Major Project Report
On

OBJECT SEGMENTATION USING REGION GROWING AND EDGE CONSTRAINTS

Submitted in partial fulfillment of the requirement
for the award of the degree of

MASTER OF TECHNOLOGY
(INFORMATION SYSTEM)

Submitted by

JYOTI SWARUP

(06/IS/10)

Under the Guidance of

SEBA SUSAN

(Asst. Professor)

Dept. of Information Technology

Department of Information Technology

DELHI TECHNOLOGICAL UNIVERSITY

Bawana Road, Delhi-110042

2010-2012

CERTIFICATE

This is to certify that the Major Report entitled as “Object Segmentation using Region Growing and Edge Constraints” is being submitted by **Jyoti Swarup**, 06/IS/10, in partial fulfillment for the award of degree of Master of Technology in Information System from Delhi Technological University, Delhi, is the original work carried out by her under the guidance of Asst. Prof. Seba Susan. The matter contained in this thesis report has not been submitted elsewhere for reward of any other degree.

(Project Guide)

Ms. Seba Susan

Assistant Professor
Dept. of Information Technology
Delhi Technological University, Delhi

ACKNOWLEDGEMENT

First of all, I would like to present my sincere thanks to my project guide Ms. Seba Susan, Asst. Prof. of Information Technology, DTU, Delhi for her insightful guidance and constant encouragement during the preparation of this project report.

Secondly, I would like to thank Dr. O.P.Verma, Head of Dept. of Information Technology, DTU, Delhi, for giving me the freedom to pursue my own research interest and study diverse topics throughout my M.Tech study. I would thank all the members of DTU and my friends who have helped me in the completion of my thesis report.

Jyoti Swarup

M.Tech (Information System)
06/IS/10

ABSTRACT

This thesis focuses upon Object Segmentation which plays an important role in the field of computer vision. The image segmentation problem is concerned with partitioning an image into multiple regions according to some homogeneity criterion. Object segmentation is used to typically locate objects and boundaries in images.

The proposed object segmentation method is an integration of region growing and edge information. This method automatically selects the initial seed and determines the threshold with the help of a 20x20 window across the center pixel for single seeded region growing. Automatic threshold is determined by the difference between mean and median of this window, whereas, minimum distance between mean and all pixels of window help in initial seed selection. The grown region is used for object segmentation by placing edge constraints over it to obtain nearest strong canny edges. Further, certain morphological operations are performed to obtain precise results. The proposed algorithm is applied to state-of-the art database PASCAL VOC 2005 and compared to the method proposed by Xavier Bresson based on the Active Contour model. This algorithm produces successful results and accompanying ground truth annotations helps in determining precision and recall parameters. On the basis of evaluation of these parameters it can be said that proposed method produce good segmentation results with high precision.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1

INTRODUCTION TO IMAGE SEGMENTATION

1.1 Image Segmentation

Image Segmentation is a classical problem in image processing which partitions an image into set of non-overlapping homogenous regions. Many researchers in image processing focus upon certain portions of an image called foreground (rest is called background) having unique and specific nature. This background needs to be extracted in order to identify and analyze object. Object segmentation consists of extracting the shape of complete physical objects and eliminating the texture edges. Object segmentation is almost impossible without semantic knowledge about the scene or when only single-image analysis is performed. Object segmentation is particularly important in computer vision applications such as medical image analysis, video surveillance, content-based image retrieval, etc.

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image. Each of the pixels in a region are similar with respect to some characteristic such as color, intensity or texture. Image segmentation has vast application in medical imaging to locate tumors and measure tissue volumes, locate objects in satellite images, and face recognition and so on. Several general-purpose algorithms and techniques have been developed for image segmentation. Since there is no general solution to the image segmentation problem, these techniques often have to be combined with domain knowledge in order to effectively solve an image segmentation problem for a problem domain.

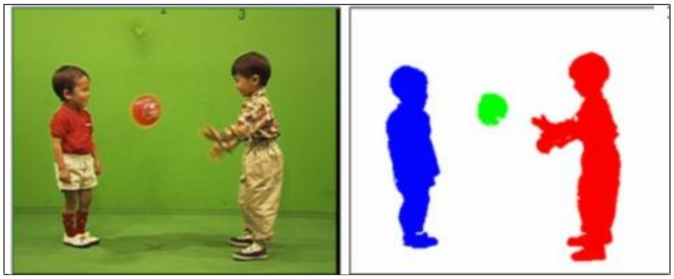


Fig. 1.1 Example of Object Segmentation

1.2 Image segmentation techniques

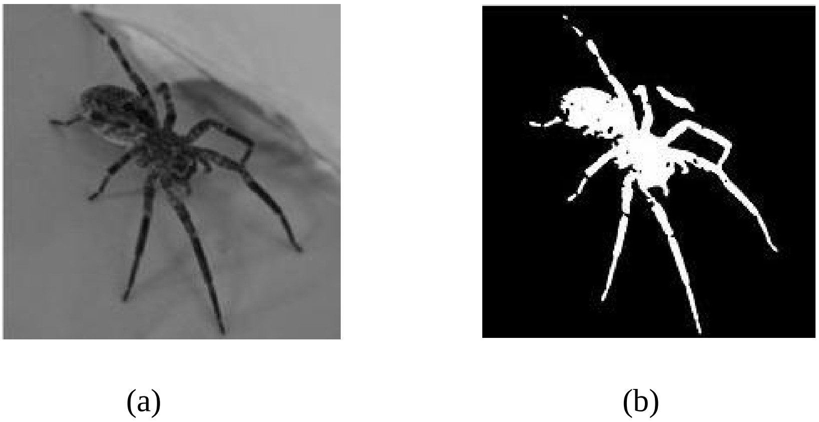
For segmentation of intensity images there are several approaches [1]-[6],[10],[11] in the literature.

THRESHOLDING METHOD

The simplest method of image segmentation is called the thresholding method. This method is based on a clip-level (or a threshold value) to turn a gray-scale image into a binary image. The key parameter in the thresholding process is the choice of the threshold value. A simple method to choose threshold value would be to choose the mean or median value, the rationale being that if the object pixels are brighter than the background, they should also be brighter than the average. In a noiseless image with uniform background and object values, the mean or median will work well as the threshold, however, this will generally not be the case. Threshold segmentation generates a binary image given as follows:

$$G(x,y) = \begin{cases} 1\ for\ f(x,y) \geq \mathrm{T} \\ 0\ for\ f(x,y) < T \end{cases} \tag{1.1}$$

Where, G(x,y) is the resultant binary image and f(x,y) is the original image. T is the threshold value.



(a)

(b)

Fig. 1.2: Example of Threshold based image segmentation. (a)Original image (b) Segmented result.

A more sophisticated approach might be to create a histogram of the image pixel intensities and use the valley point as the threshold. The histogram approach assumes that there is some average values for both the background and object pixels, but that the actual pixel values have some variation around these average values. Thresholding is called adaptive thresholding when a different threshold is used for different regions in the image. This may also be known as local or dynamic thresholding [44].

Threshold techniques [2] are based on the postulate that all pixels whose value (gray level, color value, or other) lie within a certain range, belong to one class probably the foreground and rest are background pixels. Such methods neglect all of the spatial information of the image and do not cope well with noise or blurring at boundaries. Global thresholding method can fail when background illumination is uneven. Balanced histogram thresholding method is used for automatic thresholding. It tries to find the optimum threshold level that divides the histogram in two classes.

BOUNDARY BASED METHOD

Boundary based methods are often used to look for explicit or implicit boundaries between different region. The two most commonly used boundary based methods are known as *ridge detection* and *edge-detection.*

- Ridge detection follows the peaks (local maxima) in the original image. For example, one way to outline the shape of a crater in a two-dimensional surface plot is to ``walk along'' the crest of the crater's rim. The circular path that would be traced out would be a much more compact representation of the crater than say, the set of all points inside the crater.

- When objects don't have large enough ridges at the region boundaries for ridge detection, the *gradient operator* can be used to enhance the boundaries (i.e., edges) between distinct regions. Edge detection is identical to ridge detection except that peaks are tracked in the gradient space of the image instead of the original image space. The gradient space can be computed by simply applying the gradient operator to the entire image. The two-dimensional gradient-operator is defined as,

$$\nabla I(x,y) \triangleq \vec{\imath}\frac{\partial}{\partial x}I(x,y) + \vec{\jmath}\frac{\partial}{\partial y}I(x,y). \tag{1.2}$$

When it is necessary to distinguish the inside versus the outside of the edge, the Laplacian operator can be used. In two dimensions, the Laplacian operator is defined as,

$$L[I(x,y)] \triangleq \frac{\partial^2}{\partial x}I(x,y) + \frac{\partial^2}{\partial y}I(x,y). \tag{1.3}$$

The drawbacks of ridge-and edge-detection methods are that they can produce spurious, missing, or discontinuous edges. Hence, they can suffer from inadequate sensitivity and specificity because the image in the gradient space must be thresholded or otherwise classified according to edge or non-edge membership. Also, the problem of tracking an edge that bifurcates into two or more edges is one that cannot be adequately resolved using these low-level image operators alone. Furthermore, the derivative operator is inherently noisy and will exacerbate any noise already present in the image. Hence, the images must be smoothed before applying the gradient or Laplacian operators. Unfortunately, smoothing an image can hide or blur fine structures and other subtle features.

Boundary based methods [3] use postulate that pixel values change abruptly at object boundaries. The most crucial part here is to apply a gradient operator (e.g., Sobel, Roberts filter). High values of this filter gives candidates for region boundaries. These candidates when modified to produce closed curve represents the boundaries between the regions. Boundary based methods cannot incorporate region information such as texture easily.

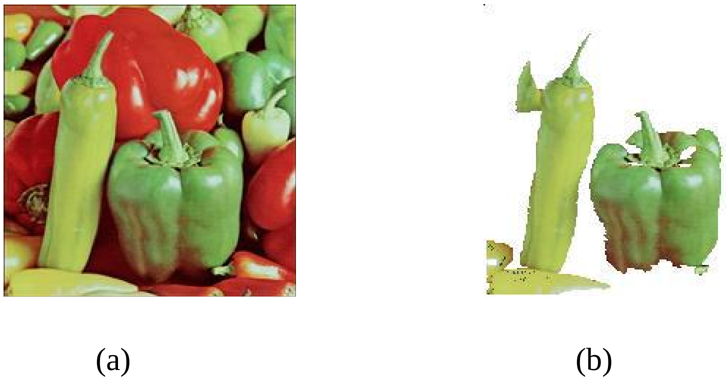
REGION BASED METHODS

The main idea here is to classify a particular image into a number of regions or classes. Thus for each pixel in the image it is needed to somehow decide or estimate that which class it belongs to. Region based methods involves the fact that all neighboring pixels of a region have similar properties. There are a variety of approaches to do region based segmentation and to our understanding the performance does not change from one method to the other considerably. Region-based segmentation methods attempt to partition or group regions according to common image properties. These image properties consist of

1. Intensity values from original images, or computed values based on an image operator.

2. Textures or patterns those are unique to each type of region.

3. Spectral profiles that provide multidimensional image data.



(a) (b)

Fig. 1.3 Example of Region based method. (a) Original image (b) Segmented image

SPLIT-AND-MERGE TECHNIQUE

Split-and-merge technique [4] is the most popular region based method for image segmentation. The split and merge algorithm have two phases: the split, and the merge phase. In the split phase we recursively split regions into four sub-regions (starting with the whole image as one region) until our homogeneity criterion is met in all sub-regions. In the merge step we check that P(Ri U Rj) =TRUE for each two neighbor regions, and merge the two regions. This step is repeated until no more changes are necessary. Splitting and merging attempts to divide an image into uniform regions. The basic representational structure is pyramidal, i.e. a square region of size m by m at one level of a pyramid has 4 sub-regions of size m/2 by m/2 below it in the pyramid. Usually the algorithm starts from the initial assumption that the entire image is a single region, and then computes the homogeneity criterion to see if it is TRUE. If FALSE, then the square region is split into the four smaller regions. This process is then repeated on each of the sub-regions until no further splitting is necessary. These small square regions are then merged if they are similar to give larger irregular regions. The problem (at least from a programming point of view) is that any two regions may be merged if adjacent and if the larger region satisfies the homogeneity criteria, but regions which are adjacent in image space may have different parents or be at different levels (i.e. different in size) in the pyramidal structure. Now all the conditions are met, and image is segmented into sub-regions. When a special data structure is involved in the implementation of the algorithm of the method, its time complexity can reach O(nlogn), an optimal algorithm of the method.

Here is a pseudo code for split and merge algorithm:

1. Initially, we have only one big region (the whole image).
2. Split: If P (R_i) = TRUE proceed to next step. Otherwise subdivide R_i to four sub-regions and perform step 2 on them.
3. Merge: If R_i and R_j are neighbors and P (R_i U R_j) = TRUE, merge the two regions, than repeat step 3. If there are no such regions we are finished.

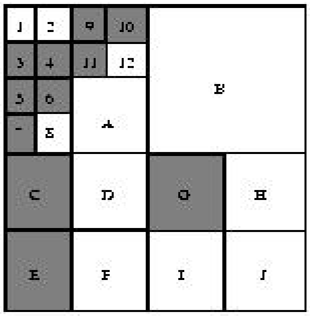
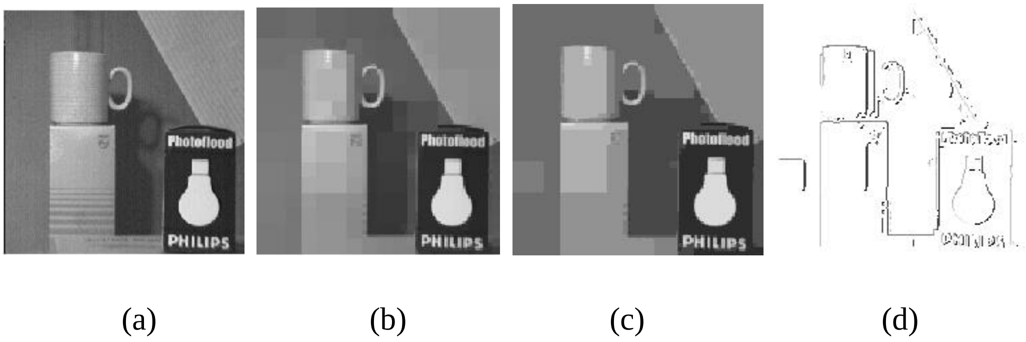


Fig. 1.3: Quad Splitting

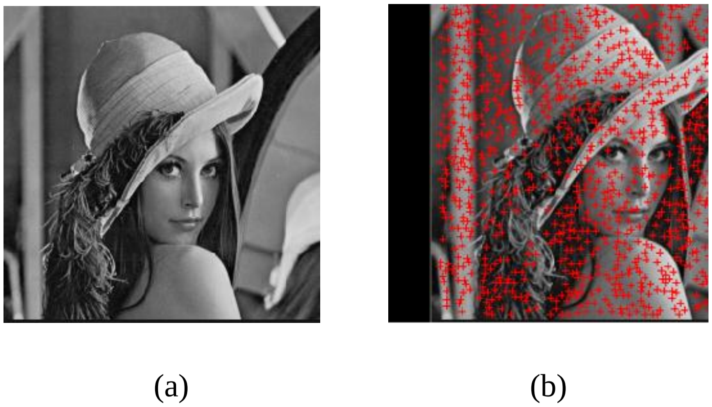


(a) (b) (c) (d)

Fig. 1.4: Example of Split-and-merging method. (a) Original image (b) After Quad splitting (c)After merging (d) Regional segmentation

SEEDED REGION GROWING

Mehnert and Jackway [34] pointed out that Seeded Region Growing (SRG) has two inherent pixel order dependencies that cause different resulting segments. The first-order dependency occurs whenever several pixels have the same difference measure to their neighboring regions. The second-order dependency occurs when one pixel has the same difference measure to several regions. They used parallel processing and re-examination to eliminate the order dependencies.



(a) (b)

Fig. 1.5: Example of SRG. (a)Original image (b) Seeds are in red color

HYBRID TECHNIQUES

The fourth type is the hybrid techniques which combine boundary and region criterion. This class includes morphological watershed segmentation [5] and variable-order surface fitting [6]. The watershed method is generally applied to the gradient of the image. This technique encounters difficulties with images in which regions are both noisy and have blurred or indistinct boundaries. This is a computationally expensive method. In [47], the image is initially partitioned into regions using surface curvatures and, then, a variable-order surface fitting iterative region merging process is initiated. In [9], the image is initially segmented using the region-based split-and-merge technique and, then, the detected contours are refined using edge information. In [48], an initial image partition is obtained by detecting ridges and troughs in the gradient magnitude image through maximum gradient paths connecting singular points. Then, region merging is applied through the elimination of ridges and troughs via similarity or dissimilarity measures. Many morphological segmentation approaches using the watershed transform have been proposed. Watersheds have also been used in multi resolution methods for producing resolution hierarchies of image ridges and valleys. Although these methods were successful in segmenting certain classes of images, they require significant interactive user guidance or accurate prior knowledge on the image structure. Kostas et al. [19] proposed a faster algorithm, which additionally maintains the so-called nearest neighbor graph, due to which the priority queue size and processing time are drastically reduced. The final segmentation provides, due to the RAG, one-pixel wide, closed, and accurately localized contours or surfaces.

CLUSTERING METHODS

In clustering process [10][11], a data set (pixels) is represented by clusters, pixels of one cluster have similar characteristics of color, texture etc. There exist two natural algorithms for clustering: *divisive clustering* and *agglomerative clustering.* The difficulty in using either of the methods directly is that there are lots of pixels in an image; larger the image size more the complexity of this algorithm. However, clustering image segmentation has many problems. The number of regions in image has to be known a priori. Selection of initial seeds greatly affects the resulting segmentation. Arnau Oliver et al. [35] proposed a method to improve these problems which begins with automatic seeds placement using contour information. It included two more restrictions with clustering algorithm: a) do not classify the pixels with a high gradient; that is, pixels that belong to a boundary region and b) do not classify the pixels far away from the center of all clusters (noise pixels).

Some of the classical clustering algorithms include K-means algorithm, Fuzzy C-Mean algorithm and Gaussian of Mixtures algorithm. The popular k-Means algorithm [36] is an error minimization algorithm where the function to minimize is the sum of squared error:

$$e^2(K) = \sum_{k=1}^{K} \sum_{i \in C_k} (x_i - c_k)^2 \tag{2.1}$$

where c_k is the centroid of cluster c_k, and K is the number of clusters (known a priori). Two factors have made the k-Means popular: it has linear time complexity and it is easy to implement [37]. One restriction of the k-Means algorithm is that it associates each pattern of the image into one, and only one, cluster. With the use of fuzzy theory, each pattern can be associated with every cluster using a membership function. Another way to allow each pattern to belong to different clusters is by using the Gaussian of Mixtures algorithm [14]. In this probabilistic model, each pattern is characterized by a set of Gaussian mixtures:

$$p(x_i; K) = \sum_{k=1}^{K} \pi_k \, g_k(x_i) \qquad (2.2)$$

where, g_i is a Gaussian distribution and π_i a prior distribution ($\sum_k \pi_k = 1$). The model parameters and the cluster membership function are determined by maximizing the log-likelihood function: $l(K) = \sum_{i \in I} \log\big(p(x_i; K)\big)$ (2.3)

where I is the whole image. This step is efficiently done by using the Expectation Maximization algorithm [38].

Most approaches to image segmentation separate segments containing pixels with similar intensity. Ehquierdo and M.Ghanbari in [12] introduced an object segmentation algorithm based on asymmetrically weighted Gaussian filtering and information extracted from previously estimated disparity fields. This reduced computational cost by applying asymmetrically weighted Gaussian filtering only along the object borders. In the remaining image areas, isotropic diffusion and thresholding are performed.

Different from region-based schemes, the edge-based schemes relying on detecting discontinuities of captured surfaces can localize surface boundaries more precisely, in comparison with the region-based algorithms [45, 46]. On the other hand, detected edges may not be completely reconstructed since the noise disturbance and limited sampling of the edge detection algorithm may divide the actual edge to many undesirable surface sections.

1.3 LITERATURE WORK

Segmentation using Region based methods

Pratikakis et al. [13] invented the idea of measuring the region weighting based on a hierarchical watershed-driven algorithm that extracts meaningful regions automatically. Carson et al. [14] proposed the Blobworld system, in which a user is required to select important regions and features. This computationally expensive method is highly user interactive and requires prior information of region of interest. Wang et al. [15] proposed an integrated matching algorithm to retrieve images from picture libraries based on region similarity in terms of information of color, shape, and texture combined all together. However, this approach does not provide a general way to measure image similarity using region spatial relationship, which is an important clue for middle-level image understanding. Hsieh and Grimson [16] proposed an image retrieval framework by region matching using spatial templates. They support one-to-many regions matching in two stages –a similarity comparison followed by a region voting. Ballard [18] used Hough transform method to detect arbitrary shapes in images. The only drawback of this method is the fact that the edge strength used to determine the candidate boundary points is not descriptive enough. Chuang et al. [17] proposed a region-based object retrieval using the generalized Hough transform (GHT) and content aware image segmentation to overcome the problem of setting single threshold to obtain a perfect segmentation result. In this paper, a training mechanism is used to adaptively select stable segmentation parameters for each database image. In addition, content-aware image segmentation is proposed to synchronize image segmentation of query and target images.

Segmentation using Edge based methods

Michael Kass et al. [27] proposed a model that snakes can be considered as active contour models. They lock onto nearby edges, localizing them accurately once they are supplied with the starting points. In this model, issues such as the connectivity of the contours and the presence of corners affect the energy functional and hence the detailed structure of the locally optimal contour. Unlike most other techniques, this model is active as it is always minimizing its energy functional and therefore exhibits dynamic behavior. The only drawback of this model is the existence of local minima in the active contour energy, which makes initial guess critical to get satisfactory results. To solve this problem by determining a global minimum of the active contour model Xavier Bresson et al. [29] proposed an approach based on the unification of image segmentation and image denoising tasks into a global minimization framework. Leventon et al. [30] developed active contours that use a shape model defined by a principal components analysis (PCA). In their approach, the active contour evolves locally based on image gradients and curvature and globally towards the maximum a posteriori (MAP) probability of position and shape of the prior shape model.

Segmentation by combining Region growing and edge detection

Pavlidis and Liow [9] presented a method that combines region growing and edge detection for image segmentation. They started with split-and-merge algorithm such that it results in over-segmentation. Then region boundaries are eliminated depending upon two criteria: a) criteria that integrate contrast with boundary smoothness, variation of the image gradient along the boundary, and b) a criterion that penalizes for the presence of artifacts reflecting the data structure used during segmentation (quad tree in this case). Bajcsy [22] showed that both edge detection and region growing processes could be integrated by making the decision whether a point is on a boundary or on a homogenous surface. Anderson and Bajcsy [23] showed a combination of edge detection and region growing where they used edge detection to initialize a region growing process based on a local similarity threshold which was used to check whether two points belong to the same region. Yau et al. [20] proposed an algorithm in which color edges in an image are first obtained automatically by combining an improved isotropic edge detector and a fast entropic thresholding technique. The centroids between these adjacent edge regions are taken as the initial seeds for seeded region growing (SRG). Gambotto [39] describes an algorithm that starts with the gradient image and an initial patch located inside the boundary of the region, thereafter iteratively merging the adjacent pixels to the region. This growth process is controlled by two criteria. First compare the image data with region model and then use gradient information to detect optimal region boundaries. Aggarwal [21] proposed an integration algorithm that enables multiple edge detection and region segmentation modules to work in parallel as front ends. The solution procedure consists of three steps. A maximum likelihood estimator provides initial solutions to the positions of edge pixels from various inputs. An iterative procedure using only local information (without edge tracing) then minimizes the contour curvature. Finally, regions are merged to guarantee that each region is large and compact. Chalan et al. [40] proposed a modified active contour model (using Bayesian approach) that uses region statistics in addition to gradient information. This method integrates region growing and edge detection in a regularization framework. Zhang Guoying et al. [41] proposed a new segmentation method based on Seed Region and Boundary Growing (SRBG) to segment blurred and conglutinated bubbles in floating image. Bright pixels on top of bubbles contribute as seed regions. The seed areas are extracted by a gray threshold method and each seed boundary is divided into four consecutive curves in the left-top, right-top, right-bottom and left-bottom directions. Each curve then grows outward until the bubble boundary conditions are satisfied. This method segments each bubble separately rather than operating on the whole image. In this thesis we have proposed a new method for object segmentation which integrates region growing with edge information.

A similar work has been done by Xuan, Adali and Wang [43] on MR brain images. Their method starts with simple region growing technique which produces over-segmented region. Edge information is then integrated to verify and, where necessary, to correct region boundaries. Their work is highly dependent on region growing. Our proposed algorithm is different from theirs in terms that our method discards region growing results once it has obtained nearest strong edges. Therefore, our method works efficiently even if region growing results are not efficient enough.

CHAPTER-2

REVIEW OF REGION GROWING AND EDGE DETECTION

2.1 Review of Region Growing

Out of the many image segmentation techniques, region growing is the most popular one. Region growing is a procedure that group pixels or sub-regions into larger regions based on predefined criteria. The basic approach is to start with a set of seed points and from these grow regions by appending to each seed those neighboring pixels that have properties similar to the seed.

Adam and Bischof [1] proposed a new ‘seeded region growing approach’ for image segmentation. It required prior knowledge of initial points called seeds to grow a region based on some similarity criteria. This method, inherently depend upon the selection of seeds and homogeneity criteria which would largely affect the segmentation of image into various regions.

Let R represent the entire image. Segmentation partitions R into **n** non-overlapping sub regions R_1, R_2,……, R_n, such that [7]

a) $U\, R_i = R$ where i=1,2,…,n

b) R_i is a connected region, i=1,2,…,n

c) $R_i \cap R_j = □$ for all i and j, i≠j

d) $P(R_i)$ = TRUE for i=1,2,…,n

e) $P(R_i U R_j)$ = FALSE for i≠j

Here, $P(R_i)$ is a predicate defined over the points in set R_i and □ is the null set. Here, conditions indicate that every pixel must be in a region. All the points in a region must be connected in some predefined sense. Regions must be disjoint. All the pixels in a region must have similar properties. Finally, two different regions should be different in the sense of predicate P.

Region growing process can be depicted pictorially in fig. 2.1

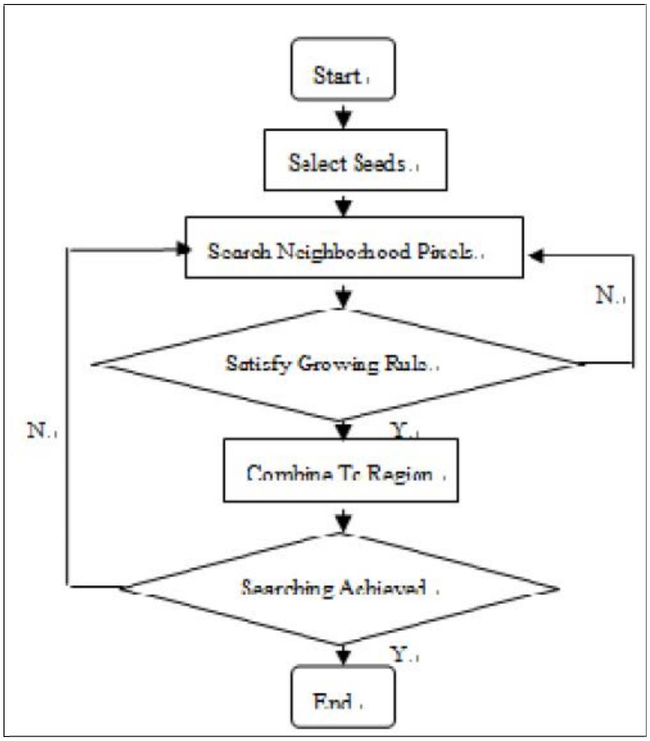


Fig. 2.1: Region growing process

The **algorithm** for implementing Seeded Region Growing (boundary flagging case) is as follow:

```
Label seed points according to their initial grouping.
Put neighbors of seed points in the stack.
While the stack is not empty:
    Remove first pixel (x,y) from stack.
    Test the neighbors of this point:
        Check for similarity condition
        If all neighbors of this pixel which are already labeled
        (other than with the boundary label) have the same label-
            Set pixel to this label.
        Otherwise
            Flag the pixel with the boundary label
```

Seeded region growing requires seeds as additional input. The segmentation results are dependent on the choice of seeds. Noise in the image can cause the seeds to be poorly placed.

Unseeded region growing is a modified algorithm that doesn't require explicit seeds. It starts off with a single region A_1 – the pixel chosen here does not significantly influence final segmentation. In each iteration, it considers the neighboring pixels in the same way as seeded region growing. It differs from seeded region growing in that if the minimum δ is less than a predefined threshold T then it is added to the respective region A_j. If not, then the pixel is considered significantly different from all current regions A_i and a new region A_{n+1} is created with this pixel. Felzenszwalb and Huttenlocher's algorithm [25] is an example of the region-growing approach, which starts with each pixel in an individual segment and iteratively merges the pixels into larger regions. Regions in this algorithm can be either high-variance, as in the case of a highly textured surface, or low-variance, such as a region of constant color, and the merging procedure automatically adjusts its criteria to maintain these properties.

One variant of this technique, proposed by Haralick and Shapiro [24], is based on pixel intensities. The mean and scatter of the region and the intensity of the candidate pixel is used to compute a test statistic. If the test statistic is sufficiently small, the pixel is added to the region, and the region's mean and scatter are recomputed. Otherwise, the pixel is rejected, and is used to form a new region.

Conventional segmentation techniques for monochromatic images can be categorized into two distinct approaches [8]. One is region based, which relies on the homogeneity of spatially localized features, whereas the other is based on boundary finding, using discontinuity measures. The two methods exploit two different definitions of a region which should ideally yield identical results. Homogeneity is the characteristic of a region and non-homogeneity or discontinuity is the characteristic of the boundary of a region [1],[8],[9].

A special region growing method is called λ-connected segmentation (see also lambda-connectedness). It is based on pixel intensities and neighborhood linking paths. A degree of connectivity (connectedness) will be calculated based on a path that is formed by pixels. For a certain value of λ, two pixels are called λ-connected if there is a path linking those two pixels and the connectedness of this path is at least λ. λ-connectedness is an equivalence relation.

2.2 Review of Edge detector

Edge detection aims at identifying points in a digital image at which the image brightness changes sharply or has discontinuities. Image Edge detection significantly reduces the amount of data and filters out useless information, while preserving the important structural properties in an image. Edge detection methods [7] locate the pixels in the image that correspond to the edges of the objects seen in the image. The result is a binary image with the detected edge pixels. Common algorithms like Sobel, Prewitt and Laplacian operators are suitable for simple and noise-free images. They often produce missing edges, or extra edges on complex and noisy images. In complex images, the edges identified are often disconnected, while closed region boundaries are required to segment an object. In a study, Maini and Aggarwal [31] conclude that canny's edge detection algorithm performs better than all other operators under almost all scenarios. The edges identified by edge detection are often disconnected. To segment an object from an image however, one needs closed region boundaries. The desired edges are the boundaries between such objects.

The Canny edge detector, developed by John F. Canny in 1986, uses a multi-stage algorithm to detect a wide range of edges in images. In his paper [32], this algorithm first smoothes the image to eliminate noise, then finds image gradient to highlight regions with high special derivatives. It is in most cases impossible to specify a threshold at which a given intensity gradient switches from corresponding to an edge into not doing so. Therefore Canny uses thresholding with hysteresis. Thresholding with hysteresis requires two thresholds – high and low. Making the assumption that important edges should be along continuous curves in the image allows us to follow a faint section of a given line and to discard a few noisy pixels that do not constitute a line but have produced large gradients.

The Canny operator works in a multi-stage process. First of all the image is smoothed by Gaussian convolution. Then a simple 2-D first derivative operator (somewhat like the Roberts Cross) is applied to the smoothed image to highlight regions of the image with high first spatial derivatives. Edges give rise to ridges in the gradient magnitude image. The algorithm then tracks along the top of these ridges and sets to zero all pixels that are not actually on the ridge top so as to give a thin line in the output, a process known as *non-maximal suppression*. The tracking process exhibits hysteresis controlled by two thresholds: T_1 and T_2, with $T_1 > T_2$. Tracking can only begin at a point on a ridge higher than T_1. Tracking then continues in both directions out from that point until the height of the ridge falls below T_2. This hysteresis helps to ensure that noisy edges are not broken up into multiple edge fragments.

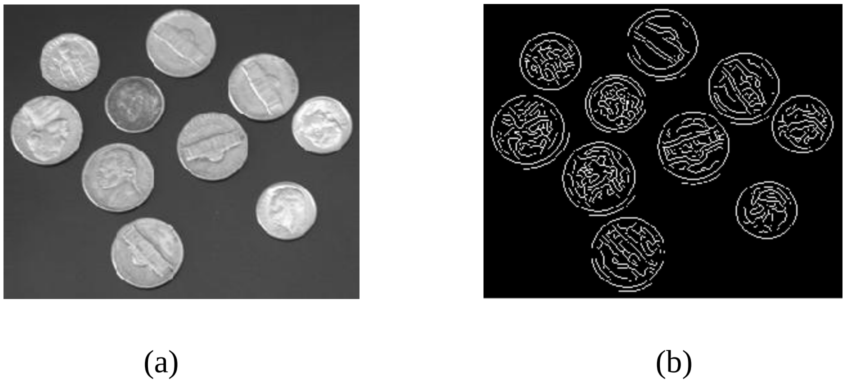


Fig. 2.2: Example of canny edge detection. (a)Original image, (b) Canny Filter

The effect of the canny operator is determined by three parameters: the width of the Gaussian kernel used in the smoothing phase, and the upper and lower thresholds used by the tracker. Increasing the width of the Gaussian kernel reduces the detector's sensitivity to noise, at the expense of losing some of the finer detail in the image. The localization error in the detected edges also increases slightly as the Gaussian width is increased. Usually, the upper tracking threshold can be set quite high and the lower threshold quite low for good results. Setting the lower threshold too high will cause noisy edges to break up. Setting the upper threshold too low increases the number of spurious and undesirable edge fragments appearing in the output.

One problem with the basic canny operator is to do with Y-junctions *i.e.* place where three ridges meet in the gradient magnitude image. Such junctions can occur where an edge is partially occluded by another object. The tracker will treat two of the ridges as a single line segment, and the third one as a line that approaches, but doesn't quite connect to, that line segment.

CHAPTER-3

PROPOSED ALGORITHM FOR OBJECT SEGMENTATION

3.1 Motivation

Region-based methods tend to be global, optimizing a functional of the image segmentation. On the other hand, they often ignore important boundary properties such as smoothness. Boundary-based approaches can treat such properties very naturally, but suffer from their own difficulties. In this thesis, a new method is proposed which integrates both region-growing and edge-constraints for object segmentation in order to modify the segmentation results.

The segmentation results are dependent on the choice of seeds. The results can be poor if seeds are placed poorly. It may give erroneous result if seed falls on the boundaries of the object. A poor threshold for stopping rule can result in either over-segmentation or under-segmentation of region. To overcome this drawback of seed selection, the proposed algorithm automatically selects the seed pixel and the threshold for stopping rule. The proposed method is capable of segmenting an object from the image even if the region is under-segmentation. For over-segmented region also, this method gives satisfactory results and can distinguish foreground from background.

3.2 Proposed Algorithm

Step 1: Calculate automatic threshold and initial seed.

(a) A 20x20 window ***w*** is chosen across the center pixel to start the automatic phase of the method.

(b) Threshold determination from window ***w***

Find the largest connected component of this window and the mean of the gray pixels in largest connected component. Absolute distance between mean of window and median of all the window pixels gives the threshold to decide whether a neighbor pixel is included into a region or not.

Threshold, T = $|mean_w - median_w|$ (3.1)

(c) Automatic initial seed selection from window

Calculate deviation of pixel values in window from the mean,

dev = $|w_{(x,y)} - mean_w|$ (3.2)

Find the coordinates (x,y) in window where deviation is minimum and embed back this window into the actual image to obtain the initial seed coordinates.

Step 2: Seeded Region growing process

In the seed based region growing method, selection of initial seed is crucial because it decides the overall segmentation by region growing technique. The initial seed obtained above is labeled as the grown region. All eight neighboring pixels are checked for the similarity criteria whether to include in this region or not. The similarity criterion is nothing but the Euclidean distance of seed and the pixel in question being less than the threshold of the image (as obtained by above process). The pixel is labeled as a region that then grows based on a similarity measure.

Euclidean distance, $E_d = \sqrt{(seed - n_p)^2}$ (3.3)

Where, E_d = Euclidean Distance

n_p= neighbor of the seed , p=1,2,3,…, 8

If E_d<=thr, Label the unlabeled pixel and push onto stack

Step 3: Find nearest strong edge for the grown region

For each pixel of grown region, first calculate its distance with canny edge pixels of the image and then mark the minimum distance as the nearest strong edge pixel.

The **algorithm EDGE** for finding nearest strong edge for the grown region is as follow:

```
Push all pixels of grown region onto stack regr and canny edge pixels onto stack edg.
While stack regr is not empty:
    Remove top pixel(x,y) from stack regr.
    While stack edg is not empty:
        Remove top pixel (a,b) from stack edg.
        Calculate absolute distance d between (x,y) and (a,b).
        Label the pixel (a,b) which gives minimum distance d corresponding to (x,y).
All these labeled pixels give nearest strong edge pixels.
```

Algorithm: EDGE

Hence, nearest strong edge segments the object present in the image. There might be present some discontinuities and holes in the object. These defects are overcome by applying certain morphological operations onto the segmented image.

Step 4: Perform morphological operations to refine the segmentation results

(a) Morphologically close image

This operation performs dilation followed by erosion on binary image, using the same structuring element for both operations.

Structuring element is an essential part of the dilation and erosion operations used to probe the input image. It is a matrix consisting of only 0's and 1's that can have any arbitrary shape and size. The pixels with values of 1 define the neighborhood. The shape and size of structuring element can be chosen depending upon the objects one has to process in input image. Here, we chose structuring element of shape of disk and size equal to 5.

(b) Trace region boundaries in binary image

It traces the exterior boundaries of objects in binary image and results in largest connected component. This function returns the row and column coordinates of border pixels of all the objects in an image.

(c) Fill image regions and holes

A hole is a set of background pixels that cannot be reached by filling in the background from the edge of the image. This operation uses an algorithm based on morphological reconstruction [42].

Complete method can be summarized pictorially in the flowchart given below.

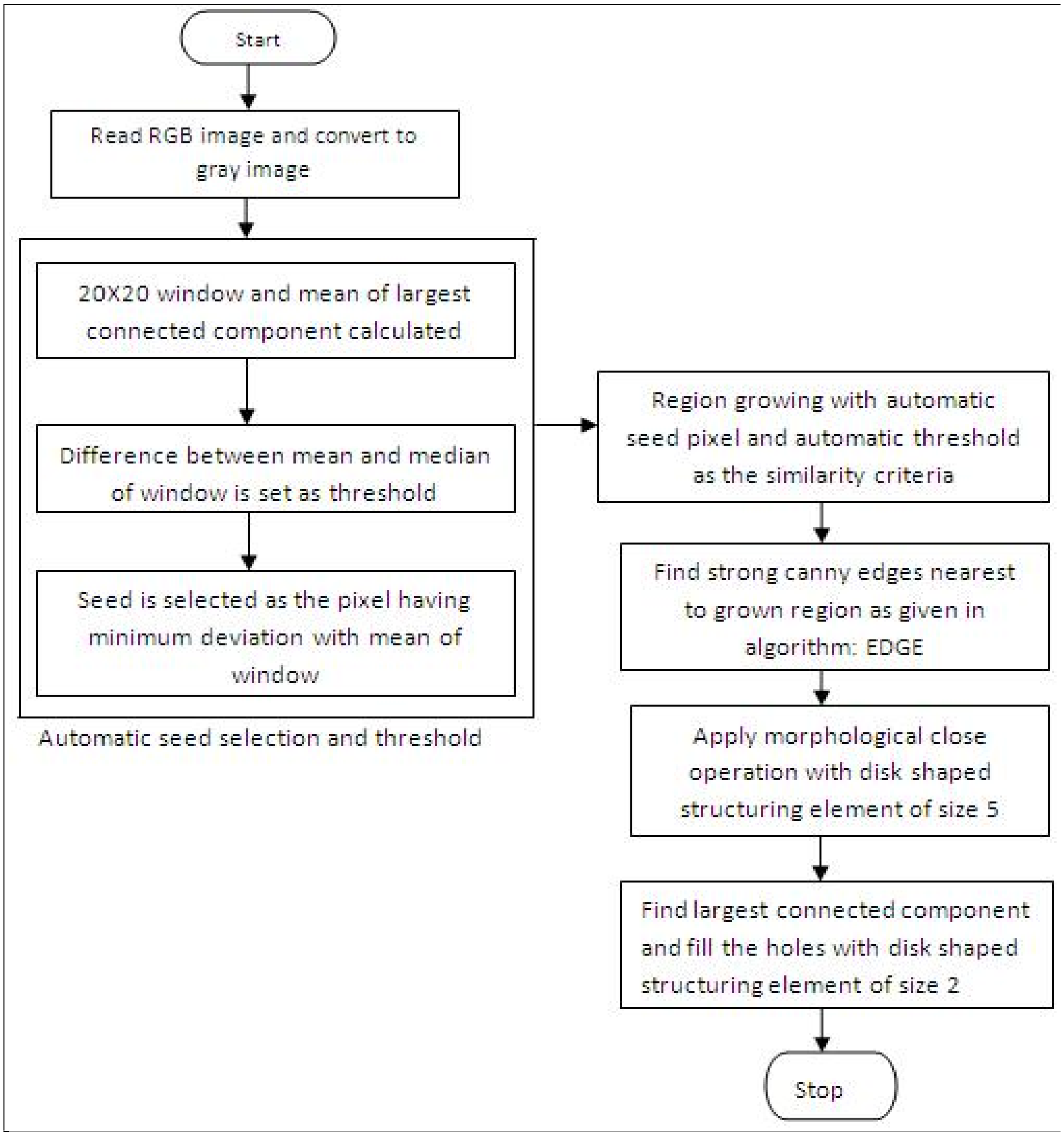

Start
Read RGB image and convert to gray image
20X20 window and mean of largest connected component calculated
Difference between mean and median of window is set as threshold
Seed is selected as the pixel having minimum deviation with mean of window
Automatic seed selection and threshold
Region growing with automatic seed pixel and automatic threshold as the similarity criteria
Find strong canny edges nearest to grown region as given in algorithm: EDGE
Apply morphological close operation with disk shaped structuring element of size 5
Find largest connected component and fill the holes with disk shaped structuring element of size 2
Stop

Fig. 3.1: Flowchart for proposed method

CHAPTER-4

RESULTS AND DISCUSSIONS

4.1 Experimental Setup

To examine the efficiency of our method we used the 97 images from **CAR** category of a challenging PASCAL VOC 2005 database [33]. The images are of varying size. To implement proposed algorithm, system configured with Intel core 2 duo processor 2GHz and 2 Gigabytes of RAM is used. Tool used is matlab version 7.9.0.529 (R2009b) for 32-bit processor. Fig. 7 demonstrates a step-by-step performance of our method for one of the test image as shown in Fig. 4.1. The results obtained after applying our method are shown in Table 4.1. We compared our results with those obtained by applying Xavier Bresson [29] code, which introduced one of the object segmentation techniques, on the same database. Results are evaluated with respect to precision and recall with respect to ground truth annotations provided in the database.

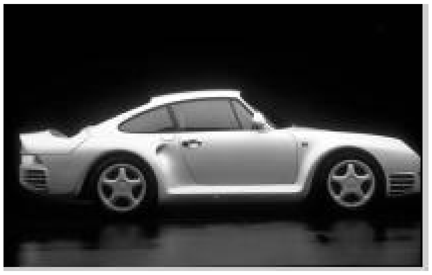


Fig. 4.1: Test image.

Step 1:

1. A 20x20 window is considered across the center of the image.

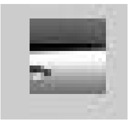


Fig. 4.2(a): window

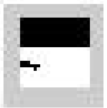


Fig. 4.2(b): largest connected component

2. Mean of the largest connected component across the window is calculated. The difference between median of window and its mean is chosen as threshold.

   Mean=198.1074
   Median=194
   Threshold=4.1074

The position of pixel having minimum deviation with the mean of window is selected as the initial seed to initiate the process of region growing.

Pixel position in window is (16,4)

Step 2:

1. Initial seed as obtained above is embedded back into the original image to calculate the actual position of the seed pixel for region growing.

   Seed is at (92,134)

2. For 8 neighbors of the seed:

   Check for similarity condition. If this pixel satisfies the condition then label the unlabeled pixel and push onto stack.

3. Pop the stack one by one and repeat step 2 by assigning the popped pixel as the new seed.

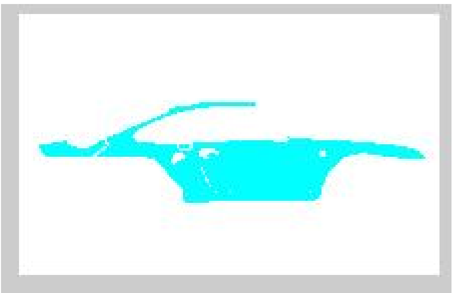


Fig. 4.3: Result of Seeded Region growing

Step 3:

For all labeled pixels of grown region, strong edges connected to the grown region are given by the minimum distance between the canny edge of the original image and the grown region.

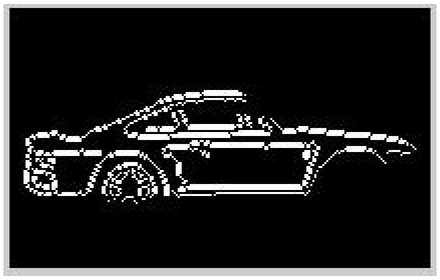


Fig. 4.4: Strong and nearest canny edge to grown region

Step 4:

Certain morphological operations are performed on the strong largest connected edge of the grown region. First of all morphologically close operation is done with disk-shaped structuring element equal to 5.

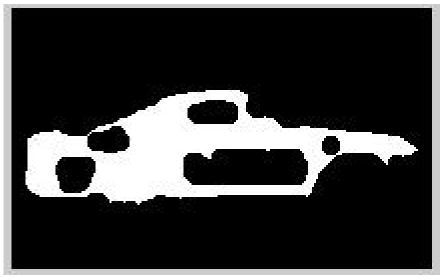


Fig. 4.5: Morphologically closed image

Trace the boundaries of image. It gives the single largest connected component of the closed image followed by filling the holes with disk-shaped structuring element equal to 2 to segment the single object in the image.

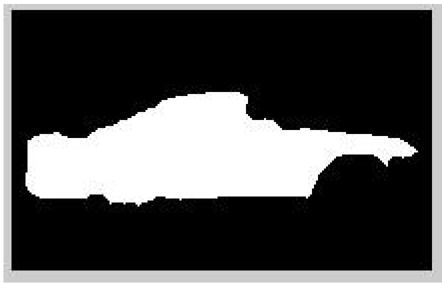


Fig. 4.6: Filled image

Finally, multiply this filled image with original image to obtain only the localized object in grayscale.

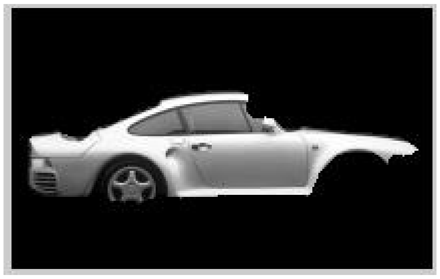
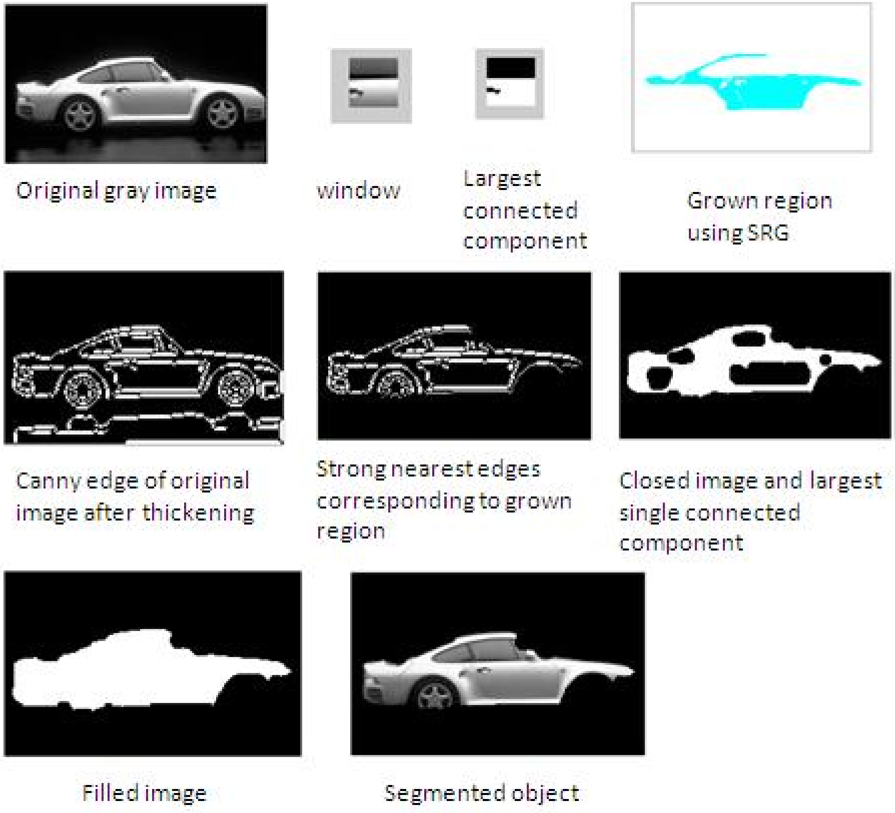


Fig. 4.7: Segmented object



Original gray image

window

Largest connected component

Grown region using SRG

Canny edge of original image after thickening

Strong nearest edges corresponding to grown region

Closed image and largest single connected component

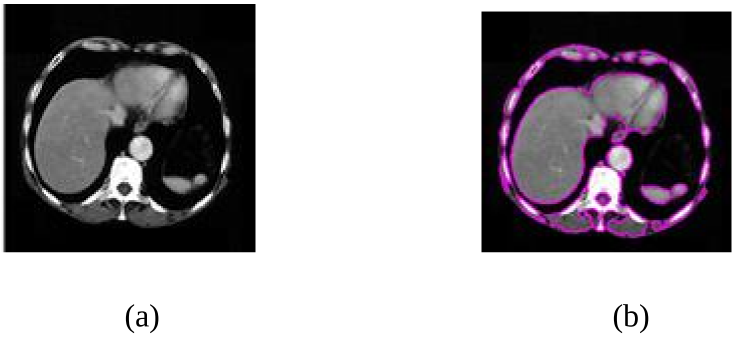
Filled image

Segmented object

Fig. 4.8: Step by step results of proposed algorithm.

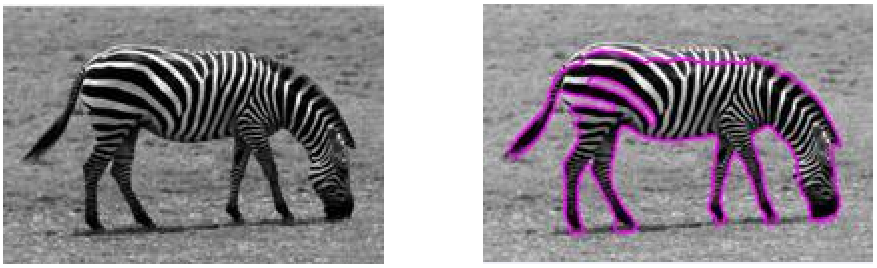
4.2 Comparison method

The active contour/snake model is one of the most successful variational models in image segmentation. It consists of evolving a contour in images toward the boundaries of objects. Its success is based on strong mathematical properties and efficient numerical schemes based on the level set method. The only drawback of this model is the existence of local minima in the active contour energy, which makes the initial guess critical to get satisfactory results. Xavier Bresson et al. [29] proposed a method to solve this problem by determining a global minimum of the active contour model. Their approach is based on the unification of image segmentation and image denoising tasks into a global minimization framework. More precisely, they propose to unify three well-known image variational models, namely the snake model, the Rudin–Osher–Fatemi denoising model and the Mumford–Shah segmentation model. They established theorems with proofs to determine the existence of a global minimum of the active contour model. From a numerical point of view, they proposed a new practical way to solve the active contour propagation problem toward object boundaries through a dual formulation of the minimization problem. The dual formulation, easy to implement, allows a fast global minimization of the snake energy. It avoids the usual drawback in the level set approach that consists of initializing the active contour in a distance function and re-initializing it periodically during the evolution, which is time-consuming. They applied their segmentation algorithms on synthetic and real-world images, such as texture images and medical images, to emphasize the performances of their model compared with other segmentation models.



(a) (b)

Fig. 4.9: Tumor example. (a) Original image (b) Segmented image



(a) (b)

Fig. 4.10: Zebra example. (a)Original image (b)Segmented image

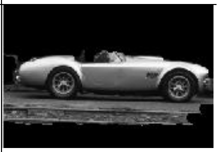
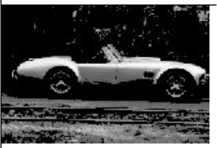
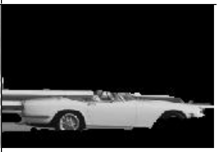
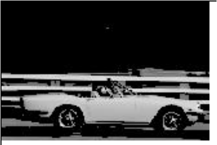
4.3 Discussion on results

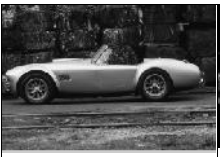
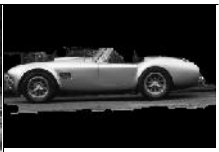
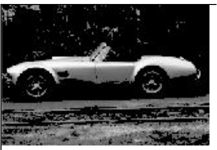
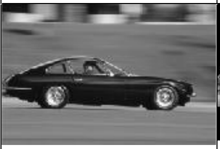
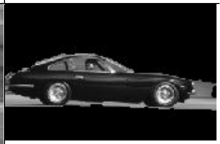
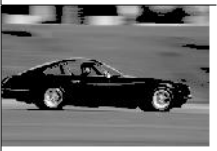
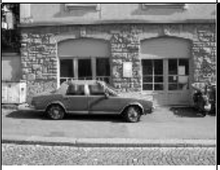
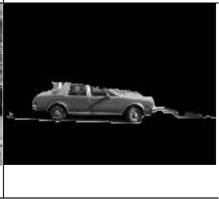
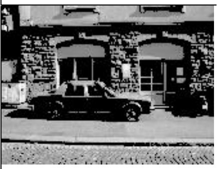
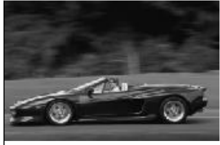
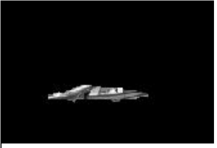
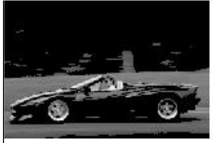
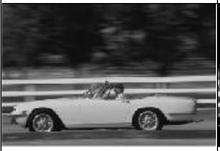
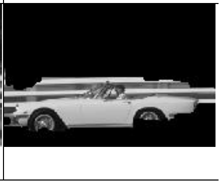
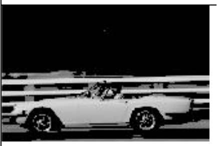
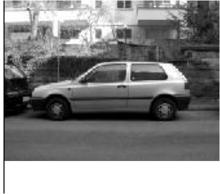
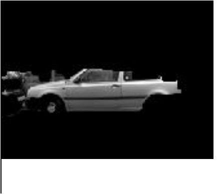
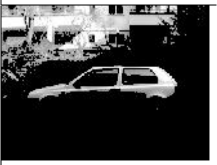
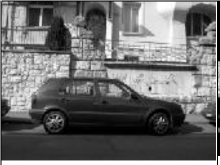
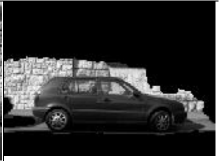
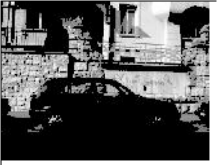
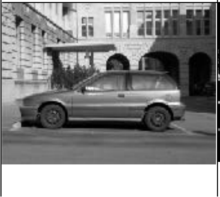
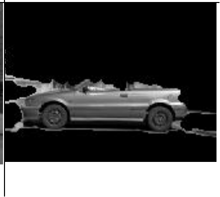
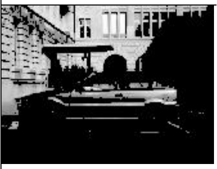
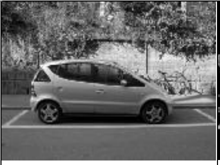
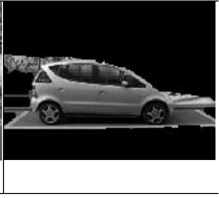
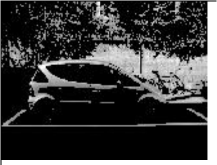
Table 4.1 below shows the results after applying our proposed method and Xavier Bresson method on database images. It also includes precision and recall values for both the methods for every image. Precision and recall can be calculated as:

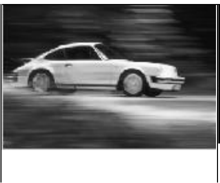
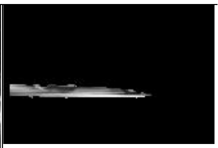
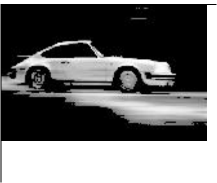
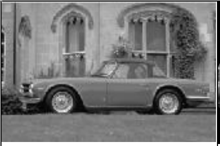
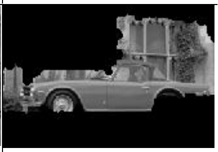
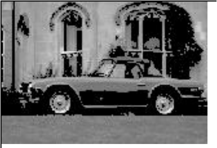
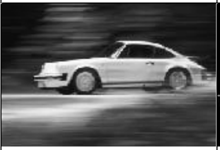
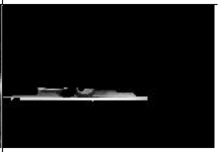
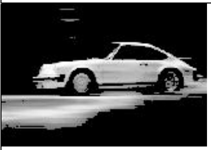
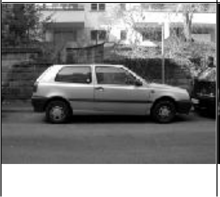
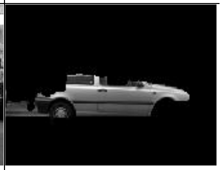
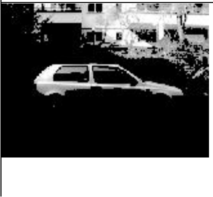
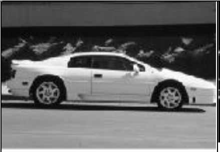
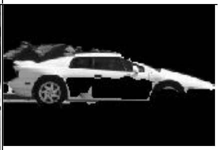
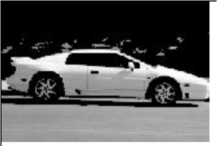
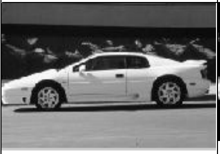
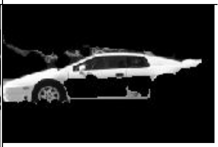
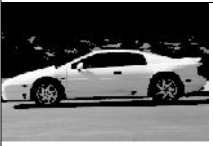
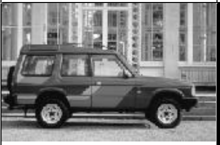
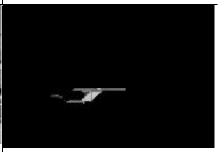
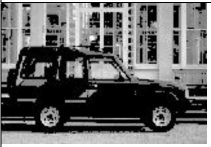
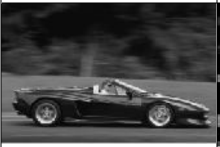
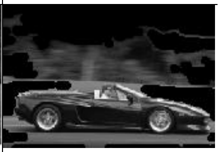
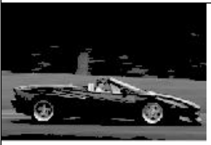
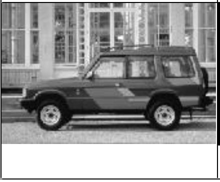
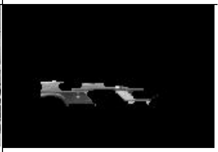
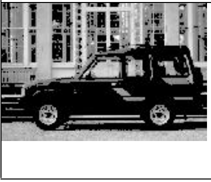
$$\text{Precision} = \frac{true\ positive}{(true\ positive + false\ positive)} \quad (4.1)$$

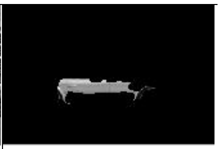
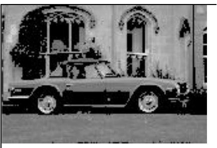
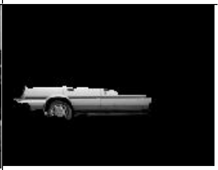
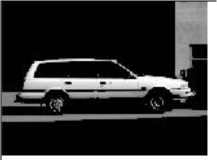
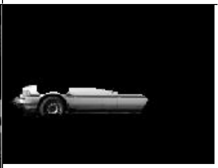
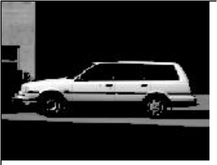
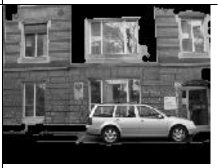
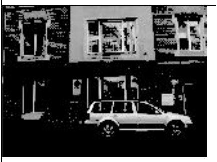
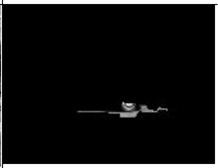
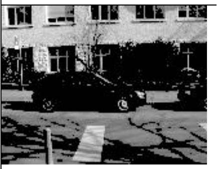
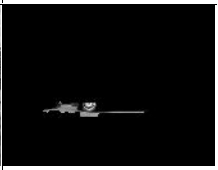
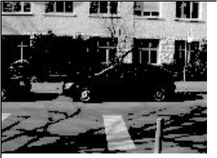
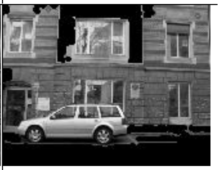
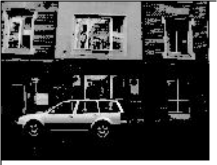
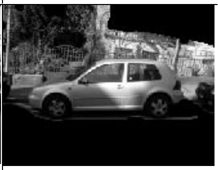
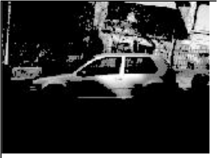
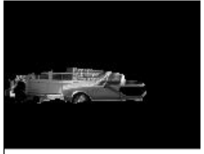
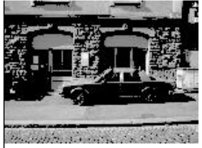
$$\text{Recall} = \frac{true\ positive}{(true\ positive + false\ negative)} \quad (4.2)$$

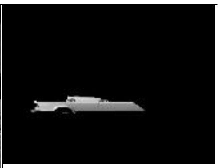
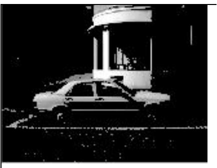
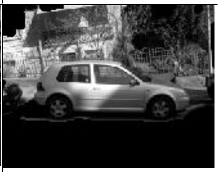
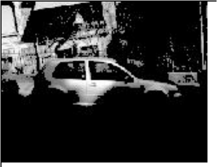
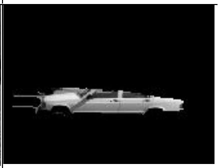
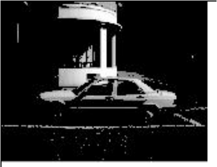
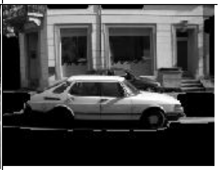
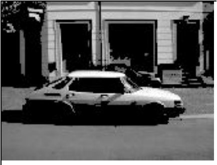
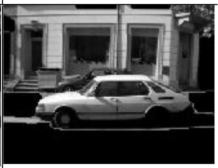
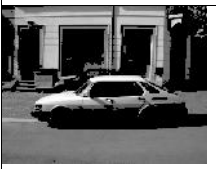
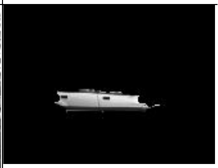
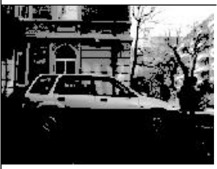
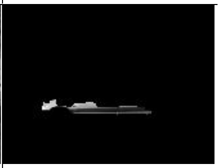
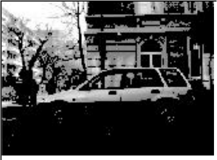
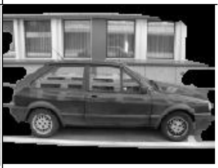
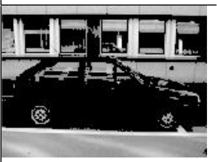
Table 4.1 : Results of proposed algorithm and Xavier Bresson Code for complete PASCAL VOC 2005 database images.

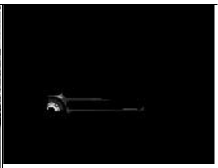
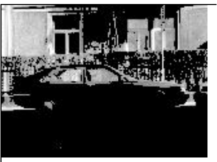
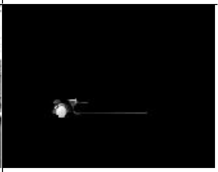
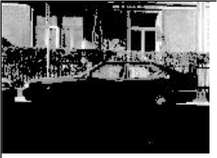
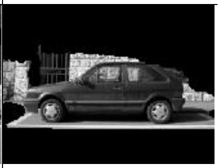
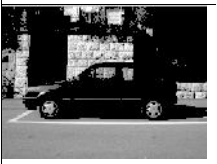
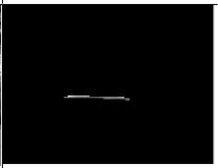
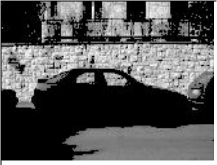
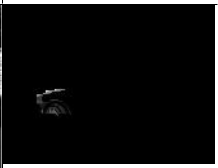
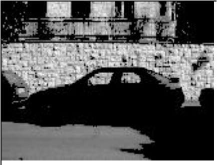
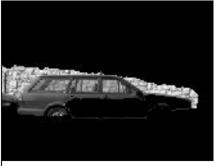
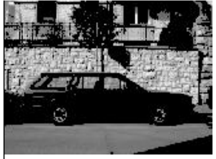
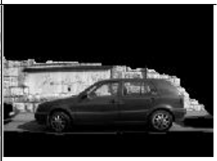
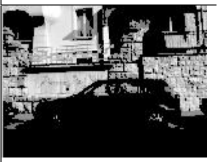
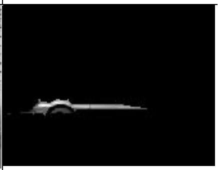
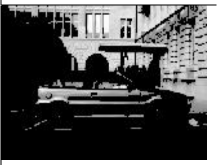
No	Name	Original	Proposed method	Xavier Bresson Code	precision	recall	Precision (Xavier)	Recall (Xavier)
1	29000-sml.png				0.564645	0.984703	0.327684	0.475219
2	29036-sml.png				0.723981	0.808274	0.427458	0.648109

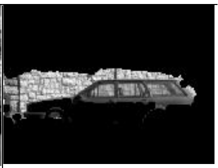
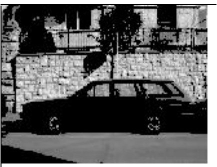
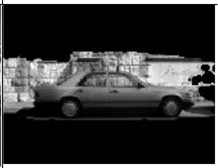
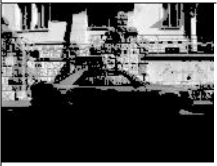
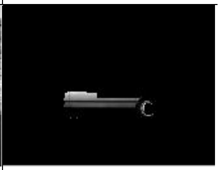
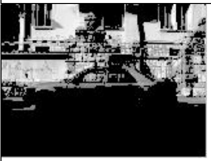
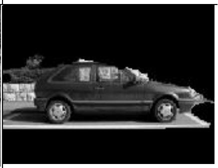
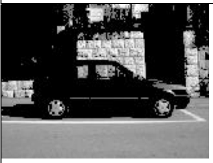
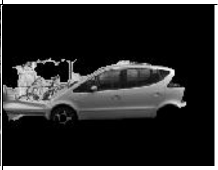
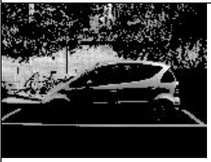
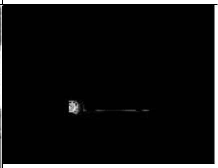
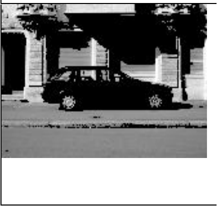
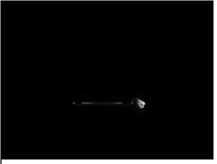
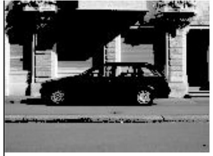
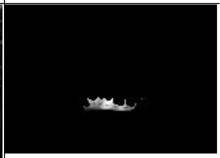
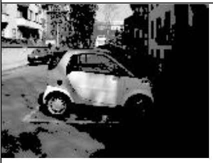
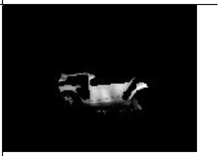
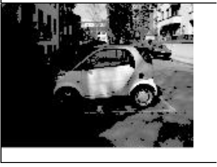
3	29000-sml-lt.png				0.675766	0.992148	0.327684	0.475219
4	29060-sml2.png				0.790801	0.998924	0.076892	0.262105
5	car-pic60-sml.png				0.840207	0.932139	0.044952	0.241556
6	29014-sml-lt.png				0.883333	0.121999	0.261223	0.348241
7	29036-sml-lt.png				0.583879	0.853664	0.427458	0.648109
8	car-pic2-sml.png				0.842176	0.572372	0.228307	0.268055
9	car-pic97-sml2.png				0.553111	1	0.041473	0.083529
10	car-pic99-sml-lt.png				0.804393	0.834559	0.148686	0.225911
11	car-pic111-sml2.png				0.695155	0.978353	0.198323	0.225407

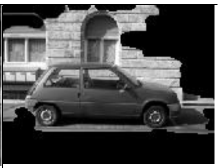
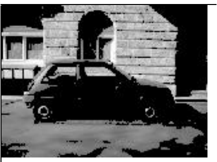
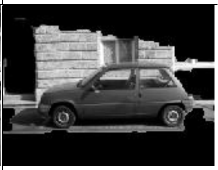
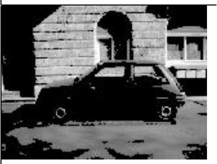
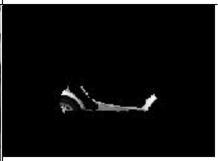
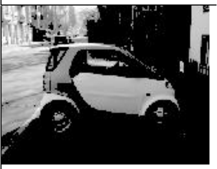
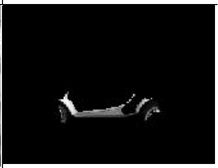
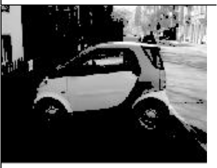
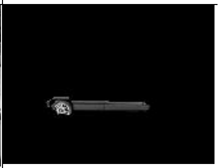
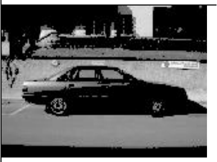
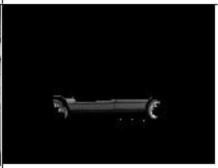
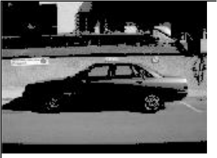
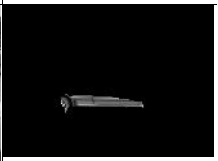
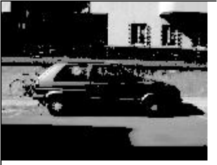
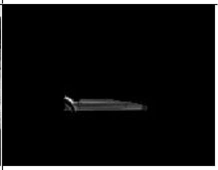
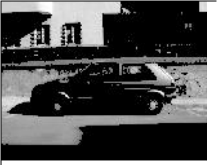
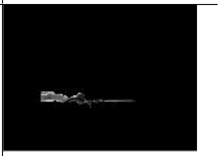
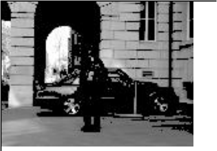
12	29045- sml.png				0.309031	0.070751	0.470738	0.59503
13	354072- sml-lt.png				0.448096	0.755687	0.160345	0.472793
14	29045- sml-lt.png				0.250266	0.044575	0.469652	0.598824
15	car-pic2- sml2.png				0.898345	0.578974	0.228307	0.268055
16	29092- sml.png				0.77947	0.60947	0.408537	0.641632
17	29092- sml-lt.png				0.837926	0.507548	0.408537	0.641632
18	354002- sml2.png				1	0.033401	0.160325	0.262331
19	29014- sml.png				0.397101	1	0.261223	0.348241
20	354002- sml.png				0.989467	0.108198	0.160325	0.262331

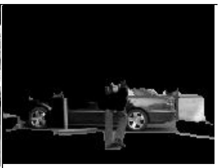
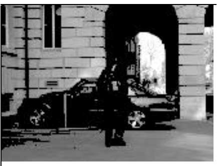
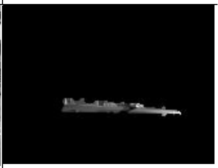
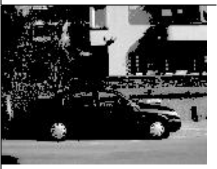
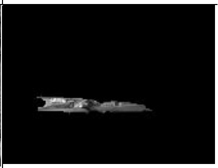
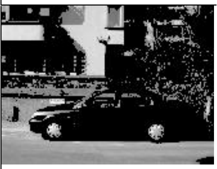
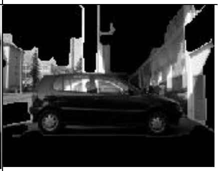
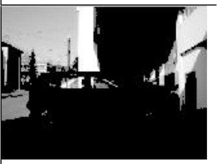
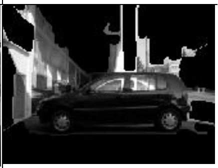
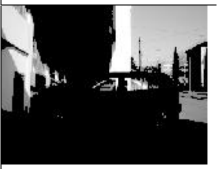
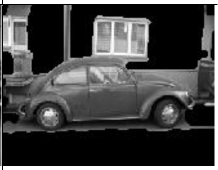
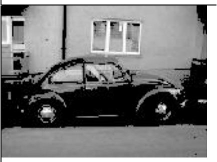
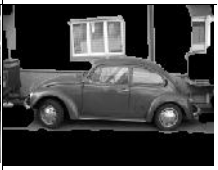
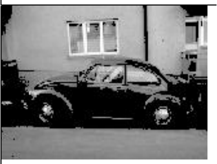
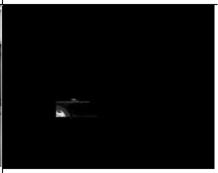
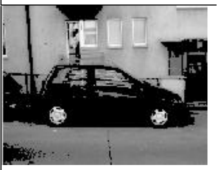
21	354072-sml.png				1	0.155727	0.160399	0.476601
22	car-pic1-sml.png				0.988628	0.397123	0.481307	0.427933
23	car-pic1-sml-lt.png				0.947733	0.345417	0.481307	0.427933
24	car-pic24-sml.png				0.405955	0.061516	0.147856	0.419153
25	car-pic22-sml.png				0.436137	0.04622	0.014863	0.06867
26	car-pic22-sml-lt.png				0.405955	0.061516	0.014863	0.06867
27	car-pic24-sml-lt.png				0.138274	0.996303	0.147856	0.419153
28	car-pic40-sml.png				0.356313	0.988823	0.310028	0.343369
29	car-pic60-sml-lt.png				0.468096	0.485056	0.044952	0.241556

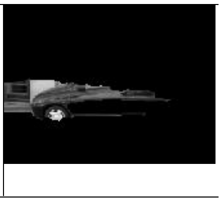
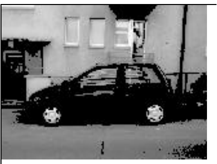
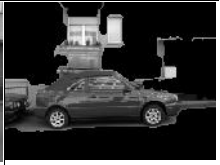
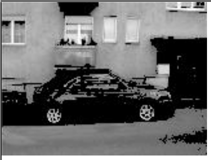
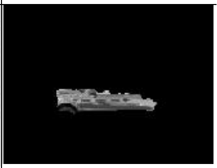
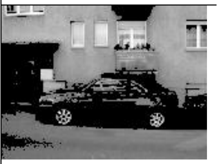
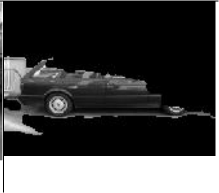
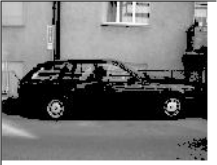
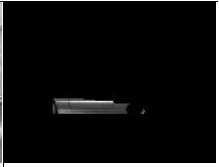
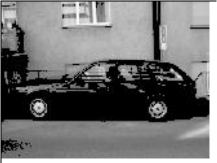
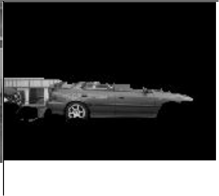
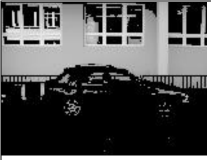
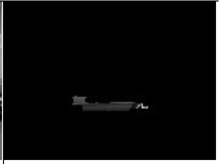
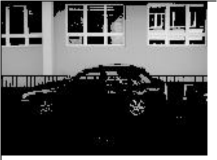
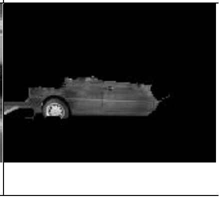
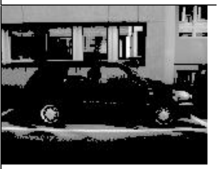
30	car-pic75-sml2.png				0.988825	0.223292	0.359043	0.314142
31	car-pic40-sml2.png				0.317294	0.989326	0.309868	0.338637
32	car-pic75-sml.png				0.886693	0.47397	0.359043	0.314142
33	car-pic78-sml.png				0.235065	0.79343	0.118934	0.370622
34	car-pic78-sml-lt.png				0.234746	0.795502	0.118934	0.370622
35	car-pic87-sml2.png				1	0.21975	0.190087	0.312549
36	car-pic87-sml.png				0.997579	0.106598	0.190087	0.312549
37	car-pic147-sml2.png				0.477287	0.9972	0.100855	0.152215

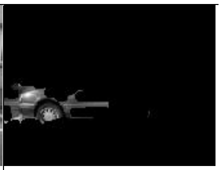
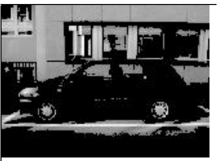
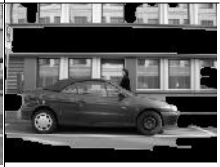
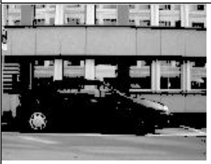
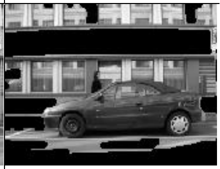
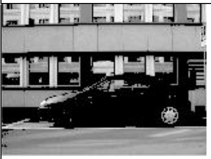
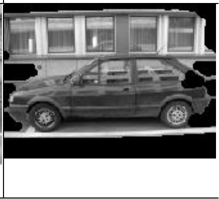
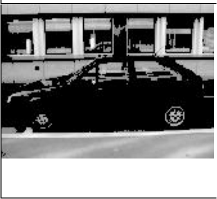
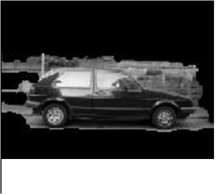
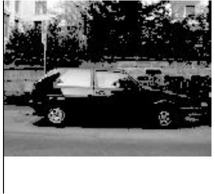
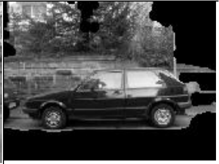
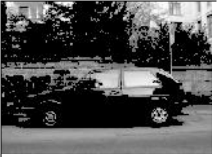
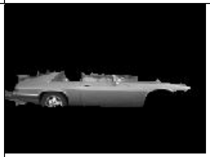
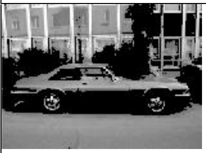
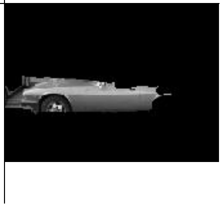
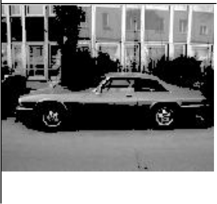
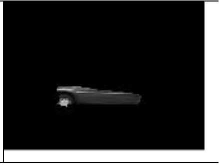
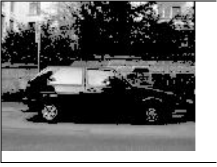
38	car-pic90-sml.png				1	0.064076	0.132937	0.194539
39	car-pic90-sml-lt.png				1	0.053617	0.135045	0.200374
40	car-pic109-sml-lt.png				0.564636	0.997677	0.06576	0.146016
41	car-pic95-sml2.png				1	0.022095	0.040632	0.103422
42	car-pic95-sml.png				0.983607	0.067695	0.040632	0.103422
43	car-pic96-sml2.png				0.639817	0.501794	0.064397	0.16401
44	car-pic97-sml.png				0.510837	1	0.041473	0.083529
45	car-pic99-sml.png				0.959003	0.107034	0.148686	0.225911

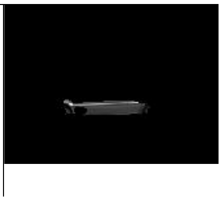
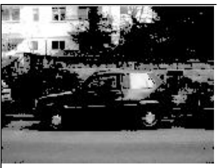
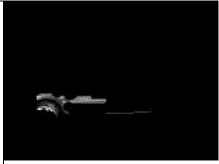
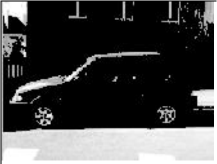
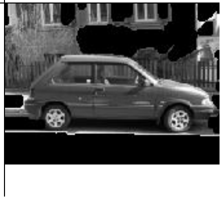
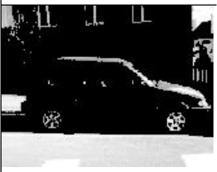
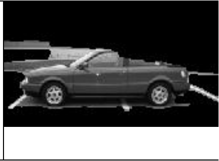
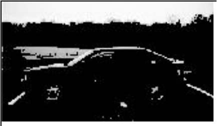
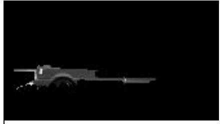
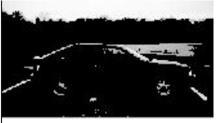
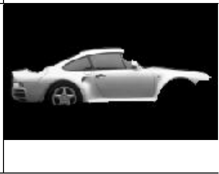
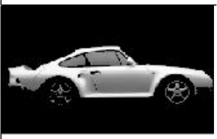
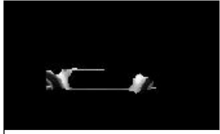
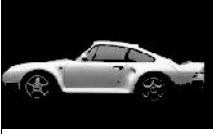
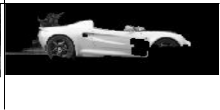
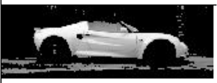
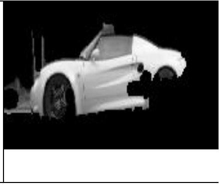
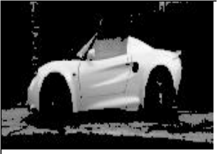
46	car-pic96-sml.png				0.490598	0.434137	0.064397	0.16401
47	car-pic101-sml.png				0.457273	0.960548	0.094097	0.200548
48	car-pic101-sml-lt.png				0.961675	0.17874	0.094097	0.200548
49	car-pic109-sml.png				0.637467	0.986873	0.06576	0.146016
50	car-pic111-sml.png				0.699348	0.759301	0.198323	0.225407
51	car-pic115-sml2.png				1	0.065193	0.034853	0.140137
52	car-pic115-sml.png				1	0.095189	0.034853	0.140137
53	car-pic116-sml2.png				1	0.060811	0.263538	0.606034
54	car-pic116-sml.png				0.97943	0.270922	0.2651	0.601174

55	car-pic117-sml2.png				0.401928	0.999813	0.089133	0.202202
56	car-pic117-sml.png				0.477498	0.988056	0.089133	0.202202
57	car-pic118-sml.png				1	0.082152	0.381981	0.514467
58	car-pic118-sml-lt.png				0.996568	0.107174	0.381981	0.514467
59	car-pic119-sml2.png				1	0.159189	0.091024	0.241147
60	car-pic120-sml.png				0.998882	0.281706	0.089609	0.235789
61	car-pic120-sml2.png				1	0.130193	0.097601	0.193718
62	car-pic120-sml.png				1	0.130496	0.097601	0.193718
63	car-pic121-sml2.png				0.871957	0.129729	0.04232	0.20432

64	car-pic121-sml.png				0.504334	0.671317	0.04232	0.20432
65	car-pic122-sml.png				1	0.16097	0.063586	0.13488
66	car-pic122-sml-lt.png				0.811159	0.145567	0.063586	0.13488
67	car-pic128-sml.png				0.448287	1	0.04069	0.060083
68	car-pic128-sml-lt.png				0.490612	1	0.04069	0.060083
69	car-pic130-sml2.png				0.503487	0.998608	0.13865	0.375522
70	car-pic130-sml.png				0.523334	0.997912	0.13865	0.375522
71	car-pic131-sml.png				1	0.044583	0.074646	0.234646

72	car-pic131-sml-lt.png				0.680971	0.331871	0.074646	0.234646
73	car-pic134-sml2.png				0.494764	0.983988	0.061369	0.234283
74	car-pic134-sml.png				0.990971	0.301866	0.061369	0.234283
75	car-pic135-sml2.png				0.716757	0.648293	0.083478	0.233735
76	car-pic135-sml.png				1	0.118984	0.083478	0.233735
77	car-pic139-sml2.png				0.728703	0.620976	0.053686	0.111582
78	car-pic139-sml.png				1	0.104084	0.053686	0.111582
79	car-pic145-sml2.png				0.935821	0.460313	0.075204	0.113364

80	car-pic145-sml.png				0.810579	0.180837	0.038828	0.113364
81	car-pic146-sml2.png				0.301226	0.998149	0.038828	0.125165
82	car-pic146-sml.png				0.307002	0.998237	0.100855	0.125165
83	car-pic147-sml.png				0.445362	0.970897	0.134166	0.152215
84	car-pic154-sml2.png				0.591574	0.998503	0.134166	0.314585
85	car-pic154-sml.png				0.342964	0.998931	0.134166	0.314585
86	car-pic157-sml2.png				0.942026	0.642346	0.143761	0.434487
87	car-pic157-sml.png				0.858977	0.502287	0.143761	0.434487
88	car-pic165-sml2.png				1	0.206616	0.095158	0.255866

89	car-pic165-sml.png				1	0.147009	0.095158	0.255866
90	car-pic172-sml2.png				1	0.077588	0.116322	0.115518
91	car-pic172-sml3.png				0.481725	0.998704	0.116322	0.115518
92	redcar005-01-sml-lt.png				0.745631	0.919491	0.09446	0.077376
93	redcar005-01-sml-rt.png				0.925098	0.157125	0.09446	0.077376
94	sportscar2-sml.png				0.971169	0.757331	0.999277	0.675464
95	sportscar2-sml-lt.png				1	0.09934	0.999276	0.674487
96	yellowcar-sml.png				0.871089	0.656877	0.711111	0.604601
97	yellowcar-sml-lt.png				0.902585	0.693882	0.712749	0.603133

Visually, our segmentation results are good for simple images having clear background which can be seen from Fig. 4.11.

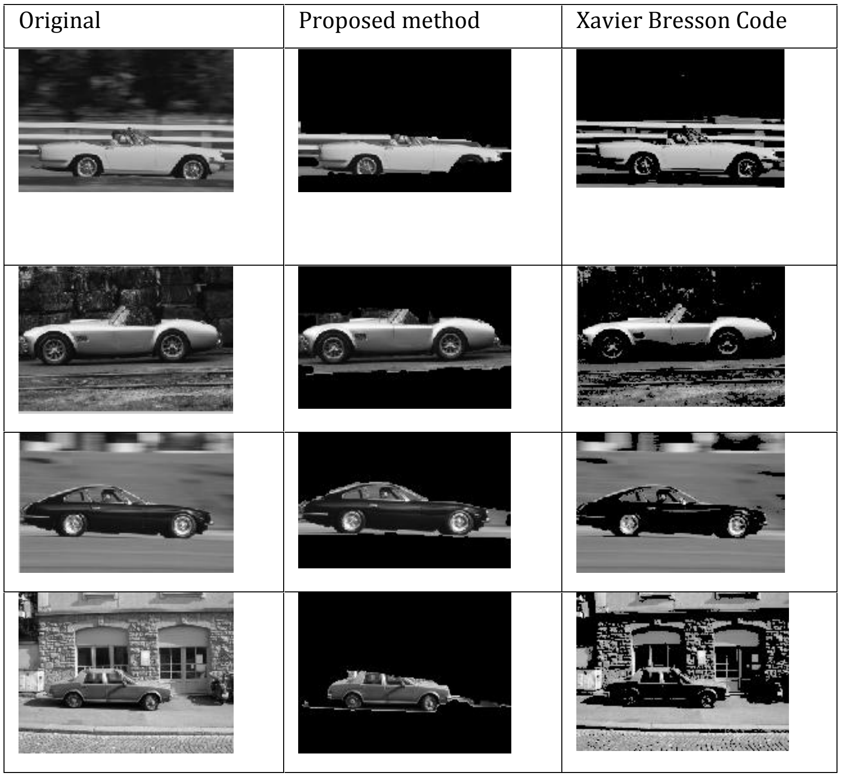


Fig. 4.11: Results for simple images.

Xavier Bresson method produced erroneous results for cluttered images but our method produced optimum output as seen in Fig. 4.12.

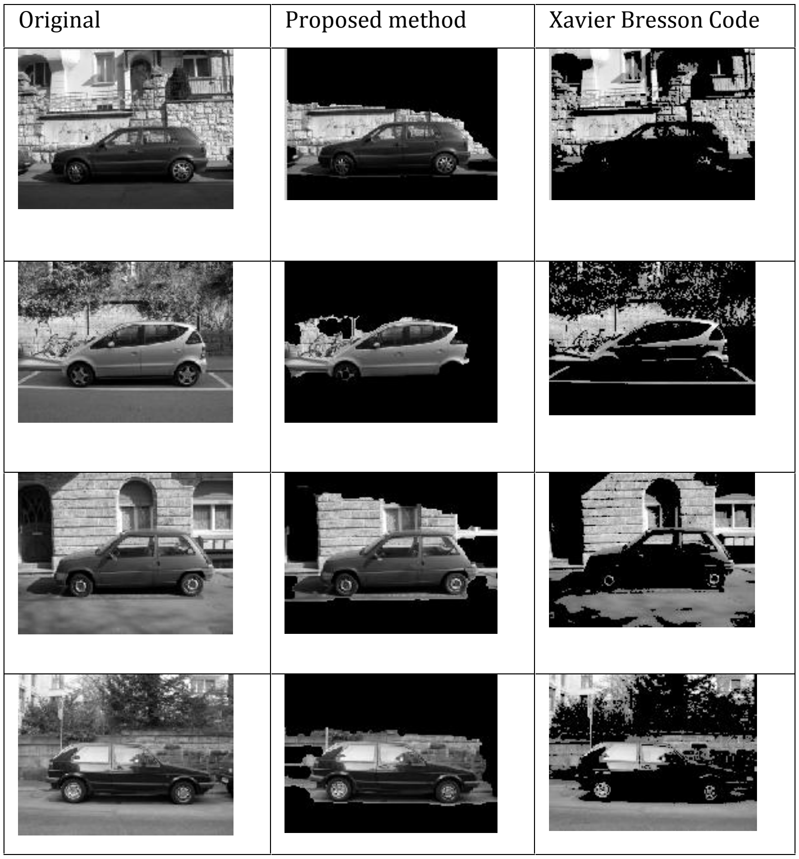


Fig. 4.12: Results for complex images.

For few complex images, some parts of car are missing but have higher precision than Xavier Bresson method. This can be verified from Fig. 4.13.

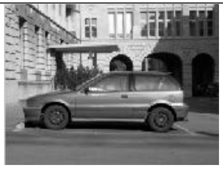
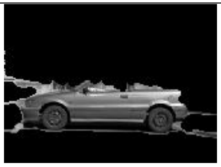
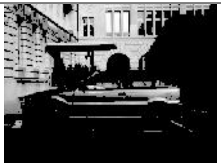
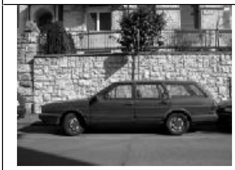
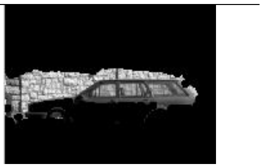
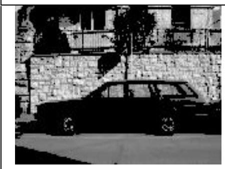
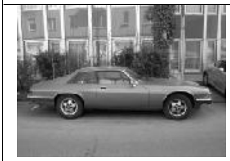
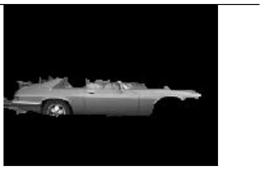
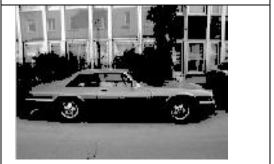
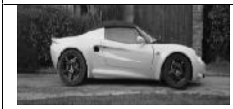
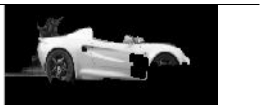
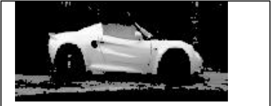
Original	Proposed method	Xavier Bresson Code
		
		
		
		

Fig. 4.13: Results where some portion of object is missing.

Fig. 4.14 shows that even though our method fails for some images and very less portion of car is segmented then also the precision of our method is high since only the car pixels are segmented out. Our proposed method is therefore efficient enough to segment the object and gives the optimum output even if the initial region growing results are not satisfactory.

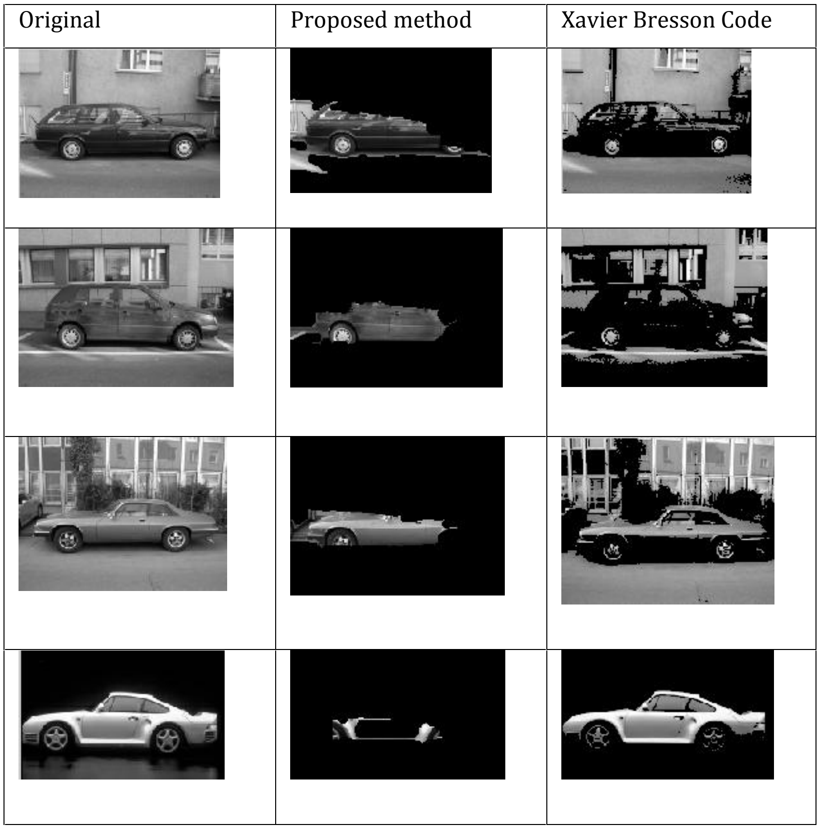
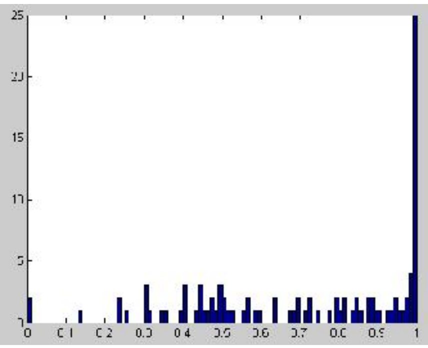
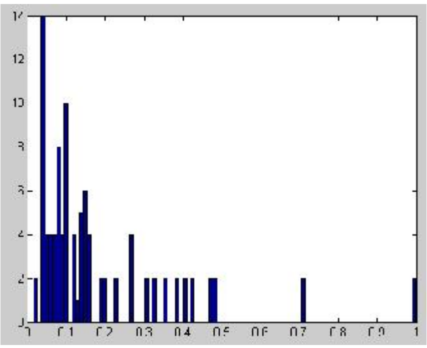
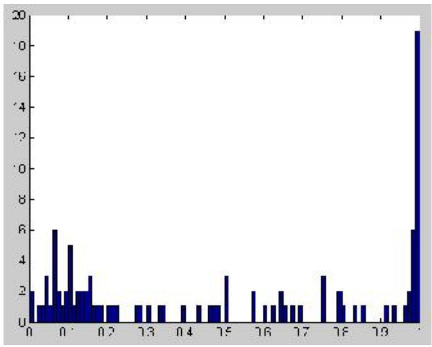
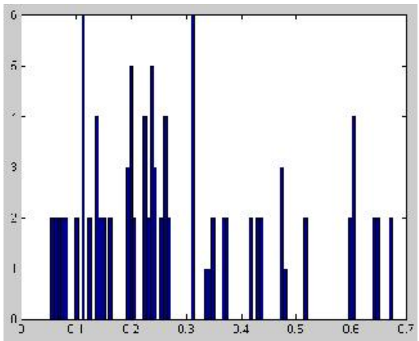


Fig. 4.14: Results where our method gave poor output.

Table 4.2: Mean of Precision and Recall for results of 97 images

	Proposed method	Xavier Bresson method
Precision	0.7190	0.1808
Recall	0.5115	0.2844

Table 4.3: Recall and Precision graph for results of 97 images

	Proposed method	Xavier Bresson code
Precision (p) $\frac{TP}{(TP+FP)}$		
Recall (r) $\frac{TP}{(TP+FN)}$		

Precision and recall parameters indicate the positive predictive value and true positive rate or sensitivity, respectively and the mean values are shown in Table 4.2. The histogram of precision and recall for 97 images is shown in Table 4.3. A right skewed precision histogram for our method and left skewed precision histogram for Xavier Bresson method clearly indicate that our method produced better segmentation results and has high accuracy.

CHAPTER-5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

This thesis provides a brief introduction to various techniques of image segmentation. In this thesis we have proposed a method for object segmentation by integrating region growing and edge constraints. This method automatically selects the initial seed and determines a threshold for growing rule with the help of a 20x20 window across the center pixel in image. Then a primitive single seeded region growing is done with the initial seed and threshold. The grown region along with edge information and certain morphological operations results in distinguishing an object from the image background. However poor the region growing result is, our method is capable of giving optimum result even for images having cluttered background. The proposed algorithm is applied to 97 images of CAR category of the PASCAL VOC 2005 database with successful results and evaluation of results is done by comparing the results with a method proposed by Xavier Bresson. Recall and precision parameters are obtained with the help of annotations that are given in database. Mean precision and recall of our method are 0.7190 and 0.5115 respectively. Mean precision and recall of Xavier Bresson method are 0.1808 and 0.2844 respectively. These parameters are used to compare the results of two methods and prove that our proposed method produced better results than Xavier Bresson method. The proposed method contributes to object segmentation in two aspects, firstly, automatic seed selection and threshold determination. Secondly, less dependence on region growing results.

5.2 FUTURE WORK

- Instead of conventional region growing method, an improved region growing technique can be used.
- The proposed method can be extended for color images also.
- This method can also use different types of edge detectors. Improved edge detector can result in more precise segmentation and less background.

References

[1] R. Adams and L. Bischof, “Seeded region growing,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 641–647, 1994.

[2] P. K. Sahoo, **S.** Soltani, and A. K. C. Wong, “A survey of thresholding techniques,” *Comput.* Vis., *Graph. Image Process***.,** vol. 41, pp. 233-260, 1988.

[3] L. S. Davis, “A survey of edge detection techniques,” *Compur. Graph. Image Process.,* vol. 4, pp. 248-270, 1975.

[4] S. L. Horowitz and T. Pavlidis, “Picture segmentation by a directed split-and-merge procedure,” *Proc. 2nd Int. Joint Conf Pattern Recognit.,* 1974, pp, 424433.

[5] F. Meyer and S. Beucher, “Morphological segmentation,” *J. Vis. Commun. Image Represent.,* vol. 1, pp. 2146, 1990.

[6] P. J. Besl and R. C. Jain, “Segmentation through variable-order surface fitting,” *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-IO, pp.167-192,1988.

[7] ‘**Digital Image Processing**’, second edition by Rafael C. Gonzales, University of Tennessee.

[8] J. Kittler, “A locally sensitive method for cluster analysis,” *Pattern Recognition,* vol. 8, pp. 23-33, 1976.

[9] T. Pavlidis and Y.-T. Liow, “Integrating region growing and edge detection,” *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 12, pp. 225–233, 1990.

[10] M. P. Pathegama and Ö. Göl, “Edge-end pixel extraction for edge based image segmentation,” in *Proc. World Acad. Sci. Eng. Technol.*, Jan. 2005, vol. 2, pp. 164–167.

[11] P. Salembier and F. Marques, “Region-based representations of image and video-segmentation tools for multimedia services,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 8, pp. 1147–1169, Dec. 1999.

[12] Ehquierdo and M.Ghanbari, “Nonlinear Gaussian filtering approach for object Segmentation,” *IEE Pmc.-yls. Image Signal Process, Vol. 146, No.* 3, *June 1999.*

[13] I. Pratikakis, I. Vanhamel, H. Sahli, B. Gatos, and S. J. Perantonis, “Unsupervised Watershed-Driven Region-Based Image Retrieval,” *IEE Proc. Vision, Images, and Signal Processing*, Vol. 153 No. 3, pp. 313-322, 2006.

[14] C. Carson, S. Belongie, H. Greenspan, and J. Malik, “Blobworld: Image Segmentation Using E-M and Its Application to Image Querying,” *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 24, pp. 1026-1038, 2002.

[15] J. Z. Wang, J. Li, and G. Wiederhold, “SIMPLIcity: Semantic–Sensitive Integrated Matching for Picture Libraries,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 23 No. 9, pp. 947-963, 2001.

[16] J.-W. Hsieh and W. Eric L. Grimson, “Spatial Template Extraction for Image Retrieval by Region Matching,” *IEEE Trans. Image Processing*, Vol. 12 No. 11, pp. 1404-1415, 2003.

[17] Chi-Han Chuang, Chin-Chun Chang, and Shyi-Chyi Cheng, “Content Aware Image Segmentation for Region-Based Object Retrieval,” *International Symposium on Intelligent Signal Processing and Communication Systems* (ISPACS2008) Swissôtel Le Concorde, Bangkok, Thailand, 2008.

[18] D.H.Ballard, “Generalizing the Hough transform to detect arbitrary shapes,” *Pattern Recogn.*, vol.13, pp.111-122, 1981.

[19] Kostas Haris, Serafim N. Efstratiadis, Nicos Maglaveras and Aggelos K. Katsaggelos, “Hybrid Image Segmentation Using Watersheds and Fast Region Merging,” *IEEE TRANSACTIONS ON IMAGE PROCESSING*, vol. 7, no. 12, December 1998.

[20] D.K. Yau, A.K. Elmagarmid, and W.G. Aref, “Automatic image segmentation by integrating color-edge extraction and seeded region growing,” *IEEE Trans. Image Processing*, Vol. 10 No. 10, pp. 1454-1466, 2001.

[21] J.K. Aggarwal, “The integration of image segmentation maps using region and edge information,” *IEEE Trans. Image Processing*, Vol. 15 No. 12, pp. 1241-1252, 1993.

[22] M. Bajcsy,M. Mink and E. Liebman, “A common Framework for edge detection and region growing,” *University of Pennsylvania GRASP Lab. Technical Report 61, 1986.*

[23] H.L Anderson, R. Bajcsy and M. Mintz, “A modular feedback system for image segmentation”, *University of Pennsylvania GRASP Lab., Tech, Report 1O, 1997.*

[24] R. M. Haralick and L. G. Shapiro, “Image segmentation techniques,” *Comput. Vis. Graph. Image Process.,* vol. 29, pp. 100-132, 1985

[25] P. Felzenszwalb and D. Huttenlocher, **“**Efficiently computing a good segmentation**”,** *Technical report,* Cornell University, 1999.

[26] J F Canny, “Finding edge and lines in image,” *M.I.T Artihcial Intell. Lab., Cambridge, MA.* Rep AI TR 720. 1983.

[27] Michael Kass, Andrew Witkin and Demetri Terzopoulos, “Snakes: Active Contour Models,” *International Journal of Computer Vision*, 321-331, 1988.

[28] Xavier Bresson, Pierre Vandergheynst and Jean-Philippe Thiran, “A Variational Model for Object Segmentation Using Boundary Information and Shape Prior Driven by the Mumford-Shah Functional,” *International Journal of Computer Vision 68(2),* 145–162, 2006.

[29] Xavier Bresson, Selim Esedoglu, Pierre Vandergheynst, Jean-Philippe Thiran and Stanley Osher, “Fast Global Minimization of the Active Contour/Snake Model,” *J Math Imaging Vis (2007)* 28: 151–167 DOI 10.1007/s10851-007-0002-0.

[30] M. Leventon, W. Grimson, and O. Faugeras, “Statistical shape influence in geodesic active contours,” *IEEE International Conference of Computer Vision and Pattern Recognition*, pp. 316–323., 2000.

[31] Raman Maini and Dr. Himanshu Aggarwal, “Study and Comparison of Various Image Edge Detection Techniques,” *International Journal of Image Processing (IJIP)*, Volume (3): Issue (1).

[32] Canny, J., “A Computational Approach to Edge Detection,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[33] A fully annotated PASCAL VOC database provides appropriate images for object segmentation. http://pascallin.ecs.Soton.ac.uk/challenges/VOC/databases.html#UIUC. [online]

[34] A. Mehnert, P. Jackway, “An improved seeded region growing algorithm,” Pattern Recognition Letters 18 (1997) 1065–1071, 1997.

[35] Arnau Oliver, Xavier Munoz, Joan Batlle, Lluis Pacheco, and Jordi Freixenet, “Improving Clustering Algorithms for Image Segmentation using Contour and Region Information,” *IEEE Trans. Image Processing,* 1-4244-0361-8/06/$20.00, 2006.

[36] J. MacQueen, “Some methods of classification and analysis of multivariate observations,” *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297, 1967.

[37] A. Jain, M. Murty, and P. Flynn, “Data clustering: A review,” ACM: Computing Surveys, vol. 31, no. 3, pp. 264–323, 1999.

[38] A. Dempster, N. Laird, and D. Rubin, “Maximum-likelihood from incomplete data via em algorithm,” *Journal of the Royal Statistical Society: Series B*, pp.1–38, 1977.

[39] Jean-Pierre Gambotto, “A new approach to combining region growing and edge detection,” *Pattern Recognition Letters,* Volume 14, Issue 11, November 1993, Pages 869−875.

[40] Vikram Chalana, Wendy Costa and Yongmin Kim, “Integrating region growing and edge detection using regularization,” *Proc. SPIE Vol. 2434, p. 262-271, Medical Imaging 1995: Image Processing, Murray H. Loew; Ed.*

[41] Zhang Guoying, Zhu Hong and Xu Ning, “Flotation bubble image segmentation based on seed region boundary growing,” *Mining Science and Technology (China) 21 (2011) 239e242.*

[42] Soille, P., “Morphological Image Analysis: Principles and Applications,” *Springer-Verlag*, 1999, pp. 173-174.

[43] Jianhua Xuan, Adali, T. and Yue Wang, “Segmentation of magnetic resonance brain image: integrating region growing and edge detection,” *Image Processing, Proceedings, International Conference,* 1995, vol.3,pp.544-547.

[44] Shapiro, Linda G. & Stockman, George C., “Computer Vision”, *Prentice Hall*, ISBN 0-13-030796-3, 2002.

[45] L. Silva, O. R. P. Bellon, and P. F. U. Gotardo, “Edge-Based Image Segmentation using Curvature Sign Maps from Reflectance and Range Images,” *Proceeding of ICIP*, pp.730-733, 2001.

[46] X. Jiang, “An Adaptive Contour Closure Algorithm and Its Experimental Evaluation,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 1252-1265, Nov. 2000.

[47] P. Besl and R. Jain, “Segmentation through variable-order surface fitting,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, pp. 167–192, Mar. 1988.

[48] L. D. Griffin, A. C. F. Colchester, and G. P. Robinson, "Scale and segmentation of grey-level images using maximum gradient paths," *Image Vis. Comput., vol. 10, pp. 389–402*, July/Aug. 1992.