Automatic Image Segmentation by Dynamic Region Growth and Multiresolution Merging

Luis Garcia Ugarriza, Eli Saber, *Senior Member, IEEE*, Sreenath Rao Vantaram, Vincent Amuso, Mark Shaw, and Ranjit Bhaskar

Abstract—Image segmentation is a fundamental task in many computer vision applications. In this paper, we propose a new unsupervised color image segmentation algorithm, which exploits the information obtained from detecting edges in color images in the CIE L ∗ a ∗ b∗ color space. To this effect, by using a color gradient detection technique, pixels without edges are clustered and labeled individually to identify some initial portion of the input image content. Elements that contain higher gradient densities are included by the dynamic generation of clusters as the algorithm progresses. Texture modeling is performed by color quantization and local entropy computation of the quantized image. The obtained texture and color information along with a region growth map consisting of all fully grown regions are used to perform a unique multiresolution merging procedure to blend regions with similar characteristics. Experimental results obtained in comparison to published segmentation techniques demonstrate the performance advantages of the proposed method.

Index Terms—Adaptive threshold generation, CIE L ∗ a ∗ b∗, color gradient, entropy, G-SEGmentation, multivariate analysis.

I. INTRODUCTION

IN recent years, automatic image segmentation has become a prominent objective in image analysis and computer vision. Image segmentation can be defined as the classification of all the picture elements or pixels in an image into different clusters that exhibit similar features. Various techniques have been proposed in the literature where color, edges, and texture were used as properties for segmentation. Using these properties, images can be analyzed for use in several applications including video surveillance, image retrieval, medical imaging analysis, and object classification.

At the outset, segmentation algorithms were implemented using grayscale information only (see [1] for a comprehensive survey). The advancement in color technology facilitated the achievement of meaningful color segmentation of images as described in [2] and [3]. Color provided significant advantages over gray-level segmentations. However, early procedures consisted of clustering pixels by utilizing only color similarity. Spatial locations and correlations of pixels were not taken into account yielding, fragmented regions throughout the image.

The problem of clustering pixels was initially confronted in the gray-level domain in [4] by extending the K-means clustering algorithm to a Gibbs Random-Fields model, obtaining spatially contiguous regions. This method was augmented to handle multichannel images by Chang *et al.* [5] by assuming that each individual channel was conditionally independent. Saber *et al.* [6] extended the algorithm in [5] by incorporating a vector-edge field and a split-and-merge procedure to obtain an improved segmentation and a linked-edge map. Jianping *et al.* [7] proposed a method to select the number of clusters in an image by acquiring the location of clusters between adjacent-edge regions. The challenge in this method was to evaluate the correct threshold(s) that distinguished true from false edge pixels. A different approach to define the number of clusters needed was instituted by Wan *et al.* [8] who developed a set of rules that split or merge the clusters to obtain a segmentation map with meaningful regions.

Liu *et al.* [9] proposed an approach using predetermined shapes to obtain a final segmentation. However, the algorithm assumed that the initial segmentation had a larger number of clusters than required, which was not always true, in general, rendering this assumption to be somewhat inconsistent. Furthermore, the aggregation of over-segmented regions rendered the algorithm vulnerable to match incorrect shapes. On the other hand, D'Elia *et al.* [10] proposed an algorithm by considering the entire input image as a single region and obtained a final segmentation map by combining a Bayesian classifier and a Split-and-Merge technique. The main drawback of the Bayesian approach was that it yielded too many segments in a patterned or textured region, which in turn, produced a cluttered final segmentation map.

Texture is another feature that has been widely used for segmenting images. Derin *et al.* [11] proposed a technique of comparing the Gibbs distribution results to known textures. This technique was not applicable in the presence of noise and had proven to be computationally prohibitive. Pyramid-structured wavelet transforms appeared in the work of Mallat *et al.* [12] and had become an important alternate approach to identify patterns. Unser *et al.* [13] used a variation of the discrete wavelet transform to characterize texture properties. In this study, he limited

his detection to a set of 12 Brodatz textures. Further analysis/extension of this wavelet theory was proposed by Fan *et al.* [14], who showed that the periodicity in natural textures resulted in dependencies across the discrete wavelet transform sub-bands. Chen *et al.* [15] extended the above mentioned approach without the use of Brodatz models by simply classifying texture into a limited number of classes: smooth, horizontal, vertical, and complex. This initial work provided the basis for the more complete study introduced in [16]. Cheng *et al.* observed the need to include two additional directions to detect texture ($+45°$, and $-45°$), and further segment the areas that have been classified as smooth. The objective of this approach is to enable the algorithm to handle a wide range of images encountered in nature, instead of sets of predetermined texture sets. Deng *et al.* [17] who introduces an alternative automated method to identify textured regions without prior models share this goal. This method, widely known as the J-SEGmentation algorithm, first quantizes the input image into a few colors and filters them individually to determine the smoothness of the local areas. The use of color quantization caused a major problem when regions of varying shades, due to illumination, appeared in the image.

The task of segmenting images in perceptually uniform color spaces is an ongoing area of research in image processing. Paschos *et al.* [18] proposed an evaluation methodology for analyzing the performance of various color spaces for color-texture analysis methods such as segmentation and classification. The work showed that uniform/approximately uniform spaces such as $L*a*b*$, $L*u*v*$ and HSV, possess a performance advantage over RGB, a non uniform space traditionally used for color representation. The use of these spaces was found to be well suited for the calculation of color difference using the Euclidean distance, employed in many segmentation algorithms. Yoon *et al.* [19] utilized this principle to propose a Color Complexity Measure (CCM) for generalizing the K-means clustering algorithm, in the CIE $L*a*b*$ space. Chen *et al.* [20] employed color difference in the CIE $L*a*b*$ space to propose directional color contrast segmentations. Contrast generation as a function of the minimum and maximum value of color difference was seen in the work of Chang *et al.* [21]. This contrast map, subjected to noise removal and edge enhancement to generate an Improved Contrast Map (ICMap), was the proposed solution to the problem of over-segmentation in the JSEG algorithm. Uniform spaces were also used for other image segmentation methodologies such as a Self-Organizing Map (SOM) proposed by Huang *et al.* [22].

In this paper, we propose a new Gradient-SEGmentation algorithm (GSEG) that automatically: 1) selects clusters for images using gradient information in the CIE $L*a*b*$ color space, 2) characterizes the texture present in each cluster, and 3) generates a final segmentation by utilizing an effective merging procedure. Using color-edge detection [23], our procedure first detects regions in the image without edges. The pixels that compose each detected region receive a label and the combination of pixels with the same label is referred as a seed. These seeds grow into the higher edge density areas, and additional seeds are created to generate an initial segmentation map. The characterization of texture is performed by color quantization in the image and evaluating the local entropic

color information present at each seed. The seeds that have similar values of color and texture are consequently merged to obtain a final segmentation map. The algorithm takes into account the fact that segmentation is a low-level procedure and as such, it should not require a large amount of computational complexity. Our algorithm is compiled in a MATLAB environment and tested over a large database ($\sim$700 images) of highly diverse images. It was also benchmarked qualitatively and quantitatively — using the Normalized Probabilistic Rand index (NPR) [32] — against several popular algorithms on the same test bed of manually segmented images (ground truth) obtained from the Berkeley database of 300 images. The results indicate that our proposed methodology performs favorably against the currently available benchmarks.

The remainder of this paper is organized as follows. In Section II, a review of the necessary background required to effectively implement our algorithm is presented. The proposed algorithm is described in Section III. The NPR technique used for evaluating various segmentation results is discussed in Section IV. Results obtained in comparison to popular segmentation methods are provided in Section V, and conclusions are drawn in Section VI.

II. BACKGROUND

The proposed algorithm uses an edge-detection algorithm that provides the intensity of edges present in an image; these help to detect the individual regions into which an image is segmented and the direction in which the region growth procedure takes place.

A. Edge Detection in a Vector Field

The detected areas with no edges inside them are the initial clusters or seeds selected to initiate the segmentation of the image. Edge detection has been extensively studied in a 2-D space and was generalized for a multidimensional space by Lee and Cok [23]. Assuming that an image is a function $f(x, y)$, the gradient can be defined as its first derivative $\nabla f = [(\delta f/\delta x); (\delta f/\delta y)]$. Choosing the magnitude of the gradient provides a rotational invariant value of the edges. For a vector field f, Lee and Cok expand the gradient vector to be

$$D(x) = \begin{bmatrix} D_1 f_1(x) & \cdots & D_n f_1(x) \\ \vdots & \ddots & \vdots \\ D_1 f_m(x) & \cdots & D_n f_m(x) \end{bmatrix} \quad (1)$$

where $D_j f_k$ is the first partial derivative of the kth component of f with respect to the jth component of x. The distance from a point x with a unit vector u in the spatial domain $d = \sqrt{u^T D^T D u}$ corresponds to the distance traveled in the color domain. The vector maximizing this distance is the eigenvector of the matrix $D^T D$ that corresponds to its largest eigenvalue. In the special case of a three-channel color image, the gradient can be computed in the following manner: let u, v, w denote each color channel and x, y denote the spatial coordinates for a pixel. Defining the following variables to simplify the expression of the final solution:

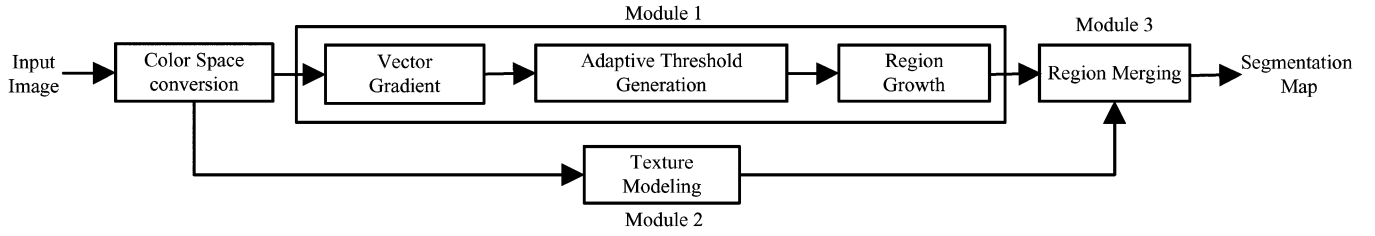$$q = \left(\frac{du}{dx}\right)^2 + \left(\frac{dv}{dx}\right)^2 + \left(\frac{dw}{dx}\right)^2 \quad (2)$$

Fig. 1. Block diagram of the proposed algorithm.

$$t = \left(\frac{du}{dx}\frac{du}{dy}\right) + \left(\frac{dv}{dx}\frac{dv}{dy}\right) + \left(\frac{dw}{dx}\frac{dw}{dy}\right) \quad (3)$$

$$h = \left(\frac{du}{dy}\right)^2 + \left(\frac{dv}{dy}\right)^2 + \left(\frac{dw}{dy}\right)^2. \quad (4)$$

The matrix $D^T D$ becomes

$$D^T D = \begin{bmatrix} q & t \\ t & h \end{bmatrix} \quad (5)$$

and its largest eigenvalue λ is

$$\lambda = \frac{1}{2}\left(q + h + \sqrt{(q+h)^2 - 4(qh - t^2)}\right) \quad (6)$$

The corresponding gradient at each location is defined as

$$G = \sqrt{\lambda}. \quad (7)$$

III. PROPOSED ALGORITHM

The proposed algorithm consists of three different modules (see Fig. 1 for a block diagram). The first module implements an edge-detection algorithm to produce an edge-map used in the generation of adaptive gradient thresholds, which in turn dynamically select regions of contiguous pixels that display similar gradient and color values, producing an initial segmentation map. The second module creates a texture characterization channel by first quantizing the input image, followed by entropy based filtering of the quantized colors of the image. Finally, the last module utilizes the initial segmentation map and the texture channel to obtain our final segmentation map. The following sections describe each of the three modules in detail.

A. Region Growth and Dynamic Seed Addition

The quality of region-growth techniques is highly dependent on the locations chosen to initialize the growth procedure. We propose an alternate process for region growth that does not depend exclusively on the initial assignment of clusters for the final segmentation. The procedure searches for regions where the gradient map displays no edges. The selected regions form the initial set of seeds to segment the image. The region growth procedure also accounts for regions, which display similar edge values throughout, by detecting unattached regions at various edge density levels. Fig. 2 displays a flowchart of this module, followed by a detailed explanation.

Adaptive Threshold Generation: The GSEG algorithm is initiated with a color space conversion of the input image from RGB to CIE $L*a*b*$. This step is vital, because the latter is a better model for the human visual perception, justified by
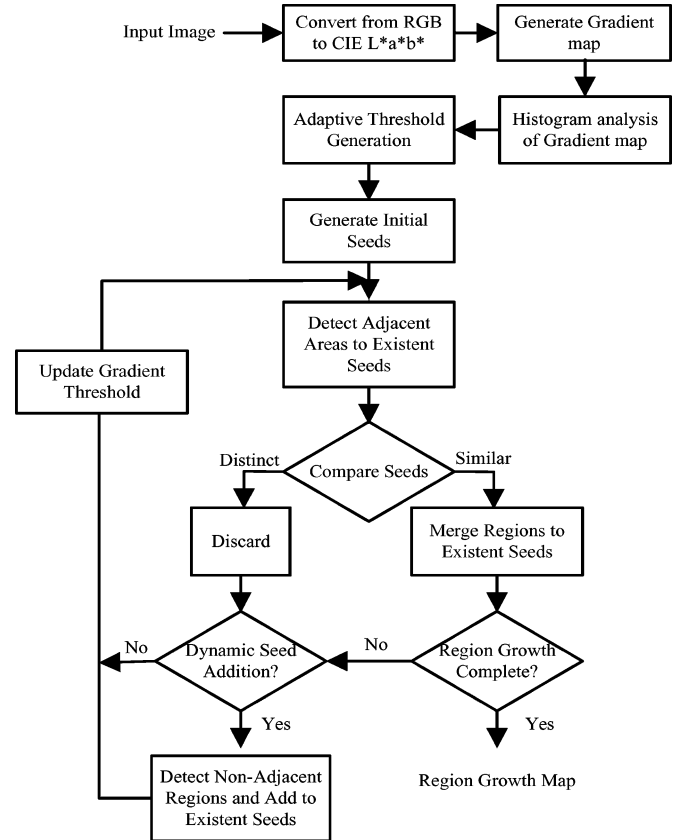

Fig. 2. Flowchart of the Region Growth procedure: Module 1.

the fact that given two colors, the magnitude difference of the numerical values between them is proportional to the perceived difference as seen by the human eye [24], a property that cannot be associated with the RGB space. Using the acquired $L*a*b*$ data, the magnitude of the gradient $G(i, j)$ of the color image field is calculated, as described in Section II-A. The histogram of this gradient map is utilized to determine the seed addition levels employed for dynamic seed addition.

Initially, the objective is to select a threshold λ for the initiation of the seed generation process. Ideally, a threshold value could be selected to provide the most edges, while ignoring the noise present in images. The problem is that the nature of images does not allow for this disposition. A single threshold that may correctly delineate the boundary of a given region may allow other regions to merge incorrectly. Due to this factor, we propose choosing one of two empirically determined threshold values for initiating the seed generation process, by validating how far apart the low and high gradient content in the image

are, in its corresponding histogram. The idea is that a high initial threshold be used for images in which a large percentage of gradient values spread over a narrow range and a low initial threshold value be used for images in which a large percentage of gradient values spread over a wide range, in comparison to the span of the histogram. The choice of λ made in such a manner ensures that all significant low gradient regions are acquired as initial seeds.

From a practical implementation standpoint, we made this decision of selecting the initial threshold by obtaining the percentage ratio of the gradient values corresponding to 80% and 100% area under the histogram curve. If 80% area under the histogram curve corresponds to a gradient value that is less than 10% of the maximum gradient value in the input image, a high threshold value is chosen, else a low initial threshold value is chosen. Keeping in view the problems posed by over and under-segmentation, the low and high threshold values were empirically chosen to be 5 and 10, respectively. The former case corresponded to images where background and foreground have largely indistinguishable gradient detail from each other. The latter case corresponded to images consisting of a slowly varying background with less gradient detail, well distinguished from prominent foreground content. Having obtained λ, all significant flat regions and its neighboring areas are generated at threshold intervals of λ and $\lambda + 5$ with varied size criterions, to form the initial seeds map.

Once the threshold for initiating the segmentation process is determined, we proceed to calculate thresholds intervals for the dynamic seed addition portion of the region growth procedure. Dynamic seed generation requires a set of threshold intervals at which additional seeds are added to the initial seeds. The threshold values selected for the addition of new seeds are adaptively determined utilizing the area under the gradient histogram that does not fall within the gradient range of the initial seeds. The first threshold for dynamic seed addition (T1) is determined by adding 10% of the histogram area greater than the maximum gradient value of the initial seeds $(\lambda + 5)$, to the cumulative area detected by $\lambda + 5$ and obtaining the corresponding gradient value. This process is continued for each new stage of the dynamic seed addition procedure where a 10% increment of the histogram area greater than the maximum gradient value of its corresponding previous stage is added to the cumulative image area detected at the end of that previous stage, as illustrated in Fig. 3. Generating the threshold values in such a manner always ensures that: 1) they are adjusted to account for the exponential decay of gradient values (as seen in Fig. 3); 2) regions of significant size are added to the segmentation map at each interval; 3) they lie within the span of the histogram, avoiding the possibility of wasted computational efficiency.

Initial Seed Generation: As mentioned previously, initial seeds are generated by detecting all the regions in the image whose gradient value fall below the initial thresholds, λ and $\lambda + 5$. If no region exists under this threshold, the edge value is increased until regions are detected. The Initial Seed Generation employs special size requirements to select the initial seeds, in order to prevent multiple seed generation within homogeneous and connected regions. The first requirement is to enforce that seeds be larger than 0.5% of the image when searching for regions with a threshold value lower than λ. The reason of this rule is that often backgrounds contain false edges generated by illumination or other varying factors in this gradient range. The
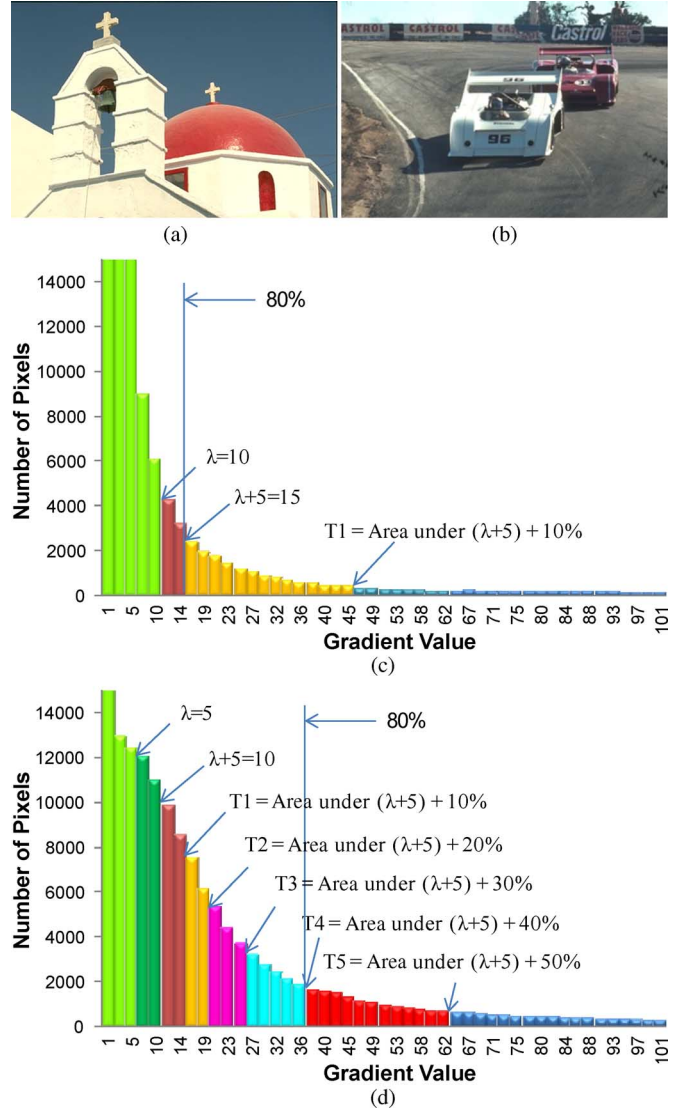

Fig. 3. Adaptive threshold, (a) and (c) $\lambda = 10$ case, (b) and (d) $\lambda = 5$ case. (a) Church images; (b) cars image; (c) edge map histogram for church image; (d) edge map histogram for cars image.

second requirement is to enforce seeds to be larger than 0.25% of the image in the range λ to $\lambda + 5$, since in this range variations may be important enough to differentiate these regions. For differentiating purposes the pixels composing each seed receive a unique label, the combined set of labeled seeds is referred to as the *Parent Seeds (PS)*, as seen in Fig. 4. The labeling procedure uses the general procedure outlined in reference [25]. It is briefly described as follows. 1) Run-length encoding of the input image. 2) Scan the runs, and assign preliminary labels and recording label equivalences in a local equivalence table. 3) Resolve the equivalence classes. 4) Relabel the runs based on the resolved equivalence classes.

Dynamic Seed Generation: In general, the dynamic seed addition procedure is carried out in five stages, where each stage corresponds to a different threshold. However, especially in the case of images with less gradient detail, a situation may arise where fewer stages are needed as is the case in Fig. 3(c). This generates an additional dimension to the algorithm, as not only are the thresholds generated adaptively but also their number

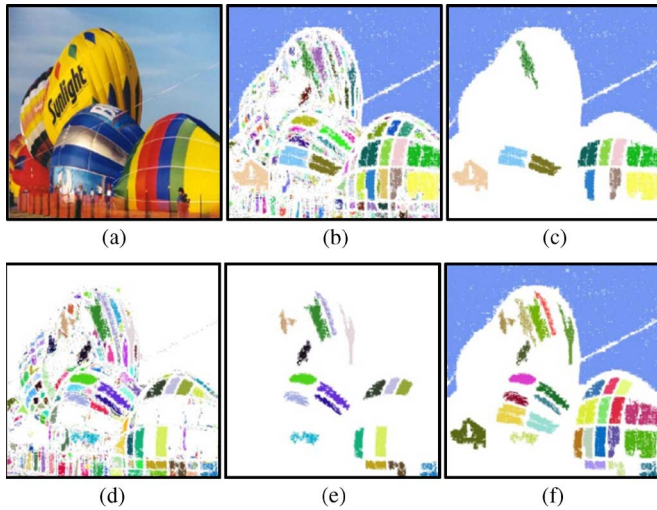Fig. 4. (a) Original, (b) seeds with $G < \lambda$, (c) seeds of b with sizes $>0.5\%$ of total image size, (d) new seeds with $G < (\lambda + 5)$, (e) seeds of d with sizes $>0.25\%$ of total image size, (f) combined seeds of b and e, or initial parent seeds.
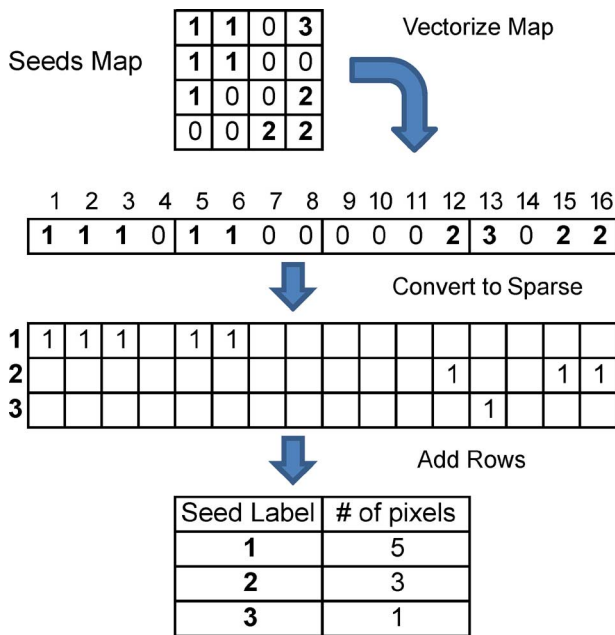

Fig. 5. Method to identify number of pixels per seed.

may vary from image to image. When the parent seeds are created in the initial seed generation, the regions represented by these seeds are characterized by areas of the image that have no texture. These areas, in general, are regions where there is either no color variance or a gradual transition from one color to the next. The dynamic addition of seeds to the PS map is designed to include the remaining regions that display different levels of gradient intensities through them but are part of the same identifiable object, as seen in Fig. 6(g). The addition of new seeds at these intervals follows the following procedure. All regions that fall below each adaptive threshold Tx are detected, where x refers to the current stage. Regions that are not attached to any parent seeds and are larger than the minimum seed size (MSS) are added to the PS map [see Fig. 6(h)]. In our algorithm, the MSS is set to 0.01% of the image. For the addition of new seeds

that share borders with existent seeds, it is required for them to meet two qualifications: 1) a group must be large enough to be considered as an independent entity and 2) the color differences between a region and its neighbors must be greater than the maximum color difference allowed.

Region Growth: In between each stage, the existent parent seeds grow by increasing the threshold a single unit at a time, for example from 20 to 21. After each increment, detection of new regions or child seeds that fall below the new threshold occurs, as shown in Fig. 6(c). These child seeds need to be classified into adjacent-to-existent or nonadjacent seeds. The nonadjacent seeds are discarded, because, they can only be added at the beginning of each adaptive threshold stage. In order to make the region growth process efficient, it is important to know the parent seed to which the child is adjacent. The objective is to be able to process all adjacent child seeds in a vectorized approach. To achieve this task, we proceed to first detect the outside edges of the PS map using a nonlinear spatial filter. The filter operates on the pixels of a 3×3 neighborhood, and its output is assigned to the center pixel of the neighborhood. The filter is defined as follows:

$$F(i,j) = \begin{cases} 0, & \text{if } PS(i,j) > 0 \\ 0, & \text{if } \sum_{(i,j) \in \beta} PS(i,j) = 0 \\ 1, & \text{otherwise} \end{cases} \quad (8)$$

where β is the neighborhood being operated on.

The result of applying this filter is a mask indicating the borders of the PS map [see Fig. 6(d)]. The detected and labeled child seeds form a child seeds map. Performing an element-by-element multiplication of the child seeds map with the PS borders mask, results in a matrix containing the labels of the adjacent child seeds. Taking all the pixels in the child seed map that matches these labels generates the *adjacent child seeds* map as shown in Fig. 6(e). For proper addition of the adjacent child seeds, it is necessary to compare their individual color differences to their parents to assure a homogeneous segmentation. To compare all generated adjacent seeds will be computationally expensive; instead, all seeds that have a size smaller than the MSS are directly merged to the adjacent parent seed. The child seed sizes are computed utilizing sparse matrix storage techniques to allow for the creation of large matrices with low memory costs. Sparse matrices store only the nonzero elements of a matrix, together with their location in the sparse matrix (indices). The size of each child seed is computed by creating a matrix of $M \times N$ columns by C rows, where M is the number of columns of pixels in the image, N the number of rows, and C the number of adjacent child seeds. The matrix is created by allocating a 1 at each column in the row that matches a pixel label. The pixels that do not have a label are ignored. By summing all the elements along each row, we obtain the number of pixels per child seed. This procedure (see Fig. 5), is useful for any operation that requires the knowledge of the number of elements per group in the segmentation algorithm.

To merge regions, an association between child seeds and their parents is required. A two-column table, the first column occupied by the labels of child seeds, and the second, by the child's respective parent labels, provides this association. The numbers of rows of this table will be equal to the number of adjacent child pixels. The first column is populated from the nonzero values detected after performing the element-by-element multiplication of the child seeds and the border mask of
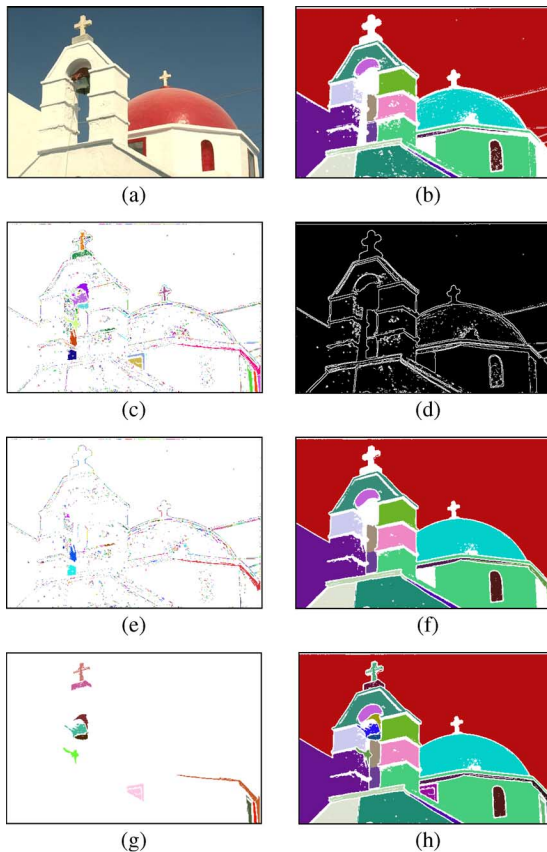
Fig. 6. (a) Original RGB image. (b) Parent seeds map. (c) New seeds after threshold increment. (d) Parent seed borders. (e) Adjacent child seeds map. (f) Seed map after one stage of the region growth procedure. (g) Seeds obtained during the dynamic seed addition procedure. (h) Parent seeds for the next stage of region growth.
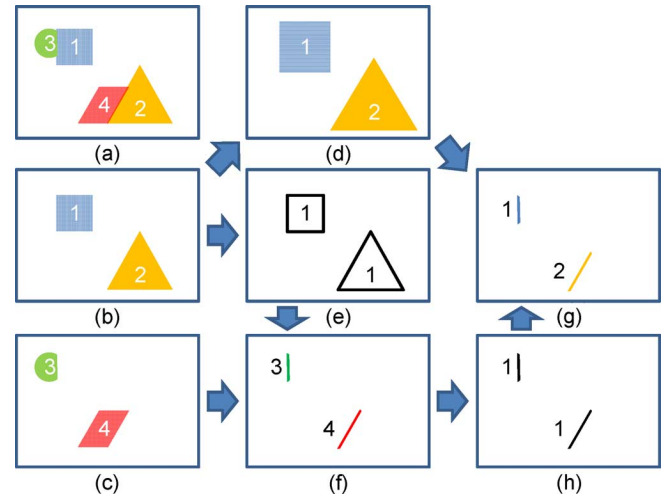

Fig. 7. (a) Parents and child seeds, (b) parent seeds, (c) child seeds, (d) result of applying max filter to b, (e) result of applying (8) to b, (f) element-by-element multiplication of c and e, (g) element-by-element multiplication of d and h, (h) conversion of f to mask.

the parent seeds [see Fig. 7(f)]. The respective parents of each child in the first column are detected using a two-step procedure. First, a filter, which returns the maximum value of a 3×3 neighborhood to each pixel location, applied to the PS map, extends the reach of the borders of each parent seed [see Fig. 7(d)]. The second step is to use the adjacent child pixels locations as a mask over the PS extended map to extract the parent's label at each child location [see Fig. 7(g)]. It is important to note that nonlinear filters provide information about the seeds, but do not to directly manipulate the image; therefore, the final result does not display any deformation effects.

The functionality of the association table is twofold, providing the number of child pixels attached to the parent seeds and identifying the child seeds that share edges with more than one parent seed. The labels of child seeds that are smaller than MSS are replaced with those of its parents using the association table. When comparison between the remaining adjacent seeds and their parents returns a large difference, these may become parent seeds themselves.

Enforcing parent and child seeds to share a border greater than five pixels, prevent comparing two seeds unnecessarily. Allowing child seeds that fall below this requirement to grow more prior to comparing them to their parents, may provide more information. To this effect, these seeds are returned to the pool of pixels that do not form part of the PS map to gain size with the increment of the edge threshold. Comparing the seeds sharing more than five pixels with their parents, allows replacing their

labels either, with their parent's labels or a new parent label, depending whether they are alike or distinct.

Given that spatial regions in images vary gradually, only the nearby area of adjacency between parent and child is compared to provide a true representation of the color difference. This objective is achieved by using a mask signifying areas of parent seeds lying in the surroundings of child seeds. A dilation of the *adjacent child seed's* map using a square structuring element with a dimension of 30 pixels obtains the regions in close proximity to existing child seeds. Regions beyond the reach of the dilated child seeds are excluded from the parent seeds map yielding a mask consisting of parent pixels close to child seeds. Only these parent pixels are used to check for color similarity. The values used in our algorithm are optimized to work with images that range from 300×300 to 1600×1600 pixels of resolution.

Color difference in a perceptually uniform color space such as CIE $L * a * b*$, can be calculated in terms of the Euclidean distance [24]. The comparison of regions is performed, using their mean color information. The reason for choosing this color space and distance metric combination is that: 1) it ensures that comparison of colors is similar to the differentiation of colors by the human eye, 2) the increased complexity of a different distance metric like the Mahalanobis distance does not improve the results, due to the small variance of the regions being compared, owed to their spatial proximity. The maximum color distance to allow the integration of a child seed to its parent was empirically chosen as 50.

Seed Growth Tracking and Classification: Region growth without growth rate feedback of each seed may cause parent seeds to overflow into regions of similar colors but different textures. Implementation of seed growth tracking helps maintain homogeneity through the segmented regions. At each dynamic seed-addition interval, the number of pixels per parent seed is calculated and stored. When the algorithm has reached the next interval, calculation of the seed-size percentage increment, using the stored size and the current size, detects regions that have slow is growth rate. The border delimited by these seeds, often correspond to boundaries of image elements that

display low levels of texture. The growth rate determined empirically to detect this behavior, is any size increment equal to or below 5%. The seeds detected by this process are temporarily replaced from the PS map into a grown seed map. This is the same as continuing the growing procedure, but inhibiting only these regions to grow any more. After the last dynamic seed generation, no additional seeds will be generated. All seeds, both in the PS map and the grown seed map are placed together into one segmentation map. At this point all remaining pixels that do not belong to this map are the edges of the segmented regions. Continuing the region growth on the new segmentation map, until all pixels in the image has a label assigned to them, generates the Region Growth Segmentation (RGS) map.

B. Texture Channel Generation

The presence of regions that contain distinct patterns and textures is the greatest source of problems in image segmentation algorithms. The issue is that patterns are composed of multiple shades of colors causing over-segmentation and misinterpretation of the edges surrounding these regions. Texture regions may contain regular patterns such as a brick wall, or irregular patterns such as leopard skins, bushes, and many objects found in nature. The presence of texture in images is so large and descriptive of its objects that ignorance of this important information is not possible and, therefore, requires being part of any meaningful segmentation algorithm.

A method for obtaining information of patterns within an image is to evaluate the randomness present in various areas of that image. Entropy provides a measure of uncertainty of a random variable [26]. If the pixel values of a region compose a random variable, entropy will define the randomness associated to the region being evaluated. Textured regions contain various colors and shades; therefore, these regions will contain a specific value of uncertainty associated with them, providing a structure to merge regions that display similar characteristics.

Information theory introduces entropy as the quantity which agrees with the intuitive notion of what a measure of information should be [26]. In accordance with this supposition, we can select a random group of pixels **s** from an image, with a set of possible values $\{a_1, a_2, \ldots, a_J\}$. Let $P(a_j)$ be the probability for a specific value a_j to occur, then $P(a_j)$ is said to contain $I(a_j)$ units of information, defined as

$$I(a_j) = \log \frac{1}{P(a_j)} = -\log P(a_j). \qquad (9)$$

The quantity $I(a_j)$ is referred to as the self-information of a_j. If k values are present in the set, the law of large numbers stipulates that the average of a random value a_j is likely to be close to the average of the whole set. Thus, the average self-information obtained from k outputs is

$$-kP(a_1)\log P(a_1) - \cdots - kP(a_J)\log P(a_J) \qquad (10)$$

and the average information per sample, or entropy of the set, is defined by

$$H(s) = -\sum_{j=1}^{J} P(a_J)\log P(a_J). \qquad (11)$$

This quantity is defined for a single random variable; the usage of multiple channels requires defining each additional
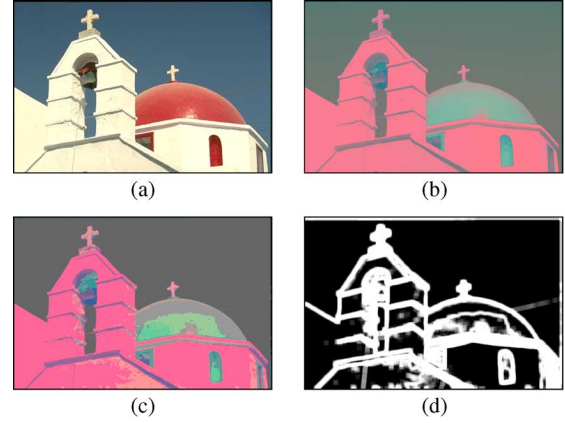

Fig. 8. (a) Original RGB image. (b) RGB to CIE $L*a*b*$. (c) Indexed image. (d) Texture channel.

channel as a separate random variable, and the entropy of a region composed of multiple channels would require the joint entropy between variables. To take advantage of the luminance and chrominance information without extending the process to compute the joint entropy, these are quantized into 6^3, or 216 different levels. This quantization can be done by uniformly dividing the 8-bit $L*a*b*$ cube into small boxes, and mapping all information that fall within each box to the respective channel values at the center of that box. The advantage of quantizing the $L*a*b*$ information rather than RGB information is that, if approximately uniform $L*a*b*$ is uniformly quantized, a constant distance between and any two quantization levels will result in small variation of perceptual color difference [27], unlike nonuniform RGB data where this variation can be very large . After the quantization process, each pixel of an image is indexed to one of the 216 representative levels [see Fig. 8(c)], effectively reducing the probability of each level occurring to a 1-D random variable. To create a texture channel, the local entropy is computed in a 9×9 neighborhood around each pixel of the indexed image, and the resulting value is assigned to the center pixel of the neighborhood, as shown in Fig. 8(d).

C. Region Merging Using Multivariate Analysis

The growth procedure primarily based on color similarity as illustrated in Section III-A, in general, results in an oversegmented seed map. We incorporate an effective method to analyze grouped data from the statistical field, to merge all over-segmented regions. This method better known as a multivariate analysis allows us to take regions that have been separated due to occlusion, or small texture differences, and merge them together. The core of a multivariate analysis lies in highlighting the differences between groups that display multiple variables to investigate the possibility that multiple groups are associated with a single factor [28].

One-Way Variance Analysis: We consider the general case in which p variables $x_1, x_2, \ldots, x_p$ are measured for each individual group, and any direction in the p-dimensional sample of the groups is specified by the p-tuple $(a_1, a_2, \ldots, a_p)$. A multivariate observation $\boldsymbol{x_i^T} = (x_{i1}, x_{i2}, \ldots, x_{ip})$ can be converted into a univariate observation $y_i = \boldsymbol{a^T x_i}$ where $\boldsymbol{a^T} = (a_1, a_2, \ldots, a_p)$. Furthermore, if a sample space is divided into g groups, it is useful to recognize each element of this space

using the notation y_{ij}, where i refers to the group that the element belongs to, and j is the location of the element in the ith group.

The objective of one-way variance is to find the optimal coefficients of the vector a, which will yield the largest differences across groups and minimize the distances of elements within a group. The between-groups sum-of-squares and products matrix B_0 and the within-groups sum-of-squares and products matrix W_0 are defined by

$$\mathbf{B_0} = \sum_{i=1}^{g} n_i(\overline{x}_i - \overline{x})(\overline{x}_i - \overline{x})^T \tag{12}$$

and

$$\mathbf{W_0} = \sum_{i=1}^{g} \sum_{j=1}^{n_i} (x_{ij} - \overline{x}_i)(x_{ij} - \overline{x}_i)^T \tag{13}$$

where the labeling x_{ij} is analogous to that of y_{ij}. The sample mean vector in the ith group is $\overline{x}_i = (1/n_i)\sum_{j=1}^{n_i} x_{if}$ and $\overline{x} = (1/n)\sum_{i=1}^{g}\sum_{j=1}^{n_i} x_{ij} = (1/n)\sum_{i=1}^{g} n_i\overline{x}_i$ is the overall sample mean vector. Since $y_{ij} = a^T x_{ij}$, it can be verified that the sum of between-groups and within-groups components become

$$SSB(a) = a^T B_0 a \text{ and } SSW(a) = a^T W_0 a. \tag{14}$$

With n sample members and g groups, there are $(g-1)$ and $(n-g)$ degrees of freedom between and within groups respectively. A test of the null hypothesis that there are no differences in mean value among the g groups is obtained from the mean square ratio

$$F = \left\{ \frac{1}{g-1} a^T B_0 a \right\} \div \left\{ \frac{1}{n-g} a^T W_0 a \right\} = \frac{a^T B a}{a^T W a} \tag{15}$$

where B is the between-group covariance matrix and W is the within-groups covariance matrix. Maximizing F with respect to a is accomplished by finding a such that differentiating F equals zero, yielding $Ba - (a^T Ba / a^T Wa)Wa = 0$, but at the maximum of F, $a^T Ba / a^T Wa$ will be a constant (let $l = (a^T Ba / a^T Wa)$), so the required value of a must satisfy

$$(B - lW)a = 0. \tag{16}$$

Equation (16) can be written as $(W^{-1}B - lI)a = 0$, indicating l and a must be an eigenvalue/eigenvector pair corresponding to the largest eigenvalue of $W^{-1}B$. This result signifies the direction in the p-dimensional data space that tends to keep the within class distances small, simultaneously maintaining large between class distances, among various groups. In the case where g or the original dimensionality is large, a single direction will provide a gross oversimplification of the true multivariate configuration. The term in (16) generally possesses more than one eigenvalue/eigenvector pair, which can be used to generate multiple differentiating directions. Let $\lambda_1 > \lambda_2 > \cdots > \lambda_S > 0$ be the eigenvalues associated with the eigenvectors $a_1, a_2, \ldots, a_s$. The new variates $y_1, y_2, \ldots$ defined by $y_i = a_k^T x_i$ are termed *canonical variates*. The eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_S$ and eigenvectors $a_1, a_2, \ldots, a_s$ are gathered together so that a_k is the kth column of a $(p \times s)$ matrix A, while λ_k is the kth diagonal element of a $(s \times s)$ diagonal matrix $\bar{L}$. Then, in matrix terms, (16) may be written as $BA = WAL$, and the collection of canonical variates is given by $y = A^T x$.
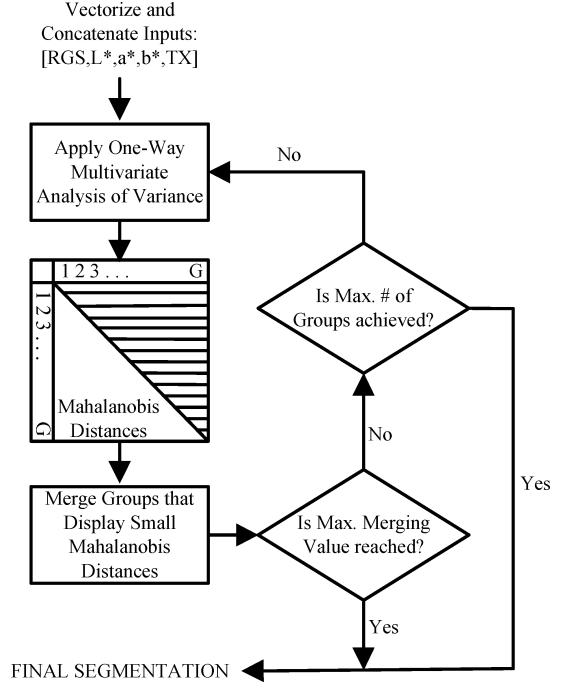

Fig. 9. Flowchart of the region merging procedure: Module 3.

The space of all vectors y is termed the *canonical variate space*. In this space, the mean of the kth group of individuals is $\overline{y}_k = A^T \overline{x}_k$.

The Mahalanobis-squared distance between the ith and j^{th} group is given by

$$D^2 = (\overline{y}_i - \overline{y}_j)^T W^{-1}(\overline{y}_i - \overline{y}_j). \tag{17}$$

Comparing (17) to the squared Euclidean distance of the group means in the canonical variate space, and substituting for $\overline{y_i}$ and $\overline{y_j}$, we obtain

$$\begin{aligned} d^2 &= (\overline{y}_i - \overline{y}_j)^T (\overline{y}_i - \overline{y}_j) \\ &= (\overline{x}_i - \overline{x}_j)^T AA^T(\overline{x}_i - \overline{x}_j) \end{aligned} \tag{18}$$

but it can be proved that $AA^T = W^{-1}$. Thus, substituting for AA^T in (18) yields (17). Hence, by constructing the canonical variate space as described above, the Euclidean distance between the group means is equivalent to the Mahalanobis distance in the original space. This distance accounts for the covariance and differential variance between variables, and is now the preferred measure of distance between two multivariate populations.

Multiresolution Region Merging: The texture channel obtained from the second module is combined with the color information to describe the characteristics of each region segmented by the region growth module. Using a multivariate analysis approach of all independent regions, the resultant Mahalanobis distances between groups is used to merge similar regions. A flowchart of the procedure is shown in Fig. 9.

Up until now, the segmentation of the input image has been performed with the absence of information about the individual regions. Now that the image has been segmented into different groups, information can be gathered from each individual region. Given that we have four sources of information (L∗, a∗, b∗, and Texture) and individual regions displaying a different

number of pixels per group, we require a suitable method to display the data in order to investigate relationships of the regions. The data can be modeled using an $(N \times P)$ matrix, where N is the total number of pixels in the image, and P is the total number of variables that contain information about each pixel. G is the total number of groups in which the image has been segmented in the region growth procedure resulting in the matrix being composed of G separate sets. The objective is to obtain a mean value for each group used for comparison with the other groups. The method used to achieve this objective is a one-way analysis of variance, explained in detail previously.

From the one-way analysis, we obtain the Mahalanobis-squared distances for each pair of groups. The algorithm uses these distances to locate and merge similar groups. Once a group has been merged, its similarity to the others is unknown but required if the new group needs to be merged to other similar groups. To prevent the need to re-evaluate the Mahalanobis distances for various groups after each stage of the region merging procedure, an alternate approach is introduced. Having the distances between groups, the smallest distance value is found, corresponding to a single pair of groups. Therefore, we increase the similarity value until a larger set of group pairs is obtained. Empirically determined, five group pairs were found to be an adequate number to reduce computation time without merging groups inadequately. We begin by merging the smallest group in this set and then continue to merge the next larger group. After the first merge, a check is performed to see if one of the groups being merged is now part of a larger group. In this case all the pair combinations of the groups should belong to the pairs selected initially in the set to be merged together. Once all the pairs of the set have been processed, the Mahalanobis distance is recomputed for the new segmentation map, and the process is repeated until either a desired number of groups is achieved or the smallest distance between groups is larger than a minimum acceptable similarity value between two arbitrary groups. The first criterion aids in achieving a workable number of groups in order to do further processing, and the second criterion assures that all images display a similar level of segmentation. We have set the desired number of groups to be 50 and the minimum acceptable similarity value to be 2; however, these could be varied depending on the application for which this algorithm is being used.

IV. EVALUATION OF SEGMENTATION ALGORITHMS

Initial evaluation methodologies were highly subjective and greatly dependent on intuition. However, the ever growing applications and diverse methods of segmenting images over the last decade, has resulted in the evolution of various evaluation metrics to analyze the performance and effectiveness of these methods. A survey of some of the earliest evaluation metrics can be found in [29], where, in general, supervised evaluations utilized one reference image for evaluation, while their unsupervised counterparts utilized requirements, utilities, complexity, etc., of algorithms.

More recently, effective evaluation methodologies have become more generic yet application specific like the one proposed by Jaime *et al.* [30]. In a broad perspective, the evaluation was based on distances between partitions with multiple

regions to overcome the limitations of some of the earlier existing techniques which dealt with only two distinct objects-foreground and background. Feng *et al.* [31] proposed an evaluation based on the arbitrary segmentation algorithm's capability of separating foreground from background objects utilizing a small number of segments.

To objectively measure the quality of our segmentation results, we have chosen a recently proposed evaluation metric called the Normalized Probabilistic Rand (NPR) index. The NPR evaluation method reported in [32] and [33] is summarized in this section. The essence and effectiveness of the NPR lies in the fact that it compares results obtained from a tested algorithm to a set of multiple manually segmented ones for a given image taking into consideration the variability across the manual segmentations. Furthermore, this set is imperative for segmentation evaluation, as there is not a single correct outcome for an image; consequently, the set of multiple perspectives of correct segmentations becomes the corresponding groundtruths for that image. The NPR technique is very generic and is not application specific like some of the earlier ones, modeled to account for variability in human perception.

As its name implies, the NPR is a normalization of the Probabilistic Rand (PR) index. The PR index allows comparison of a test segmentation result to a set of multiple ground-truth segmentation images through a soft nonuniform weighting of pixels pairs as a function of the variability in the ground-truth set [33]. Assume that the ground truth set is defined as $\{S_1, S_2, \ldots, S_K\}$ of an image $X = \{x_1, x_2, \ldots, x_N\}$ consisting of N pixels. Let S_{test} be the segmentation that is to be compared with the manually labeled set. We denote the label of pixel x_n as l_n^{test} in the test segmentation and as $l_n^{S_k}$ in the kth manual segmentation. The PR models label relationships for each pixel pair, where each human segmenter provides information (c_{ij}) about each pair of pixels (x_i, x_j) as to whether the pair belongs to the same group or belongs to different groups. The set of all perceptually correct segmentations defines a Bernoulli distribution for the pixel pair, giving a random variable with an expected value denoted as $p_{i,j}$. The set $\{p_{i,j}\}$ for all unordered pairs (i,j) defines a generative model of correct segmentations for the image X.

The Probabilistic Rand index is defined as

$$PR\left(S_{\text{test}}, \{S_k\}\right) = \frac{1}{\binom{N}{2}} \sum_{\substack{i,j \\ i<j}} \left[p_{ij}^{c_{ij}} (1 - p_{ij})^{1-c_{ij}} \right]. \quad (19)$$

This measure takes values between 0 and 1, where 0 means $S_{\text{test}}, \{S_1, S_2, \ldots, S_K\}$ have no similarities, and 1 means all segmentations are equal. Although the summation in (19) is over all possible pairs of N pixels, Unnikrishnan *et al.* [33] showed that the computational complexity of the PR index is $O(KN \sum_k L_k)$.

The NPR index establishes a comparison method, which meets the following requirements for comparison correctness: 1) Images whose ground-truth segmentations are not well defined cannot provide abnormally large values of similarity. 2) The comparison does not assume equal labeling of groups or same number of groups. 3) Boundaries that are well defined by the human manual segmentation are given greater importance than those regions that contain ill-defined boundaries. 4) Scores provide meaningful differences between segmentations of different images and segmentations of the same image.

The NPR is defined as

$$NPR = \frac{PR - E[PR]}{\max[PR] - E[PR]} \qquad (20)$$

where the maximum possible value of the PR is 1 and the expected value of the PR index is computed as

$$
\begin{aligned}
E\left[PR\left(S_{\text{test}}, \{S_k\}\right)\right] \\
= \frac{1}{\binom{N}{2}} \sum_{\substack{i,j \\ i<j}} \Big\{ E\left[I\left(l_i^{S_{\text{test}}} = l_j^{S_{\text{test}}}\right)\right] p_{ij} \\
+ E\left[I\left(l_i^{S_{\text{test}}} = l_j^{S_{\text{test}}}\right)\right] (1 - p_{ij}) \Big\} \\
= \frac{1}{\binom{N}{2}} \sum_{\substack{i,j \\ i<j}} p_{ij}^T p_{ij} + \left(1 - p_{ij}^T\right)(1 - p_{ij}) \qquad (21)
\end{aligned}
$$

where $p_{ij}^T = E[I(l_i^{S_{\text{test}}} = l_j^{S_{\text{test}}})]$ is defined as the weighted sum of $PR(S_k^\phi, \{S_k\})$. Let Φ be the number of images in a data set and K_ϕ the number of ground-truth segmentations of image ϕ. Then, p_{ij}^T can be expressed as

$$p_{ij}^T = \frac{1}{\Phi} \sum_\phi \frac{1}{K_\phi} \sum_{k=1}^{K_\phi} I\left(l_i^{S_k^\phi} = l_j^{S_k^\phi}\right). \qquad (22)$$

The normalization of the Probabilistic Rand is important, since it provides a meaningful value of similarity when segmentations of the same image are compared and a low value of similarity for segmentations of different images.

V. RESULTS AND DISCUSSIONS

Until recently, different segmentation algorithms demonstrated their effectiveness by displaying results obtained on a limited set of images. To the contrary, we evaluate the quality of our segmentation results by the calculating the NPR index as summarized in Section IV. Due to the fact the calculated NPR index varies depending on the image's segmentation ground-truth utilized in the evaluation process, they are required to display the following characteristics: 1) They cannot be chosen selectively to favor a given algorithm. 2) They display various scenarios with multiple levels of complexity. 3) They contain more than one individual perspective. 4) They can be accessed by anyone for performing the same test on diverse algorithms.

Such a set has been made publicly available by the University of California at Berkeley. This database provides 1633 manual segmentations for 300 images created by 30 human subjects [34]. Moreover, for our evaluation, state-of-the-art algorithms were chosen to furnish a fair indication of the quality of segmentation results achieved. These segmentation techniques are fusion of color and edge information for improved segmentation and edge linking (GRF) [6], the unsupervised segmentation of color-texture regions in images and video (JSEG) [17], and our new algorithm automatic image segmentation by dynamic region growth and multiresolution merging (GSEG). To prevent any discrepancies, all the available images were segmented using the above mentioned segmentation algorithms on
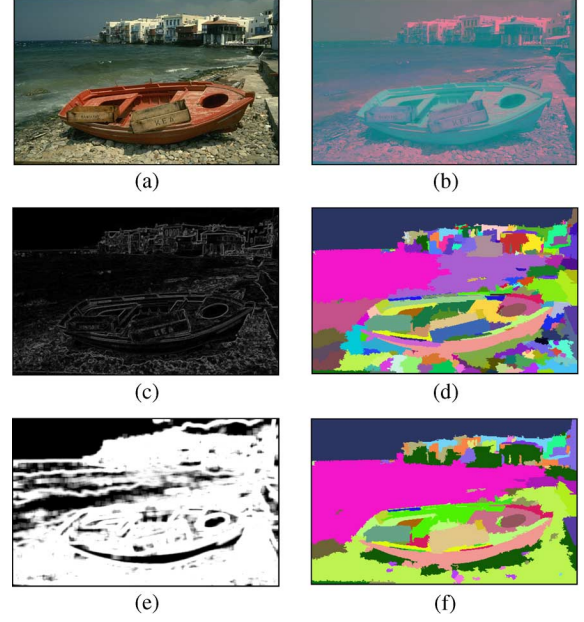

Fig. 10. (a) Original RGB image. (b) Color space conversion. (c) Gradient map. (d) Seed map after region growth. (e) Texture channel. (f) Final segmentation map after region merging.

the same machine. The testing computer has a Pentium 4 CPU 3.20GHz, and 3.00 GB of RAM. The GRF and JSEG algorithms were run from the executable file provided by the Rochester Institute of Technology and the University of California at Santa Barbara, respectively. The proposed method was implemented using MATLAB version R2007a.

The results of the GSEG algorithm at different stages are presented in Fig. 10(a)–(f). The input RGB image and its CIE $L * a * b*$ counterpart are shown in Fig. 10(a) and (b), respectively. The outcome of gradient computation on the color converted input image, shown in Fig. (10c). The seed map at the end of the region growth procedure, obtained utilizing thresholds that are generated adaptively, is displayed in Fig. 10(d). Observe that this interim result is oversegmented, due to reasons specified in Section III-B. The texture channel generated using color quantization and local entropy calculation is depicted in Fig. 10(e). The segmentation map at the end of the region merging algorithm is shown in Fig. 10(f).

Results obtained from the GSEG in comparison to the previously mentioned segmentation methods, are shown in Figs. 11–15. An image with less complex content is shown in Fig. 11(a). The image has relatively uniform color information without much variation due to illumination, and the absence of texture in all its regions, renders it to be a simple image to segment. The results shown in Fig. 11(b)–(d), obtained from the GRF, JSEG and GSEG for this image appear to be similar. Conversely, it can be seen that in case of the GRF and the JSEG segmentation results, the foliage is partitioned as two different regions in spite of their color of being similar. This is overcome by our new multiresolution merging procedure that merges these regions which are almost separate in the image. Clear advantages can be seen on the level of detail achieved in our segmentation results in comparison to the GRF and JSEG, for images with higher complexity.
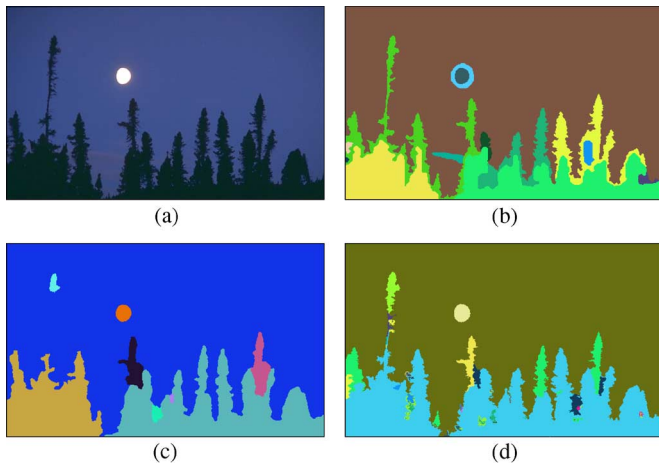
Fig. 11.   Moon light results: (a) Original, (b) GRF, (c) JSEG, (d) GSEG.

Fig. 12.   Face results: (a) Original, (b) GRF, (c) JSEG, (d) GSEG.
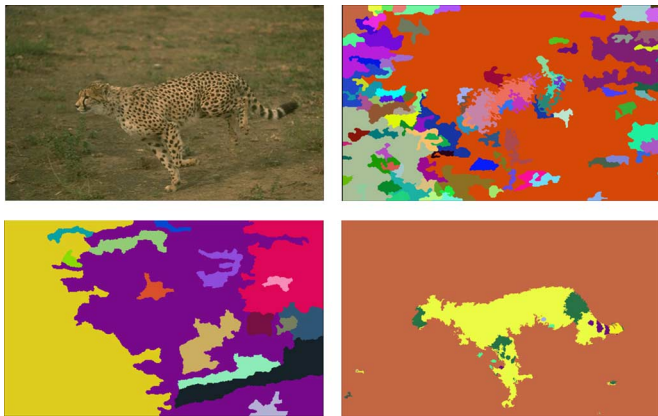

Fig. 13.   Cheetah results: (a) Original, (b) GRF, (c) JSEG, (d) GSEG.

The "Face" image in Fig. 12(a) represents a moderately complex image with dissimilar texture content associated with the skin, hat and robe of the person. Observed that in Fig. 12(b) and (c), the GRF and JSEG algorithms over segment this image due to the texture and illumination disparity seen in various regions. However, our algorithm employs the CIE $L*a*b*$ color space where the $L*$ channel contains the luminance information in the image, incapacitates the illumination problem. Furthermore, our texture model has been effective in handling different textures as seen in Fig. 12(d). Segmenting textured regions becomes a harder challenge when regions with diverse textures are extremely similar in color. In such scenarios, a good texture descriptor is indispensible. Fig. 13(a) represents an image of a Cheetah which has a skin tone that almost matches its background making it extremely difficult to segment it based on just color information. The GRF and JSEG results shown in Fig. 13(b) and (c) illustrates the effect of an indistinct texture

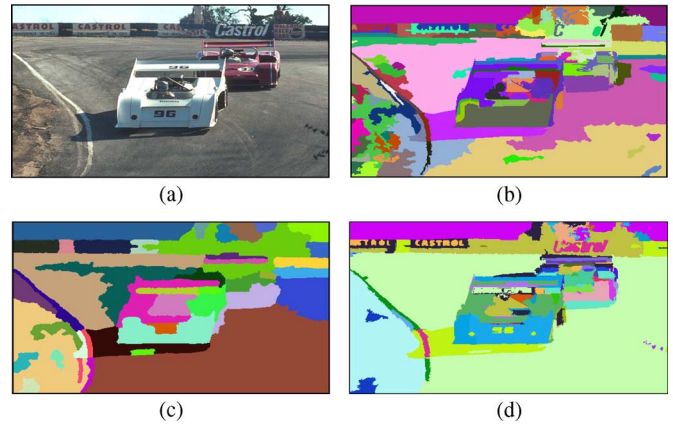Fig. 14.   London results: (a) Original, (b) GRF, (c) JSEG, (d) GSEG.

Fig. 15.   Cars results: (a) Original, (b) GRF, (c) JSEG, (d) GSEG.

descriptor for segmentation. The GSEG algorithm in comparison has been able to achieve a good segmentation illustrating the robustness of our texture model in handling highly complex images.

Results obtained for the following set of images demonstrate the advantages of our initial clustering technique to overcome problems of varying illumination, as discussed previously. These images represent instances of largely varying illumination conditions in comparison to previous examples, as well as occlusions by foreground objects. Segmentation results of the "London" image are depicted in Fig. 14. Observe that although the GRF algorithm produces a clear segmentation where the boundaries of the segments match the boundaries of the object as seen in Fig. 14(b), it suffers from over segmentation in the sky region. This is due to the use of predetermined number of clusters for segmentation. This phenomenon of over segmentation can also be observed with the JSEG algorithm [see Fig. 14(c)] owing to the fact that the initial clusters are based on the quantization of colors that causes varying shades of colors to produce a number of clusters. In this image the sky has been over segmented, because the change of light provides different shades of blue. This problem of varying illumination has yet again been well tackled by the GSEG algorithm which employs a region growth procedure that does not exclusively depend on the initial assignment of clusters, as seen in Fig. 14(d). This aspect of the GSEG algorithm has also enabled it to segment even fine details such as text with great efficiency unlike the GRF and JSEG, as illustrated in results in Fig. 15.

Observe that the word "Castrol" as seen in Fig. 15(a) is segmented out at multiple locations with near perfection by the GSEG algorithm as seen in Fig. 15(d). Yet again, the GRF and JSEG cause over segmentation in regions representing the motorway due to varying illumination, as seen in Fig. 15(b) and (c).
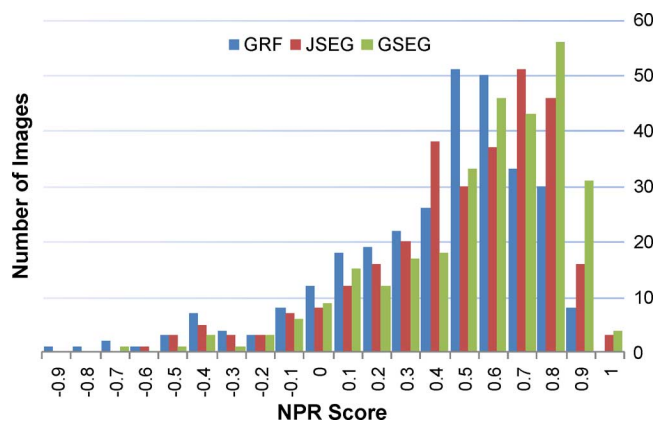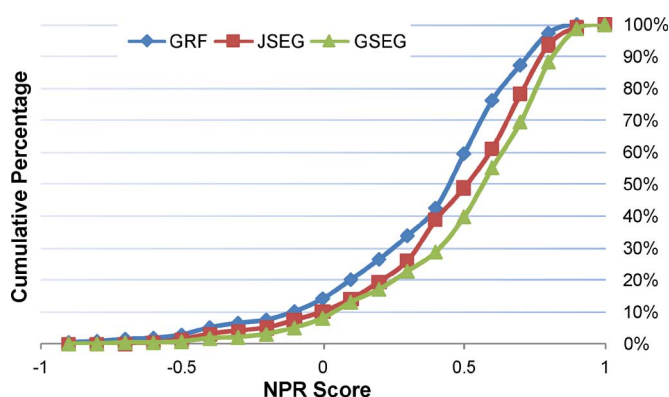
Fig. 16.   Distribution of NPR scores.

Fig. 17.   Cumulative percentage of NPR Scores.

Fig. 18.   Parachute results: (a) Original, (b) GRF, (c) JSEG, (d) GSEG.

Fig. 19.   Mask results: (a) Original, (b) GRF, (c) JSEG, (d) GSEG.

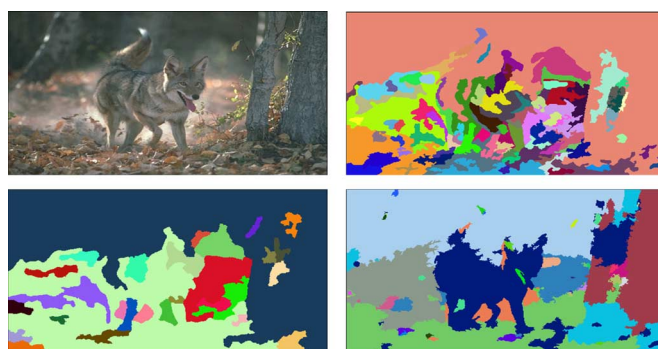Fig. 20.   Safari results: (a) Original, (b) GRF, (c) JSEG, (d) GSEG.

Fig. 21.   Fox results: (a) Original, (b) GRF, (c) JSEG, (d) GSEG.

In addition, the efficiency of the GSEG algorithm in handling the background occlusion problem discussed earlier is emphasized in the "London" and "Cars" results. In Fig. 14(d) the sky region inspite of being completely occluded by the clock tower has been segmented out as a single region and in Fig. 15(d) the motorway has been segmented as one region even though the cars occlude it almost entirely. In these results, multiple regions are assigned to the same class, yielding a reduction in the total number of classes without losing information obtained from multiple similar regions that are not adjacent to each other. This has been possible only due to our unique region merging procedure.

The normalization factor was computed by evaluating the Probabilistic Rand (PR) for all available manual segmentations, and the expected index $(E[PR])$ obtained was 0.6064. A comparison of our evaluation, for the segmentation results obtained from the three methods, is displayed in Table II. This table shows that our algorithm has the highest average NPR score, and the variance of the NPR scores has the narrowest spread, illustrating that our algorithm performs consistently better than the other algorithms qualitatively and quantitatively. In addition, the GSEG has an average computation time comparable to the JSEG, considering the different environments in which they were developed.

The distribution of the NPR scores for the Berkeley database is displayed in Fig. 16. This figure illustrates our algorithm peaking at a NPR score of 0.8, ahead of JSEG's 0.7 and GRF's 0.55. When using the NPR scores to create a cumulative percentage of images below a specific NPR score (see Fig. 17), the curve representing the GSEG method always stays to the right. This shows that for any NPR score, the GSEG method has a larger number of images segmented with a higher NPR value than the JSEG and GRF methods.

Figs. 18–21 provide additional results to observe the algorithm performance on various scenes. Table I exhibits the NPR score comparison of all displayed results. Additionally, our algorithm was tested on a database of several hundred images with varying complexity and similar results were observed, indicating the algorithm's robustness to various image scenarios.

TABLE I
NPR Score Comparison for the Displayed Images

IMAGE NAME	GRF	JSEG	GSEG
Boat	0.670	0.708	0.769
Moon light	0.765	0.766	0.790
Face	0.626	0.637	0.814
Cheetah	-0.488	-0.519	0.745
London	0.672	0.642	0.801
Cars	0.210	0.287	0.390
Parachute	0.407	0.246	0.875
Mask	0.632	0.831	0.903
Safari	-0.063	0.139	0.869
Fox	0.040	-0.007	0.635

TABLE II
Evaluation of Segmentation Results

	GRF	JSEG	GSEG
Avg. Time (sec)	240	16	24
Avg. NPR	0.357	0.439	0.495
Std. Dev. NPR	0.345	0.318	0.306
Environment	C	C	MATLAB

VI. Conclusion

This work presents a computationally efficient method designed for automatic segmentation of color images with varied complexities. The GSEG algorithm is primarily based on color-edge detection, dynamic region growth, and culminates in a unique multiresolution region merging procedure. The algorithm has been tested on a large database of images including the publicly available Berkeley database, and the quality of results show that our algorithm is robust to various image scenarios and is superior to the results obtained on the same image when segmented by other methods, as can been seen in the results displayed.

Acknowledgment

References

[1] H. Cheng, X. Jiang, Y. Sun, and J. Wang, "Color image segmentation: Advances & prospects," *Pattern Recognit.*, vol. 34, no. 12, pp. 2259–2281, Dec. 2001.
[2] J. Wu, H. Yan, and A. Chalmers, "Color image segmentation using fuzzy clustering and supervised learning," *J. Elect. Imag.*, vol. 3, no. 4, pp. 397–403, Oct. 1994.
[3] P. Schmid, "Segmentation of digitized dermatoscopic images by two-dimensional color clustering," *IEEE Trans. Med. Imag.*, vol. 18, no. 2, pp. 164–171, Feb. 1999.
[4] T. Pappas, "An adaptive clustering algorithm for image segmentation," *IEEE Trans. Signal Process.*, vol. 40, no. 4, pp. 901–914, Apr. 1992.
[5] M. Chang, M. Sezan, and A. Tekalp, "Adaptive bayesian segmentation of color images," *J. Elect. Imag.*, vol. 3, no. 4, pp. 404–414, Oct. 1994.
[6] E. Saber, A. Tekalp, and G. Bozdagi, "Fusion of color and edge information for improved segmentation and edge linking," *Imag. Vis. Comput.*, vol. 15, no. 10, pp. 769–780, Jan. 1997.
[7] J. Fan, D. Yau, A. Elmagarmid, and W. Aref, "Automatic image segmentation by integrating color-edge extraction and seeded region growing," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1454–1466, Oct. 2001.
[8] S. Wan and W. Higgins, "Symmetric region growing," *IEEE Trans. Image Process.*, vol. 12, no. 9, pp. 1007–1015, Sep. 2003.
[9] L. Liu and S. Sclaroff, "Region segmentation via deformable model-guided split and merge," in *Proc. Int. Conf. Comp. Vis.*, Vancouver, BC, Canada, Jul. 2001, vol. 1, pp. 98–104.
[10] C. D'Elia, G. Poggi, and G. Scarpa, "A tree-structured markov random field model for bayesian image segmentation," *IEEE Trans. Image Process.*, vol. 12, no. 10, pp. 1259–1273, Oct. 2003.
[11] H. Derin and H. Elliott, "Modeling and segmentation of noisy and textured images using gibbs random fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, no. 1, pp. 39–55, Jan. 1987.
[12] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, Jul. 1989.
[13] M. Unser, "Texture classification and segmentation using wavelet frames," *IEEE Trans. Image Process.*, vol. 4, no. 11, pp. 1549–1560, Nov. 1995.
[14] G. Fan and X. Xia, "Wavelet-based texture analysis and synthesis using hidden markov models," *IEEE Trans. Circuits Syst.*, vol. 50, no. 1, pp. 106–120, Jan. 2003.
[15] J. Chen, T. Pappas, A. Mojsilovic, and B. Rogowitz, "Adaptive image segmentation based on color and texture," in *Proc. Int. Conf. Image Process.*, Jun. 2002, vol. 3, pp. 777–780.
[16] J. Chen, T. Pappas, A. Mojsilovic, and B. Rogowitz, "Adaptive perceptual color-texture image segmentation," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1524–1536, Oct. 2005.
[17] Y. Deng and B. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 8, pp. 800–810, Aug. 2001.
[18] G. Paschos, "Perceptually uniform color spaces for color texture analysis: An empirical evaluation," *IEEE Trans. Image Process.*, vol. 10, no. 6, pp. 932–937, Jun. 2001.
[19] J. Yoon and I. Kweon, "Color image segmentation considering the human sensitivity for color pattern variations," in *Proc. SPIE*, Oct. 2001, vol. 4572, pp. 269–278.
[20] H. Chen, W. Chien, and S. Wang, "Contrast-based color image segmentation," *IEEE Signal Process. Lett.*, vol. 11, no. 7, pp. 641–644, Jul. 2004.
[21] Y. Chang, D. Lee, and Y. Wang, "Color-texture segmentation of medical images based on local contrast information," in *Proc. IEEE Symp. Comput. Intel. Bioinf. Comput. Bio.*, Apr. 2007, pp. 488–493.
[22] H. Huang, Y. Chen, and W. Hsu, "Color image segmentation using a self-organizing map algorithm," *J. Elect. Imag.*, vol. 11, no. 2, pp. 136–148, Apr. 2002.
[23] H. Lee and D. Cok, "Detecting boundaries in a vector field," *IEEE Trans. Signal Process.*, vol. 39, no. 5, pp. 1181–1194, May 1991.
[24] P. Green and L. MacDonald, *Color Engineering*. New York: Wiley, 2002.
[25] R. Haralick and L. Shapiro, *Computer and Robot Vision*. New York: Addison-Wesley, 1992, vol. 1, pp. 28–48.
[26] T. Cover and J. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.
[27] C. Chou and T. Wu, "Embedding color watermarks in color images," *EURASIP J. Appl. Signal Process.*, vol. 2003, no. 1, pp. 32–40, Oct. 2003.
[28] W. J. Krzanowski, *Principles of Multivariate Analysis*. Oxford, U.K.: Oxford Univ. Press, 1988, ch. 11.
[29] Y. J. Zhang, "A survey on evaluation methods for image segmentation," *Pattern Recognit. Soc.*, vol. 29, no. 8, pp. 1335–1346, 1996.
[30] J. S. Cardoso and L. Corte-Real, "Toward a generic evaluation of image segmentation," *IEEE Trans. Image Process.*, vol. 14, no. 11, pp. 1773–1782, Nov. 2005.
[31] F. Ge, S. Wang, and T. Liu, "New benachmark for image segmentation evaluation," *J. Elect. Imag.*, vol. 16, no. 3, Jul.–Sep. 2007.
[32] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "Toward objective evaluation of image segmentation algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 929–944, Jun. 2007.
[33] R. Unnikrishnan and M. Hebert, "Measures of similarity," in *Proc. IEEE Workshop Comp. Vis. Appl.*, Jan. 2005, vol. 1, pp. 394–400.

[34] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. IEEE Int. Conf. Comput. Vis.*, Vancouver, BC, Canada, Jul. 2001, vol. 2, pp. 416–423.

Luis Garcia Ugarriza was born in Bogota, Colombia, in 1982. He received the B.S. degree in electrical engineering and the M.S. degree with a focus on signal and image processing from the Rochester Institute of Technology, Rochester NY, in 2004 and 2007, respectively.

He is currently with Zoran Corporation, Burlington, MA, in the Imaging Department, developing technology to improve the quality and speed of printer output. His research interests include computer vision, digital image segmentation, and classification of objects. He has worked on problems related to wireless audio data transportation, image segmentation, and video surveillance.

Eli Saber (SM'00) received the B.S. degree in electrical and computer engineering from the University of Buffalo, Buffalo, NY, in 1988, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Rochester, Rochester, NY, in 1992 and 1996, respectively.

He is an Associate Professor in the Electrical Engineering Department, Rochester Institute of Technology (RIT). He was with Xerox Corporation from 1988 to 2004 in a variety of positions ending as Product Development Scientist and Manager in the Business Group Operations Unit. From 1997 to 2004, he was an adjunct faculty member in the Electrical Engineering Department, RIT, and in the Electrical and Computer Engineering Department, University of Rochester. His research interests are in the areas of digital image and video processing, including image/video segmentation, object tracking, content-based video analysis and summarization, multicamera surveillance video processing, and image understanding.

Dr. Saber is a senior member of the Electrical Engineering Honor Society Eta Kappa Nu and the Imaging Science and Technology Society (IS&T). He is an Associate Editor of the IEEE Transactions on Image Processing, the *Journal of Electronic Imaging*, and guest editor for the special issue on Color Image Processing for the *IEEE Signal Processing Magazine*. He is the former Chair of the IEEE Technical Committee (TC) on Industry DSP Technology and a member of the Image and Multi-Dimensional Digital Signal Processing TC. He was the finance chair for the 2002 International Conference on Image Processing, the general chair for the 1998 Western New York Imaging Workshop, and a member of several conference committees.

Sreenath Rao Vantaram was born in Visakhapatnam, India, in 1984. He received the B.Tech. (equivalent B.S.) degree in electronics and communication engineering from the Jawaharlal Nehru Technological University (JNTU), Andhra Pradesh, India, in 2006. He is currently pursuing the M.S. degree in electrical engineering, with a focus area in signal, image, and video processing, at the Rochester Institute of Technology (RIT), Rochester, NY.

His research interests are in the areas of computer vision and digital image/video processing including image/video segmentation, image/video analysis, and object recognition.

Vincent Amuso received the B.S.E.E. degree from Western New England College in 1987, the M.S.E.E. degree from Syracuse University, Syracuse, NY, in 1989, and the Ph.D. degree in electrical engineering from the Rensselaer Polytechnic Institute, Troy, NY, in 1998.

He is an Associate Professor and Head of the Electrical Engineering Department at the Rochester Institute of Technology. Prior to that, he was employed by the Sensis Corporation (1997-1999) as a senior radar systems engineer, Mohawk Valley Community College (1994-1997) as a professor, the General Electric Company (1988-1994) as an electrical/systems engineer, and the Rome Air Development Center (1987-1988) as an electronics engineer in the Advanced Concepts and Analysis Branch. His research area includes the design and analysis of deep Ground Penetrating Radar (GPR) systems, as well as three-dimensional SAR target modeling. He is also actively working both as a researcher and consultant in the area of waveform diversity and design.

Mark Shaw received the B.S. degree in graphic arts printing technology from the University of Hertfordshire, U.K., and the M.Sc. degree in color science from the Munsell Color Science Laboratory, Rochester Institute of Technology, Rochester, NY, in 1999.

He is currently with the Color Laserjet Division, Hewlett Packard Company, Boise, ID, as a Color Imaging Expert. He has ten years experience in the color and imaging industry, having previously worked for other major corporations, including 3M and Xerox. His interests include multispectral color reproduction, color modeling, HDR image rendering, color management, and image understanding.

Dr. Shaw was awarded the Grum Scholarship from the Munsell Color Science Laboratory in 1999, presented bi-annually to outstanding graduate students for leadership, dedication, and academic performance. He has also been awarded the Varns Excellence in Printing Award, the AGFA Printing Award, and the Institute of Printing Bronze Medal.

Ranjit Bhaskar received the B.S. degree in mechanical engineering from General Motors Institute in 1986 and the M.S. degree in imaging science from the Rochester Institute of Technology, Rochester, NY, in 1990.

He is an Image Scientist on Hewlett Packard's Color and Imaging Team, Vancouver, WA. Prior to joining Hewlett Packard, he was a contractor at the Johnson Space Center, Houston, TX, between 1990 and 1996. He served in numerous roles at NASA, including stints as a technical lead in the Space Shuttle Safety Program, as well as coordinating image requirements for flight experiments during joint USA/Russia missions to the Mir Space Station. Prior to that, he was a consultant to Xerox Corporation in their scanner group. His research interests include image understanding and analysis, as well as image classification and segmentation.