
5. Canny, J.: A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-8**(6), 679–698 (Nov 1986)
6. Chen, C., Shao, Y., Bi, X.: Detection of anomalous crowd behavior based on the acceleration feature. IEEE Sensors Journal **15**(12), 7252–7261 (Dec 2015)
7. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(5), 603–619 (May 2002)
8. Deng, Y., Manjunath, B.S.: Unsupervised segmentation of color-texture regions in images and video. IEEE Transactions on Pattern Analysis and Machine Intelligence **23**(8), 800–810 (Aug 2001)
9. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. International Journal of Computer Vision **59**(2), 167–181 (Sep 2004), https://doi.org/10.1023/B:VISI.0000022288.19776.77
10. Freixenet, J., Muñoz, X., Raba, D., Martí, J., Cufí, X.: Yet another survey on image segmentation: Region and boundary information integration. In: 7th Eur. Conf. Comput. Vis. (2002)
11. Grady, L.: Random walks for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(11), 1768–1783 (Nov 2006)
12. Huang, K., Tan, T.: Vs-star: A visual interpretation system for visual surveillance. Pattern Recognition Letters **31**(14), 2265 – 2285 (2010), http://www.sciencedirect.com/science/article/pii/S0167865510001868
13. Jr., J.H.W.: Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association **58**(301), 236–244 (1963), https://www.tandfonline.com/doi/abs/10.1080/01621459.1963.10500845
14. Levinshtein, A., Stere, A., Kutulakos, K.N., Fleet, D.J., Dickinson, S.J., Siddiqi, K.: Turbopixels: Fast superpixels using geometric flows. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(12), 2290–2297 (Dec 2009)
15. Li, S., Wu, D.O.: Modularity-based image segmentation. IEEE Transactions on Circuits and Systems for Video Technology **25**(4), 570–581 (April 2015)

16. Linares, O.A.C., Botelho, G.M., Rodrigues, F.A., Neto, J.B.: Segmentation of large images based on super-pixels and community detection in graphs. IET Image Processing **11**(12), 1219–1228 (2017)
17. Liu, G., Zhang, Y., Wang, A.: Incorporating adaptive local information into fuzzy clustering for image segmentation. IEEE Transactions on Image Processing **24**(11), 3990–4000 (Nov 2015)
18. Liu, T., Seyedhosseini, M., Tasdizen, T.: Image segmentation using hierarchical merge tree. IEEE Transactions on Image Processing **25**(10), 4596–4607 (Oct 2016)
19. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proc. 8th Int'l Conf. Computer Vision. vol. 2, pp. 416–423 (July 2001)
20. Meilă, M.: Comparing clusterings by the variation of information. In: Schölkopf, B., Warmuth, M.K. (eds.) Learning Theory and Kernel Machines. pp. 173–187. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)
21. Murshed, M., Teng, S.W., Lu, G.: Cuboid segmentation for effective image retrieval. In: 2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA). pp. 1–8 (Nov 2017)
22. Ren, Malik: Learning a classification model for segmentation. In: Proceedings Ninth IEEE International Conference on Computer Vision. pp. 10–17 vol.1 (Oct 2003)
23. Sharon, E., Galun, M., Sharon, D., Basri, R., Brandt, A.: Hierarchy and adaptivity in segmenting visual scenes. Nature **442**(7104), 810–813 (2006), https://doi.org/10.1038/nature04977
24. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(8), 888–905 (Aug 2000)
25. Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) Computer Vision – ECCV 2006. pp. 1–15. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
26. Steven L. Horowitz, T.P.: Picture segmentation by a tree traversal algorithm. Journal of the ACM **23**(2), 368–388 (Apr 1976)
27. Syu, J., Wang, S., Wang, L.: Hierarchical image segmentation based on iterative contraction and merging. IEEE Transactions on Image Processing **26**(5), 2246–2260 (May 2017)
28. Tania, S., Murshed, M., Teng, S.W., Karmakar, G.: Cuboid colour image segmentation using intuitive distance measure. In: 2018 International Conference on Image and Vision Computing New Zealand (IVCNZ). pp. 1–6 (Nov 2018)
29. Tapia, E.: A note on the computation of high-dimensional integral images. Pattern Recogn. Lett. **32**(2), 197–201 (Jan 2011), http://dx.doi.org/10.1016/j.patrec.2010.10.007
30. Ugarriza, L.G., Saber, E., Vantaram, S.R., Amuso, V., Shaw, M., Bhaskar, R.: Automatic image segmentation by dynamic region growth and multiresolution merging. IEEE Transactions on Image Processing **18**(10), 2275–2288 (Oct 2009)
31. Unnikrishnan, R., Pantofaru, C., Hebert, M.: Toward objective evaluation of image segmentation algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(6), 929–944 (June 2007)
32. Vincent, L., Soille, P.: Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE Transactions on Pattern Analysis and Machine Intelligence **13**(6), 583–598 (June 1991)
33. Wang, X., Tang, Y., Masnou, S., Chen, L.: A global/local affinity graph for image segmentation. IEEE Transactions on Image Processing **24**(4) (Apr 2015)