Bayesian Flooding for Image and Video Segmentation

Elias Grinias

Computer Science Department

University of Crete

A thesis submitted for the degree of

PhilosophiæDoctor (PhD)

2009 February

Abstract

Image segmentation is one of the fundamental problems in image processing and computer vision. Segmentation is also one of the first steps in many image analysis tasks. Image understanding systems such as face or object recognition often assume that the objects of interest are well segmented. Different visual cues, such as colour and texture in still images and motion in image sequences, help in achieving segmentation. Segmentation is also goal dependent, subjective, and hence ill-posed in a general set up. However, it is desirable to consider generic methods that can be applied to a large variety of images and can be adapted for specific applications. This thesis work focuses on developing such segmentation methods that work on natural images.

Segmentation is based on a statistical framework. Visual feature description is unified under a statistical point of view. The first part of the framework proposes a new, block based clustering method for visual content classification. Starting with the computation of visual cues, statistics of blocks are estimated and a k-means algorithm is employed for classification of blocks in a number of classes, using statistical dissimilarity criteria. A novel distance metric between affine models of optical flow is also described. Towards the automation of clustering, a method for computing the number of visual classes as well as a feature selection procedure are also proposed.

Second part of the framework explores region based segmentation, given the statistical description of classes. Initial regions of high confidence per class are determined and two new region

growing algorithms are proposed to expand initial regions. The strong relation of region growing on statistical surfaces to the connectivity percolation process is also underlined.

To test the effectiveness of these new techniques, extensive tests are conducted on the Berkeley segmentation data set and the associated ground truth, using colour and texture. Furthermore, segmentation of independently moving objects using interframe difference and colour as well optical flow based segmentation in image sequences is also described. Finally, an application is shown, in which the proposed framework is used for extracting left ventricle in medical cardiac images.

Περίληψη

Η τμηματοποίηση εικόνων είναι ένα από τα θεμελιώδη προβλήματα στην επεξεργασία εικόνας και τη μηχανική όραση. Αποτελεί επίσης ένα από τα πρώτα βήματα σε πολλές εφαρμογές ανάλυσης εικόνας. Σε συστήματα κατανόησης εικόνων όπως η αναγνώριση προσώπων ή αντικειμένων, συχνά γίνεται η υπόθεση ότι τα αντικείμενα ενδιαφέροντος είναι καλά τμηματοποιημένα. Τα διάφορα οπτικά χαρακτηριστικά, όπως το χρώμα και η υφή στατικών εικόνων καθώς και η κίνηση στις ακολουθίες εικόνων, βοηθούν στην επίτευξη της τμηματοποίησης. Η τμηματοποίηση είναι ένα όχι καλά ορισμένο πρόβλημα, καθώς είναι γνωστό ότι εξαρτάται από την εφαρμογή και το επιδιωκόμενο αποτέλεσμα είναι υποκειμενικό. Εντούτοις, είναι αναγκαίο να εξετασθούν γενικές μέθοδοι που βρίσκουν εφαρμογή σε μεγάλη ποικιλία εικόνων και μπορούν να προσαρμοστούν σε συγκεκριμένες εφαρμογές. Η εργασία εστιάζει στην ανάπτυξη μεθόδων τμηματοποίησης αυτού του τύπου για εφαρμογή σε φυσικές εικόνες.

Η τμηματοποίηση βασίζεται σε ένα πλαίσιο στατιστικής περιγραφής των χαρακτηριστικών. Η περιγραφή αυτή επιτρέπει την ενιαία αντιμετώπιση των χαρακτηριστικών από στατιστικές κατανομές. Στο πρώτο μέρος του πλαισίου προτείνεται μια νέα μέθοδος ομαδοποίησης των μπλοκ της εικόνας βασισμένη στο οπτικό περιεχόμενο. Ξεκινώντας με τον υπολογισμό των οπτικών χαρακτηριστικών, εκτιμάται η στατιστική περιγραφή των χαρακτηριστικών των μπλοκ. Στη συνέχεια, εφαρμόζεται ένας k-means αλγόριθμος για την ταξινόμηση των μπλοκ σε ένα αριθμό κλάσεων, χρησιμοποιώντας αποστάσεις ανομοιότητας κατανομών. Επίσης, περιγράφεται μια νέα μετρική

απόστασης μεταξύ αφινικών μοντέλων οπτικής ροής. Για την αυτοματοποίηση της ομαδοποίησης, προτείνονται μια μέθοδος υπολογισμού του αριθμού των κλάσεων, όπως επίσης και μια τεχνική επιλογής χαρακτηριστικών.

Στο δεύτερο μέρος του πλαισίου, διερευνάται η τμηματοποίηση σε περιοχές, με δεδομένη τη στατιστική περιγραφή των κλάσεων. Το πρώτο βήμα της προσέγγισης είναι ο προσδιορισμός αρχικών περιοχών μεγάλης εμπιστοσύνης για κάθε κλάση, οι οποίες επεκτείνονται στη συνέχεια από δύο νέους αλγορίθμους επέκτασης περιοχών. Επίσης, υπογραμμίζεται η ισχυρή σύνδεση που υπάρχει μεταξύ της επέκτασης περιοχών σε στατιστικές επιφάνειες και τη θεωρία διήθησης.

Ο έλεγχος της αποτελεσματικότητας των νέων τεχνικών βασίζεται σε εκτεταμένα πειράματα στο γνωστό σύνολο εικόνων του Berkeley, βάσει της τμηματοποίησης από ανθρώπους που συνοδεύουν αυτό το σύνολο. Επιπλέον, περιγράφεται η εφαρμογή των τεχνικών στην τμηματοποίηση αντικειμένων που κινούνται ανεξάρτητα από την κίνηση της κάμερας, χρησιμοποιώντας είτε τη διαφορά φωτεινότητας μεταξύ διαδοχικών καρέ video σε συνδυασμό με το χρώμα είτε την οπτική ροή που εξάγεται από διαδοχικά καρέ. Τέλος, αναλύεται η εφαρμογή του προτεινόμενου πλαισίου στην εξαγωγή της αριστερής κοιλίας σε ιατρικές εικόνες καρδιάς.

Contents

List of Figures

LIST OF FIGURES

List of Tables

LIST OF TABLES

Chapter 1

Introduction

1.1 Problem Formulation

Image segmentation is a key step in many computer vision analysis and interpretation tasks. Segmentation of colour textured images has become a necessity for many applications, such as content based image retrieval and object recognition. Video segmentation is a key step in determining the motion features, as well as the position and 2D shape of the scene objects. Such a description may be used either for coding purposes in order to reduce storage and transmition requirements or for indexing and retrieval purposes in order to improve the content description and storage reduction of visual databases. The development of the corresponding international standards MPEG-4 for coding and MPEG-7 for visual content description, which both rely on the concept of audio/visual objects, has raised the importance of these methods [6]. Finding semantic regions is the ultimate goal of segmentation for image understanding.

Let Λ be the image lattice and $\mathbf{I}_\Lambda$ the image data defined on it. The image segmentation refers to partitioning the lattice into an unknown number M of *meaningful*, disjoint regions

$$\Lambda = \cup_{m=0}^{M-1} R_m, \quad R_m \cap R_l = \emptyset, \quad \forall m \neq l. \tag{1.1}$$

The known ambiguity in segmentation problems derives by the term "mean-

ingful" in this definition. In low level image processing tasks, an acceptable segmentation result extracts compact image regions, whose sites are characterized by high similarity according to image cues or the statistical description of derived features. In higher level tasks, towards image understanding, image regions correspond to semantic objects, that may be described by visual feature classes and the segmentation task in that case is to localize or discriminate the objects of interest from their background, in an image or a video sequence.

Despite the plethora of methodologies for image segmentation, a generic, admitting some user-tuned parameters, algorithm that addresses the whole range of segmentation problems and applications does not exist. This is due to the complexity and ill-posedness of segmentation problems and the fact that often an unambiguous ground truth does not exist, instead the plausible result is dependent on subjective interpretation. In light of these considerations, interactive segmentation techniques are also frequently employed [7]. However, referring to segmentation applications, ambiguity is more or less resolved without the need of user interaction, in segmentation problems of real life objects, or classes of objects, that are efficiently described by visual features. The description refers either to the objects themselves or their behavior. Known problems of that kind, is frontal human face localization and independently moving objects localization.

1.2 Relevant Work

Considering image segmentation as an optimization problem we should introduce four kinds of constraints: boundary, shape, region and topology constraints. Boundary constraints refer to cues presenting significant differences between neighboring sites belonging to different regions. The objective could be limited to detect boundaries, without labelling regions. Soft shape constraints lead to boundary regularization, while hard shape conditions could address specific deformable object localization. In any case an accurate spatial localization is required. Region constraints refer to pixel grouping according to class properties. The objective is to obtain regions

uniform and homogeneous with respect to the selected features. Finally, topology constraints should limit the number of connected components, in order to eliminate irrelevant segments.

We refer to very few well-known algorithms for illustrating the role of different constraints in region boundaries localization or image segmentation. In edge detection [8] only boundary constraints are taken into account, while contrast constraints can be integrated in region growing techniques [9]. The last category of techniques incorporates soft topology constraints in the propagation process. Boundary constraints combined with geometric shape constraints lead to geodesic active contours [10], where a global optimization method is applied. This approach is generalized in [11] giving a powerful method, because it introduces and deals with boundary, shape and region constraints.

However, it should be noticed that, even implicitly, topology is always taken into account by the algorithms mentioned above and the same holds for the graph theoretic approaches which, effectively, in the last years, revealed suitable for efficiently handling the spatial coherence. Visual groupings can be captured by mapping the perceptual features and coherence constraints to a graph. Shi and Malik [12] considered image segmentation as a graph partitioning problem in order to obtain perceptual groupings by focusing on local features and their consistencies in the image data. They used a *normalized cut* criterion for measuring both the dissimilarity between the different groups as well as the similarity within the groups. Felzenszwalb and Huttenlocher [13] proposed an efficient graph-based method using a *minimum spanning tree* (MST) algorithm taking simple greedy decisions, while respecting global properties for image segmentation. Zabih and Kolmogorov [14] proposed a segmentation algorithm that operates simultaneously in feature space and in image space. An energy function is defined on graph and the *min cut* algorithm provides the optimum spatially coherent grouping. Recently, Ding, Ma and Chen [15] proposed a *connected coherence tree* algorithm (CCTA) for image segmentation, based on the definition of a neighbor coherence criterion.

Also in the last years, Ion, Kropatsch and Haxhimusa [16] have shown the

similarity of the MST construction with the watershed transform, while Falcao et al. [17] considered a *minimum-cost path forest* in a graph for designing image processing algorithms taking into account the connectivity with possible use in image segmentation.

1.3 The Proposed Schema

As it is depicted in Fig. 1.1, the proposed framework in this thesis, may be roughly separated in two main components, namely, *feature extraction* and *classification* in the visual space, using image data information, and *Bayesian flooding* and *merging* in spatial domain, where the connectivity is based on the computed features of classes. That way, our method belongs to techniques that involve *global feature space analysis* under *spatial coherence constraints*, such as that in [18], while our framework achieves these objectives sequentially, in order to regularize the segmentation map by topological constraints.

Bayesian flooding relates to connectivity percolation theory [19] and is implemented using *region growing* (RG). A survey of seeded region growing algorithms is given in [20]. Merging is the last, post-processing step after region growing and it is used in order to improve the segmentation result, based on size, boundary and statistical information of neighboring regions, that belong to different classes. In the next two Subsections, the parts of our framework are described respectively, in more detail.

1.3.1 Feature Classification

In the pattern classification framework adopted in our work, the number of classes, as well as the visual cues selected for region discrimination are either provided by user or automatically extracted using novel cluster validity methods [21]. The extracted features are classified using *unsupervised clustering* in an image block basis to provide the required stability of classes features estimation. Referring to Fig. 1.1 and given as input

- the number of classes and

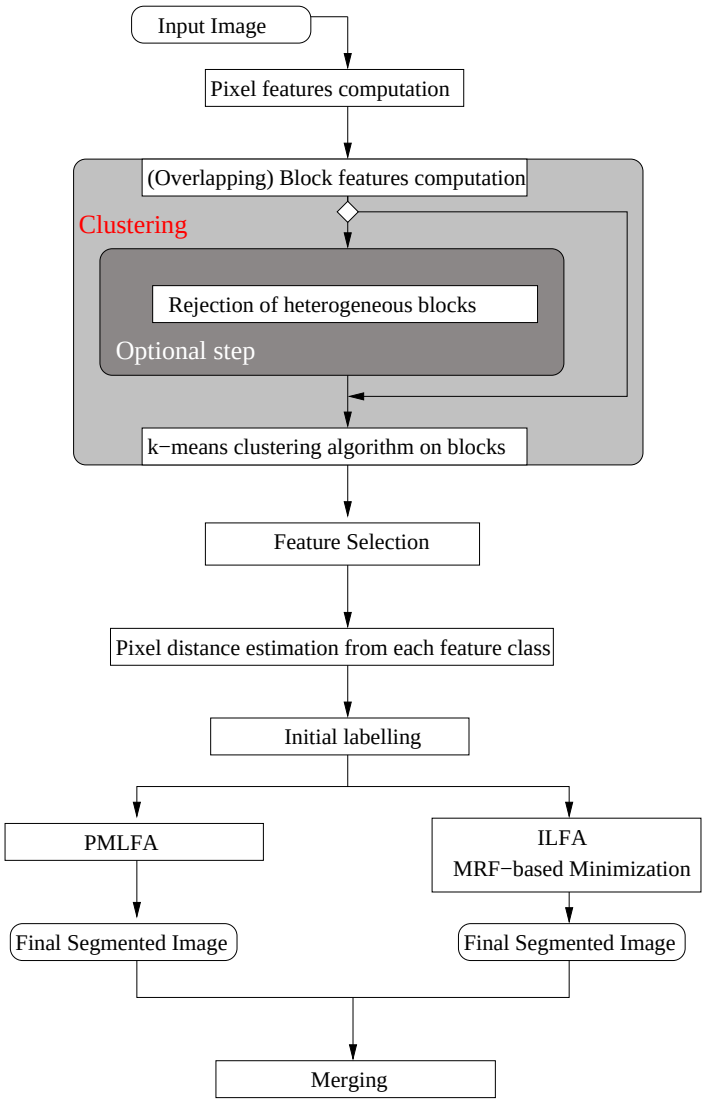


Figure 1.1: Flowchart of the proposed segmentation framework.

- the selection of classification and segmentation features,

the derived pixel features are computed. The next step consists of computing the corresponding features per block and, optionally, rejecting heterogeneous blocks, since these are likely to cross a boundary between different segments. Based on the per block features, a distance measure is computed for each pair of the remaining blocks and a *k-means* clustering procedure is applied using these distances.

When features are not selected by user, automatic feature selection takes place after clustering is performed in the way described above, for a number of different feature combinations. The best feature combination is selected using cluster validity and statistical feature selection techniques. Based on the resulting clustering of best combination, a number of feature classes are extracted, one per cluster. Pixel based algorithms that follow, effectively rule out the artifacts in the pixel based accuracy of segmentation results that are often caused by the block based feature classification.

Depending on the features under consideration, the block based clustering method may be replaced by mixture decomposition analysis using for example *Expectation Maximization (EM)*. This approach has been followed for the statistical description of the interframe difference in change detection based, object localization.

1.3.2 Graph Theoretic Algorithms and Bayesian Flooding

For a site $u \in \Lambda$, let $\boldsymbol{\xi}_u$ be a feature vector extracted by image cues $\mathbf{I}_\Lambda$ and $p(\boldsymbol{\xi})$ the *probability density function (pdf)* of it computed over Λ. We suppose as before that the image lattice Λ is decomposed in M disjoint regions R_m and that, in addition, each region belongs to a class C_k (or k in short) according to the feature vector $\boldsymbol{\xi}$. Given the number K of classes, the statistical description in the feature space, derived by the feature classification

method, is given by the mixture

$$p(\boldsymbol{\xi}) = \sum_{k=0}^{K-1} P_k p_k(\boldsymbol{\xi}),$$

where P_k and $p(\boldsymbol{\xi}|k) = p_k(\boldsymbol{\xi})$ are the corresponding estimates of *prior probability* and *conditional probability density function (pdf)* of class k. According to the *Bayesian rule*, the posterior probability of a site to belong to class k is given by

$$\Pr\{k|\boldsymbol{\xi}_u\} = \frac{P_k p_k(\boldsymbol{\xi}_u)}{\sum_{l=0}^{K-1} P_l p_l(\boldsymbol{\xi}_u)}, \tag{1.2}$$

In this thesis, flooding methods are proposed to obtain the partition:

$$\Lambda = \cup_{m=0}^{M-1} \hat{R}_m, \quad \hat{R}_m \cap \hat{R}_l = \emptyset \quad \forall m \neq l, \tag{1.3}$$

on lattice Λ, as close as possible to the unknown, original partition given by Eq. (1.1), based on the computed pdf for each class and using only the information of Eq. (1.2) for each site. In that case, wrong labelling in region R_m belonging to class k, appears exclusively by the *Bayesian classification error* caused to class k due to the presence of the other classes, in the feature space. The effect of this type of error is to disconnect R_m in smaller connected components. The purpose of the methods presented herein, is to reconnect the regions using the statistical information of classes. The formation of a region by unifying smaller connected components under a statistical point of view is studied by the well known in computational physics, *connectivity percolation theory* [19]. Indeed, the image segmentation problem as it is formulated herein, is a connectivity percolation process. Although percolation theory describes the problem, an accurate segmentation solution is not guaranteed without applying topological constraints.

To accurately extract $\hat{R_m}$, probabilistic distances of $\Pr\{k|\boldsymbol{\xi}_u\}$ are used to select and label a set of pixels that belong to class k with high confidence, thus providing an initial map of correctly labelled pixels. Having available the initial map of correctly labelled pixels, we propose two new flooding algorithms in order to fill the initial map of decisions, using statistical dissimilarity cri-

teria. The new algorithms are called *Independent Label Flooding Algorithm* (ILFA) and *Priority Multi-label Flooding Algorithm* (PMLFA) respectively. Because ILFA is characterized by relaxed shape constraints and not strong topology constraints, it is followed by a MRF-based minimization algorithm (ILFMA). PMLFA imposes strong topology constraints and constructs the segmentation map with linear computational cost to image dimensions.

1.4 Contributions

In the proposed clustering and segmentation schema, treatment of visual features is unified using their statistical description. Assuming that natural objects could be described by a characteristic feature distribution, segmentation using the proposed framework on different visual cues becomes feasible, either in still images or video, as it is described in subsequent Chapters of the thesis. Visual features used for segmentation in this thesis are intensity, colour, texture, interframe differences and optical flow. Considering classification of visual data, new, robust methods are proposed for automating statistical clustering as well as the selection of segmentation features. Robustness of these methods is proved in practice by using them in segmentation of natural images. Furthermore, a novel distance metric between affine models of data sets is also defined and used in block based clustering of optical flow.

Having at hand a confident description of visual classes, implied topology constraints guide the segmentation process for obtaining semantic regions, without sacrificing topology flexibility. This is the objective of Bayesian flooding algorithms described in this thesis. Since, any visual cue may be imperfect with respect to semantic content, incorporating topology constraints in the segmentation process allows for limiting the effect of visual imperfections. Usage of topological constraints is admitted in such a way that also allows topological flexibility, while region homogeneity is also preserved.

1.5 Thesis Organization

This thesis is organized as follows: in Chapter 2 the feature prototypes extraction technique is presented. In Chapter 3 the relation between image segmentation and percolation theory is examined. In Chapter 4 the new flooding algorithms are described in detail. In Chapter 5 we present the application of algorithms to image segmentation tasks by the fusion of luminance, texture and colour information. The application of the proposed framework in colour based face detection is also described. In Chapter 6 colour and change detection based flooding segmentation is applied in image sequences. Furthermore, optical flow segmentation using the novel distance between affine models of data sets is described in this Chapter. Finally, in Chapter 7 the application of PMLFA in the crucial task of left ventricle segmentation in medical images, is presented in detail.

Chapter 2

Automatic Feature Extraction

Visual features considered for segmentation are intensity, chromaticity, texture, optical flow and interframe difference either in their own or in combination. In this Chapter, semi and fully unsupervised clustering of intensity, chromaticity and texture is described, although, for features, such as intensity and chromaticity, mixture analysis could also be used, if image region statistics are described by a reliable and tractable mixture model, which, furthermore, is enough to appropriately separate image regions.

In the general case, clustering is basically performed semi-automatically, in the sense that a number of parameters has to be provided given the image data, such as the combination of visual cues or the number of classes that are present in the image. In order to determine the features of classes, the image is divided into blocks that may overlap. The block size is related to the image size and the selected image features. Segmentation features are computed for each block and, optionally, heterogeneous blocks are rejected, using an inhomogeneity detection procedure that relies on feature dependent statistical criteria. A *k-means* algorithm is used to cluster the remaining blocks to obtain classes corresponding to region prototypes. The number K of labels-classes is assumed to be either user given or automatically provided.

To perform clustering automatically, important parameters of classification such as the combination of features and the number of classes for each combination should be considered as decision tasks that guide the cluster-

ing procedure, without user interaction. In what follows, two methods to automate clustering are described. The first one computes the appropriate number of classes K, given as input to k-means for a given combination of features, while the second one selects the best combination of features for segmentation, after clustering has been performed for all combinations.

Furthermore, we describe in detail the statistical modeling of features as well as the feature dependent dissimilarity metrics, which are used to classify image blocks as well as for deciding the number of classes and for selecting the appropriate visual features. Probabilistic criteria are used to measure the distance between texture, intensity and colour features of data sets. Finally, the clustering of blocks to labels is discussed in extent.

However, although clustering refers to visual features of still images in this Chapter, features that are extracted by image sequences could be considered for classification as well, using the same, block based clustering technique. Clustering of optical flow vectors using affine modeling of data sets and a novel distance metric, is described in detail in Section 6.3, using the block based method of this Chapter.

2.1 Visual Features

In this Section, the statistical modeling of intensity, chromaticity and texture visual features is described. Statistical modeling of features is then used to describe and compare data sets. With the term "data set" we refer to connected components of sites on the grid, such as blocks, regions or the whole image, treated as unordered site sets. Sites of any data set are used to sample its statistical distribution, for the given features.

This way, visual features of data sets are replaced by their probability density functions. Intensity and chromaticity are described by histograms, while the statistical modeling of texture derives by known statistical properties of texture analysis components. In the case of intensity and chromaticity, a method to automatically compute histogram bins is also described in the following Subsection.

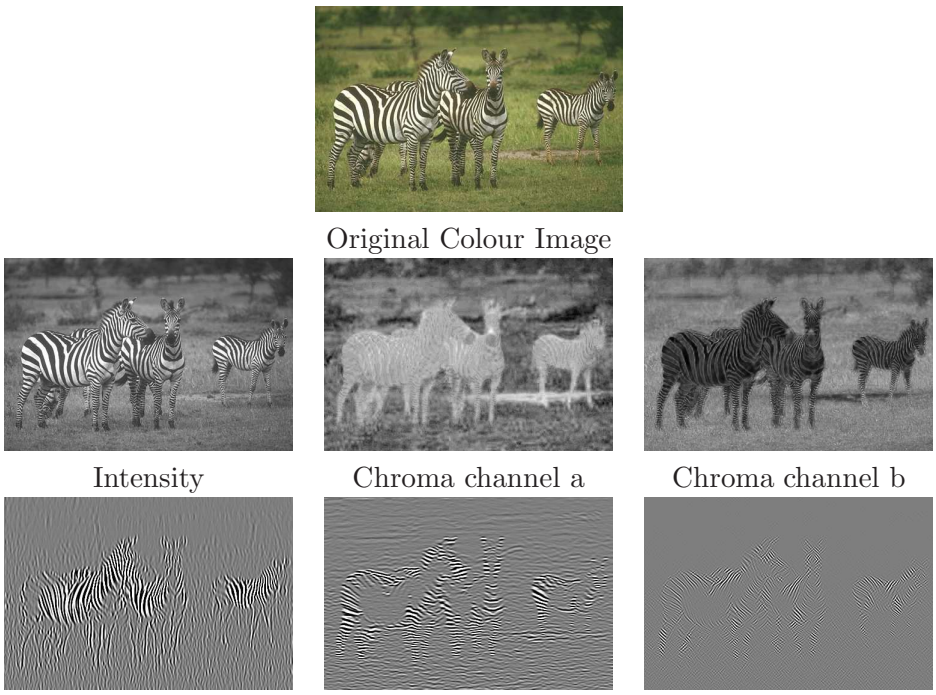


Figure 2.1: Intensity, chromaticity (2^{nd} row) and texture (3^{nd} row) cues for the colour image of the first row.

2.1.1 Colour Features

The colour of images is described by the intensity and the chromaticity components (a, b) of Lab colour space. This way, the RGB colour image

$$\mathbf{c}_\Lambda = (R_\Lambda, G_\Lambda, B_\Lambda)$$

is transformed to image

$$\mathbf{c}_\Lambda = (I_\Lambda, (\mathrm{a}, b)_\Lambda).$$

In Fig. 2.1, the extracted cues for the image of the first row are shown. In the second row of the Figure, the intensity and chromaticity channels $(\mathrm{a}, b)_\Lambda$ of Lab colour space are depicted. For the needs of demonstration, chromaticity channels have been scaled to the range [0 1] and displayed as grayscale images.

In order to extract the statistical description of image, intensity I_Λ as well as the chromaticity image $(\mathrm{a}, b)_\Lambda$ are quantized using k-means algorithm. Quantization reduces the spatial uncertainty of features and speeds up the computation of their statistics during clustering. The number of bins for intensity and chromaticity quantization by k-means is automatically determined using a method which takes under consideration the local standard deviation of features.

The proposed schema is described for a 1D feature ξ and the same approach is followed separately in each dimension, in the case of more than one feature dimensions. The method detects the average step of bins in an adaptive manner, based on the local spatial distribution of feature values. The local standard deviation of the feature for each image site s is defined as

$$\sigma_s = \sqrt{\mathcal{E}\left\{\xi_s^2\right\} - \left(\mathcal{E}\left\{\xi_s\right\}\right)^2},$$

where ξ is the feature of image and $\mathcal{E}$ is the mean operator involving the sites in a disk centered at s. The average step of bins is considered to be the median μ_ξ of local standard deviation and the number of quantization values

K_ξ is given by the equation

$$K_\xi = \left\lceil \frac{\max\{\xi\} - \min\{\xi\}}{\mu_\xi} \right\rceil . \tag{2.1}$$

Intensity data are quantized in K_I bins by Eq. (2.1), according to the computed average step μ_I. If the frequency of a bin is lower than a threshold, it is rejected as statistically negligible and the number of bins K_I is decreased by one. The k-means algorithm that follows, computes the final bin centers and maps image sites to bins according to their intensity value, using the city-block distance between samples.

In the case of chromaticity, average steps μ_a and μ_b are determined separately and K_a, K_b are computed by Eq. (2.1). The 2D chromaticity space is quantized in

$$K_{ab} = K_a \cdot K_b$$

2D bins. Statistically negligible bins are rejected, as in the case of intensity. Having available the final number of not rejected bins K_{ab}, the mapping of sites to bins according to their chromaticity value is performed by k-means, using the Euclidean distance.

2.1.2 Texture Details

Multichannel filtering approaches for texture analysis have been proposed, using filter banks of Gabor filters [22, 23] or wavelet packet frame decomposition [24]. In [5], many different multichannel filtering approaches have been compared. In our work we mainly use a Discrete Wavelet Frames (DWF) filter bank [25], which appears to give a good scale-space image analysis.

Texture analysis gives a set of K_ζ detail components for image grid Λ,

$$\boldsymbol{\zeta}_\Lambda = \{\zeta_{i,\Lambda}, \quad 1 \le i \le K_\zeta\}.$$

Because of their orthogonality, texture details are statistically independent. Furthermore, it is assumed that the distribution of each component is described by a zero mean Gaussian model and thus, only variance σ_i^2 needs to

be computed in order to get the statistical description of the i^{th} component, in a given data set.

Images of the third row in Fig. 2.1 display, from left to right, the vertical, horizontal and diagonal details respectively, of the second layer (scale) obtained by the DWF texture analysis of the image. As for chromaticity channels, texture details have been scaled to the range [0 1] and displayed as grayscale images.

2.2 Dissimilarity Metrics on Data Sets

Several metrics have been proposed in the literature to measure the dissimilarity between data sets. Well known statistical metrics are the Mahalanobis, Bhattacharyya and χ^2 metrics [26]. The robust, but in some cases computationally intensive Mallows distance has also been used in image classification tasks [27].

In our work, the metric used to measure the dissimilarity between the statistical description of data sets $\mathcal{D}_1$ and $\mathcal{D}_2$, described by distributions p_1 and p_2 respectively, is the Bhattacharyya distance, denoted herein as J^B. The definition as well as the properties of this statistical metric are outlined in Section A.1 of Appendix. The distance is based on the Bayesian classification error of distributions p_1 and p_2, it can be applied on parametric as well as on non parametric statistical modeling of data and it is computationally tractable.

An example is given in order to explain how Bhattacharyya distance between blocks is computed. In image of Fig. 2.2, the division of colour image of Fig. 2.1 in 32×32, non overlapping blocks is depicted. Dissimilarity measurement involves the *Red* and the *Blue* block of the image. It is assumed that visual features extracted for the image are intensity, chromaticity channels and the three texture components of Fig. 2.1. Furthermore, texture details are assumed zero mean, Gaussian distributed. The last important assumption made is that the extracted features are statistically independent.

Automatic quantization of intensity and chromaticity for this image (Subsection 2.1.1) results in $K_I = 27$ and $K_{ab} = 98$ bins, respectively. These bins

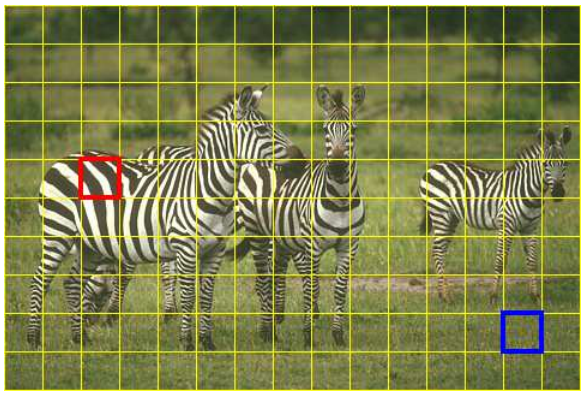
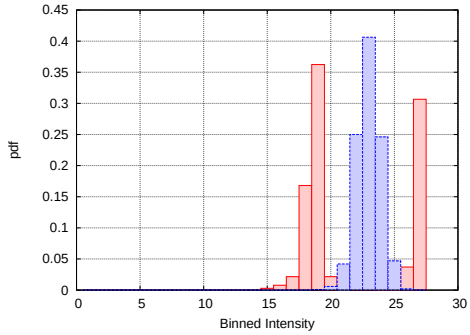


Figure 2.2: Division of image in 32×32 blocks. Yellow lines delineate block borders. *Red* and *Blue* blocks are compared using Bhattacharyya distance (see text).

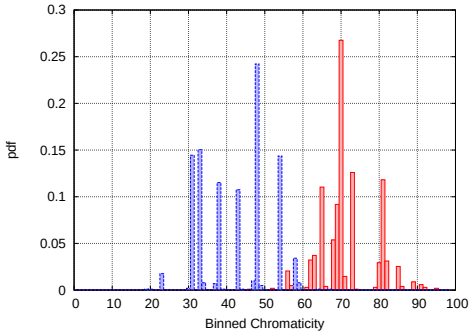
are used for sampling intensity and chromaticity pdfs of blocks by the corresponding histograms, as it is illustrated in plots (a) and (b) of Fig. 2.3. Furthermore, variance of the three texture components is also sampled in each block and the Gaussian distributions per component are shown in plots (c)-(e) of Fig. 2.2, for the two blocks. Red (blue) coloured pdfs, are those of *Red* (*Blue*) block of Fig. 2.2. The statistical dissimilarity of the two blocks in that case is the sum of three terms, namely, the Bhattacharyya distance J_I^B between histogram computed distributions of intensity bins, obtained by Eq. (A.2), the Bhattacharyya distance J_{ab}^B between histogram computed distributions of chromaticity bins, using once again Eq. (A.2) and finally, the Bhattacharyya distance J_ζ^B in the form of Eq. (A.5), between the corresponding variances of texture components.

When data sets are blocks of the image, problems may arise in the case of statistical modeling by histograms, due to the sampling of distributions by bins for the relatively small number of block sites. Thus, the measured statistical distance between two blocks may be infinite, even if their distributions are similar. This behavior has to be detected and corrected, since most of the decisions made by the classification schema in this chapter are based on the distance between blocks.

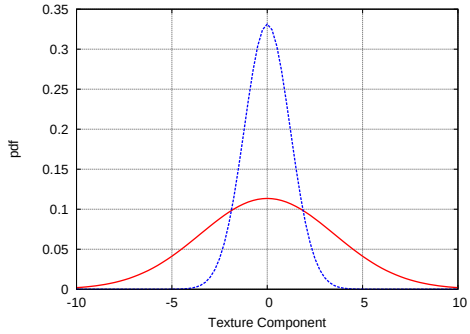
The solution to this problem comes from the property of Bhattacharyya



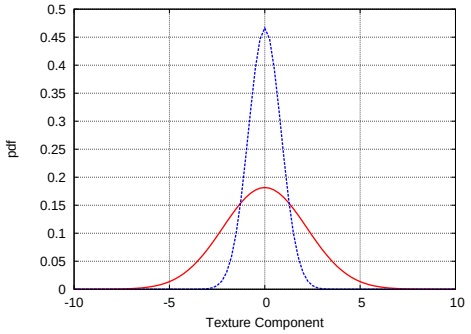
(a) Intensity pdfs



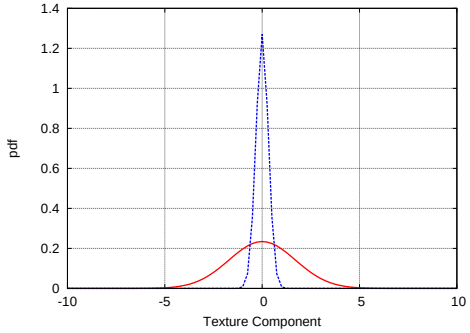
(b) Chromaticity pdfs



(c) 1^{st} texture component pdfs



(d) 2^{nd} texture component pdfs



(e) 3^{nd} texture component pdfs

Figure 2.3: Intensity, chromaticity and texture components distributions of red and blue blocks of Fig. 2.2.

distance, which applies to parametric as well as to non parametric statistical models. Thus, if the distance between the histograms is infinite, the two distributions are assumed Gaussians. In that case the mean and variance per block are computed, assuming statistical independence of feature dimensions and Bhattacharyya metric in the form of Eq. (A.4) is used to compare the distributions. This solution is in fact a method to smooth the histograms, but only when such a process is really necessary.

2.3 Clustering

In this Section, statistical clustering of visual features is discussed in extend. Clustering problem and the algorithms that have been used are described in Subsection 2.3.1. Clustering is performed in a block basis, using Bhattacharyya metric to measure the statistical distance between blocks. A k-means algorithm is used to classify image blocks for a given combination of features (Subsection 2.3.4), preceded by an optional method to reject heterogeneous blocks (Subsection 2.3.3). Towards the full automation of the classification procedure, a method to automatically determine the number of classes for each combination is described in Subsection 2.3.2, while a cluster validity metric to select the best combination of features is presented in Subsection 2.3.5. In the first case "Clustering" module of flowchart in Fig. 1.1, becomes that of Fig. 2.4. This clustering module is used for selecting the best combination of features.

2.3.1 Clustering Algorithms

The goal of any clustering technique is to obtain a $K \times n$ partition matrix or map $U(X)$ of a data set

$$X = \{x_1, x_2, \ldots, x_n\} \in \mathbb{R}^N.$$

The matrix U represents the partitioning of data in a number K of clusters $\{C_1, C_2, \ldots, C_K\}$ and the notation $U = [u_{kj}]$ is used to denote the mem-

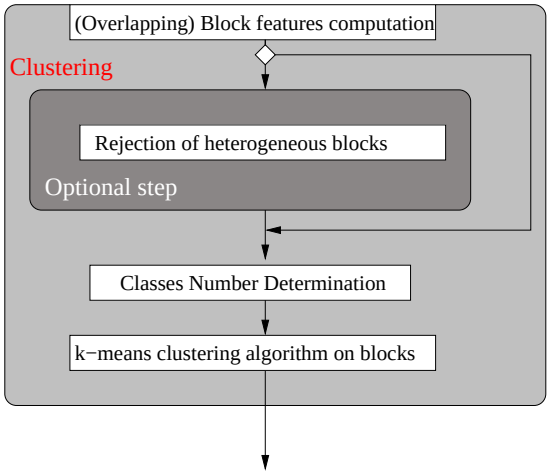


Figure 2.4: Flowchart of the proposed clustering framework with automatic classes number determination.

bership of data vector x_j to cluster C_k. In the case of hard partitioning, $u_{kj} = 1$ if data sample j belongs to cluster k. Otherwise, $u_{kj} = 0$. Clustering algorithms broadly fall in two categories, partitional and hierarchical [28]. The widely used *k-means* algorithm belongs to the first category, while *average linkage* is the hierarchical algorithm, used herein to define an evaluation metric curve, in order to determine the number of classes.

Average linkage is a non iterative method, which creates an hierarchical, binary tree T of clusters using distances d_{ij} between samples x_i and x_j for $i, j = 1, 2, \ldots n$ and $i \neq j$. The leafs of the tree are the data samples which are considered singleton clusters. In average linkage, the distance of cluster pairs k and l is given by equation:

$$\bar{D}_{kl} = \frac{1}{n_k n_l} \sum_{i=1}^{n_k} \sum_{j=1}^{n_l} d_{ij} \tag{2.2}$$

where n_k, n_l are the numbers of samples in clusters k, l respectively. Beginning at the lower level of leafs, at each step of the method the two nodes-clusters with minimum distance are merged and this way, a new node of the tree is constructed, which represents the newly created cluster that replaces

the merged pair. The distance of the remaining clusters to the newly created one is recomputed as above and merging continues bottom-up in this manner, until a single cluster comprising all the samples is formed.

2.3.2 Number of Clusters Determination

A crucial parameter of the automatic clustering scheme, is the number of feature classes K, used in k-means. It is well known in clustering theory that the number of classes, or clusters, is difficult to be correctly determined without pre-existing knowledge of the data or the presence of a user with sufficient knowledge of the domain. Due to these impractical requirements, several approaches have been proposed in the literature in order to automatically determine a reasonable number of classes for a data set. The method described below belongs to the "knee locating" methods [29], applied on an evaluation metric curve, known as *evaluation graph*. The "knee" of a curve is loosely defined as the point of maximum curvature. As it is shown in Fig. 2.5, given the distances between all block pairs, evaluation graph is the outcome of average linkage, which is given as input to the "knee locating" method in order to compute the appropriate number of classes.

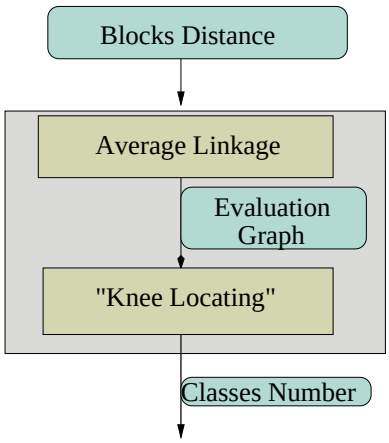


Figure 2.5: Flowchart of classes number determination module of Fig. 2.4.

Comparison of m different clustering maps $U_1, U_2, \ldots, U_m$ of X, constructed by a clustering algorithm, is based on validity measures V_i, $(i =$

$1, 2, \ldots, m$). V_i measures the discrimination achieved by the corresponding clustering U_i. In the case of average linkage, the validity measure V_i of a partition U_i in a number i of clusters is given by the average distance of the two sub-trees, last merged in order to get clustering U_i. Although this is a greedy computation, it performs reasonably in practice and it is a fast method to obtain a number of clustering maps. The evaluation graph is the xy-plot of the evaluation metric values V_i (y-axis) vs. the number of classes i (x-axis).

The evaluation graph for the clustering of blocks of a colour image using average linkage, is depicted in the plot of Fig. 2.6. Chromaticity, **ab** histograms have been used as classification feature and Bhattacharyya distance $J^B(p_i, p_j)$ is used to measure the dissimilarity d_{ij} between block histograms p_i and p_j.

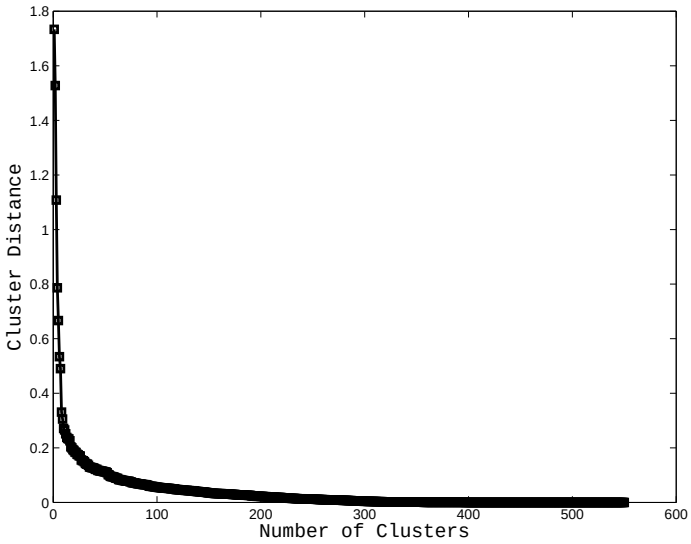


Figure 2.6: Evaluation curve obtained by average linkage for the blocks of an image, using **ab** histograms and Bhattacharyya distance between blocks.

The reasonable number of clusters K is determined on the evaluation graph of T for partitions U_i with $i = 2, \ldots, m$, i.e. excluding the trivial case of one cluster. In many cases, these curves are noisy and trying to locate the maximum curvature is a difficult task.

The method described below, is a robust "knee locating" technique. It

locates the knee as the point of maximum angle between two best-fitted lines to the curve, taking into account the least square error of that fit, as it is the case in [29].

Given the graph, first, the line l_{tot} that best fits the overall metric curve V_i, $(i = 2, \ldots, m)$, is computed. If the slope of this line is less than a predefined threshold, the discrimination between the different clustering maps is low and thus we set $K = 2$. Otherwise, for each cluster number $i = 3, m-1$, best-fit lines l_i^- and l_i^+ are computed in intervals $[1, 2, \ldots, i]$ and $[i, i+1, \ldots, m]$ using the corresponding values of V_i. The angle ϕ_i, as well as the x-coordinate q_i of the intersection point between the two lines are retained, in each case. In addition, the sum e_i of least square errors e_i^- and e_i^+ corresponding to each line fitting is also considered.

To determine the appropriate number of classes, two cluster numbers are checked as x-coordinate candidates for the knee point on the curve. That of maximum angle i_1 and the number of minimum square error i_2. In many cases the two numbers may be the same. Otherwise, i_2 is considered better candidate if ϕ_{i_2} is large enough, (larger than a threshold) and error e_{i_1} is significantly greater than e_{i_2}. How large the angle ϕ_{i_2} should be, depends on the slope of l_{tot}. This way, the knee point location is robustly specified, as that of a fairly large angle between the lines that best fit the curve.

Furthermore, if we denote as i' the decision made between numbers i_1 and i_2, the exact number of classes is set to $K = \lfloor q_{i'} \rfloor + 1$ if the abrupt change in metric values appears at $\lfloor q_{i'} \rfloor + 1$, rather than at $\lfloor q_{i'} \rfloor$. This heuristic locates the knee of the curve with increased accuracy.

Fig. 2.7 demonstrates the method as it is applied to determine the number of classes using values $V_2, V_3, \ldots, V_{11}$ of the evaluation graph in the plot of Fig. 2.6. Evaluation points are depicted as blue squares in Fig. 2.7. The red lines in that Figure, are those of maximum angle, indicating that $K = 5$.

A novel center-based clustering method, proposed in [30], has also been used for clustering in this work. Given as input the distances between data point pairs, as other clustering approaches, it reduces clustering to a well defined (but NP-hard) minimization problem, solved by making use of dual Linear Programming (LP) relaxation. Under this formulation, the stability

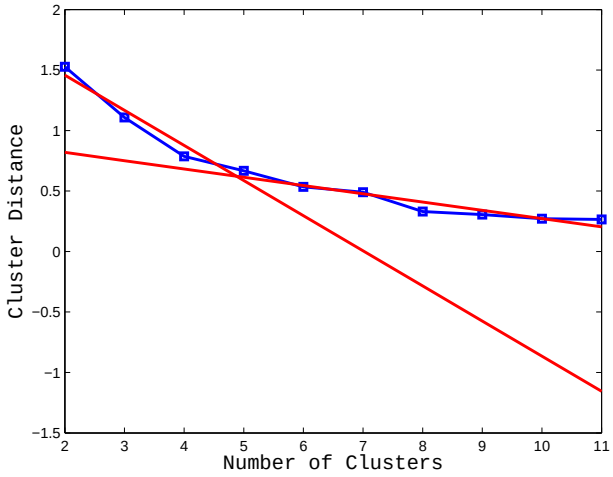
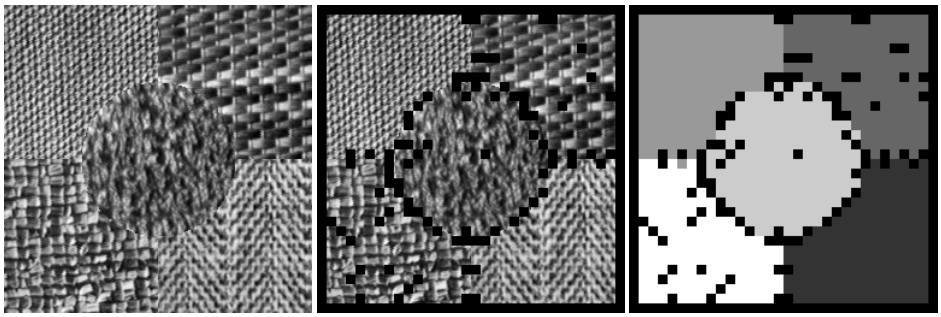


Figure 2.7: The two lines of maximum angle, indicating the appropriate number of 5 classes, given the values V_2 up to V_{11} of the evaluation graph in Fig. 2.6.

of each data point to be selected as cluster center, is defined as an LP based quantity. Optimization is simultaneously performed over both the cluster centers and the number of clusters, using the input distances. Thus, when this algorithm is used herein to determine the number of clusters, the outcoming clustering map and cluster centers are taken under consideration as well, avoiding completely k-means clustering.

2.3.3 Rejection of Heterogeneous Blocks

Heterogeneous blocks may optionally determined and rejected in order to prevent the corruption of the feature classification process (k-means), which follows. Rejection is based on an edge detection method for blocks, performed separately in block rows and columns. For each block $B_{i,j}$ in a row i, the Bhattacharyya distance $J^B(p_{i,j-1}, p_{i,j+1})$ is computed as a measure of gradient for $B_{i,j}$, based on the statistical description $p_{i,j-1}$, $p_{i,j+1}$ of blocks $B_{i,j-1}$, $B_{i,j+1}$, respectively. The same computation is made in columns. Then, row (column) blocks are sorted according to their gradient and a user given percentage of blocks (typically 10%) with the highest gradient determines



(a) (b) (c)

Figure 2.8: Block rejection (b) and classification (c) for the *five regions* image (a).

a cut-off gradient threshold for block rejection. Afterall, maxima suppression is performed in each row (column) to reject only the blocks with locally maximum gradient and local maxima above the cut-off threshold.

The procedure is depicted in Figures 2.8, 2.9 for the image of Fig. 2.8(a). The features used in that case are intensity 1-D histograms and Discrete Wavelet Frames (DWF) texture analysis [25] of the image up to three layers. Texture details are assumed to be zero-mean, Gaussian distributed and uncorrelated, and therefore they are described by the variance of the texture components. The image is divided in 8×8 blocks and the statistical description of block features is computed. Bhattacharyya distance between blocks is used to measure the statistical gradient, by summing the distance in the form of Eq. (A.2) for intensity histograms and the distance in the form of Eq. (A.5) for the texture details of the block pair. In Fig. 2.9 the frequency of the row blocks according to their statistical gradient is shown. The dashed line in that plot corresponds to the cut-off threshold computed by the 10% of blocks with the highest gradient. In Fig. 2.8(b) the rejected 8×8 blocks are shown on the image.

However, it is noticed that block rejection is used only to demonstrate the method in that case, since the result of k-means that follows is good even without applying block rejection.

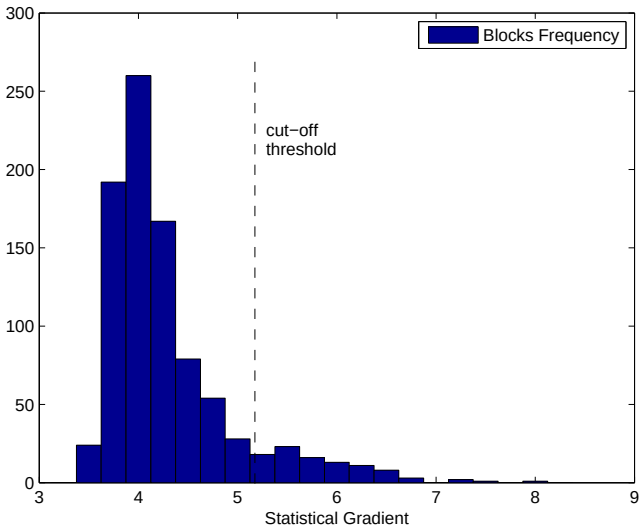


Figure 2.9: Frequency of block rows statistical gradient for the *five regions* image.

2.3.4 Block Based Feature Clustering

A *k-means* algorithm is employed for grouping feature vectors and for the computation of feature prototypes of classes, for a given combination of features. Heterogeneous blocks are optionally determined and rejected as described in the previous Subsection. The retained blocks are given as input to a *k-means* initialization algorithm proposed by Kauffman and Rousseeuw [31]. The initial clustering is obtained by the successive selection of prototype blocks until K instances have been found, where K is the number of clusters, given by the user or automatically extracted using the method of Subsection 2.3.2. The first representative block is the most centrally located block in the feature space. The rest of representative blocks are selected according to the heuristic rule of choosing the blocks that promise to have around them a higher number of the rest of blocks. The output of initialization is a block-prototype for each class.

Finally, *k-means* is applied and each block is assigned to its most similar class. Depending on the statistical description of features, the feature vector

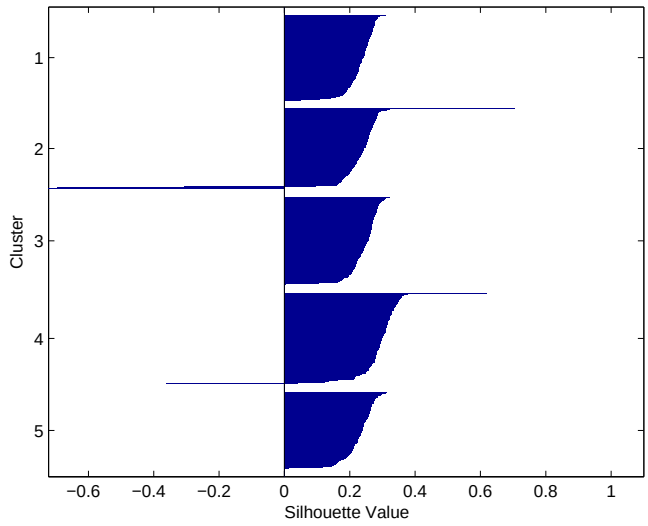


Figure 2.10: *Silhouette* plot for the block classification of the *five regions* image.

of the most centrally located block (centroid) of each class may be used as "mean" feature vector, at each iteration of *k-means*. That way, the features describing each class are obtained at the end of block clustering.

In Fig. 2.8(c), the clustering of not rejected blocks in five classes, is depicted for the *five regions* image, using centroids of clusters as "mean" feature vectors. Bhattacharyya distance is used as the distance between instances in *k-means*. The efficiency of the proposed approach is demonstrated by the so called *silhouette* plot [31] of Fig. 2.10. In that plot, the silhouette value of not rejected blocks is depicted for each class. Silhouette value indicates the separability of each block from the blocks of the other classes and ranges from -1 to 1, i.e. from low to high separability. As a second step of rejection, blocks with negative silhouette value may also rejected and not used for feature statistics computation of their class.

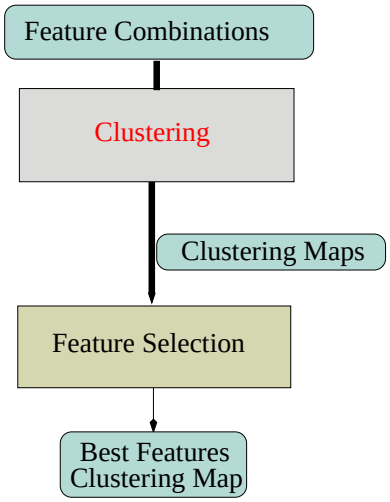


Figure 2.11: Flowchart of automatic feature selection using clustering module of Fig. 2.4. Bold lines indicate that clustering is performed for all feature combinations.

2.3.5 Feature Selection

A generic method to select among the seven combinations of intensity, chromaticity and texture features the best one according to cluster separability, is described hereafter. We denote each one of the $2^3 - 1$ combinations as a binary vector $\mathbf{b} = (b_I, b_{ab}, b_\zeta)$. If b_ξ is zero, the corresponding feature ξ is not used in the combination. Otherwise, it is used.

The method belongs to the category of cluster validity methods [21, 32, 33], which are used for measuring "goodness" of a clustering result comparing to other ones which were created by other clustering algorithms, or by the same algorithm but using different parameter values. Cluster validity is a hard problem, since the notion of "clusters" is not well defined in the general case and measuring cluster quality is subjective.

As it is shown in the flowchart of Fig. 2.11, for each combination b of features, clustering module of Fig. 2.4 is applied to determine the number of classes $K_\mathbf{b}$, followed by k-means on a block basis to obtain map U_b. Given the clustering maps for all combinations, feature selection module of Fig. 2.11 is then used for deciding the best combination of features.

First step of feature selection consists of computing a separability index for each feature combination using the combination dependent, statistical description of blocks and the corresponding clustering map. For each resulting k-means map U_b, the compactness of each cluster k is measured by the mean distance in the interior of the cluster

$$\bar{d}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} J^B(\bar{p}_k, p_i)$$

where n_k is the number of blocks belonging to cluster k, $\bar{p}_k$ denotes the mean feature vector of the cluster, p_i is the feature vector of the i^{th} block belonging to cluster k and J^B is the Bhattacharyya distance between vectors. The distance between clusters k and l is defined as the average distance $\bar{D}_{kl}$ between their samples-blocks (Eq. (2.2)). The cluster separability index for the clustering under consideration is defined as:

$$SI_{\mathbf{b}} = \frac{1}{K_{\mathbf{b}}} \sum_{k=1}^{K_{\mathbf{b}}} \frac{P_k}{\bar{d}_k} \sum_{\substack{l=1 \\ l \neq k}}^{K_{\mathbf{b}}} \bar{D}_{kl}$$

where $P_k = n_k/n$ and n is the total number of blocks.

In the general case, metric SI is higher for clustering maps with compact, well separated clusters, giving emphasis to the separability of clusters with more members or, in other words, with higher P_k. Furthermore, factor $1/K_{\mathbf{b}}$ normalizes the contribution of the number of clusters to the index value of combinations.

After the computation of SI for all combinations, combination $\mathbf{b}$ of features which leads to the clustering with maximum $SI_{\mathbf{b}}$ value is selected as the best one for segmentation. The statistical description of classes, used by the flooding algorithms, is computed using this mapping of blocks to clusters and the features that gave this mapping.

A variant of this method is to decide first if intensity, chromaticity or both of them are going to be used. Then, given the selected combination of intensity and chromaticity, the decision of including or not texture with that combination, could be performed using the cluster validity method described

above. In this work, to decide the combination of intensity/chromaticity, a multimodality test is performed. The statistical extend of each colour channel in Lab colour space is computed, using the maxima of channel histograms. Statistical extend (or strength) for each channel is defined as the distance between the first and last histogram maxima or modes. If this distance is greater than a threshold, the channel is a candidate to participate in combination, otherwise it is not. Intensity is selected if L channel is a candidate, while chromaticity participates in combination if a, b or both channels are candidates. This way, selection of intensity/chromaticity is pixel based and it is applied without using clustering information. However, due to the statistical properties of texture details, such a technique is not feasible for the second decision, i.e. for selecting or not texture.

Chapter 3

Percolation Theory and Image Segmentation

In this Chapter, we show that the solution of the image segmentation problem described in Section 1.3, being considered as a grouping process, could be viewed as a connectivity percolation process. In the first Section the main aspects of the connectivity percolation theory are extensively described, while in the second Section, the main tools of the theory are used in the classification context in order to show possible links between the the image segmentation problem and the percolation theory. Furthermore, the strong relation to the flooding algorithms of the next Chapter is also underlined.

3.1 Critical Connectivity Percolation

We denote as $\Lambda_{inf} = \{\mathcal{V}, \mathcal{E}\}$ the infinite 2D regular lattice $\mathbb{Z}^2$, where $\mathcal{V}$ is the set of vertices-sites and $\mathcal{E}$ the set of edges-bonds between vertices. We also assume that vertices are labelled according to the set $\{0, 1\}$ of labels by *Bernoulli* trials with $Pr\{v = 1\} = p$ and $Pr\{v = 0\} = 1 - p$, for each $v \in V$, $0 \leq p \leq 1$. Sites belonging to class 1 are considered *occupied* or *open*. Percolation theory [19] states that an *infinite* or *giant* connected component is *almost surely* formed for label 1 in the random lattice, if, and only if, $p > p_c$, where p_c is the so called *critical probability* of site percolation. In

that case, we say that label 1 percolates. In the considered case of 2D lattice, extensive simulations have shown that $p_c \simeq 0.59275$. In Fig. 3.1, the giant component of label 1, as well as the tiny in size components of label 0, are shown for an 8×8 lattice, with $p = 0.8$.

Several estimates have been defined to accurately measure p_c and study the properties of the connected components of the percolating label [34]. Among the known percolation *observables* Q_p are the size of the largest component and the average size of the connected components of label 1, as a function of occupation $p \in [0\ 1]$. In fact, Q_p has to be interpolated in an increasing set of values of $p \in [0\ 1]$. Observables like these, necessitate the implementation of fast algorithms to compute the connected components of occupied sites at p, since, due to the difficulty of the problem and the lack of sufficient rigorous mathematical results, *Monte Carlo* simulations are often used to measure Q_p in $L \times L$ lattices of varying dimension L, over a large number T of iterations. Two well known algorithms in connected components computation are those of Hoshen and Kopelman [35] and Newman and Ziff [36].

Newman and Ziff proposed an, almost linear cost, "73 lines C code" algorithm, to compute the percolation quantity $Q^t_{p_n}$ of interest for the set of increasing probabilities $p_n = \frac{n}{L^2}$, $n = 0, 1, \ldots L^2$, at each Monte Carlo iteration t. Beginning at $n = 0$ with a configuration of L^2 unoccupied sites in the lattice, at each step $n > 0$ one of the $L^2 - n + 1$ unoccupied sites is randomly peeked and occupied. The vast improvement is that the connected components of label 1 are not recomputed from scratch after each occupation, as it was the case in [35]. Instead, the newly occupied site u is grouped to its neighboring component, if only one such component exists. In the case that more than one connected components are in the 4-neighborhood of u, the components are unified in a new one, including u. Otherwise u is a new one-element component by itself.

Using "Union/Find" algorithms [37] to perform these operations of site grouping, the cost for a single Monte Carlo iteration becomes *almost linear* to L^2. Estimate $Q^t_{p_n}$ is then computed, using the new configuration of the connected components. The final observable estimate given as output by the

Monte Carlo simulation is:

$$Q_{p_n} = \frac{1}{T}\sum_{t=1}^{T} Q_{p_n}^{t},$$

and $Q = \{Q_{p_0}, Q_{p_1}, \ldots Q_{p_{L^2}}\}$ is an interpolated set of the observable curve through the computation time line n or occupation sequence p_n.

As it is noticed by Newman and Ziff, the occupation order could equivalently be obtained by a two steps procedure for each iteration t. The procedure consists of assigning weights $r_u \in [0\ 1]$ randomly for all u in the finite lattice and sorting the weights in increasing order at each t. Although that was a common practice before their work, sorting requires $O(L^2 \log_2 L^2)$ time, which is more than the cost of all the other operations per iteration t and according to the arguments of many other authors, this technique is proved to be useless in practice, for critical percolation simulations. Newman and Ziff proved this argument in practice, giving accurate results for known observables, as well as for new rigorous mathematical efforts in percolation.

We denote by $W(p)$ the probability of appearance of a *spanning* component, at given concentration p. As spanning we define the component that spans the finite lattice from the left to right side. The fraction $P(p)$ of the occupied sites belonging to the spanning component may be considered as a *connectivity* or *strength* measurement of the percolating system. The two quantities are depicted in the plots of Fig. 3.2(a) and Fig. 3.2(b), respectively, for three values of L. The fast Monte Carlo algorithm of [36] has been implemented and used for the computation on average of these quantities using 10^5 iterations. The relation between these curves and linear system size L, has been extensively studied in the literature. In general, the smooth phase transitions near p_c become more abrupt as L increases, in perfect accordance with the fact that an abrupt transition phenomenon would be observed, if we had an infinite system and not the simulations of it by finite size ones.

Average component size is one of the *observables* used to measure the percolation threshold and the behavior of the system very close to it. The

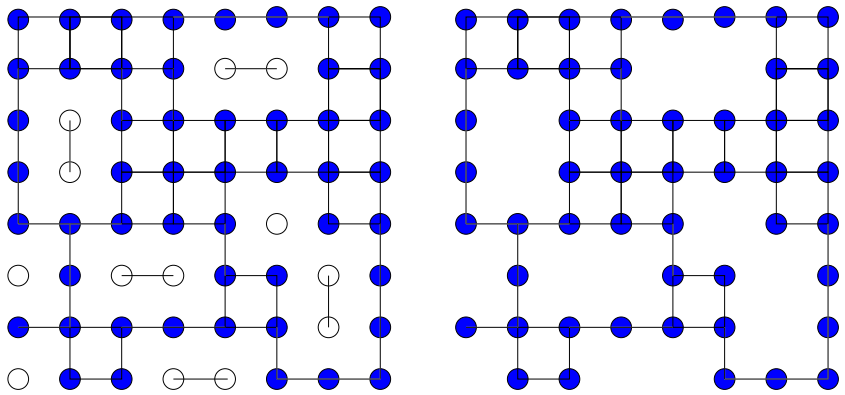


Figure 3.1: Connected components (a) and giant component (b) for $p = 0.8$.

mean component size S of the percolating label is defined as [34]

$$S(p) = \frac{\sum_s s^2 n_s}{\sum_s s n_s} \tag{3.1}$$

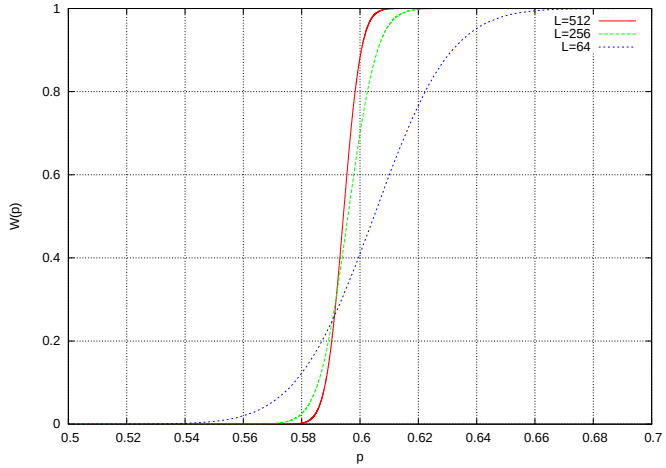
where n_s is the number of components of size s per lattice site:

$$n_s = \frac{N_s}{sL^2} = \frac{p_s}{s},$$

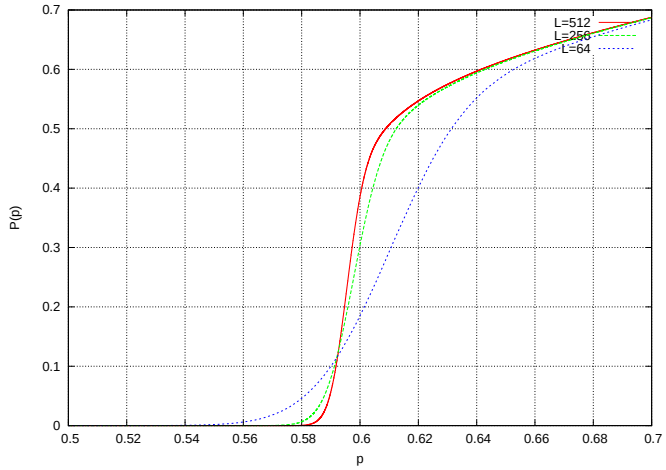
and N_s is the number of sites belonging to components of size s. To avoid the diversion of S above p_c due to the presence of the infinite component, the summation in Eq. (3.1) does not include the largest component. An alternative definition for the average component size is given by

$$S_{\mathcal{S}}(p) = \frac{\sum_s s n_s}{\sum_s n_s} = \frac{\sum_s N_s}{|\mathcal{S}|}, \tag{3.2}$$

where $|\cdot|$ denotes cardinality and $\mathcal{S}$ is the set of components of label 1. The two different definitions of average size lead to different curves, as it is shown in the plots (a) and (b) of Fig. 3.3, for $L = 512$ and 10^5 iterations. The largest component has been excluded in all cases. Furthermore, the curve of the first plot reaches its maximum at p_c and the accuracy in estimation of p_c using this heuristic gets finner as both the iterations of Monte Carlo



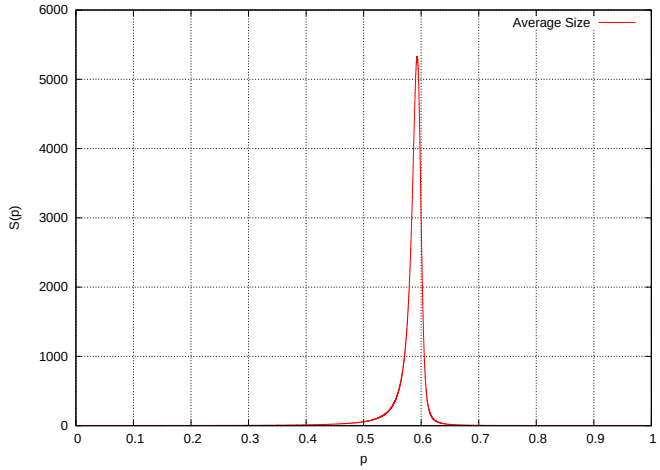
(a)



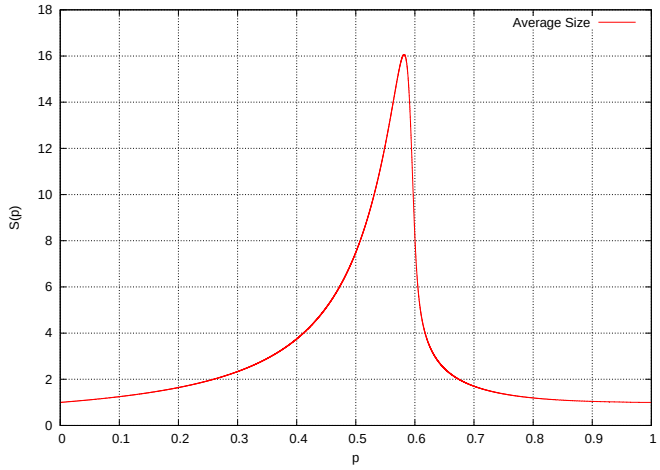
(b)

Figure 3.2: Probability of the spanning component W (a) and connectivity P of the occupied sites (b).

simulation and L increase.



(a)



(b)

Figure 3.3: Mean (a) and average (b) component size of the occupied sites.

If we measure the observables defined above for the zero class as a function of p, we see that the zero class does not survive for $p > 1-p_c$ and furthermore all the corresponding curves of class 0 are in perfect symmetry to those of 1, around the point $p = 1 - p = 0.5$. The symmetry in behavior of the two classes, indicates an uncertainty interval for $p \in (1 - p_c, p_c)$. As p increases in that interval, the average size $S_0(p)$ of 0-regions becomes smaller, while the size $S_1(p)$ of 1-regions increases and we have $S_1(p) > S_0(p)$ for $p > 0.5$.

This fact is graphically depicted in the plot of Fig. 3.4, where the probability $P(p)$ of a pixel to belong to the largest connected component of classes 0 and 1 respectively, as a function of p, is shown for $L = 256$. We use the same notation for the spanning and the largest region since, with high probability, the spanning region is the largest one.

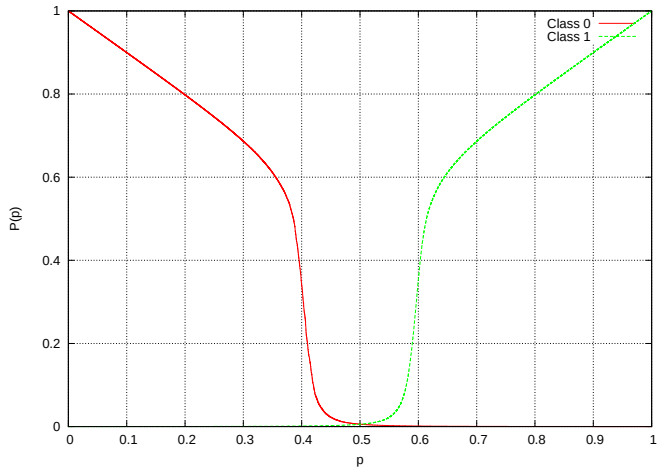


Figure 3.4: Probability $P(p)$ of the largest connected component of classes C_0 and C_1.

Furthermore, the average size of "holes" in the largest component as a function of p is maximum at p_c, due to the *topological* properties of the percolating component. It is well known that the *backbone* is an important structure of the component. It is formed by the union of all self-avoiding walks connecting two points of the lattice. The most promising model for the backbone was introduced by Stanley [38] and is known as the *nodes-links-blobs* model. The model is depicted in Fig. 3.5. As we see in this picture, the backbone consists of a network of nodes connected by "one-dimensional" links, which are often separated by multi-connected pieces or blobs. Thus, the backbone may be viewed as a topologically linear string of blobs of all possible sizes. In general, there are sites that, when removed, split the spanning component into pieces. Hence, just before p_c there are many components of label 1 with small "holes" in them, which are interconnected through *singly connected* edges to form the backbone at p_c [34]. Just after p_c the average size

of "holes" in the spanning component is maximum and gradually decreases for $p > p_c$, as the spanning component becomes a compact region. This heuristic constitutes one more observable for estimating p_c, although it is harder to be computed by Monte Carlo methods compared to the average size of components.

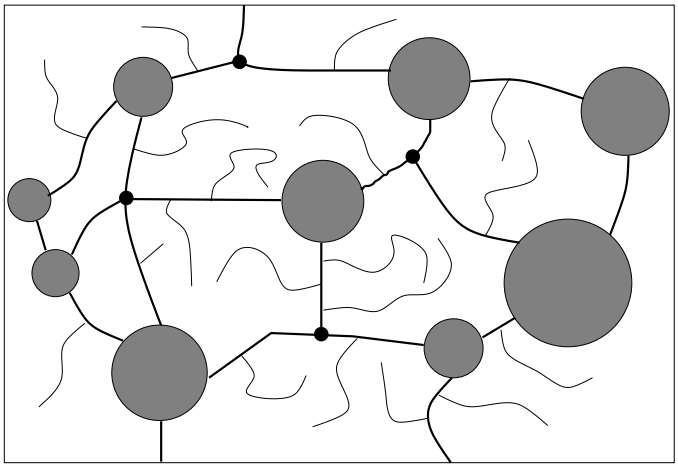


Figure 3.5: Topological model of the spanning component at p_c, composed by links which interconnect blobs (shown circular here). The crossing points of the links (nodes) as well a small number of the *dead ends* (thin lines) are also shown.

3.2 Image Segmentation and Connectivity Percolation Process

3.2.1 Percolation in a Known Region

We consider the segmentation problem defined in Subsection 1.3. According to the *Maximum A Posteriori (MAP)* criterion we may assign the pixel u to the class that maximizes Eq. (1.2). The labelling according to the MAP criterion for the *five regions* image is shown in Fig. 3.6. For this example, image intensity has been used as classification feature ξ and the pdf of classes is given by normalized histograms of intensity.

In Fig. 3.7, the same labelling is depicted in white colour per class. In each region R_m, the correct labelling is corrupted by the statistical errors due to the presence of the other classes $k \neq m$ in the feature space. Any site of region R_m is assigned to a wrong label, if, and only if,

$$\max_{k \neq m} p(\boldsymbol{\xi}_u|k) > p(\boldsymbol{\xi}_u|m).$$

This fact is graphically depicted in the plot of Fig. 3.8(a), where the pdf for an 1D feature space is shown for a class m, together with the curve of $\max_{k \neq m} p(\boldsymbol{\xi}_u|k)$ and the area of erroneous classification in the feature space. The probability of correct labelling in R_m under this setting is:

$$p_m = Pr(C_m|R_m) = \int_{\Xi} p(\boldsymbol{\xi}|m), \tag{3.3}$$

where

$$\Xi \equiv \{\boldsymbol{\xi} : \ p(\boldsymbol{\xi}|m) > p(\boldsymbol{\xi}|k), \quad k \neq m\}.$$

The probabilities $Pr(C_k|R_m)$ of erroneous assignments in R_m due to C_k, are computed the same way for $k \neq m$. This is the probability of "holes" in the region and the total probability of erroneous labelling in R_m is

$$p_e = \sum_{k \neq m} Pr(C_k|R_m).$$

Thus, the extend of the perceived error in the region, depends on the spatial correlation in Λ and the statistical modeling of features in the feature space, which affect the discrimination capability of the features themselves. The extend of error is what we observe as black "holes" in the five images of Fig. 3.7. Furthermore, in each row of Table 3.1, the probability $p_m = Pr(C_m|R_m)$ together with the probabilities of "holes" in R_m, are depicted per class. In each region the correct class occupies the majority of sites, while the other classes appear in smaller percentages. In addition, with a closer look at each region, the majority class is spread in the whole area of the corresponding region, in many connected components of all sizes compared to the size of R_m.

In fact, what we see in all regions is a snapshot of a *connectivity perco-*

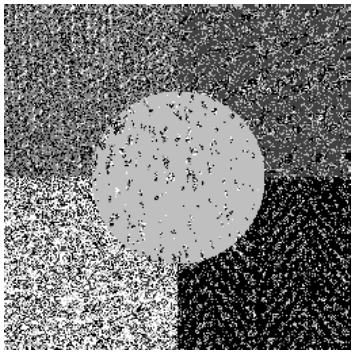


Figure 3.6: Labelling to class (a) $m = argmax\{p(\boldsymbol{\xi}|k)\}$ for the *five regions* image example.

R_m	$Pr(C_1\|R_m)$	$Pr(C_2\|R_m)$	$Pr(C_3\|R_m)$	$Pr(C_4\|R_m)$	$Pr(C_5\|R_m)$
R_1	0.4760	0.0618	0.0535	0.2426	0.1661
R_2	0.0427	0.5938	0.0194	0.1288	0.2153
R_3	0.0667	0.1477	0.4214	0.1677	0.1965
R_4	0.0545	0.0577	0.0205	0.7457	0.1216
R_5	0.0269	0.0056	0.0240	0.0386	0.9049

Table 3.1: Probabilities of class appearance per region.

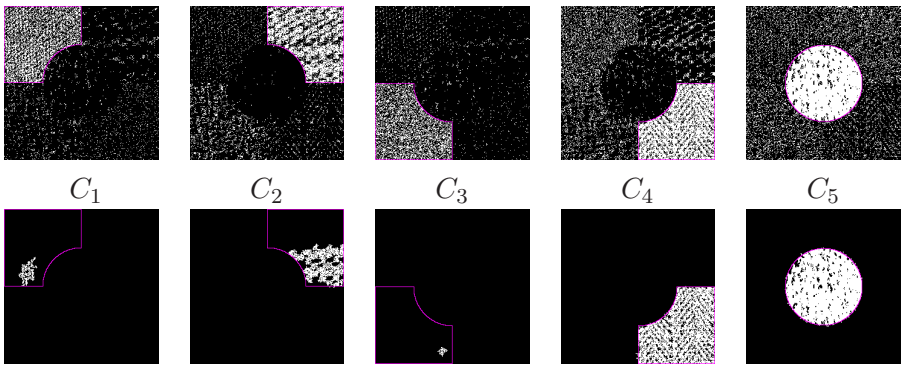


Figure 3.7: First row: map of the sites in which C_m dominates according to MAP criterion for $m = 1, 2, 3, 4, 5$. Second row: largest connected component of C_m in the image. Pink line delineates the boundary of the corresponding region R_m.

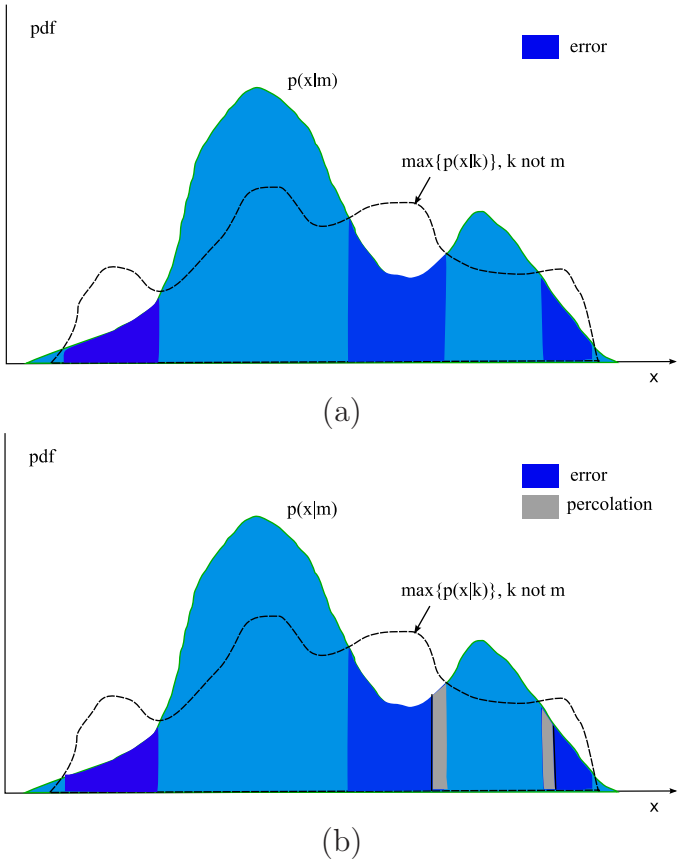


Figure 3.8: (a) Probability density function (pdf) of a class m in an 1D feature space. The error area for the class, caused by the presence of the other classes in the feature space, is also depicted in dark blue. (b) The effect of region reconnection using a percolation process is depicted in gray colour.

lation process at $p = p_m$, for the majority class m. Using the posteriors of m, a surface is constructed in R_m with values $f_u^m = p(m|\boldsymbol{\xi}_u)$, $\forall u \in R_m$. In the absence of other classes, $f_u^m = 1$ and all the sites are classified to label m in R_m. The introduction of the other classes, lowers the values of f_u^m for some u, due to the overlapping in the feature space as already explained and when $f_u^k > f_u^m$ a wrong assignment appears in R_m. If we think of a wrong assignment as a disconnection of u from its 4-neighbors, then what we see at $p = p_m$ is the "dilution" of R_m after a percentage of $1-p$ such disconnections have been applied. As stated by percolation theory, in cases where $p_m > p_c$, a giant region $\hat{R}_m$ still remains, as we observe in the images of Fig. 3.7, for the fifth region of class C_5. When $p_m < p_c$, the connectivity of R_m is more or less broken and a giant component is almost surely absent in R_m. To reconnect the region, we should perform the percolation process in the opposite direction, i.e. from $p = p_m$ towards $p = 1$, examining disconnected sites in decreasing posterior probability. The effect of this process is to move the decision boundaries displayed in Fig. 3.8(a) towards the error area according to the Bayesian rule, until the corresponding region retrieves its connectivity. This effect of boundaries movement is shown in gray in Fig. 3.8(b).

Supposing that the only knowledge available is the posteriors of classes, we could initiate the percolation process for all classes at $p = 0$. Doing so, we are in position to observe a phase transition phenomenon that appears in R_m, if we group sites to connected components using only the posteriors of class m, in a way similar to that of percolation simulations. The term "phase transition" refers to the connection of many small components to a unique, large one. To be consistent with percolation terminology, for each site $u \in \Lambda$, we compute the *resistance* or *height* of u as a decreasing function $h_u^m = h(p(\boldsymbol{\xi}_u|m))$ of the posterior probability given the class m. This way, a surface $\mathbf{h}_\Lambda^m$ is constructed for each class. To observe the phase transition, the sites in R_m are sorted in increasing order according to their h value and this ordering is used to construct the connected components of class m. At each step n, a site $u_n \in R_m$ is occupied in that order. After each occupation, we compute the connected components of occupied sites, whose resistance is less than or equal to $h_{u_n}^m$ and we monitor the mean size S of them at occupancy

$p_n = n/|R_m|$, excluding the size of the largest one. At the occupancy p_c^m of maximum S, a phase transition appears on the lattice and the construction of a unique, large component for region R_m is detected.

In figures 3.9, 3.10, the detection of the phase transition for regions R_3 and R_5 respectively is graphically depicted. In plots (a) and (b) of each figure, the average size S of connected components and the size of the largest one are respectively shown as a function of occupation. In plots (b), the phase transition probabilities p_c^m, defined to be those of maximum S are also depicted. Transitions occur at $p_c^3 = 0.58$ and $p_c^5 = 0.57$. In images (c) and (d) the largest connected component before and after transition are shown. In the first example, the heuristic detects that a giant component exists exactly when it is constructed, while in the second, a large connected component exists in R_5 before it is detected by the heuristic of average size.

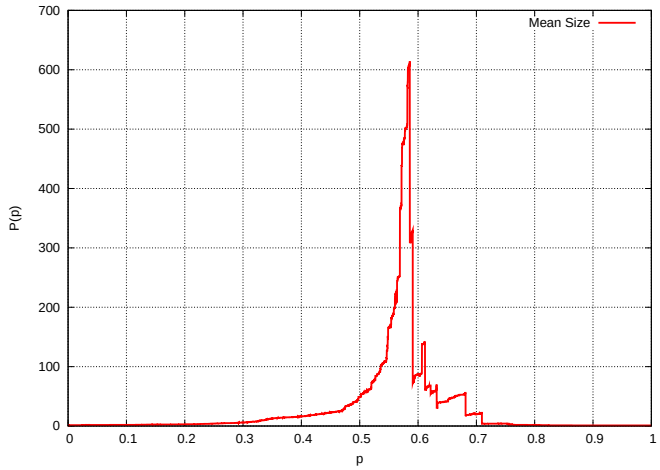
The mean size, as well as any other percolation observable, could be monitored as a function of resistance h^m, instead of occupation. In that case we could monitor the resistance h'_m at which the existence of the "giant" component is detected. In particular, we can interpolate the observable for a number N of increasing values of h,

$$\{h_1, h_2, \ldots, h_N\}.$$

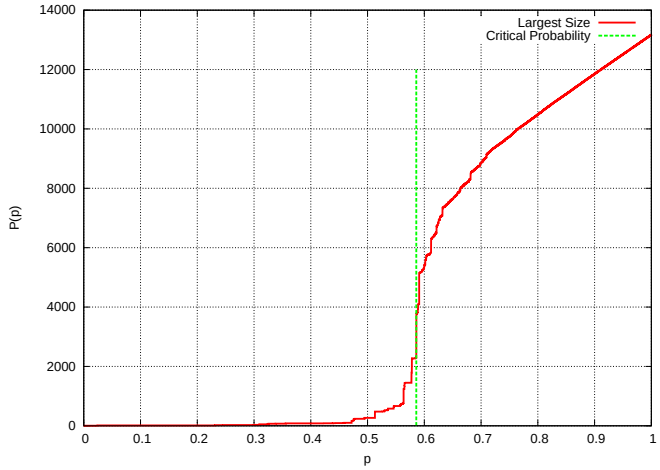
At each step n of the process the connected components of sites u with $h_u^m \leq h_n$ are constructed and the observable for h_n is computed. The probability of sites with $h_u^m \leq h_n$ is $P(h_n|m, R_m)$, where $P(h|m, R_m)$ is the *cumulative distribution function (cdf)* of the random variable h given the class m, in region R_m. Hence, equivalently, we can say that observable is estimated at occupancies

$$\{P(h_1|m, R_m), P(h_2|m, R_m), \ldots, P(h_N|m, R_m)\}.$$

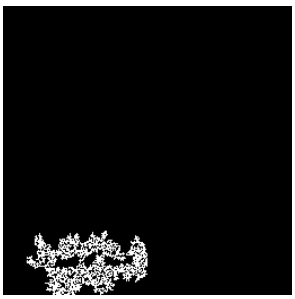
The cdf $P(h|m, R_m)$ for $h = 1 - f$ is depicted in the plot of Fig. 3.11, for the 3^{nd} region of the *five regions* image. Probability $P(h|m, R_m)$, that drives the percolation process, is an estimation of the cdf $P(h|m)$ of the random



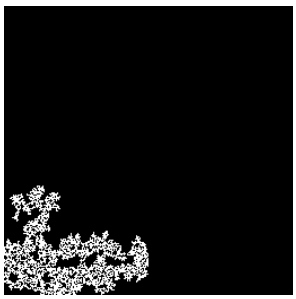
(a)



(b)

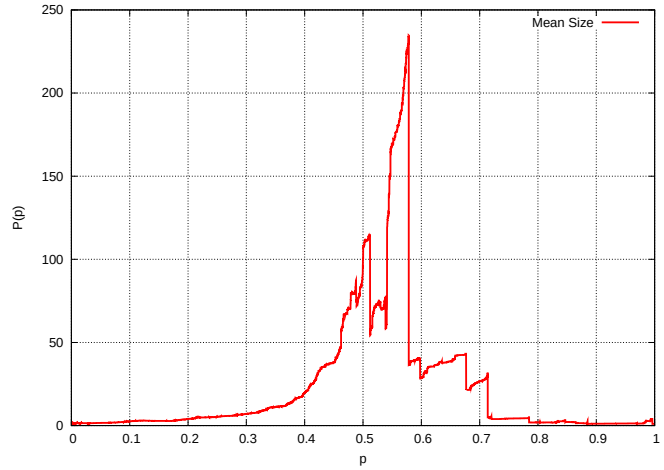


(c)

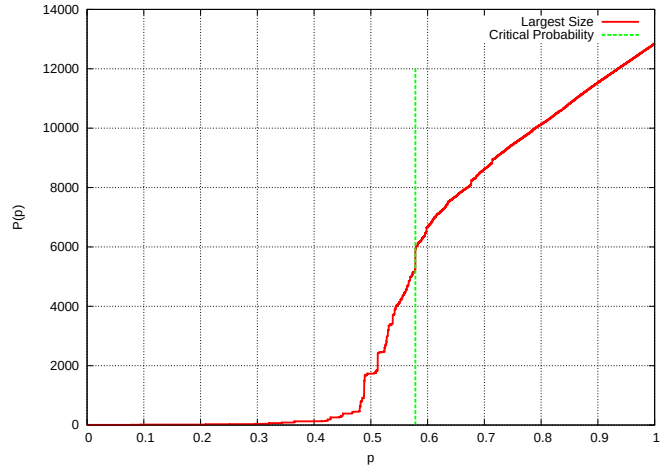


(d)

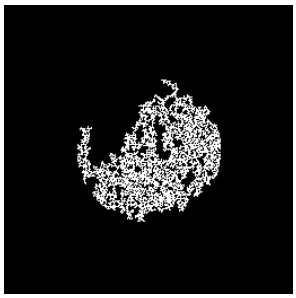
Figure 3.9: Critical connectivity phase transition for the 3^{nd} class in region R_3. In plot (a), the average size S of connected components and in plot (b), the size of the largest component are depicted as functions of occupation p. In images (c) and (d) the largest connected component before and after transition are shown respectively.



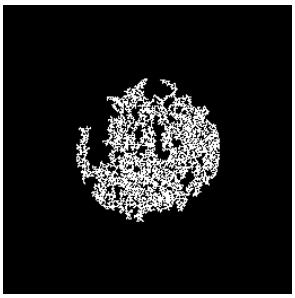
(a)



(b)



(c)



(d)

Figure 3.10: Critical connectivity phase transition for the 5^{th} class in region R_5. Description is similar to that of Fig. 3.9.

variable h given the class m. The cdf of h for each class could be computed in the feature space as well. Thus, we can detect the unique "giant" component of R_m after $p' \simeq p_c$ of its sites have been occupied, by constructing the components of sites u with $h_u^m \leq h'_m$, where h'_m is determined by the equation $p' = P(h'_m|m)$. This approach is described in the next Section, in order to detect "large" connected components of the classes in the image, without any knowledge of their location.

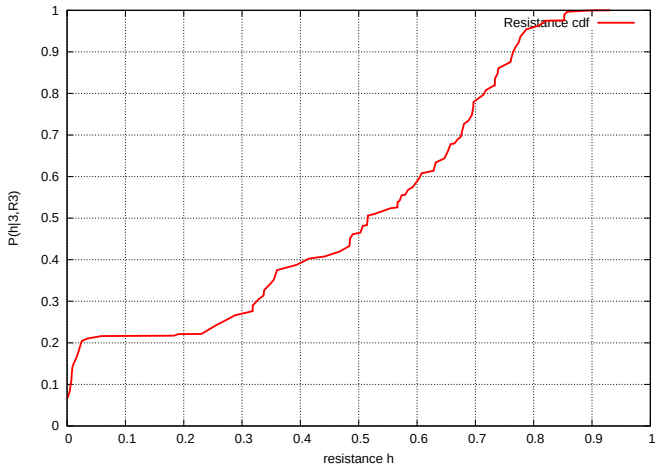


Figure 3.11: Cumulative distribution function of resistance h computed in region R_3 for C_3.

3.2.2 Initial "Large" Regions by a Percolation Process

As it is evident by the previous Section, we can retrieve large connected components of an image region R_m described by class m, if we group sites to components in decreasing order of their posterior probability. In other words, percolation process could be used to locate initial regions of high confidence per class. These initial regions are the parts of image that are better described by the corresponding class, compared to the other classes.

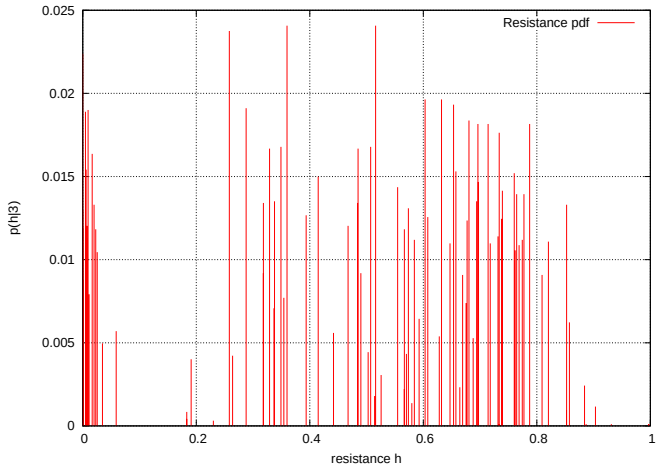
We describe a procedure to get a large region for each class, using only the surface $\mathbf{h}_\Lambda^m$, the pdf of classes and the fact that a "giant" connected component exists or has began to be formed at $p' \simeq p_c$. Given the surface

$\mathbf{h}_\Lambda^m$, we construct the connected components of class m until $p' \simeq p_c$ sites of the regions have been examined in increasing order of their h^m value. Assuming that the resistance h^m is a random variable h, the stopping value h'_m of components construction, is determined by the pdf $p(h|m)$ of resistance h given the class m, which in turn derives by the pdf of features given the class. Threshold value h'_m is computed in the feature space by the equation:

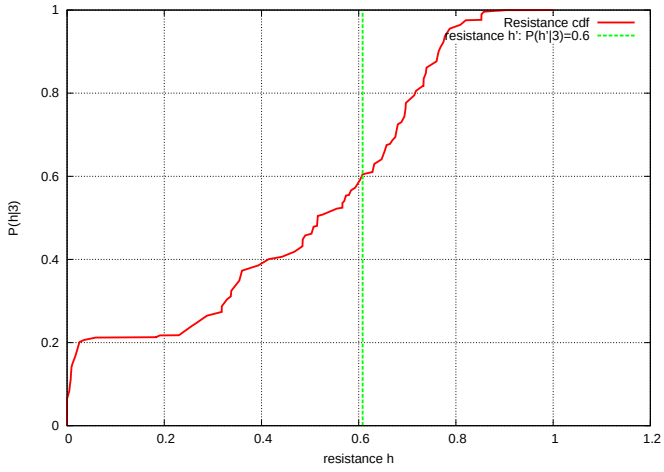
$$p' = P(h'_m|m) = \int_0^{h'_m} p(h|m)\mathrm{d}h \ ,$$

where $P(h|m)$ is the cdf of resistance h for class m. In Fig. 3.12, the pdf as well as the cdf of $h^3 = 1 - f^3$ are depicted for C_3, together with the computed stopping value $h'_3 \simeq 0.61$ for $p' = 0.6$. If we compare this cdf against the cdf $P(h|3, R_3)$ of Fig. 3.11, we note that the two distributions are almost similar, a fact that implies the similarity in components construction by the percolation process. The sites u of image with $h_u^3 < h'_3$ are shown in white colour in Fig. 3.13(a). Since the pdf of classes is a normalized histogram of intensities, the pdf as well as the cdf of h are easily obtained in that case. The same holds for a resistance that derives by Laplacian distributions of classes, as it is described in Chapter 6.

In Fig. 3.13(b), the largest components of sites u with $h_u^m < h'_m$ are depicted, for each class m, using $p' = 0.6$. Components of small size are considered to be formed due to the error that class m causes to the regions of the other classes and are ignored. The "largest" component might be considered as the initial region of high confidence, which could be then grown by flooding as it is discussed in the next chapter, under the restrictions mentioned hereafter. Until now, it is considered that each class corresponds to one region in the image. Obviously, this assumption does not hold in physical images. Instead, in many cases more than one large connected components of a class appear in an image. The procedure described above in order to extract the largest component per class, could be applied in the case of multiple regions as well, since the distribution of resistance is the same for each one of them. What is needed in addition, is a threshold to reject components that are considered "small". We demonstrate the method using the color

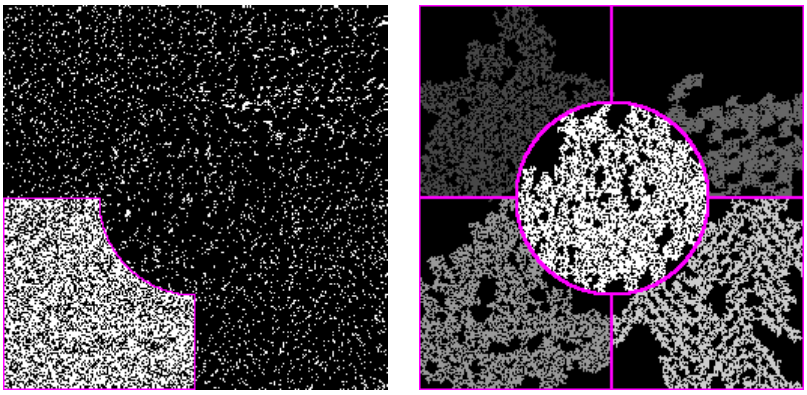


(a)



(b)

Figure 3.12: Probability distribution function (a) and cumulative distribution function (b) of resistance h computed by the corresponding pdf of class C_3.



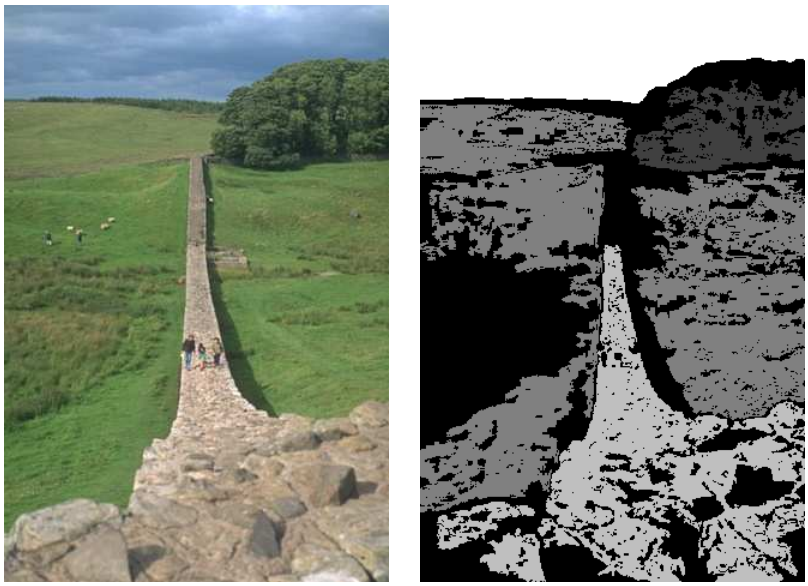
(a) (b)

Figure 3.13: (a) Sites u of image that satisfy $h_u^3 < h'_3$ and (b) largest component of sites satisfying $h_u^m < h'_m$.

image of Fig. 3.14(a). Intensity and the chromaticity coordinates (a, b) of Lab have been used in that case to classify blocks to four classes, as it is described in the previous Chapter. The pdfs of classes are computed using histograms and assuming statistical independence of intensity and chromaticity. In Fig. 3.14(b), the "large" connected components per class are shown, for $h_u^m < h'_m$. Components of size less than the 0.5% of image size have been rejected as "small".

Although in these figures almost perfect initial regions are formed, "large" components cannot be used to initialize the labelling of regions to the corresponding class m, since boundary sites of them, in the form of small or larger blobs, may be placed in the interior of other neighboring regions, thus affecting the accuracy of labelling from the beginning. The extend of this type of errors relates exclusively to the spatial correlation of the errors caused in the feature space from class m to the other classes.

Solutions to this problem are given by the flooding algorithms described in the next chapter. Obviously, the "large" components constructed by the percolation process are characterized by the maximum resistance h_{max} of their sites. If we define the cost of any path between two sites in the component to be the maximum resistance of path sites, then for all paths between two sites the cost is less than or equal to $h_{max} \leq h'_m$. This means that, if



(a) (b)

Figure 3.14: (a) Color image and (b) large components of acceptable size, satisfying $h_u^m < h'_m$.

we define a "central" site in the component, the other sites are reached by this in cost less than or equal to h_{max}. The flooding process described in the next Chapter constructs the component beginning at one, or more than one "centers" of sites that belong to the corresponding class with high confidence, which are then grown using a distance defined by the same cost.

Unlike the percolation process described in this Section, an initialization algorithm is described in the next chapter (Section 4.1) that gives initial "centers" which are placed both into the "large" component and in the interior of the whole region with high probability, while topological restrictions imposed by the existence of the neighboring regions may also be applied in flooding. In fact, all we have to do in order to get initial sites which are *surely* (with probability 1) placed on the "large" component and *almost surely* in the interior of the region, is to apply the initialization process of Section 4.1 considering only the sites of the "large" component. However, initialization for flooding could be applied for all sites in the image, without any knowledge about the large component and this is the approach that has been followed

herein.

Chapter 4

Flooding Process for Label Propagation

In this Chapter, we describe in detail the new flooding algorithms, as well as their relation to *Minimal Spanning Tree* (MST) construction and *Watershed* methods. The relations of the two last algorithmic approaches have been presented in [16]. Label initialization is discussed at first in the next Section, since it is a pre-request for both of the flooding algorithms.

4.1 Label Initialization

The output of *label initialization* is a set of spatially connected regions of pixels, which are classified to class l with high confidence, using statistical tests. For each pixel site s and class l , the distances in a disk Δ_r of radius r are averaged, resulting to the metric:

$$d_l^{SB}(s) = \sum_{z \in \Delta_r} d_l^B(s+z),$$

with

$$d_l^B(s) = -\ln \Pr\{l|\boldsymbol{\xi}(s)\} = -\ln \frac{P_l p_l(\boldsymbol{\xi}(s))}{\sum_{k=0}^{K-1} P_k p_k(\boldsymbol{\xi}(s))},$$

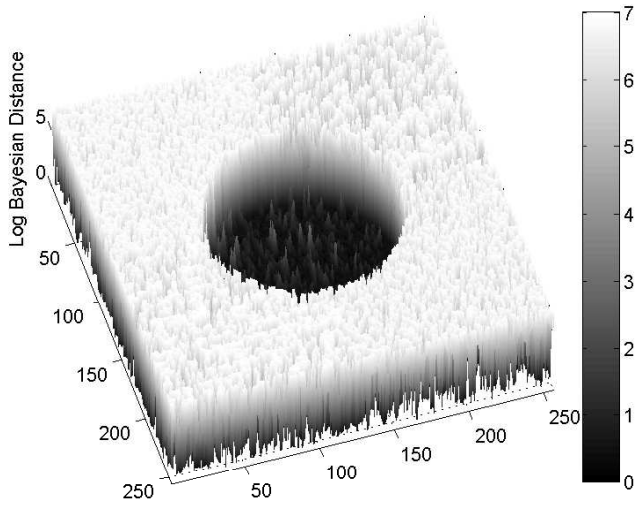


Figure 4.1: Topographic surface for the 4^{th} class of *five regions* image.

where P_k is the *a priori* probability of class k, and $\boldsymbol{\xi}(s)$ is the feature vector at site s. Distance $d_l^B(s)$ for the central circular region of *five regions* image is shown in Fig. 4.1. Label $l = 4$ has been assigned to this region by *k-means*. The profile of this distance for the row 128 of the image is depicted in Fig. 4.2. All the classes are assumed equiprobable in this case and only intensity histograms of classes are used as segmentation features, computed by the not rejected pixels of Fig. 2.8(c). Therefore, as it is clear by the two figures, for each l a topographic surface on a discrete grid is defined, considering *4-connected* pixels. The initially labelled pixels are defined to be at the zero level, while the height of the unlabelled pixels is given by the Bayesian rule. Indeed, $d_l^B(s)$ are always non negative.

After the computation of $d_l^{SB}(\cdot)$ on the surface of all classes, image pixels are sorted in ascending order according to that metric and a user given percentage of the sites with minimum average distance are retained and get labelled. Initial regions (seeds) obtained this way for the *five regions* image using $r = 10$ are shown in Fig. 4.3. These regions correspond to 1% of the pixels that belong to the corresponding class according to $d_l^{SB}(\cdot)$.

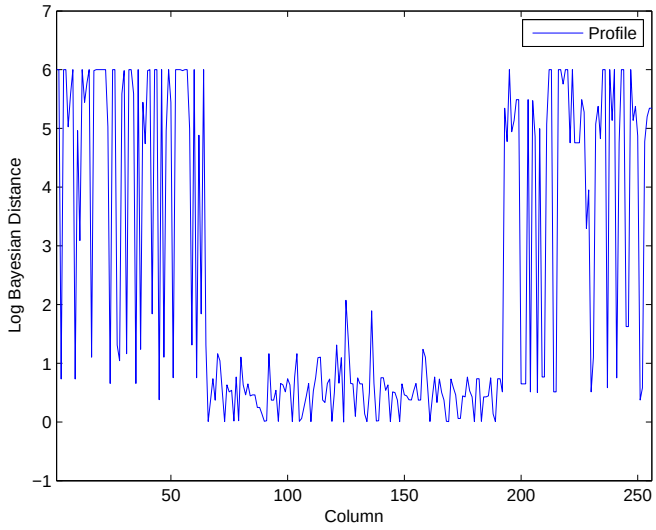


Figure 4.2: Profile of (middle) row 128 taken by the topographic surface of Fig. 4.1.

This method may also be considered as an algorithm to determine initial regions of high confidence for the construction of minimum spanning tree, for each label. Indeed, metric $d_l^{SB}(s)$ could be interpreted as the weight of the spanning subtree, which is constructed using all the pixels of disk Δ_r. Pixels s of minimum $d_l^{SB}(s)$ are placed on topographic valleys of minimum height, thus constituting the better initialization option for label flooding. The strong relation between flooding and both of MST construction and topographic computations is described in the next Subsection.

4.2 A min-max criterion for labelling

It is now assumed that an arbitrary number of pixel regions has been assigned to visual classes with high confidence. Pixels initialized this way are hard constrained to belong to the selected class, admitting that the decision is sure or almost sure. Let $S = \bigcup_{l=0}^{K-1} S_l$ be the set of initially labelled pixels. For any unlabelled pixel s we can consider all the paths linking it to a labelled set

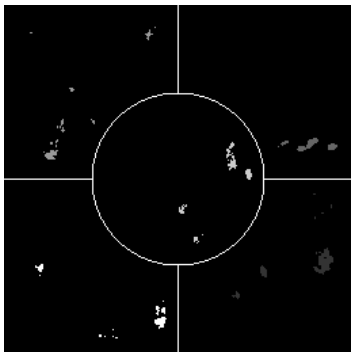


Figure 4.3: Initial regions (seeds) per class for the *five regions* image.

or region. A path $\mathcal{C}_l(s)$ is a sequence of adjacent pixels $\{s_0, \ldots, s_n\}$, where $s_n = s$ and all pixels of the sequence are unlabelled, except s_0 which has label l. The cost of a particular path is defined as being equal to the maximum cost of a pixel classification according to the Bayesian rule and along the path

$$\max_{i=1,\ldots,n} d_l^B(s_i).$$

Finally, the labelling problem becomes equivalent to search for the shortest path under the above cost, as we can define the distance of any unlabelled pixel from the different classes as being the lowest height to climb for reaching site s,

$$\delta_l(s) = \min_{\mathcal{C}_l(s)} \max_{s_i \in \mathcal{C}_l(s)} d_l^B(s_i). \tag{4.1}$$

Therefore the decisions are topology constrained.

If we consider the graph of unlabelled sites with *4-connections* and the labelled connected components, we can define an edge weight as follows

$$w(s_{i-1}, s_i) = \max(d_l^B(s_{i-1}), d_l^B(s_i)).$$

According to the cycle property applied to the MST of this graph the heaviest edge of a path belonging to the MST is lighter than the heaviest edge of any other path connecting two vertices. Therefore paths defined by (4.1) belong to the MST of the graph defined above and the computation of $\delta_l(s)$ neces-

sitates the construction of this MST. Prim's algorithm, with computational cost O(N log(N)) [39], could be used.

Under this formulation, the distance computation is one variant of percolation process, called *Invasion Percolation (IP)* [40]. In a variant of the IP itself, given the surface of resistances **h**, the percolating label begins to invent the lattice at a central site in it, following paths of minimum cost, where the cost is defined by the resistance, exactly as here. The purpose of the simulation in IP is to determine up to what value of resistance the "giant" component is formed. The distance function defined in Eq. (4.1) has to be computed in that case as well, in order to determine this maximum value of resistance. The connection between the computation of the largest component in IP and MST is due to Barabási [41]. The extended use of MST for fast IP simulations, is introduced in [42]. The flooding processes that are described in the Sections that follow, are IPs with topological constraints.

On the other hand, it is very interesting to remark that the labelling problem, as posed here, consists of constructing a topographic surface, as that for finding watershed lines [43]. Hence, we can use a region growing procedure, like the immersion (flooding) algorithm [44], for computing the above defined heights and distances and for classifying pixels, taking into account region features and topology constraints. In Sections 4.3 and 4.4 two algorithms, based on the principle of the *min-max* Bayesian criterion for labelling, are presented in detail. The main difference between them is that in the former topological constraints are limited to the above min-max criterion and the final labelling is left to a MRF-based minimization method, while in the second strong topological constraints for labelling are imposed. In both cases, initial labelled regions of high confidence have to be provided. In our approach an automatic initialization is performed as described is Section 4.1.

4.3 Independent Label Flooding Algorithm and MRF-Based Minimization

4.3.1 Independent Label Flooding Algorithm

The **Independent Label Flooding Algorithm (ILFA)** determines the optimum label as that minimizing $\delta_l(s)$. K independent flooding procedures are needed and the best label is selected. The algorithm follows the principle of *Region Growing* (RG) [9, 45], where however the growing procedure is used to compute the distances, and not to directly label pixels. Among all neighboring pixels to the set S_l, that are unlabelled and of unknown distance from label l, the nearest pixel is found, according to Equation (4.1). Growing proceeds until no more pixels can be added to the expanding regions, because their propagating contour reaches only initial pixels with different labels.

The computed distances $\delta_4(\cdot)$ by ILFA for the 4^{th} class of *five regions* image are depicted in Fig. 4.4. The profile of this surface for row 128, is illustrated in Fig. 4.5. Seed pixels of zero height are also indicated in that figure. In Fig. 4.6, the labelling of pixels to the class k of minimum $d_l^B(\cdot)$ and $\delta_l(\cdot)$, computed by ILFA are shown in images (a) and (b), respectively.

4.3.2 MRF-Based Minimization

Given the region growing measurements derived in Eq. (4.1), we then propose to optimize a discrete *MRF* in order to decide what the final labels should be. In this manner, we aim at capturing the local interactions between pixels, which will help us to refine and correct the labels that were assigned during the previous stage of our algorithm. In general, the problem of optimizing a discrete pairwise MRF can be formulated as follows: we are given a weighted graph $\mathcal{G}$ (with nodes $\mathcal{V}$, edges $\mathcal{E}$ and weights w_{sz}), and we seek to assign a label l_s (from a discrete set of labels $\mathcal{L}$) to each node $s \in \mathcal{V}$, so that the following cost is minimized:

$$\sum\nolimits_{s\in\mathcal{V}} \delta_{l_s}(s) + \sum\nolimits_{(s,z)\in\mathcal{E}} w_{sz} d^P(l_s, l_z). \tag{4.2}$$

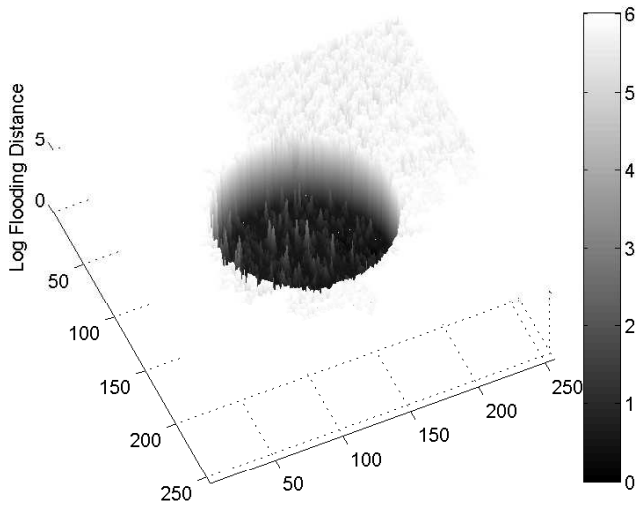


Figure 4.4: Flooding distance surface for the 4^{th} class surface of *five regions* image.

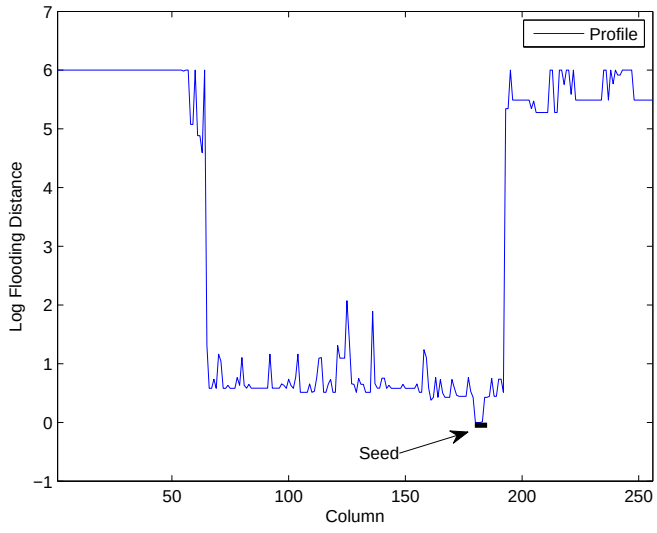
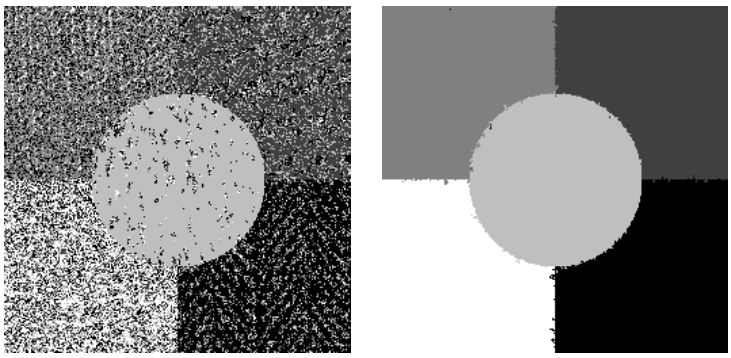


Figure 4.5: Profile of row 128 of the surface in Fig. 4.4.



(a) (b)

Figure 4.6: Labelling to class (a) $k = argmin\{d_l^P(s)\}$ and (b) $k = argmin\{\delta_l(s)\}$, where $\delta_l(\cdot)$ is computed by ILFA, for the *five regions* image example.

Here, $\delta_{l_s}(\cdot)$, $d^P(\cdot,\cdot)$ determine the singleton and pairwise MRF potential functions respectively.

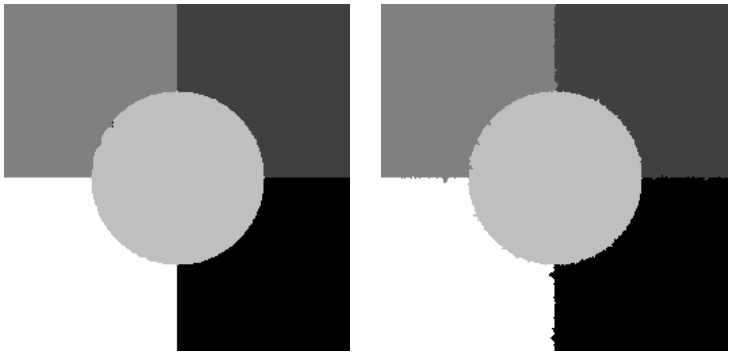
In our case, the pairwise potentials will be set according to the Potts function, i.e.:

$$d^P(l_s, l_z) = \begin{cases} 1, & l_s \neq l_z \\ 0, & l_s = l_z \end{cases} \tag{4.3}$$

Furthermore, all weights w_{sz} will be set equal to a user-specified constant.

For minimizing the MRF energy in Eq. (4.2), we will make use of the recently proposed *primal-dual* method in [46], which casts the MRF optimization problem as an integer program and then makes use of the duality theory of linear programming in order to derive solutions that are provably almost optimal. Furthermore, that algorithm proves to be a lot faster than the state of the art MRF optimization techniques, while it applies to a very wide class of MRFs.

The labelling result obtained by ILFA followed by MRF based Minimization using an *8-neighborhood* system, with $w_{sz} = 5$ in Eq. (4.2), is illustrated in Fig. 4.7(a) for the *five regions* example.



(a) (b)

Figure 4.7: Segmentation results using ILFA and MRF based Minimization (a) and PMLFA (b) for the *five regions* image example.

4.4 Priority Multi-Label Flooding Algorithm

The **Multi-Label Flooding Algorithm (MLFA)** imposes strong topology constraints. All initially labelled regions are propagated simultaneously. Hereafter we describe in detail a specific version of this algorithm, called Priority Multi-Label Flooding Algorithm (PMLFA).

In PMLFA, labels correspond to K classes of features and each class is represented by one or more spatially connected regions on image plane. These labelled regions form the initial correctly labelled pixels. Then, the contour of each initial region is propagated towards the space of unlabelled image pixels, according to dissimilarity criteria, which are based on the label and the segmentation features. Contour pixels s are sorted according to their dissimilarity $\delta_{l(R)}(s)$ from the label $l(R)$ of regions R they adjoin and at each step, a *group* of contour pixels of minimum dissimilarity are set to the label that most probably belong. Contrary to ILFA, all labels are propagated simultaneously, and pixels are directly labelled.

In addition, group labelling refers to pixels s, which are placed on region contours at an instance of the propagation progress and their metric $\delta_l(s)$ against label l is quite the same. This fact, implies the *quantization* of distance metric $d_l^B(\cdot)$, which in turn leads to the reduction of spatial redundancy that often appears during growing. Spatial coherence arises when

neighboring pixels have almost equal distance value or equivalently, as it is clear by the definition of $d_l^B(\cdot)$, when neighboring pixels have almost equal posterior probabilities for a class given their feature vector. Furthermore, two decision thresholds, namely $\mathcal{T}_l^{high}$ and $\mathcal{T}_l^{low}$, may be defined on the metric, when $d_l^B(s)$ measures the uncertainty of the decision to assign pixel s to label l. The terms *high* (resp. *low*) refer to corresponding probabilities, such that $\mathcal{T}_l^{high} < \mathcal{T}_l^{low}$. Given the thresholds, if $d_l^B(s) < \mathcal{T}_l^{high}$, pixel s belongs to l with great certainty and should be labelled as soon as it is scanned. On the opposite, if $d_l^B(s) > \mathcal{T}_l^{low}$, pixel s unlikely belongs to l and its labelling should be postponed until eventually, as growing proceeds, a better choice for s is obtained by the algorithm.

In order to incorporate the behavior just described in region growing terms, in PMLFA, contour pixels s of classes l that satisfy inequalities $d_l^B(s) < \mathcal{T}_l^{high}$, $(0 \leq l < K)$, are inserted in *high priority* simply connected lists (HPL_l) once they are scanned. Similarly, pixels s for whom $d_l^B(s) > \mathcal{T}_l^{low}$, are inserted in *low priority* simply connected lists (LPL_l). By contrary, the rest of contour pixels are inserted in *normal priority* simply connected lists, denoted as $NPL_{l,i}$ $(0 \leq l < K)$, each one corresponding to the i^{th} quantization interval of dissimilarity criteria values, according to a given quantization step Δ_Q:

$$i = Q(d_l^B) = \left\lfloor \frac{d_l^B - \min\{d_l^B\}}{\Delta_Q} \right\rfloor \tag{4.4}$$

In that way, an array of simply connected lists is enough to keep the dissimilarity information of pixels, resulting in a sorting procedure which resembles the *non-comparison sorting* algorithms (Chapter 8 of [39]). The quantization intervals as well as the priority decision thresholds are depicted graphically in Fig. 4.8.

At each step, first the items of HPL_l for $0 \leq l < K$ (if any) are popped and assigned to the corresponding label. Otherwise, if high priority lists are empty, the items of lists $NPL_{l,i}$ of minimum i are popped and get labelled. Last, if normal priority lists are empty too, the items of lists LPL_l are examined and processed by the algorithm.

Then, for each *safe*, popped and labelled pixel z, its unlabelled neighbors

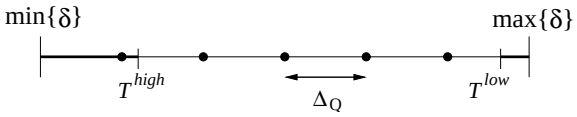


Figure 4.8: Quantization of dissimilarity metric and low/high priority thresholds.

are considered and are inserted in the corresponding list if they are not already in one of them. Safety of z tests the local topology of decisions in the neighborhood of z, at the time that z is assigned to label k. In the current implementation, z is assumed as safe pixel, if, and only if, it adjoins labelled pixels of label k and only of that.

Depending on the application, one of the two or both priority thresholds may not be used, where, in the later case, all the unlabelled pixels are handled by the normal priority lists. Furthermore, if $[\min\{d_l^B\}, \max\{d_l^B\}]$ is an interval of interest in the domain of d_l^B, we may set $\mathcal{T}_l^{high} = \min\{d_l^B\}$, $\mathcal{T}_l^{low} = \max\{d_l^B\}$ and quantize only that interval.

The computational cost of the proposed algorithm is (subject to a constant factor)

$$T = ((1 - \phi)I_Q + \phi)N_U,$$

where N_U is the number of initially unlabelled pixels, I_Q is the number of quantization intervals and ϕ is the percentage of initially unlabelled pixels that are inserted in *low/high priority* lists. From Equation (4.4) I_Q is a function of Δ_Q and since I_Q is kept fixed, the computational time T of the growing process becomes a linear function of N_U only. In addition, the constant factor of T decreases as ϕ increases.

In what follows, we describe the labelling of initially unlabelled pixels using PMLFA, where low and high priority thresholds are used on the dissimilarity metric. In order to define priority thresholds $\mathcal{T}_l^{high}$, $\mathcal{T}_l^{low}$ and forgetting for a while the competitive nature of growing, we note that according to MAP criterion, a pixel s surely belongs to class l, if $\Pr\{l|\boldsymbol{\xi}(s)\} > 0.5$, while it does not belong to l, if $\Pr\{l|\boldsymbol{\xi}(s)\} < \frac{1}{K}$. On the other hand, each initially

unlabelled pixel s_i placed on a path τ denoted as

$$\tau = s_0 \rightarrow s_1 \rightarrow \cdots \rightarrow s_n,$$

which connects the seed pixel s_0 with the boundary pixel s_n of label k, recursively satisfies the following condition

$$k = \arg\min_{l} \left\{ \max_{\{s_i \in \tau\}} d_l^B(s_i) \right\}. \tag{4.5}$$

Thus, in accordance with the ascertainment made above for the MAP criterion, the final labelling of RG will not change if we set $\mathcal{T}_l^{high} = \ln 2$. Indeed, if a region of label k reaches pixels s with $d_k^B(s) < \ln 2$ during growing, then those pixels surely belong to class k, without violating Eq. (4.5) and could be assigned to k as soon as possible. On the opposite, threshold $\mathcal{T}_l^{low}$ relates to pixels s that unlikely belong to regions of class l, which reach them during growing. This pixel category consists of *boundary* and *hole* pixels and according to the Bayesian criterion, we could set $\mathcal{T}_l^{low} \geq \ln K$.

The segmentation result obtained by PMLFA for the *five regions* image is shown in Fig. 4.7(b). 50 bins were used for height quantization in $\ln(50)$ intervals. We also set $\mathcal{T}_l^{high} = \ln(2)$ and $\mathcal{T}_l^{low}$ to a high value.

Chapter 5

Texture and Colour Segmentation

In this chapter, we describe the application of our general framework to typical segmentation problems, such as intensity, colour and texture segmentation. In each case, statistical clustering methods are used to obtain an initial map of high confidence labelled, pixel regions. The probabilistic dissimilarity metrics for the expanding regions are defined according to the considered segmentation problem. PMLFA or ILFA followed by MRF based minimization are applied as the last step of our framework, to give the final segmentation solution map.

Our general segmentation framework, permits the usage of texture, luminance and colour data either on their own, if the corresponding features are sufficient to discriminate the classes, or in combination to get improved segmentation results.

In Section 5.1, the features used for segmentation are described. Segmentation results of the proposed methods on synthetic texture images are given in Section 5.2. In Section 5.3, the segmentation framework is applied on the well known Berkeley Segmentation DataSet and performance evaluation results as well as the results of comparison against freely available segmentation algorithms are reported. In the last Section of this Chapter, detection of face colour areas is described using colour histograms and PMLFA.

5.1 Texture and Colour Analysis

Many segmentation algorithms have been proposed in the recent past in order to provide efficient tools for colour/texture based region extraction in still images. In our work we mainly use a Discrete Wavelet Frames (DWF) filter bank [25], which appears to give a good scale-space image analysis.

Wavelet frames representation decomposes the image into orthogonal texture components in different scales and orientations. The translation invariance of the representation is a desired property, when quite precise boundary localization is required. However, in cases where texture analysis in more orientations is needed, a Gabor filter bank is used herein as well.

Methods that combine texture and colour information for segmentation have been proposed in the literature [47] using Hue-Saturation-Value colour space and primitive texture features, which are classified by point-wise expectation maximization.

Texture analysis gives a set of K_ζ detail components for each pixel s, $\boldsymbol{\zeta}_s = \{\zeta_{i,s}, 1 \leq i \leq K_\zeta\}$. The chromaticity coordinates $\mathbf{c}_s = (a_s, b_s)$ of the *Lab* colour space are used for colour representation, when colour information is taken under consideration, because this representation has good uniform colour space properties.

Texture details of each class l, $0 \leq l < K$, are assumed to be zero-mean, generalized Gaussian distributed and uncorrelated and they are represented by the variance of the texture components of pixels belonging to label l, while intensity and colour of the class are represented by 1D and 2D histograms of intensity and chromaticity of classes pixels, respectively. In each case, the corresponding numbers of histogram bins K_I, K_{ab} may be provided by the user or the method described in Subsection 2.1.1 could be used, to automatically estimate distributions.

Features of classes are estimated by the feature classification method described in Chapter 2. Eq. (A.5) is used to measure the statistical distance of the K_ζ high frequency components of texture blocks, while Eq. (A.2) is used for dissimilarity measurement of intensity or colour features of image blocks. The block sets obtained by *k-means* are used for the calculation of

the prototype feature vectors, as it is described in Subsection 2.3.4.

5.2 Texture Segmentation

Image	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
Texture Analysis	3	3	4	4	4	4, 6	3	3	3
Block Dimension	32	32	32	32	32	16	32	16	16
Block Overlapping	16	16	16	16	16	0	16	0	8

Table 5.1: Classification parameters for images from [5] (p. 300)

Image	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	Mean
Error (PMLFA)	0.4%	0.28%	0.41%	0.36%	0.44%	0.44%	0.54%	0.32%	0.49%	0.41%
Error (ILFMA)	0.04%	0.04%	0.07%	0.04%	0.04%	0.1%	0.11%	0%	0.05%	0.05%
Error (MLFM)	1.65%	1.32%	1.36%	1.58%	1.66%	1.72%	1.83%	0.9%	0.87%	1.43%

Table 5.2: Error percentage results on images from [5] (p. 300)

In [48] are reported results of texture classification on nine synthetic texture mosaics using various techniques mainly based on frame representations. We illustrate experimental results on these images to evaluate the performance of our method.

Intensity and texture are used in unsupervised clustering of the images. The classification parameters for texture analysis and block dimension and overlapping extend that are selected in each case are given in Table 5.1. DWF analysis is used for all images except (f), for which a Gabor filter bank is employed. Texture scale levels used are shown in Table 5.1 for DWF analysis and the number of scales and orientations for the Gabor analysis of image (f). For each image except (f) all the intensity values are used to compute the 1D intensity histograms. In the case of (f), intensity values are quantized in 180 bins, using *k-means*. Furthermore, edge based block rejection is performed only for image (f). In block clustering (Subsection 2.3.4), the most centrally located block (centroid) of each class is used as "mean" feature vector, at each iteration of *k-means*, leading to a great reduction of *k-means* iterations

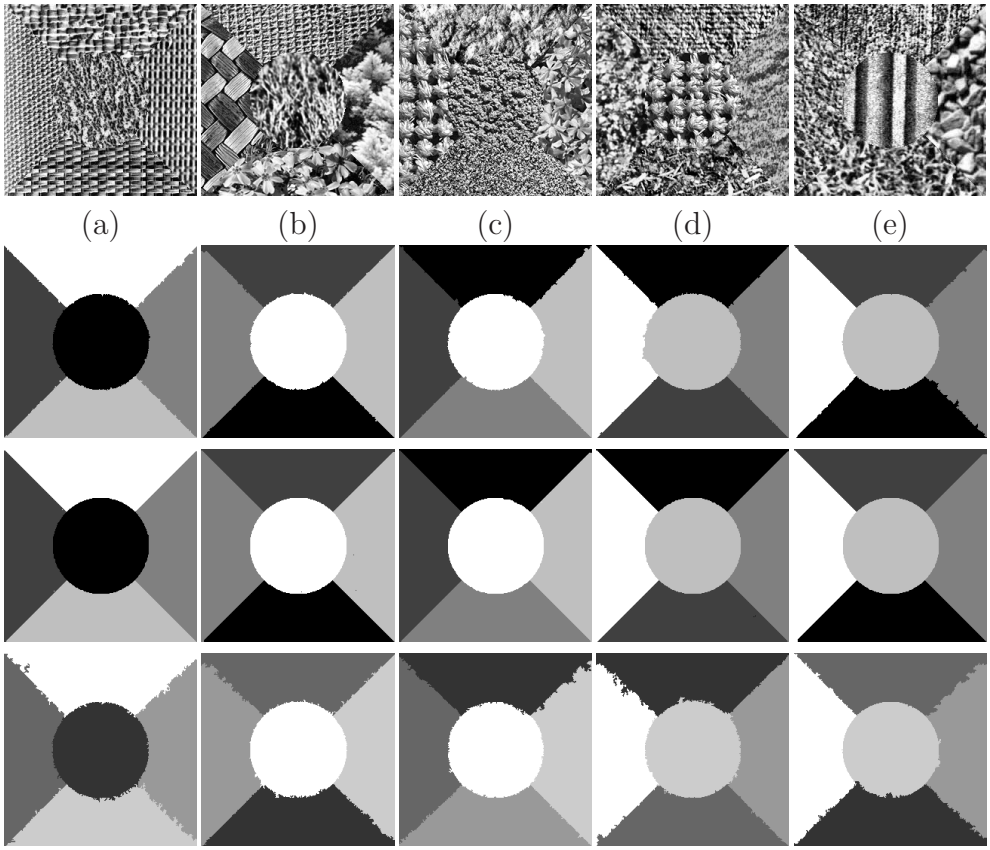


Figure 5.1: Texture segmentation for the 5 natural textures of the first row. From the second to fourth row: segmentation results using PMLFA, ILFMA and MLFM respectively.

(less than 5), although computation of centroids makes each iteration more time consuming.

In a previous work [1], segmentation results were obtained for the texture images of [48] using Multi-label Fast Marching (MLFM) algorithm. Results were presented using both unsupervised and supervised features computation. Our results are summarized in Table 5.2 for PMLFA and MRF based Minimization (ILFMA) and compared to those of MLFM. The original images and the corresponding segmentation results are given in Figures 5.1, 5.2, 5.3 and 5.4.

In all cases 1D histograms of all the intensity values are used to model the statistics of classes. Initial regions are extracted by the 1% of the pixels of high confidence per class, using a disk of radius 10 per pixel. Keeping the initialization percent low, is crucial in order to avoid erroneous segmentation results from the beginning, specially in the presence of many classes. Furthermore, the results indicate that the new algorithms are insensitive to the size of seeds. MRF based minimization is performed using an *8-neighborhood* system for all images except (h) and (i), for whom a *4-neighborhood* system is considered. In all cases we set $w_{sz} = 5$. PMLFA is applied by quantizing heights in 50 bins of ln(50) intervals and setting $\mathcal{T}_l^{high} = \ln(2)$ and $\mathcal{T}_l^{low}$ to a high value.

We also give approximate running times in the case of 5 classes images of Fig. 5.1. The methods are executed on an Intel Centrino 1.6GHz machine with 1GB RAM, running under Linux 2.6.24OS. The computational time of the overall framework is mainly consumed by the automatic feature extraction stage and the most time consuming operations are the distance computation between block pairs and k-means. The overall running time before flooding and ILFMA is about $5 - 10$ sec. The running time of PMLFA is less than 0.05 sec and the execution time of ILFA per label is of the same order, about 0.1 sec. MRF minimization is performed in about 1 second.

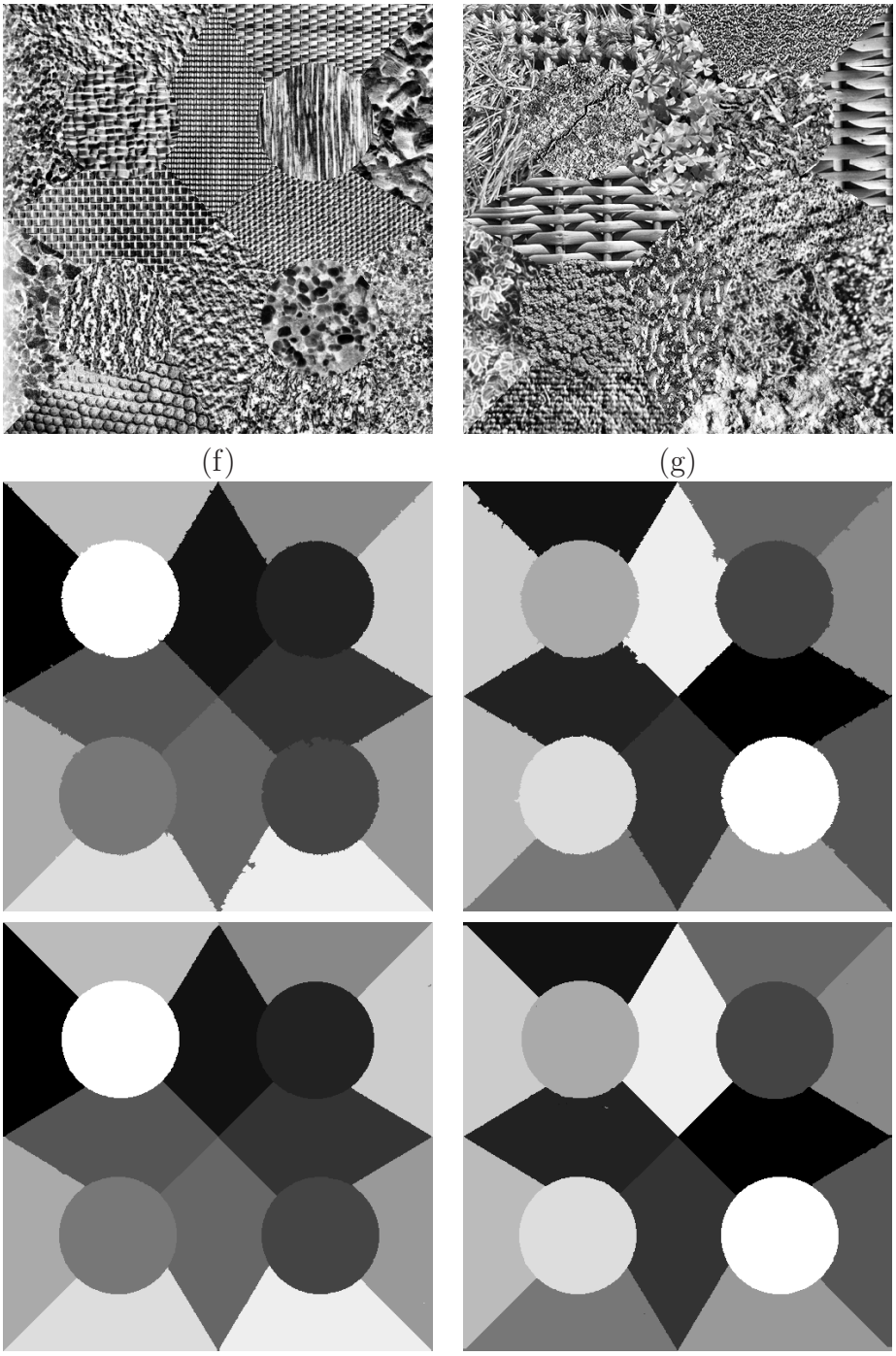
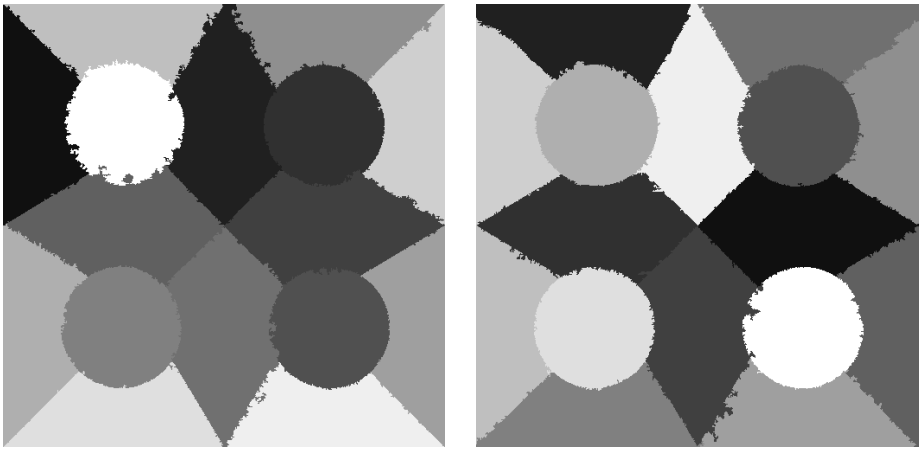


Figure 5.2: Texture segmentation for 16 natural textures. In second and third rows: segmentation results using PMLFA and ILFMA respectively.



(f) (g)

Figure 5.3: Texture segmentation for 16 natural textures using MLFM [1].

5.3 Evaluation on Berkeley Segmentation DataSet (BSDS)

In this Section, several experiments towards the performance evaluation of the segmentation framework on natural images are presented. Experiments were performed on the Berkeley Segmentation DataSet (BSDS) [2]. BSDS consists of natural colour images, of various degrees of colour and texture complexity, divided in a "train" (TR) and a "test" (TE) subset. We treat the two subsets as a unique dataset, since, in the case of clustering and flooding algorithms, tuning of parameters does not involve any training to a specific set of images. Furthermore, the freely available ground truth segmentation maps for each image of BSDS, make possible the performance evaluation of segmentation results and the comparison against other segmentation algorithms using the supervised metrics of Appendix B. However, performance evaluation using an unsupervised metric, in the sense that no ground truth data is required, has been also conducted in Subsection 5.3.1.

The purpose of these tests was twofold: first, to quantitatively measure the impact of parameter tuning on the accuracy of segmentation results and second, to determine the performance of the framework when parameter val-

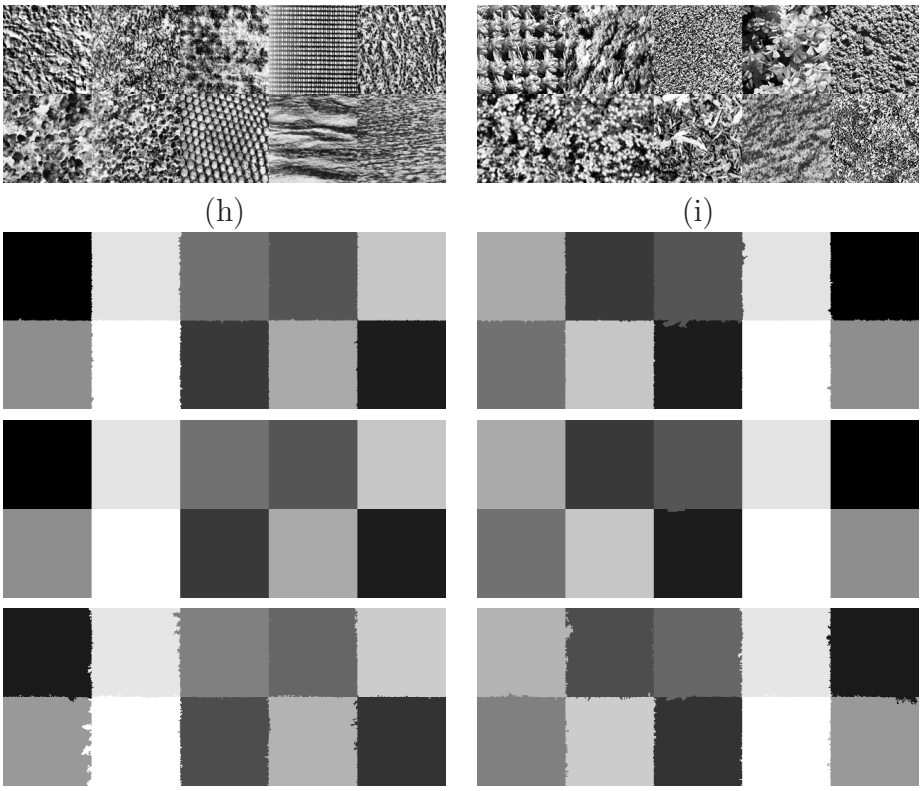


Figure 5.4: Texture segmentation for 10 natural textures. From the second to fourth row: segmentation results using PMLFA, ILFMA and MLFM respectively.

ues are kept fixed or they are automatically adapted to visual content, using the techniques of Chapter 2. Hence, in Subsection 5.3.1 supervised clustering and segmentation are performed in order to determine the sensitivity of the segmentation framework to parameter tuning. The performance of segmentation when important clustering parameters are automatically computed is quantitatively evaluated in Subsection 5.3.2. Finally, in Subsection 5.3.3 we present quantitative comparison results of the proposed segmentation framework against the colour-texture based JSEG [49] algorithm and the colour based Felzenszwalb and Huttenlocker (FH) [13] algorithm.

Performance evaluation shows that merging is an important step of our framework, which leads to improved results when segmentation is performed automatically. Merging methods are performed on the segmentation results of flooding algorithms. Merging is used to eliminate "holes" whose size is smaller than a threshold. Such "holes" appear mainly in the segmentation map of ILFMA and they are merged with their surrounding region. Furthermore, neighboring regions of classes, whose boundary energy is low, are also merged after "holes" elimination. The aim in that case is to merge neighboring regions of different classes, when no physical boundary really exists between them. Typically, such boundaries appear due to oversegmentation of image areas which undergo gradual transitions or, more generally, their feature description does not follow the statistical modeling of visual data described herein. Before that, merging of the two regions with minimum Bhattacharyya distance of their statistical description is also applied, when this distance is lower than a threshold.

5.3.1 Parameter Determination

In the experiment described in this Subsection, supervised segmentation is performed, without merging. Segmentation results using intensity, colour and texture features to characterize classes, are given in Figures 5.5, 5.6. The classification parameters used in each case are given in Tables 5.3 and 5.4 respectively.

Feature selection in block classification plays an important role in the

overall system. The result of Fig. 5.5(a) for example, could not be achieved, without the usage of texture details. On the other hand, since texture analysis involves computations in a window for each pixel, object boundaries are sometimes less accurately localized, as it is the case in Fig. 5.5(a). However, as it shown in the next Subsection, automatic segmentation for the same image gives a more accurate result, because, in that case, intensity is automatically selected as segmentation feature and regions of the clouds are merged by the boundary based, merging method. Initial regions are extracted by the 5-50% of the pixels of high confidence per class, using a window of dimension that ranges from 11 to 31 pixels. For each image, the segmentation results using PMLFA are illustrated. PMLFA is applied by quantizing heights in 50 bins and setting $\mathfrak{T}_l^{high} = \ln(2)$ and $\mathfrak{T}_l^{low} = -\ln(0.2)$. The curves of pink colour in the images of those Figures, delineate the boundaries between the regions of classes.

As it is obvious by the results, our effort is to adequately describe general feature classes which lead in a meaningful region decomposition. Furthermore, in most of results in Fig. 5.5, regions which correspond to objects, or classes of objects, have been successfully extracted, since the automatically computed statistics of the selected cues are enough to describe their class as well as to distinguish them from their background.

Considering the clustering parameters used for each image, less than 10 classes are needed and only 2 different block dimensions are used in practice. Block dimension relates to image and object size. What varies in each case is the combination of features used and the parameters of their statistical description. These observations have been taken under consideration, for parameter tuning towards the automation of the segmentation framework, as it is described in the following Subsection.

As BSDS is extensively used in literature, it is possible to compare different algorithms, although totally acceptable objective criteria are not really defined. In a recent article [50], a review of criteria for unsupervised image segmentation evaluation is presented. The sum of layout and region entropy is a simple criterion with good properties when applied for fixed number of regions. Recently Ding, Ma and Chen [15] published results on a subset of

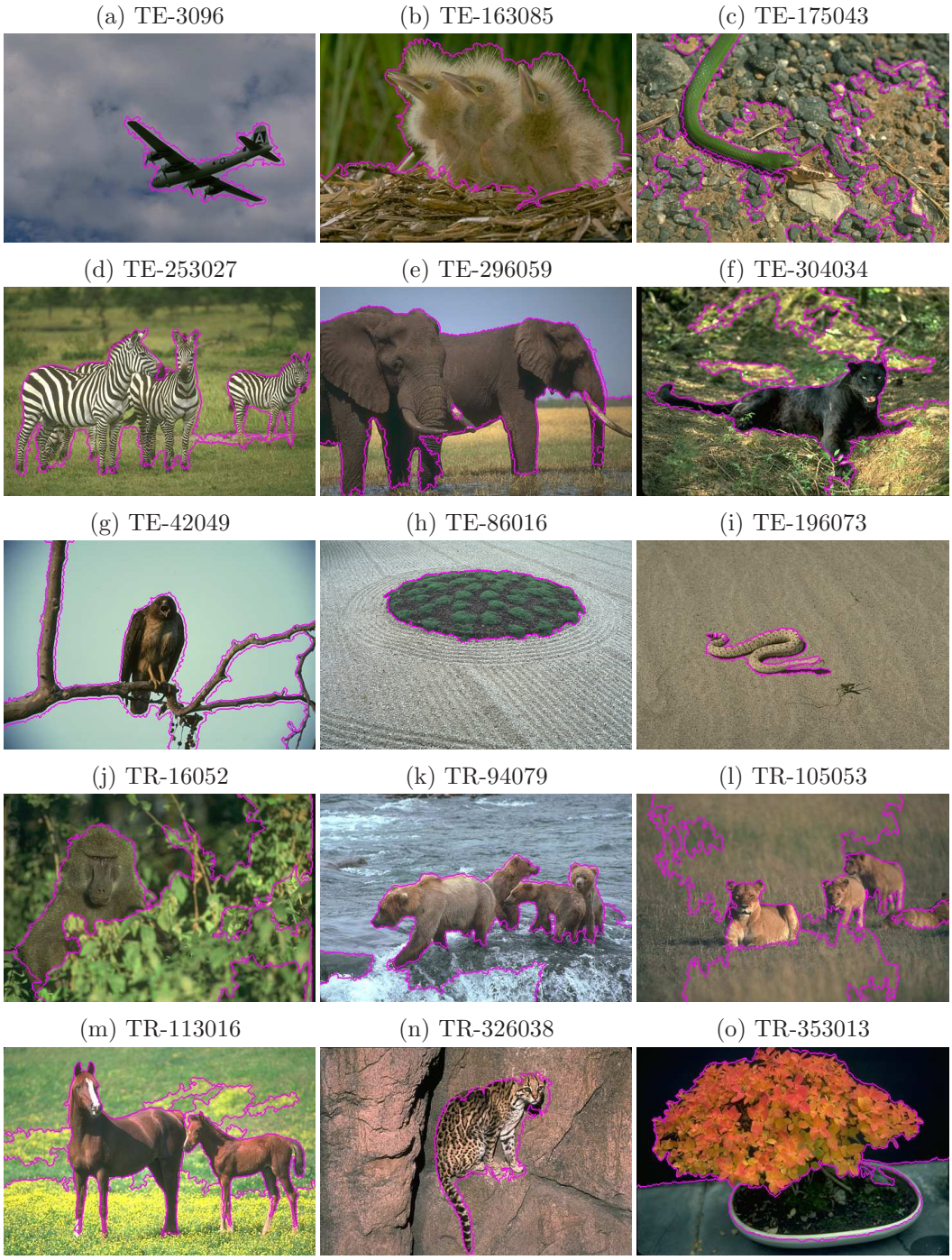


Figure 5.5: Segmentation results for a part of the Berkeley Segmentation Dataset [2], using PMLFA. Results (g), (h) and (i) are used in entropy criterion comparisons.

Figure 5.6: Segmentation results for the Berkeley Segmentation Dataset (cont.). Results (i)-(l) are used in entropy criterion comparisons.

Image	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	(m)	(n)	(o)
Classes	2	3	3	2	3	3	2	2	2	4	3	3	3	2	4
Intensity Bins	256	32	256	32	32	32	16	16	16	32	32	32	32	32	32
Chromaticity Bins	0	64	256	64	32	64	64	64	64	64	64	64	64	64	64
Texture Analysis	1	2	0	3	1	0	1	1	1	3	0	0	0	3	0
Block Dimension	32	32	32	32	32	32	16	16	16	32	32	32	32	32	32
Block Overlapping	16	16	16	16	16	0	8	8	8	16	16	16	16	16	16

Table 5.3: Classification parameters for images of Fig. 5.5.

Image	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)
Classes	4	5	5	5	4	4	4	4	2	7	3	3
Intensity Bins	32	128	64	16	32	32	32	32	16	16	16	16
Chromaticity Bins	64	256	64	16	64	128	64	64	64	64	64	64
Texture Analysis	0	0	0	2	2	2	1	0	1	1	1	1
Block Dimension	32	32	32	32	32	32	32	32	16	16	16	16
Block Overlapping	0	16	16	16	16	16	16	16	8	8	8	8

Table 5.4: Classification parameters for images of Fig. 5.6.

BSDS shown that their method (CCTA) gives better segmentation on this subset according to the entropy criterion than that obtained by the normalized cut algorithm [12] and the algorithm of [13]. We give in Table 5.5 our result and that of [15] on the subset of images where the comparison is possible. We obtain a better result on 11 over 16 images. C is the number of regions, as defined in [15] for the CCTA and as the number of the resulting connected components for our algorithm with K classes.

5.3.2 Performance Evaluation vs. Humans

The proposed methods have been tested against the ground truth maps given by humans for each image of BSDS. The metrics used for the comparison against human provided maps are: the Probabilistic Rand (PR) index and Normalized PR (NPR) index, as they are defined by Equations (B.3) and (B.4) respectively, and the region differencing measures LCE and GCE. The definitions and the properties of these metrics are given in Appendix B. For each image, algorithm and metric, the segmentation result of each algorithm is compared against each one of the ground truth results $\{S_k\}$ that

Image	Image Entropy	Number C CCTA	Entropy Segmentation CCTA	Number K PMLFA	Number C PMLFA	Entropy Segmentation PMLFA
3096	4.1849	2	4.3810	2	2	4.3065
14037	4.7252	5	5.3142	4	5	5.1935
24063	5.0626	5	5.6329	4	5	5.4656
55067	4.7245	6	5.0916	6	6	4.9650
60079	4.2383	3	4.6981	3	3	4.6211
118035	4.3159	4	5.1829	3	5	4.5650
119082	5.2815	26	6.6993	9	25	7.1583
135069	3.9342	2	4.1646	2	2	4.0094
159091	5.3569	3	5.5587	3	3	6.0035
167062	3.2160	3	3.5814	3	3	3.3224
176035	5.2215	4	5.6635	4	4	5.8649
198023	4.9806	9	5.6459	7	10	5.8267
241004	4.8029	10	5.7628	6	9	5.6119
253036	5.0771	3	5.2776	3	3	5.2598
271031	4.9824	2	5.1025	2	2	4.9903
374067	4.8794	7	5.6148	4	7	6.0724

Table 5.5: Comparison using the entropy criterion.

have been provided by humans, for the image. The median of metric scores for set $\{S_k\}$ is used to measure the performance of the algorithm on that image.

In the first experiment, which is termed "Case Study I", the K_I and K_{ab} intensity and chromaticity bins respectively, are automatically computed using the method of Subsection 2.1.1. DWF texture analysis in one level of details is performed, since it is found enough for this dataset. Clustering as well segmentation parameters have been provided by hand in each case.

Compared to the first experiment, in the second one, called "Case Study II", the number of classes is automatically determined in addition, by the method of Subsection 2.3.2. In "Case Study III", the method described in [30] is used to automatically obtain clustering maps. Furthermore, automatic feature selection takes place per image using the second method of Subsection 2.3.5, and thus, all important parameters of clustering are automatically determined, while all the other parameters are set to values shown in Table 5.6 for the overall dataset. Automatic selection of features involves six of the seven combinations of intensity, colour and texture, excluding the case of selecting texture details on their own. Finally, merging techniques are applied on the segmentation result of flooding algorithms in that case.

The results of performance evaluation for each experiment are depicted

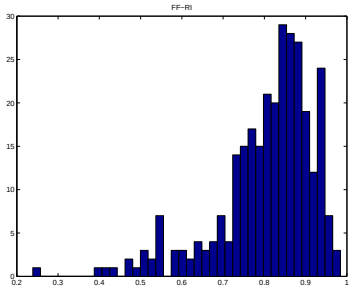
Block Dimension	32
Block Overlapping	16
Initialization Disk Radius	10
Initialization Percentage	1%
$\mathcal{T}_l^{high}$	ln(2)
$\mathcal{T}_l^{low}$	∞
MRF neighborhood	8
MRF tolerance w_{sz}	2

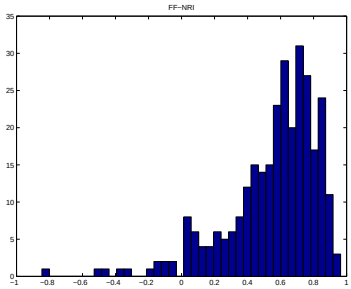
Table 5.6: Fixed parameter values of block based clustering and Bayesian flooding algorithms, for "Case Study III".

in Figures 5.7, 5.8 and 5.9 respectively. In each Figure, comparison results for PMLFA in subfigure (a) and ILFMA in subfigure (b), are given for the four metrics PR, NPR, GCE and LCE, in left-right, top-bottom order respectively. The frequency of images at a given metric value is depicted by the corresponding histogram for each metric. Furthermore, average (mean) and median performance for each metric over all BSDS images are illustrated under the histogram.

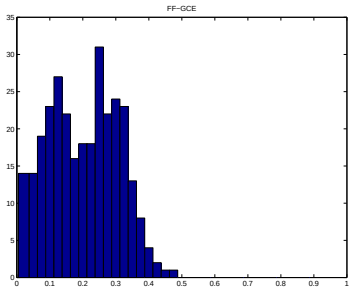
Performance evaluation against humans shows that the segmentation results scale well to human perception of regions for this dataset. Furthermore, the outcome of performance evaluation, indicates that the automatic computation of bins for intensity and colour, sufficiently adapts to image content and there is no need to be provided by hand for each image. Feature selection as well as the number of classes are robustly computed by the corresponding methods, for the majority of images.

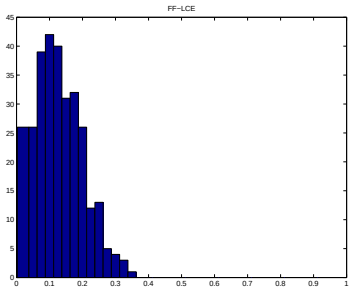
It should be noticed that although supervised evaluation metrics permit the assessment of segmentation results and the comparison between segmentation algorithms, their results are always biased because of the decisions made during ground truth extraction, often by human operators, as in the case of BSDS. However, the definition of NPR metric in Appendix B uses these decisions to judge how well ground truth maps capture the visual content complexity. In fact, this approach leads to a quantitative estimation of the hardness of segmentation task per image and hence, NPR metric normalizes Rand Index (RI) values given the complexity of visual content in an image basis. In that sense, NPR may be considered the more qualitative and





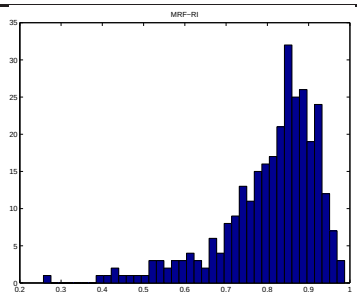
PR: $Mean = 0.8055$, $Median = 0.8333$ NPR: $Mean = 0.5554$, $Median = 0.6203$

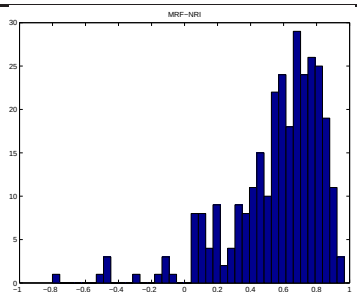




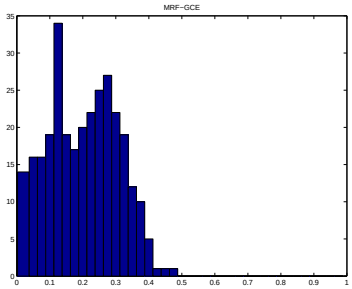
GCE: $Mean = 0.2039$, $Median = 0.2086$ LCE: $Mean = 0.1304$, $Median = 0.1211$

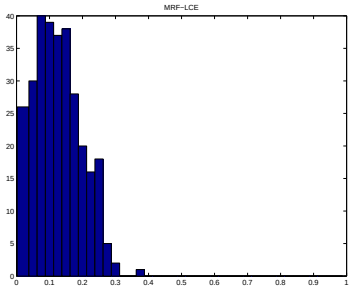
(a) PMLFA





PR: $Mean = 0.8073$, $Median = 0.8393$ NPR: $Mean = 0.5598$, $Median = 0.6247$

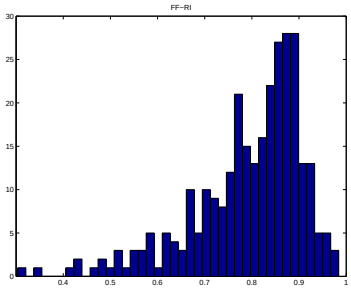


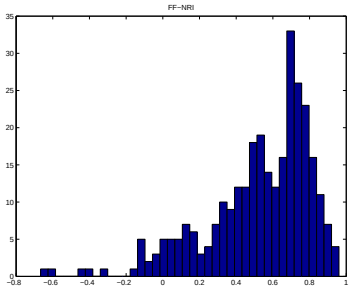


GCE: $Mean = 0.2033$, $Median = 0.2051$ LCE: $Mean = 0.1283$, $Median = 0.1204$

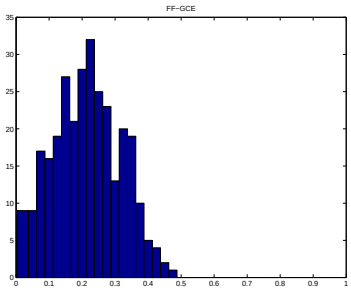
(b) ILFMA

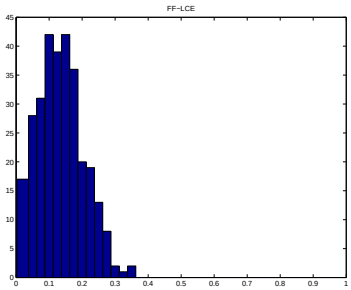
Figure 5.7: Case Study I: Histogram, mean and median metric values for the 300 images of BSDS, with automatic computation of K_I and K_{ab} intensity and chromaticity bins respectively and one DWF level of texture details. The combination of features as well as the number of classes for each image are given by user.





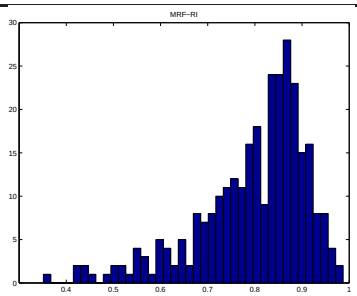
PR: $Mean = 0.7936$, $Median = 0.8265$ NPR: $Mean = 0.5244$, $Median = 0.5898$

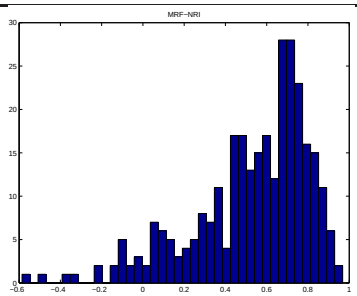




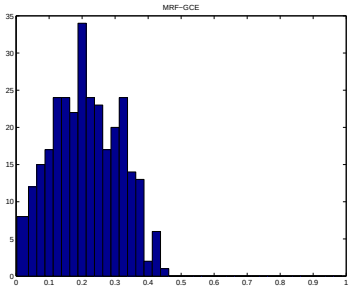
GCE: $Mean = 0.2161$, $Median = 0.2172$ LCE: $Mean = 0.1367$, $Median = 0.1317$

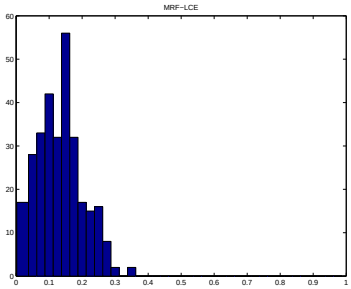
(a) PMLFA





PR: $Mean = 0.7958$, $Median = 0.8296$ NPR: $Mean = 0.5295$, $Median = 0.5978$

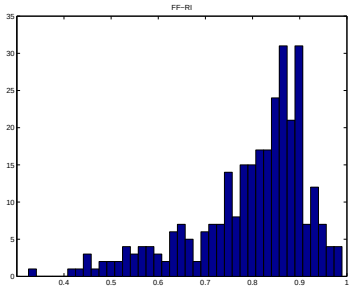


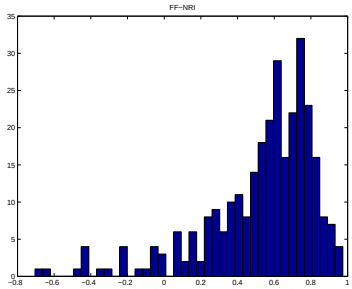


GCE: $Mean = 0.2134$, $Median = 0.2078$ LCE: $Mean = 0.1357$, $Median = 0.1353$

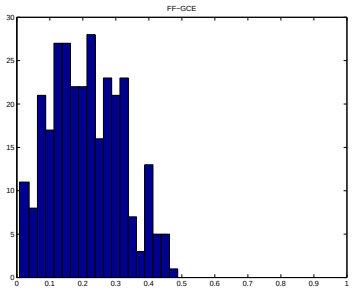
(b) ILFMA

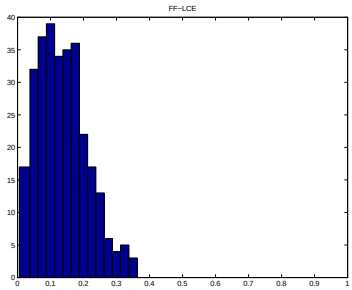
Figure 5.8: Case Study II: Histogram, mean and median metric values for the 300 images of BSDS, with automatic computation of K_I and K_{ab} intensity and chromaticity bins respectively and one DWF level of texture details. In addition, number of classes is automatically computed for the user given combination of features per image.





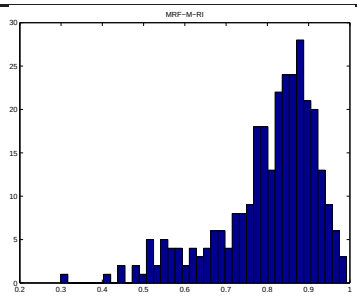
PR: $Mean = 0.7949$, $Median = 0.8296$ NPR: $Mean = 0.5308$, $Median = 0.6101$

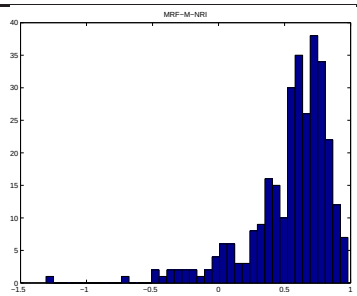




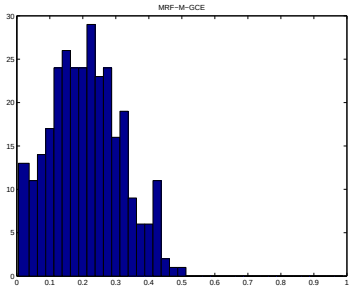
GCE: $Mean = 0.2114$, $Median = 0.2037$ LCE: $Mean = 0.1374$, $Median = 0.1278$

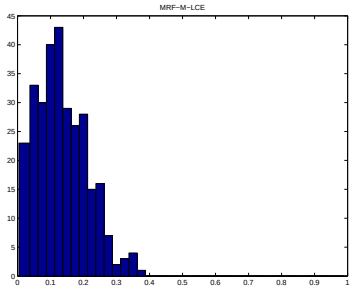
(a) PMLFA





PR: $Mean = 0.8027$, $Median = 0.8354$ NPR: $Mean = 0.5435$, $Median = 0.6213$





GCE: $Mean = 0.2120$, $Median = 0.2117$ LCE: $Mean = 0.1367$, $Median = 0.1274$

(b) ILFMA

Figure 5.9: Case Study III: Histogram, mean and median metric values for the 300 images of BSDS, with automatic computation of K_I and K_{ab} intensity and chromaticity bins respectively and one DWF level of texture details. In addition, feature combination and the number of classes are automatically computed per image.

	(a) PMLFA	(b) ILFMA	JSEG	FH
PR	0.8895	0.5008	0.4927	0.3531
NPR	0.4942	-1.3070	-1.3427	-2.0011
GCE	0.0769	0.0850	0.0635	0.0355
LCE	0.0446	0.0593	0.0590	0.0313

Table 5.7: Performance values for the segmentation results of Fig. 5.10. Expected PR index is greater than 0.78 for all human operators in that case.

explanatory criterion compared to the other three.

The way that NPR metric value scales according to the colour visual content and the ground truth maps, is explained using the example of Fig. 5.10. In images (a)-(d) of that Figure, segmentation maps are illustrated in pseudocolour for the colour image of the first row, obtained by PMLFA, ILFMA, JSEG and FH algorithms, respectively. The five ground truth maps are also depicted in that Figure. Colour content complexity is low for this image and the expected index of all ground truth maps, given the colour visual content, is greater than 0.78. High values of expected index indicate that colour visual content has been captured by human operators. Performance metric values for all results and metrics are shown in Table 5.7. In that Table, high NPR value is achieved by PMLFA result only, while the other three results are heavily penalized, because they do not agree with the complexity of visual content in that image. By contrary, according to GCE and LCE metrics, performance becomes misleadingly better (lower metric values), as the number of regions in each map increases, since these metrics favour oversegmentation. This is not the case for PR and thusfore may be considered a more objective evaluation metric compared to GCE and LCE.

5.3.3 Comparison to Other Segmentation Methods

In this Subsection, the proposed method is qualitatively and quantitatively compared against the unsupervised segmentation algorithms JSEG and FH. Considering our framework, comparison results only for "Case Study III" are illustrated, since clustering and segmentation are fully automatic in that case.

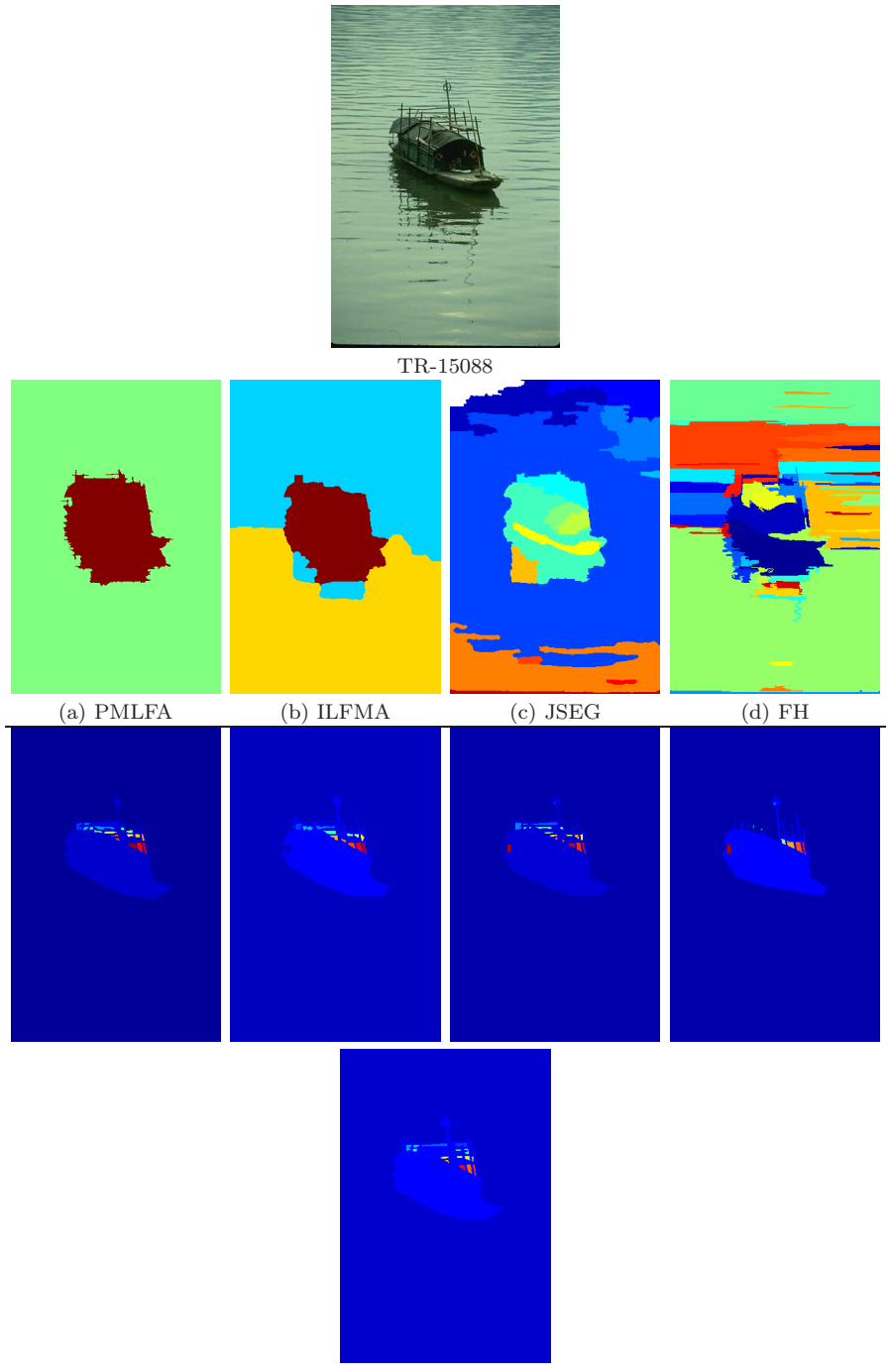
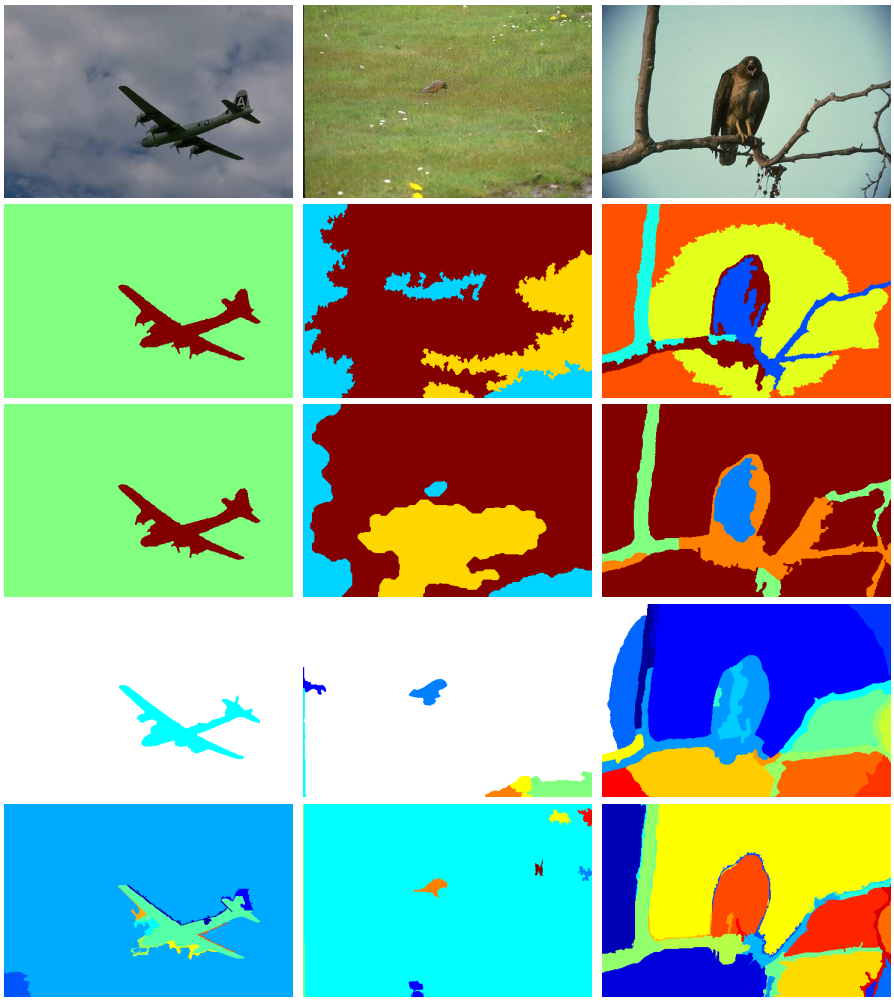


Figure 5.10: Segmentation results of (a) PMLFA, (b) ILFMA, (c) JSEG and (d) FH algorithms and ground truth maps for the colour image of the first row. Results (a) and (b) are of "Case Study III".

Segmentation results of PMLFA and ILFMA of "Case Study III" and of JSEG and FH methods using their default parameter values, are depicted in images of Figures 5.11- 5.13. Segmentation maps for each image and algorithm are displayed in pseudocolour in these Figures. In PMLFA and ILFMA segmentation maps, each colour corresponds to a class, rather than to a region label. These images are characterized by high diversity of visual content or gradual transitions and reflections, which, in general, make segmentation a difficult task. Although gradual transitions of visual cues are not modeled by our framework, boundary based merging that is applied after PMLFA and ILFMA, gives the semantically correct segmentation. In fact, boundary based merging improves the segmentation results in many cases, where there is no physical boundary between regions of classes. Characteristic examples of improved segmentation results, after merging has been applied, are those of images Fig. 5.11(a), Fig. 5.12(b) and Fig. 5.13(b). Considering the other two algorithms, gradual transitions and reflections are effectively handled by JSEG, as it shown for example by the corresponding segmentation maps of Fig. 5.11(a) and Fig. 5.12(a), but this is not always the case for FH. However, both algorithms usually lead to oversegmentation, specially when they are applied on images of high visual content diversity, as it is obvious by most examples presented herein.

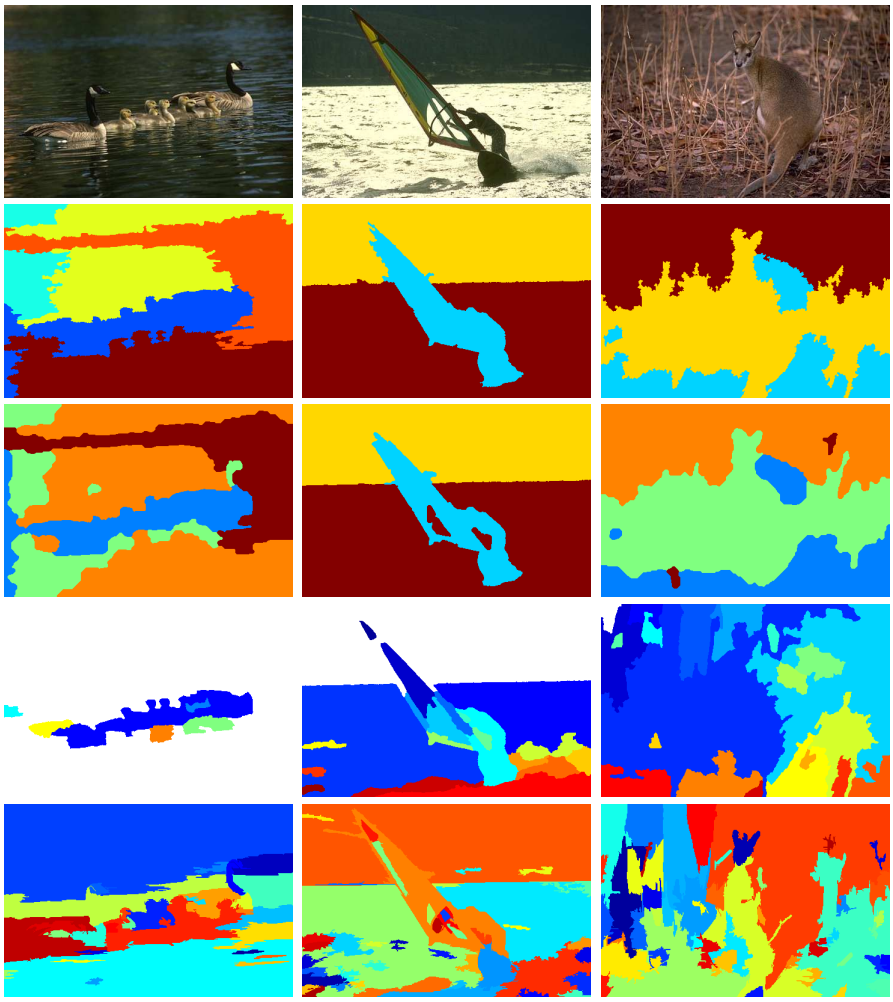
Considering our block based clustering method, problems arise in capturing the statistical description of small or thin objects, specially when their statistical description is similar to that of their background. Thusfore, extraction of bird from its background in image of Fig. 5.11(b) is a hard segmentation task for our method. However, semantically correct segmentation is achieved by our framework, even for images of high diversity, if the statistical description of objects, or classes of objects, is enough to discriminate them from their background. .

Algorithms have been also quantitatively compared, using metrics PR, NPR, GCE and LCE as before (Appendix B). As in the previous Subsection, for each image, algorithm and metric, segmentation map S is compared against each one of the ground truth results $\{S_k\}$ that is given for the image by humans. The median of metric scores against humans is used to measure



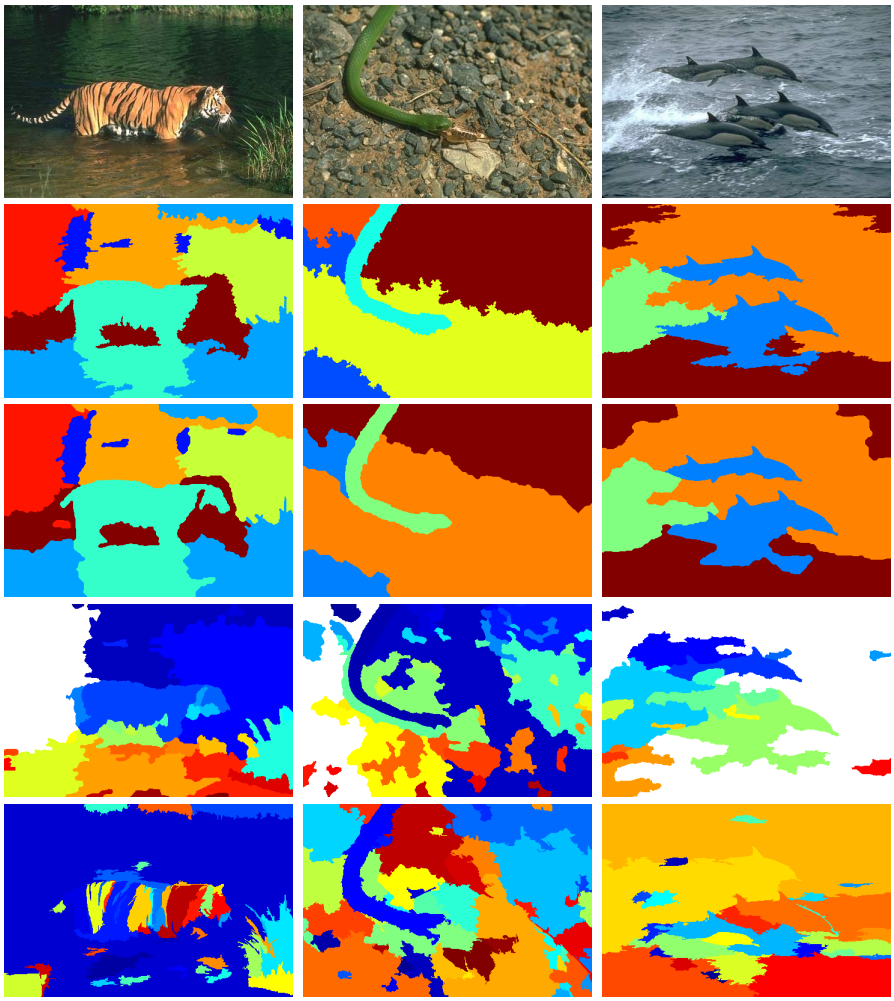
(a) TE-3096 (b) TR-28096 (c) TE-42049

Figure 5.11: In each column, in top-down order, original image and segmentation results of PMLFA, ILFMA of "Case Study III", JSEG and FH respectively.



(a) TR-43070 (b) TE-62096 (c) TE-69040

Figure 5.12: In each column, in top-down order, original image and segmentation results of PMLFA, ILFMA of "Case Study III", JSEG and FH respectively.



(a) TR-108073 (b) TE-175043 (c) TR-314016

Figure 5.13: In each column, in top-down order, original image and segmentation results of PMLFA, ILFMA of "Case Study III", JSEG and FH respectively.

the performance of the algorithm for each image.

Average and median performance per metric for the overall dataset are given in Tables 5.8 and 5.9 respectively. The first outcome of the comparison is that the performance of ILFMA is slightly better than that of PMLFA, since ILFMA not only smooths noisy region boundaries but, in cases as that of images Fig. 5.11(b) and Fig. 5.12(b), succeeds in describing with increased accuracy the objects of interest. Compared to the other two segmentation algorithms, ILFMA gives better average and median performance scores for PR and NPR, while its performance is worse according to GCE and LCE metrics. This is an expected result since GCE and LCE favour oversegmentation, which is avoided by our method.

	PR	NPR	GCE	LCE
PMLFA	0.7949	0.5308	0.2114	0.1374
ILFMA	0.8027	0.5435	0.2120	0.1367
JSEG	0.7927	0.5219	0.1930	0.1336
FH	0.8016	0.5404	0.1930	0.1309

Table 5.8: Average of performance metrics for the 300 colour images of BSDS.

	PR	NPR	GCE	LCE
PMLFA	0.8296	0.6101	0.2037	0.1278
ILFMA	0.8354	0.6213	0.2117	0.1274
JSEG	0.8296	0.6043	0.1830	0.1265
FH	0.8310	0.6316	0.1810	0.1257

Table 5.9: Median of performance metrics for the 300 colour images of BSDS.

5.4 Colour Based Face Detection

The proposed segmentation schema could be used to determine the existence or not of regions that belong to a visual class of features. The detection of that kind could be used to annotate images to the corresponding class. A recent approach, based on regions detection for video annotation is given

in [51]. In what follows we describe the application of PMLFA in order to extract image regions belonging to the class of face colours in $YCbCr$ colour space. The method could be used as the first stage of a face detection system, based on the shape and texture of the extracted regions, as it was the case in [52].

5.4.1 Segmentation Framework for Face Colours Detection

The segmentation framework includes

- the feature extraction and classification for each one of the segmentation classes,
- the labelling of initial pixel regions of high confidence per class using its statistics and
- the expansion of initial regions of classes by the flooding algorithm to fill the segmentation map of labels.

This framework is applied on the image in order to compute a binary segmentation map which represents the image regions in which face colours are present. Segmentation involves the label 0 of *non face* colours and the label 1, which corresponds to the class of *face* colours in the image. The feature we use is the colour of pixels $\mathbf{c}$ in $YCbCr$ colour space. We assume that the distribution of colours in the image is described by the mixture

$$p(\mathbf{c}) = P_0 p(\mathbf{c}|0) + P_1 p(\mathbf{c}|1),$$

where P_0 and P_1 are the *a priori* probabilities of the two hypotheses, which in our case are set equal to 0.5.

In the case of face colour regions detection, the conditional probability distributions of the mixture are computed as follows: face colours probability distribution $p(\mathbf{c}|1)$ is represented by the 3D histogram of randomly sampled pixels in manually selected face areas. In order to compute the probability

distribution $p(\mathbf{c}|0)$ of non face colours and instead of using the assumption that their colours are uniformly distributed, we first compute the probability $p(\mathbf{c})$ as 3D colour histogram over all image pixels and then we set

$$p(\mathbf{c}|0) = \begin{cases} \alpha p(\mathbf{c}), & \text{for } \mathbf{c} : p(\mathbf{c}|1) \neq 0 \\ \beta p(\mathbf{c}), & \text{otherwise} \end{cases} \tag{5.1}$$

where $0 \leq \alpha \leq 1$ is a user given factor and

$$\beta = \frac{1 - \alpha \sum_{\mathbf{c}:p(\mathbf{c}|1)\neq 0} p(\mathbf{c})}{\sum_{\mathbf{c}:p(\mathbf{c}|1)=0} p(\mathbf{c})}.$$

We shall illustrate the role of factor α and that of the computed factor β by a graphical example for one dimensional distributions. In Figure 5.14, probability distribution functions $p(x)$ and $p(x|1)$ of x are depicted. In order to compute $p(x|0)$, we lower the values of $p(x)$ in which $p(x|1) \simeq 0$ by α and we set $p(x|0) = \alpha p(x)$. Then, we distribute the subtracted probability to x values for which holds that $p(x|1) > 0$, by raising them by β, which is always greater or equal to 1. We set $p(x|0) = \beta p(x)$ in that case. Factor β makes the computed function $p(x|0)$ to be a probability distribution function. In the artificial example of Fig. 5.14, α has been set to 0.3.

Using this heuristic, probabilty $p(\mathbf{c}(s)|0)$ of sites s of the image that likely belong to the class of face colours is lowered, avoiding their classification to the class of non face colours.

Having available the statistical description of classes, the next step is to compute a set of spatially connected regions of pixels, which are classified to classes $\{0, 1\}$ with high confidence, using statistical tests. We define the statistical dissimilarity between pixel s and class l by the Bayesian rule:

$$d_l^B(s) = -\ln \Pr\{l|\boldsymbol{c}(s)\} = -\ln \frac{p(\boldsymbol{c}(s)|l)}{p(\boldsymbol{c}(s)|l) + p(\boldsymbol{c}(s)|1-l)},$$

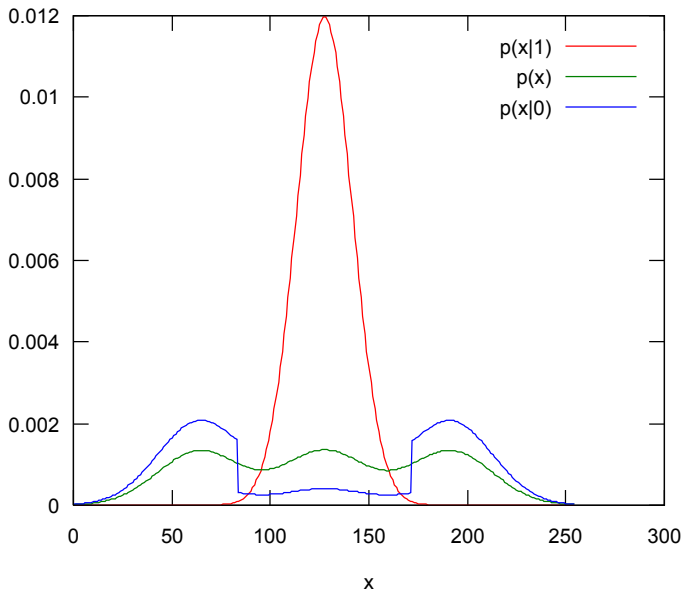


Figure 5.14: Definition of conditional probability distribution functions for face segmentation.

where $\mathbf{c}(s)$ is the colour of s. Then, for each pixel s and class l , the distances in a window Π_W of dimension $(2W+1)^2$ are averaged, resulting to the metric:

$$d_l^{SB}(s) = \sum_{z \in \Pi_W} d_l^B(s+z).$$

Image pixels s with $d_l^{SB}(s)/(2W+1)^2 < 0.5$ are sorted in ascending order according to that metric and a user-given percentage of the sorted sites with minimum average distance are retained and get labelled. If for one of the two classes (usually for the class of face colours) no initial regions are extracted by the initialization process, the flooding process is cancelled. By contrary, in the case that both classes are initialized, the initial regions of the two classes are grown by PMLFA. The output of PMLFA is the map of the face colours regions in the image. In Fig. 5.15, the initial labelled regions (b) and the final segmentation result (c) for image (a) are depicted. Labelled pixels are displayed in black for "non face" and white for the "face" label respectively, while in gray are shown the initially unlabelled pixels.

(a) (b) (c)

Figure 5.15: From left to right: original image, initialization and final segmentation map.

Chapter 6

Video Segmentation

Localization and tracking of video objects constitute critical steps in many computer vision applications. Features of visual objects are extensively used in international coding and storage standards MPEG-4 [53, 54] and MPEG-7 [55], as well as in surveillance vision tasks [56], such as the determination of patterns of activity [57].

An overview of segmentation tools, as well as of region-based representations of image and video, are presented in [6]. The video object extraction could be based on change detection and moving object localization, or on motion field segmentation, particularly when the camera is moving. Segmentation of image sequences is achieved using techniques mentioned at the introduction of the thesis, such as *Active Contour Models* [58], *Geodesic Active Contours* and *Level Sets* [59], *Region Growing* [9]. Several methods have been also proposed for foreground detection in particular, using background/foreground modelling of changes for cues such as colour, gradients or motion vectors.

Modelling of changes may be parametric, using for example mixtures of Gaussian probability density functions [57], or non-parametric using histograms [60, 61]. In [62], we proposed an object localization algorithm in which change detection is based on Bayesian tests that are applied on the inter-frame difference, while object localization is improved using the object colour information in a post-processing step. Change detection segmentation

was based on the *Multi-label Fast Marching* algorithm, introduced in [63] as an application of the general segmentation framework of *Bayesian Level Sets* [64].

In this Chapter, three methods are described in order to extract moving objects in video sequences. In the Section that follows, a segmentation method based on the change detection and colour cues of pixels is discussed in detail. In the second method, the change detection statistics are computed for the uniform in colour image regions instead of the pixels (Section 6.2) and merging using a modified Seeded Region Growing [9] algorithm is performed to label colour regions. The last method relies on the statistical description of optical flow information between successive sequence frames and it is described in detail in Section 6.3.

6.1 Change Detection and Colour based Segmentation

The approach of the Section is based on change detection, while moving objects discrimination is further supported by the statistical description of background/foreground colours using histograms. Pixel-based *mixture analysis* using *Expectation-Maximization (EM)* is employed to compute the change detection parameters. Change detection-based probabilistic distances are then used to determine and label pixels that belong to one of the classes "changed"/"unchanged" with high confidence.

Having available the data modelling and the initial map of correctly labelled pixels, foreground objects are extracted using PMLFA (Section 4.4). PMLFA completes the labelling of map using statistical, change detection and colour dissimilarity criteria.

6.1.1 Problem Formulation

In our approach, change detection statistics, in combination with background/foreground colour information are used for foreground objects detection. The simple

inter-frame gray level difference

$$d(s) = I_{t+1}(s) - I_t(s)$$

is considered as change detection, segmentation feature. Let $p_0^{CD}(d)$ (resp. $p_1^{CD}(d(s))$) be the probability density function of the observed inter-frame difference under the hypothesis of a pixel to be "unchanged" (resp. "changed"). These probability density functions are assumed to be zero-mean Laplacian for both hypotheses $(l = 0, 1)$ [65]:

$$p_l^{CD}(d(s)) = \frac{\lambda_l}{2} e^{-\lambda_l |d(s)|}. \tag{6.1}$$

Let P_0 (resp. P_1) be the *a priori* probability of the two hypotheses. Then, the probability density function is given by the mixture

$$p^{CD}(d(s)) = P_0 p_0^{CD}(d(s)) + P_1 p_1^{CD}(d(s)) \tag{6.2}$$

In this mixture distribution $\{P_l, \lambda_l,\ l = 0, 1\}$ are unknown parameters and are estimated using the *EM* algorithm, according to the *Maximum Likelihood* principle.

Furthermore, the colour coordinates of pixels s, $\mathbf{c}(s) = (L(s), a(s), b(s))$ of the *Lab* colour space are used for colour modelling. Probability density functions $p_0^C(\mathbf{c}(s))$, $p_1^C(\mathbf{c}(s))$ of background and foreground colours respectively, are represented by histograms, estimated by the pixels which have been classified as "unchanged"/"changed", in the previous change map.

6.1.2 Flooding Process for Label Propagation

6.1.2.1 Label Initialization

An initial map of labelled sites is obtained using *change detection* only, statistical tests. The first test detects "changed" pixels with high confidence, using the threshold [62]:

$$T_1 = \frac{1}{\lambda_0} \ln \frac{1}{P_{FA}},$$

upon the inter-frame difference of each pixel, where P_{FA} is a given small false alarm probability. Despite the pixel-based initialization of "changed" class, the initial "unchanged" regions of high confidence are determined using change detection information in pixels neighborhood. For each pixel s, the distances in a window Π_W of dimension $(2W+1)^2$ are averaged, resulting to the metric [62]:

$$d_0^{SB}(s) = \sum_{z \in \Pi_W} d_0^{CD}(s+z),$$

where $d_0^{CD}(\cdot)$ is the Bayesian change detection distance defined for class "unchanged" in Subsection 6.1.2.2. Then, image pixels are sorted in ascending order according to that metric and a user-given percentage of the sites with minimum average distance are retained and get labelled.

6.1.2.2 Flooding Dissimilarity Criteria

A dual type of dissimilarity metrics arises in the case of a two-class segmentation problem as it is the case of background/foreground discrimination. The difference of heights (distances) of a pixel s against the two classes ($l = 0, 1$) could be then used and measured as

$$\Delta d_l^B(s) = -\ln\left(\frac{\Pr\{l|\boldsymbol{d}(s)\}}{\Pr\{1-l|\boldsymbol{d}(s)\}}\right). \tag{6.3}$$

Assuming that colour and change detection features are independent variables and ignoring the *a-priori* probabilities, since they are only estimates and not *a-priori* knowledge, Eq. (6.3) becomes the sum of a change detection term and a colour term:

$$\begin{aligned} \Delta d_l^B(s) &= -\ln\left(\frac{p_l^{CD}(d(s))}{p_{1-l}^{CD}(d(s))}\right) - \ln\left(\frac{p_l^{C}(\mathbf{c}(s))}{p_{1-l}^{C}(\mathbf{c}(s))}\right) \\ &= d_l^{CD}(s) + d_l^{C}(s). \end{aligned} \tag{6.4}$$

PMLFA is applied using only thresholds $\mathfrak{T}_l^{high}$, for $l = 0, 1$. These thresholds and quantization step Δ_Q of PMLFA are tuned using only the change detection statistics. If flooding was meant to be based only on the change de-

tection statistics, the dissimilarity of a site s from a label l could be measured as

$$\begin{aligned} DIS_l(s) &= \frac{\sum_{k\neq l} Pr(k(s)|d(s))}{Pr(l(s)|d(s))} \\ &= \frac{\sum_{k\neq l} p(d(s)|k(s))Pr(k(s))}{p(d(s)|l(s))Pr(l(s))}, \end{aligned}$$

or equivalently

$$\begin{aligned} dcd_l(s) &= \ln DIS_l(s) \\ &= \ln \frac{\sum_{k\neq l} Pr(k(s)|d(s))}{Pr(l(s)|d(s))} \\ &= \ln(\sum_{k\neq l} p(d(s)|k(s))Pr(k(s))) \\ &\quad - \ln(p(d(s)|l(s))Pr(l(s))). \end{aligned}$$

In our case of change detection the metric for label 0 becomes

$$dcd_0(s) = \ln(p(d(s)|1)P_1) - \ln(p(d(s)|0)P_0)$$

and under the assumption of Laplacian distributions this gives

$$dcd_0(s) = -\ln \frac{\lambda_0 P_0}{\lambda_1 P_1} + (\lambda_0 - \lambda_1)|d(s)| \tag{6.5}$$

and $dcd_1(s) = -dcd_0(s)$. Since P_l, $(l = 0, 1)$ are only estimates and not a-priori knowledge, they have been set to 0.5 in the current implementation of criterion dcd_0. Apparently,

$$dcd_0(s) = -\alpha + \beta|d(s)|, \tag{6.6}$$

where $\alpha = \ln \frac{\lambda_0}{\lambda_1}$ and $\beta = \lambda_0 - \lambda_1$. In Fig. 6.1, dcd_l is plotted against $|d|$ for $\lambda_0 = 1.5$ and $\lambda_1 = 0.05$ respectively, for $l = 0, 1$. As we see, for $|d| = \frac{\alpha}{\beta}$ we get the decision point $dcd_0 = dcd_1 = 0$ between the two classes.

The fundamental principle of the new algorithm is that it labels groups of

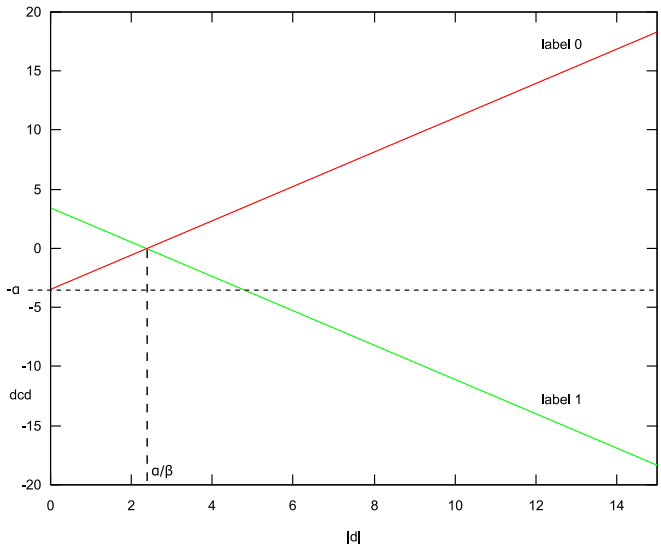


Figure 6.1: Dissimilarity, change detection based, criteria.

yet unlabelled pixels, at each execution step. Group labelling refers to pixels, which are placed on the neighborhood of already labelled pixels at an instance of the propagation progress and their metric dcd_l against label l is quite the same. This fact, implies the *quantization* of dissimilarity metric according to the specific characteristics of change detection driven propagation. Towards this direction, by Eq. (6.6) the random variable $dcd_0(s)$ is a linear function of $|d|$ and by Eq. (6.2) the probability density function $p(dcd_0(s))$ is given by the mixture

$$\begin{aligned} p(dcd_0(s) = y) &= P_0 p(dcd_0(s) = y|static) + \\ &\quad P_1 p(dcd_0(s) = y|mobile), \end{aligned} \tag{6.7}$$

where $y \geq -\alpha$. Using Eq. (6.1), holds that

$$p(dcd_0(s) = y|\Theta(s) = l) = \frac{\lambda_l}{\beta} e^{-\frac{\lambda_l}{\beta}(y+\alpha)}, \quad y \geq -\alpha \tag{6.8}$$

is an exponential distribution for $l = 0, 1$ and Eq. (6.7) becomes

$$p(dcd_0(s) = y) = P_0 \frac{\lambda_0}{\beta} e^{-\frac{\lambda_0}{\beta}(y+\alpha)} + P_1 \frac{\lambda_1}{\beta} e^{-\frac{\lambda_1}{\beta}(y+\alpha)} \tag{6.9}$$

for $y \geq -\alpha$.

Similarly, for $y = dcd_1(s)$, we get

$$p(dcd_1(s) = y) = P_0 \frac{\lambda_0}{\beta} e^{-\frac{\lambda_0}{\beta}(-y+\alpha)} + P_1 \frac{\lambda_1}{\beta} e^{-\frac{\lambda_1}{\beta}(-y+\alpha)} \tag{6.10}$$

for $y \leq \alpha$. However, since we refer to contour sites, P_k $(k = 0, 1)$ vary during growing and are not the same for the two labels. Thus, for each label l and at growing step T, P_k in Equations (6.9), (6.10) are dynamically replaced by the percentage $P_{l,k}^T$ of sites which are placed on the contour of regions of label l and according to change detection statistics belong to label k, for $l, k \in \{0, 1\}$. This fact does not affect the statistical analysis for dissimilarity criteria dcd_l $(l = 0, 1)$ that follows, since it depends only on the exponential probability density functions of mixtures, without using percentages $P_{l,k}^T$. In Fig. 6.2, the probability density functions of the mixtures of Eq. (6.9) and Eq. (6.10) respectively, are depicted graphically for $l = 0, 1$ and $\lambda_0 = 1.5$, $\lambda_1 = 0.1$.

As a consequence, if for a contour pixel s of a "changed" region holds that $dcd_1(s) < -\alpha$, then s can be labelled as "changed" with high confidence, since unlikely belongs to an "unchanged" region. Indeed, since

$$P_F(t) = e^{-\frac{\lambda_0}{\beta}(t+\alpha)}, \quad t \geq -\alpha$$

is the *false alarm* probability of labelling a pixel as "changed" while it is "unchanged", then if we set $t_1 = -\alpha$, for $t \geq -t_1$ holds that

$$P_F(t) \leq e^{-2\alpha\frac{\lambda_0}{\beta}},$$

which is a fairly small false alarm probability. In Fig. 6.1, certainty limit $y = -\alpha$ is depicted by the horizontal dashed line. By analogy, a threshold t_0 for labelling contour sites s of "unchanged" regions is set, according to

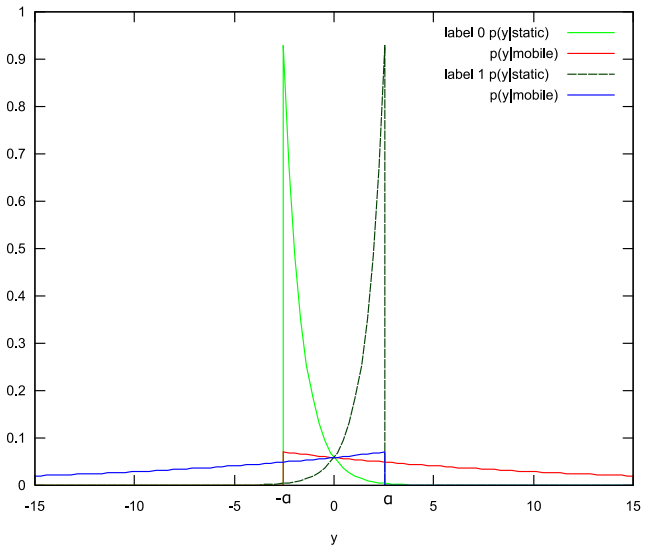


Figure 6.2: Probability density functions for dcd_0, dcd_1.

equation

$$t_0 = -\alpha - \frac{\beta}{\lambda_1} \ln(1 - P_{ND}),$$

where P_{ND} is a given small probability of not detecting a "changed" pixel. Thus, contour sites s, which are placed on the border of an "unchanged" region, with $dcd_0(s) < t_0$, can be labelled as "unchanged" with high confidence.

However, a memory term has to be introduced in the dissimilarity criterion of the "changed" class, to cope with the disadvantages in propagate "changed" regions contours, due to possible uniform areas and imperceptible motion of moving objects. After all, the dissimilarity metric of contour pixel s against the "changed" class becomes

$$\delta_1(s) = d_1^{CD}(s) + d_1^{C}(s) - \rho(s)\sigma(s), \tag{6.11}$$

where $\rho(s)$ is the distance of point s from the border of the previous "changed" mask and $\sigma(s)$ is an optional sigmoid function of local intensity variance for amplifying memory factor $\rho(s)$ in homogeneous moving areas. Colour dis-

similarity $d_1^C(s)$ is measured as

$$d_1^C(s) = -\ln \frac{p_1^C(\mathbf{c}(s))}{p_0^C(\mathbf{c}(s))}$$

For clarity, we also set $\delta_0(s) = d_0^{CD}(s) + d_0^C(s)$, where $d_0^C(s) = -d_1^C(s)$.

6.1.3 Experimental Results

We present experimental results for well known, benchmark image sequences as well as for webcam and compressed videos. In Fig. 6.3 we see the steps of our method, as it is applied on frame (a) of sequence *Erik*. Initialization is given in Fig. 6.3(b), where black colour depicts "unchanged" pixels, white denotes "changed" pixels and gray colour is used to depict the initially unlabelled pixels. Colour histogram is computed using 32 bins per channel, or $32 \times 32 \times 32$ bins in total, although, depending on the image sequence, no more than 5000 bins appear in practice. In Figures 6.4, 6.5, the marginal probabilities $p_0^C(L, a)$, $p_1^C(L, a)$ and $p_0^C(L, b)$, $p_1^C(L, b)$ are depicted respectively, as they are computed by the background and foreground 3D colour histograms. The coloured surface in those plots, corresponds to background marginal probabilities $p_0^C(\cdot, \cdot)$, while the white surface depicts foreground statistics. As it is evident, colour information improves the discrimination of the two classes. In Fig. 6.3(c), the foreground detection after the flooding process, using the change detection and colour cues, is shown.

In Fig. 6.6, foreground detection results for image sequence *Mother*, are depicted and in Figures 6.7, 6.8, foreground extraction results for the webcam video *Emilio* [3] and the MPEG-2 compressed video *TV*, are shown respectively. According to the results, the method performs reasonably in situations of low motion, automatic lighting correction and compression artifacts. Its robustness could be further improved by considering a "memory" of background/foreground colours in a short period of previous frames, instead of using only the colour statistics of the current frame.

(a) (b) (c)

Figure 6.3: Initialization (b) and foreground detection (c) for the frame (a) of image sequence *Erik*.

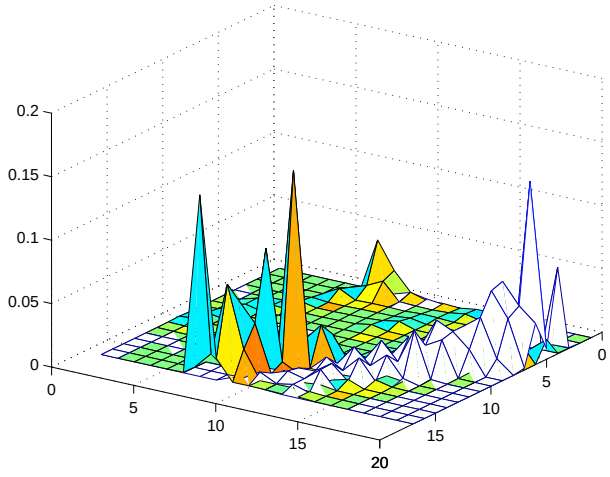


Figure 6.4: Marginal probabilities $p_0^C(L,a)$, $p_1^C(L,a)$.

6.2 Region-level Moving Object Segmentation by Graph Labelling

6.2.1 Introduction

In a number of video segmentation methods, object extraction is applied on the spatial partition of the image in homogeneous regions (region-level instead of pixel-level based extraction) in order to reduce the spatio-temporal redundancy of video images and to speed up and robustify the computations [66–68]. Video object extraction could be based then on change detec-

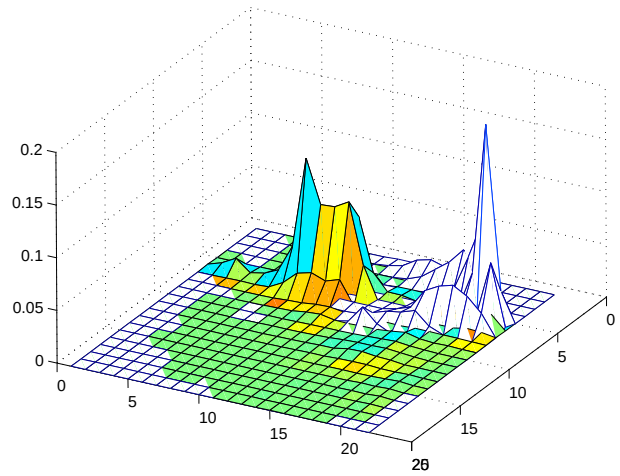


Figure 6.5: Marginal probabilities $p_0^C(L,b)$, $p_1^C(L,b)$.

tion and moving objects localization or on motion field segmentation of the spatial regions.

We follow the region-level approach in the proposed segmentation system which is depicted in Fig. 6.9. The system is divided in three layers of computation. In the first one (top-down order), the basic segmentation characteristics are evaluated by the corresponding modules in the order implied by the arrows in Fig. 6.9. Hence, (i) first the change detection mixture parameters are computed using the pixel inter-frame difference over all the image pixels, (ii) the colour regions of the current image are extracted and after that (iii) the region-based change detection statistics are computed using a region-based change detection feature.

The second level in Fig. 6.9 serves as the intermediate level between the first and the third one, since it uses the change detection parameters and colour regions, which have been extracted by the first level in order to produce an initial labelling, which will be expanded by the third. The top middle level task is to split the colour regions obtaining sub-regions which can be labelled as "changed" with high confidence. The pixel-based change detection statistics are used for this splitting. The second module involves the labelling of the remaining (sub-) regions as "unchanged" using the region-

Figure 6.6: Foreground extraction for image sequence *Mother*.

Figure 6.7: Foreground detection for *Emilio* webcam video [3].

Figure 6.8: Foreground detection for the MPEG-2 compressed video *TV*.

based statistics of the previous level.

Finally, the last level consists of the initial labels expansion task. The overall system is explained in detail below beginning with the colour regions extraction method, since the computation of the pixel-based change detection statistics is that of [69].

6.2.2 Region Inter-Frame Difference

The first step involves the partitioning $\mathbf{R}$ of the image in N homogeneous colour components, in the $YCbCr$ colour space. The well known k-means algorithm is employed to compute the dominant $YCbCr$ colours, which are then used to extract colour regions R_i of the image. Denoting by Λ the overall image points, the following equations hold for the final partition:

$$R_i \cap R_j = \emptyset \text{ and } \Lambda = \bigcup_{i=1}^{N} R_i \tag{6.12}$$

The segmentation algorithm is mainly based on *change detection*. The appropriate statistics involve not only the inter-frame difference, as was the

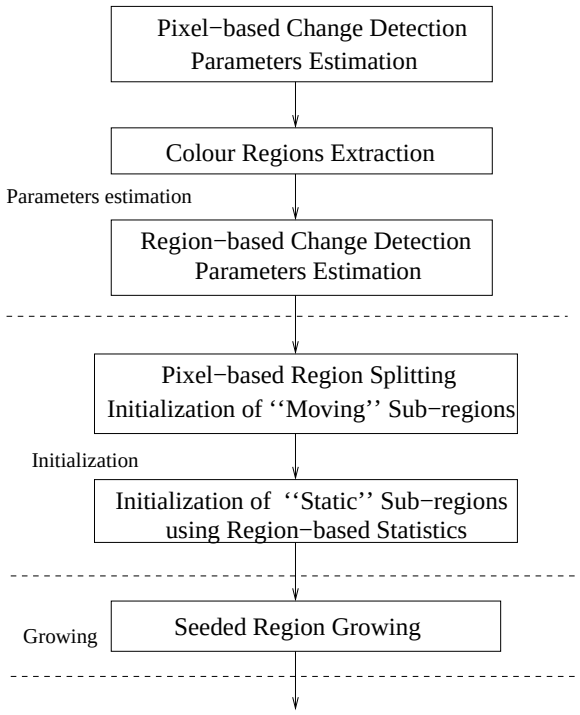


Figure 6.9: System framework.

case in [69], but also the mean of absolute differences of colour regions R_i:

$$d(R_i) = \frac{1}{|R_i|} \sum_{(x,y)\in R_i} |I_t(x,y) - I_{t+\Delta t}(x,y)| \tag{6.13}$$

where $|R_i|$ denotes the cardinality of region R_i and I_t (resp. $I_{t+\Delta t}$) is the intensity frame at time t (resp. $t + \Delta t$).

As it is described in [69], the two classes of "changed"/"unchanged" *pixels* are modeled by two *Laplace* distributions. Experimental results have shown that in the case of the mean absolute difference of colour regions, the two classes of "changed"/"unchanged" regions follow the *Gamma* distribution.

Let $D = \{d(R_i), 1 \leq i \leq N\}$ denote the mean of gray level differences of each colour region. The change detection problem consists of determining a binary label $\Theta(R_i)$ for each region R_i of the image. We associate the random field $\Theta(R_i)$ with two possible events, $\Theta(R_i) = static$ ("unchanged" region), and $\Theta(R_i) = mobile$ ("changed" region). Let $p_{D|static}(d|static)$ (resp. $p_{D|mobile}(d|mobile)$) be the probability density functions of the observed mean absolute inter-frame region difference under the H_0 (resp. H_1) hypothesis. These probability density functions are assumed to be *Gamma* for both hypotheses ($l = 0; 1$):

$$p(d(R_i)|\Theta(R_i) = l) = \frac{d(R_i)^{a_l} e^{-\frac{d(R_i)}{\beta_l}}}{\Gamma(a_l + 1)\beta_l^{a_l+1}}$$

Let P_0 (resp. P_1) be the a priori probability of hypothesis H_0 (resp. H_1). Thus the probability density function is given by

$$p_D(d) = P_0 \;\; p_{D|0}(d|static) + P_1 \;\; p_{D|1}(d|mobile) \tag{6.14}$$

In this mixture distribution $\{P_l, a_l, \beta_l, l \in \{0, 1\}\}$ are unknown parameters.

The experimental results in our effort for a robust region-based mixture decomposition, shown that it is sufficient to investigate only integer values of a_l ($l = 0, 1$). Nevertheless, in most cases holds that $a_0 \geq a_1$. These observations lead us to a straightforward method for the estimation of mixture

distribution parameters, by evaluating the χ^2 criterion between the histogram of D and the mixture distribution (eq. (6.14)) that is obtained for a finite set $\mathbf{a} = \{0, 1, \ldots, A\}$ of integer values of a_l, under the restriction that $a_0 \geq a_1$. Furthermore, the principle of Maximum Likelihood is used to obtain an estimate for P_l and β_l, $(l = 0, 1)$, for each investigated pair of a_0, a_1. The set of parameters $\{\hat{P}_l, \hat{a}_l, \hat{\beta}_l, l \in \{0, 1\}\}$ which minimizes the χ^2 metric is selected as the better estimate for the mixture distribution of eq. (6.14).

6.2.3 Change Detection using SRG on Regions

In what follows, we describe an extension of the well known *Seeded Region Growing (SRG)*[9] algorithm. In the extended algorithm the classes that are to be grown, are classes of regions instead of pixels and the same holds for the initially unlabelled items which are regions and not pixels. This modified algorithm is used to segment the image in the two classes of "changed"/"unchanged" regions, as it is described below.

6.2.3.1 Initialization

The growing algorithm requires a number of initial correctly labelled items. In our case, these are the colour regions which may be considered "static" or "moving" with high confidence. The confidence measurements are performed in both the pixel and region-based change detection statistics.

The first observation is that some "static" regions may contain a number of subregions with high inter-frame difference due to their overlapping with "mobile" regions and thus have to be split further. The splitting is performed using the pixel-based statistics of change detection. As in [69], the pixels that may be considered "changed" with high confidence are determined using the decision threshold:

$$T_1 = \frac{1}{\lambda_0} \ln \frac{1}{P_F},$$

where P_F is the given small false alarm probability and λ_0 is the estimated Laplacian parameter of "unchanged" pixels. Then, the connected "changed" pixels of each region are grouped to form new regions, which constitute the

"changed" sub-regions of high confidence. The remaining region pixels are also grouped in connected sub-regions, leading to a new image partition

$$\mathbf{R}' = \{R'_i, 1 \leq i \leq M\},$$

where M is the number of image regions. Next, the "unchanged" regions with high confidence are determined among the non "changed" sub-regions, using the decision criterion $d(R'_i) \leq T_0$, where T_0 satisfies the equation

$$P_{ND} = Pr\{d \leq T_0 | mobile\},$$

for a given small probability P_{ND} of not detecting a "changed" region.

6.2.3.2 Growing

The modified *SRG* algorithm is applied on the initial labelled regions in order to "grow" them. *Growing* refers now to regions instead of pixels and its effort is to assign the label "changed" or "unchanged" to the initially unlabelled regions. Each one of the two labels is grown according to dissimilarity criteria which are based on the label, the mean absolute difference and the boundary information of regions.

A label-depended term is set according to the *a-posteriori* probability principle. Assuming that the change detection statistics of each label follow the Gamma distribution, the dissimilarity of a colour region R from a label l is measured as

$$DIS_l(R) = \frac{1}{Pr(l(R)|d(R))} \tag{6.15}$$

Using the Bayes rule

$$Pr(l(R)|d(R)) = \frac{p(d(R)|l(R))Pr(l(R))}{\sum_k p(d(R)|k(R))Pr(k(R))}$$

which gives

$$DIS_l(R) = 1 + \frac{\sum_{k \neq l} p(d(R)|k(R))Pr(k(R))}{p(d(R)|l(R))Pr(l(R))}$$

Ignoring the constant term in the last equation and taking the logarithm of the second term gives

$$\begin{aligned} dcd_l(R) &= \ln(\sum_{k \neq l} p(d(R)|k(R))Pr(k(R))) \\ &\quad - \ln(p(d(R)|l(R))Pr(l(R))) \end{aligned}$$

In our case of change detection the metric for label 0 becomes

$$dcd_0(R) = ln(p(d(R)|1)Pr(1)) - ln(p(d(R)|0)Pr(0))$$

and under the Gamma distribution assumption this gives

$$\begin{aligned} dcd_0(R) &= (a_1 - a_0)\ln d(R) + d(R)(\frac{1}{\beta_0} - \frac{1}{\beta_1}) \\ &\quad + \ln\Gamma(a_1+1)Pr(1) + (a_1+1)\ln\beta_1 \\ &\quad - \ln\Gamma(a_0+1)Pr(0) - (a_0+1)\ln\beta_0 \end{aligned}$$

and $dcd_1(R) = -dcd_0(R)$. Since $Pr(l)$, $(l = 0, 1)$ are only estimates and not a-priori knowledge, they have been set to 0.5 in the current implementation of criterion dcd_0.

Furthermore, a boundary term dbd_l has been added to the label growing criterion:

$$dbd_0(R) = -\frac{b_0 - b_1}{\sqrt{|R|}}$$

and $dbd_1(R) = -dbd_0(R)$, where b_0 (resp. b_1) is the common boundary length between R and the regions that have been labelled as "unchanged" (resp. "changed") while $|\cdot|$ denotes the cardinality of its argument. The effect of the boundary term is to bypass the difficulties that arise in uniform regions which are parts of moving objects although their mean inter-frame difference is low. By minimizing dbd_l locally, the total common boundary between the two classes tends to be minimized.

The total dissimilarity $\delta_l(R)$ is then defined as

$$\delta_l(R) = f_{dcd}dcd_l(R) + f_{dbd}dbd_l(R)$$

where f_{dcd} is defined as:

$$f_{dcd} = \begin{cases} \frac{|R|}{100}, & \text{if } |R| \leq 5 \\ 1, & \text{otherwise} \end{cases}$$

and is used to decrease the effect of the "change detection" measurement in small regions, where mean difference estimation is often insufficient. By contrary, f_{dbd} is a binary decision factor:

$$f_{dbd} = \begin{cases} 1 & \text{if } |R| \leq 500 \\ 0, & \text{otherwise} \end{cases}$$

which implies that large enough regions cannot be treated in the same way that boundary pixels are used in order to enforce the smoothness of the boundary between the growing classes. It should be noticed that when $f_{dbd} = 0$, the overall criterion is solely based on "change detection" statistics, since the measurements in that case can be considered accurate. Apparently from the limitations that are imposed in the size of regions above, the criterion $\delta_l(\cdot)$ tends to give more emphasis to the "change" detection statistics as the size of regions becomes larger, since the boundary term $dbd_l(\cdot)$ decreases with size. This is an admirable property of the overall criterion which is achieved without any further tuning.

Furthermore, the distance $a(R)$ between the center of mass of region R and the boundary of the previous "change" mask has been introduced in metric $\delta_1(R)$ which after all becomes:

$$\delta_1(R) = dcd_1(R) + dbd_l(R) - a(R).$$

This "memory" term is used only for regions that their area was exclusively included in the "moving" objects of the previous "change" mask. The distinction between the two labels is justified by the fact that in frames, which undergo a small motion and contain large uniform background areas, a large part of the "unchanged" area is labelled at the initialization stage. By contrary, the "changed" label is initialized in small regions leading to a mismatch

for that label at the "growing" stage. Labels are expanded using the *SRG* algorithm for regions instead of pixels. For the implementation of *SRG* a list that keeps its members ordered according to the dissimilarity criterion $\delta(\cdot)$ is used, traditionally referred to as Sequentially Sorted List (SSL). In addition, $\mathbf{R}'$ is represented as a set of nodes in a connected undirected graph called *region adjacency graph* (RAG). Two nodes g_i and g_j of the graph are connected by an edge, if and only if, the corresponding regions R_i and R_j are adjacent. Finally, we define the set of indices $\mathbf{L} = L_0, \ldots, L_M$ to the class $L_i = l$ whose statistics give the minimum $\delta_l(\cdot)$ value for the region R'_i, $(1 \leq i \leq N)$. The complete SRG algorithm is as follows:

S1 Label the initial colour regions of classes 0 and 1 (initialization stage).

S2 Insert all the unlabelled spatial neighbors of the initial regions into the SSL. If they adjoin both the two classes use as $\delta_l(\cdot)$ that with the minimum value. Update properly their L value.

S3 While the SSL is not empty:

S3.1 Remove the first region y from the SSL and label it according to its L label.

S3.2 Test the neighbors of y and update the SSL:

S3.2.1 Add to the SSL neighbors of y which are neither already labelled nor already in the SSL, according to their value of $\delta(\cdot)$. If they adjoin and the other class, use as $\delta_l(\cdot)$ that with the minimum value. Update properly their L value.

S3.2.2 Test for neighbors of y which are already in the SSL and promote them accordingly in the SSL:

S3.2.2.1 if they border on and the other class, insert them in the SSL using as $\delta_l(\cdot)$ that with the minimum value,

S3.2.2.2 otherwise, insert them using the $\delta_l(\cdot)$ of y' s label.

Update properly their L value.

Each step of the modified SRG algorithm labels the minimum element-region of SSL y and a number of tests on yet unlabelled neighbors of y are

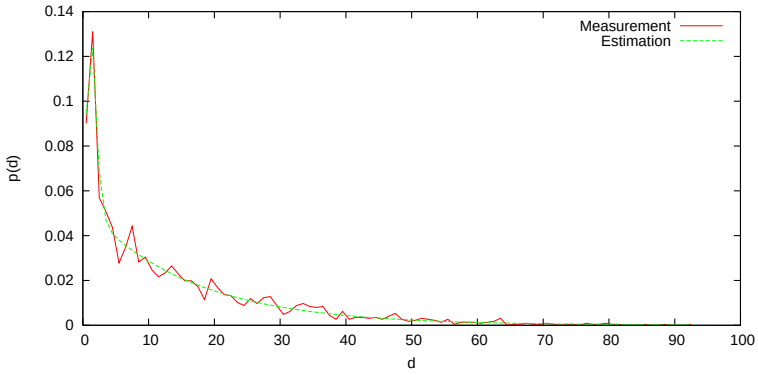


Figure 6.10: Region-based mean absolute inter-frame difference statistics for frame 42 of the sequence *"Mother"*.

performed followed by a constant number of insertions and deletions in SSL. Although in the first implementation [9] the unlabelled items were inserted only once in the SSL and their $\delta(\cdot)$ value was not updated until their labelling, the computational cost of the modified SRG algorithm that we present still remains low since (a) SSL is implemented using AVL trees in which the computational cost of insertions and deletions for M items is $O(M \log M)$, (b) the number of colour regions M is small –a few hundreds– compared to that of pixels and the number of unlabelled regions is even smaller, (c) the number of the neighbors of each region is usually in the same order of the eight neighbors of pixels, when 8-connectivity is considered and (d) the criterion value $dcd_l(\cdot)$ can be computed only once per each unlabelled region and kept in memory, since it remains unchanged during SRG iterations and the same holds for the "memory" term $a(\cdot)$ and finally, (e) the dynamically updated local boundary term $dbd_l(\cdot)$ is computed sufficiently in low cost.

6.2.4 Experimental Results

In what follows, we present the results that were obtained by the object detection system for the image sequences *"Mother"* and *"HallMonitor"*, which have been included in the COST testing data set. The camera in both sequences is static, while in *"HallMonitor"* the background is known.

Figure 6.11: Initialization (up) and moving objects extraction result (down) for frame 42 of the sequence *"Mother"*.

As we can see in the localization result of Fig. 6.11, *"Mother"* is characterized by low spatial detail while the two objects move very slowly or do not move at all during a large number of frames. The large homogeneous regions together with the low objects movement lead to low discrimination of the statistics of classes "changed" and "unchanged", which in turn affects the efficiency of algorithms that are based on change detection. The proposed system overcomes this difficulty by using regions instead of pixels in order to initialize the "unchanged" regions under the Gamma distribution assumption. In Fig. 6.10, "Measurement" refers to the histogram of the mean absolute difference of the 2300 regions that were extracted by frame 42 of *"Mother"*, while "Estimation" is the computed mixture of eq. (6.14). The initially labelled regions for this frame are shown in black ("unchanged") and white ("changed") in the upper image of Fig. 6.11, while the gray regions are initially unlabelled. The light gray curves depict the boundary of regions. The memory term $a(\cdot)$ that has been introduced in the change detection part $dcd_1(\cdot)$ of the "changed" class growing criterion retains the moving objects classification to the "changed" class for a number of frames in which appear to be stationary. Thus, the objects are extracted efficiently, as it is shown by the bottom image of Fig. 6.11. The white curves in the images represent the boundary between the classes "changed" and "unchanged". However, since this method relies on change detection in order to determine the moving objects, the result has to be improved in the case of larger objects motion, in order to be able to cope with occlusions. For that purpose, the colour based objects localization method described in [69] may be applied on the output change detection map of our algorithm.

Finally, in the image of Fig. 6.12 we see the localization result for frame 148 of *"HallMonitor"*, while the plot of the figure refers to the inter-frame difference statistics of frame' s regions. Since the background of the sequence is known, the curve "Measurement" of the plot corresponds to the histogram of region differences between frames 148 and 0. For the same reason, the bag shown in the bottom result of Fig. 6.12, is bounded as "changed". The "growing" of classes is performed without using the "memory" term, because the shape and the position of the two humans of the sequence changes among

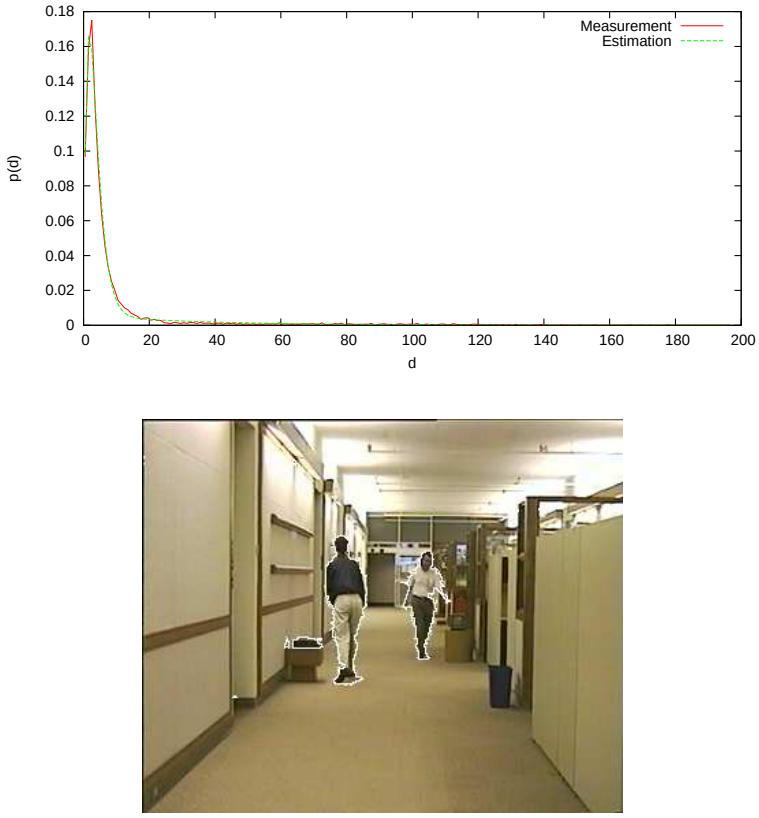


Figure 6.12: Inter-frame regions mean absolute difference statistics (up) and moving objects extraction result (down) for frame 148 of the sequence *"HallMonitor"*.

frames.

6.3 Optical Flow Based Segmentation

In this Section, segmentation as it is applied on the optical flow, computed by successive video frames, is described. It is assumed that the optical flow of regions is better described by affine models. New distance metrics of image motion data are introduced for affine models, as a special case of a novel, generic optimization framework of linear mapping models [70]. We describe below the theoretical background of linear mappings and the way that this theory is used to define distance metrics upon models of data sets, i.e. blocks

or regions. The metrics are used to compute the affine parameters of classes by the block based method of Subsection 2.3.4. Then, sites which belong to one of the classes with high confidence are determined and the flooding algorithms are applied to give the final segmentation result.

6.3.1 Linear Mapping Distance Metrics on Data Sets

Given a finite set of data pairs

$$\mathcal{D} = \{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^{m} \in S \times \mathbb{R}^d$$

where S is an arbitrary set, we seek to model the mapping $\mathbf{x}_k \mapsto \mathbf{y}_k$ using a function $\mathbf{f_w}\colon\ S \to \mathbb{R}^d$ of the form

$$\mathbf{f_w}(\mathbf{x}) = \sum_{i=1}^{n} \beta^{(i)}(\mathbf{x}) w^{(i)} = \begin{pmatrix} \sum_{i=1}^{n} \beta^{(i,1)}(\mathbf{x}) w^{(i)} \\ \sum_{i=1}^{n} \beta^{(i,2)}(\mathbf{x}) w^{(i)} \\ \vdots \\ \sum_{i=1}^{n} \beta^{(i,d)}(\mathbf{x}) w^{(i)} \end{pmatrix}.$$

That is, the model function $\mathbf{f_w}$ is a linear combination of the n vector valued basis functions $\beta^{(i)}\colon\ S \to \mathbb{R}^d$, weighted by the respective scalar parameters $w^{(i)} \in \mathbb{R}$, where $\beta^{(i,l)}(\mathbf{x})$ in the preceding equation refers to the l-th element of the basis function $\beta^{(i)}(\mathbf{x})$. Defining the $d \times n$ *base matrix* $\mathbf{B}(\mathbf{x})$ and the *parameter vector* $\boldsymbol{w} \in \mathbb{R}^n$ as

$$\mathbf{B}(\mathbf{x})_{li} = \beta^{(i,l)}(\mathbf{x})$$

$$\boldsymbol{w} = (w^{(1)} \quad w^{(2)} \quad \dots \quad w^{(n)})^T$$

the model function can be expressed in compact form as

$$\mathbf{f_w}(\mathbf{x}) = \mathbf{B}(\mathbf{x})\boldsymbol{w}.$$

Given a parameter vector $\boldsymbol{w}$ let the *error functional* $E_{\mathcal{D}}(\boldsymbol{w})$ be defined

as

$$E_{\mathcal{D}}(\boldsymbol{w}) = \sum_{(\mathbf{x};\mathbf{y})\in\mathcal{D}} \|\mathbf{y} - \mathbf{B}(\mathbf{x})\boldsymbol{w}\|_2^2.$$

This is easily recognized as a linear least squares optimization problem. Adopting the notation

$$\boldsymbol{\mathcal{B}}_{\mathcal{D}} = \sum_{(\mathbf{x};\mathbf{y})\in\mathcal{D}} \mathbf{B}(\mathbf{x})^T\mathbf{B}(\mathbf{x})$$

$$\psi_{\mathcal{D}} = \sum_{(\mathbf{x};\mathbf{y})\in\mathcal{D}} \mathbf{B}(\mathbf{x})^T\mathbf{y}$$

the error functional can be rewritten as

$$E_{\mathcal{D}}(\boldsymbol{w}) = \sum_{(\mathbf{x};\mathbf{y})\in\mathcal{D}} \|\mathbf{y}\|_2^2 - 2\psi_{\mathcal{D}}^T\boldsymbol{w} + \boldsymbol{w}^T\boldsymbol{\mathcal{B}}_{\mathcal{D}}\boldsymbol{w}.$$

There exists an optimal value for the parameter vector $\boldsymbol{w}$, say $\boldsymbol{w}^*_{\mathcal{D}}$, for which the error functional $E_{\mathcal{D}}(\boldsymbol{w})$ is minimized. The optimal parameter vector is unique, if, and only if, the symmetric, non-negative definite matrix $\boldsymbol{\mathcal{B}}_{\mathcal{D}}$ is non singular. In such a case, $\boldsymbol{w}^*_{\mathcal{D}}$ is given by the system of normal equations

$$\boldsymbol{\mathcal{B}}_{\mathcal{D}}\boldsymbol{w}^*_{\mathcal{D}} = \psi_{\mathcal{D}},$$

which in turn leads to the equation

$$E_{\mathcal{D}}(\boldsymbol{w}^*_{\mathcal{D}}) = \sum_{(\mathbf{x};\mathbf{y})\in\mathcal{D}} \|\mathbf{y}\|_2^2 - \psi_{\mathcal{D}}^T\boldsymbol{w}^*_{\mathcal{D}}$$

Given two data sets $\mathcal{D}_1$, $\mathcal{D}_2$ let $\boldsymbol{w}_1$, $\boldsymbol{w}_2$ denote their corresponding optimal parameter vectors under the same set of basis functions. Let $\mathcal{D}_{12}$ be the data set resulting by the concatenation of $\mathcal{D}_1$, $\mathcal{D}_2$ and $\boldsymbol{w}_{12}$ its optimal parameter vector. In accordance to the similarity influencing factors previously mentioned and the potential use of a compatible metric as a region metric criterion, we consider $\mathcal{D}_1$ and $\mathcal{D}_2$ to be similar if the error norm of their joint optimal modelling $E_{\mathcal{D}_{12}}(\boldsymbol{w}_{12})$ is comparable to the error norms of their individual optimal models $E_{\mathcal{D}_1}(\boldsymbol{w}_1) + E_{\mathcal{D}_2}(\boldsymbol{w}_2)$. We quantify this by introducing

the dissimilarity criterion

$$\mathtt{D}_{12} = E_{\mathcal{D}_{12}}(\boldsymbol{w}_{12}) - E_{\mathcal{D}_1}(\boldsymbol{w}_1) - E_{\mathcal{D}_2}(\boldsymbol{w}_2) \tag{6.16}$$

which is always non negative and expresses the penalty inflicted upon the overall approximation error by the merging of $\mathcal{D}_1$ and $\mathcal{D}_2$.

We ultimately define the distance of $\mathcal{D}_1$ and $\mathcal{D}_2$ to be

$$J^{LM} = \sqrt{\frac{|\mathcal{D}_1| + |\mathcal{D}_2|}{|\mathcal{D}_1||\mathcal{D}_2|}\mathtt{D}_{12}} \tag{6.17}$$

Using the square root of $\mathtt{D}_{12}$ effectively makes the proposed metric to scale linearly with uniform scaling of the $\mathbf{y}$ values. The normalization factor $(|\mathcal{D}_1|+|\mathcal{D}_2|)/(|\mathcal{D}_1||\mathcal{D}_2|)$ makes the metric independent of the overall sample density, while preserving the effect of the individual distributions and the relative density of the two data sets.

This intuitively defined metric reduces to the very simple analytic expression

$$J^{LM} = \|\boldsymbol{\Psi}_{12}(\boldsymbol{w}_2 - \boldsymbol{w}_1)\|_2, \tag{6.18}$$

where $\boldsymbol{\Psi}_{12}$ is a symmetric, positive definite matrix, dependent on the distributions of $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_{12}$, which is defined as

$$\boldsymbol{\Psi}_{12}^2 = \mathbf{R}_1\mathbf{R}_{12}^{-1}\mathbf{R}_2 \qquad \mathbf{R}_l = \frac{1}{|\mathcal{D}_l|}\boldsymbol{\mathcal{B}}_{\mathcal{D}_l}.$$

The matrix $\mathbf{R}_l$ can be interpreted in the case of a scalar base (d=1) as the correlation matrix of the basis used, evaluated at all points of $\mathcal{D}_l$. For a vector valued basis, $\mathbf{R}_l$ equals the sum of all cross-correlation matrices between all n^2 ordered pairs of dimensions of the basis used. In either case, the value of $\mathbf{R}_l$ depends only on the $\mathbf{x}$ values of the elements of P_l and the basis used. We conjecture that the matrix $\boldsymbol{\Psi}_{12}$ conveys all the essential information about the distribution of the elements in $\mathcal{D}_1$, $\mathcal{D}_2$ with respect to the optimization process used. An interesting observation regarding Eq. (6.18) is the separation of the distribution information, described by the matrix $\boldsymbol{\Psi}_{12}$

and the model information, given by the difference $(\boldsymbol{w}_1 - \boldsymbol{w}_2)$, which is not always sufficient to characterize similarity on its own.

6.3.2 Optical Flow Features

Motion based classification is obtained assuming that the optical flow of clusters is better described by the affine model rather than by a constant 2D motion vector. Feature classification is obtained by the method described in Subsection 2.3.4, for block based classification. What differs, is the distance used to estimate the dissimilarity between data blocks or a block and a feature class in k-means, since the novel linear mapping distance, described in the previous Subsection, is used to measure the distance between optical flow data sets. The affine model used in optical flow segmentation is a simple linear map from $\mathbb{R}^2$ to $\mathbb{R}$ described by the respective basis matrix

$$\mathbf{B}_{aff}(x, y) = [\ 1 \quad x \quad y\ ]$$

and is the same for each one of the two optical flow vector components v_x, v_y of image point $s = (x, y)$. Given two data sets of optical flow vectors $\mathcal{D}_1$ and $\mathcal{D}_2$ with optimal parameter vectors

$$\boldsymbol{w}_1 = [\boldsymbol{w}_{1x} \quad \boldsymbol{w}_{1y}]^T$$

and

$$\boldsymbol{w}_2 = [\boldsymbol{w}_{2x} \quad \boldsymbol{w}_{2y}]^T,$$

their distance from Eq. (6.16) becomes

$$\begin{aligned} \mathtt{D}_{12}^{aff} \;=\; & (\boldsymbol{w}_{1x} - \boldsymbol{w}_{2x})^T \boldsymbol{\mathcal{B}}_{\mathcal{D}_1} \boldsymbol{\mathcal{B}}_{\mathcal{D}_{12}}^{-1} \boldsymbol{\mathcal{B}}_{\mathcal{D}_2} (\boldsymbol{w}_{1x} - \boldsymbol{w}_{2x}) + \\ & (\boldsymbol{w}_{1y} - \boldsymbol{w}_{2y})^T \boldsymbol{\mathcal{B}}_{\mathcal{D}_1} \boldsymbol{\mathcal{B}}_{\mathcal{D}_{12}}^{-1} \boldsymbol{\mathcal{B}}_{\mathcal{D}_2} (\boldsymbol{w}_{1y} - \boldsymbol{w}_{2y}) \end{aligned} \tag{6.19}$$

where

$$\boldsymbol{\mathcal{B}}_{\mathcal{D}_m} = \sum_{(x,y)\in\mathcal{D}_l} \mathbf{B}_{aff}(x, y)^T \mathbf{B}_{aff}(x, y)$$

for $m = 1, 2$ and

$$\boldsymbol{\mathcal{B}}_{\mathcal{D}_{12}} = \boldsymbol{\mathcal{B}}_{\mathcal{D}_1} + \boldsymbol{\mathcal{B}}_{\mathcal{D}_2}.$$

Substituting $\mathtt{D}_{12}^{aff}$ in Eq. (6.17) we get the distance J^{aff} used in the case of affine motion models of classes.

6.3.3 Label Propagation

We describe hereafter, the distance that is used to define the probability of pixel s as

$$p_l(\boldsymbol{\xi}(s)) = e^{-d_l^B(s)},$$

which in turn is used to determine initial regions of high confidence and to measure the dissimilarity criterion in the flooding algorithms.

Label propagation in the case of affine modeling of optical flow classes, is based on the statistical description of the 2D residual vectors for the least-squares solution $\boldsymbol{w}^*_{\mathcal{D}_l}$ that is obtained by the affine modeling for each class. Following the terminology of Subsection 6.3.1 and given the data set $\mathcal{D}_l$ of class l and the basis function $\mathbf{B}_{aff}(s)$ for pixel $s = (x, y)$, we define the deviation of s from the optimal model induced by $\mathcal{D}_l$ as

$$\mathbf{r}_l(s) = \mathbf{v}(s) - \mathbf{B}_{aff}(s)\boldsymbol{w}^*_{\mathcal{D}_l} \tag{6.20}$$

We note, that by the theory of linear mappings, the mean of residuals $\mathbf{r}_{s,l}$ is known to be zero in the case of affine models. We further assume that the residuals of each class are Gaussian distributed. Then, the minus log-likelihood distance of a pixel s, with residual feature vector $\boldsymbol{\xi}(s)$ computed by Eq. (6.20), is given by the special case of Eq. (A.7) for $\boldsymbol{\mu}_l = \mathbf{0}$:

$$d_l^B(s) = \frac{1}{2}\ln|\boldsymbol{\Sigma}_l| + \frac{1}{2}\mathbf{r}(s)^T\boldsymbol{\Sigma}_l^{-1}\mathbf{r}(s) \tag{6.21}$$

where $\boldsymbol{\Sigma}_l$ in that case, is the covariance matrix of the residuals of class l.

6.3.4 Experimental Results

Optical flow segmentation has been performed on a subset of the optical flow, ground truth data of [4]. The dataset has been devised to evaluate optical flow computation algorithms. For each sequence, ground truth optical flow for a pair of frames is freely available in this dataset. In Fig. 6.3.4, the first frame of the pair used for optical flow segmentation is shown per sequence. Sequences (a) and (b) of Fig. 6.3.4 are characterized by hidden texture, sequences (c)-(f) are artificially devised synthetic videos and the last example (g) is a stereo pair.

(a) RubberWhale (b) Dimetrodon (c) Grove2

(d) Grove3 (e) Urban2 (f) Urban3

(g) Venus

The 2D motion field for each pair is shown in the left images of each row in Figures 6.14 and 6.15 respectively. The mapping of motion vectors to colours in these images is graphically depicted in Fig. 6.13. Eq. (6.21) is then used to define the distance of pixels against classes. Based on these statistical distances, initial regions of high confidence are determined and

PMLFA and ILFMA are applied to fill the segmentation maps, for each sequence. Segmentation results are satisfactory, although, in some cases affine modelling may not be appropriate to describe the optical flow classes. However, we note that any other model described by the linear mapping theory, could be used as well. An example of such a model, which has been used to describe camera motion, is given in [71]. What changes in that case is the base matrix $\mathbf{B}(\mathbf{x})$. As a last remark, linear mapping models could be used to fit the data of regions that are not described by a constant vector of features, such as regions of objects that their intensity or colour changes gradually due to illumination conditions.

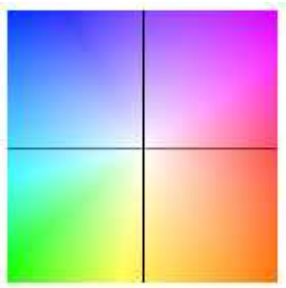
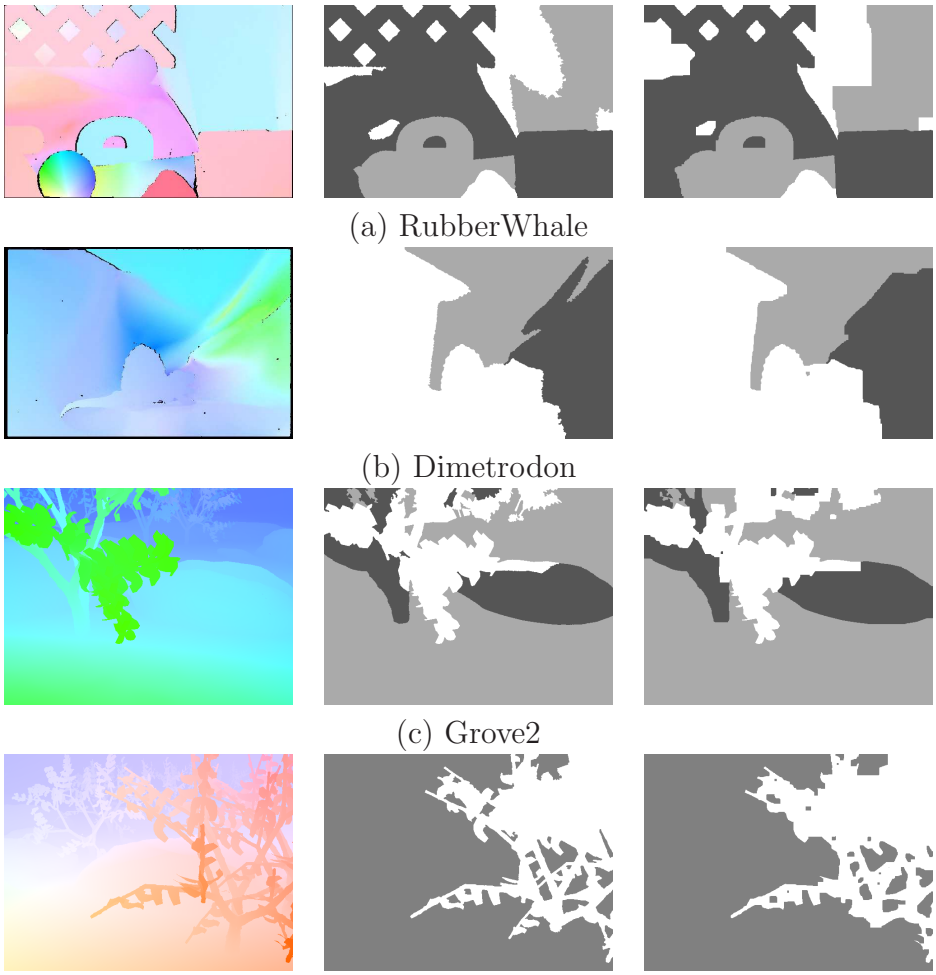


Figure 6.13: Mapping of 2D motion vectors to colours.

The segmentation results for each sequence pair using PMLFA and ILFMA are displayed in the middle and right images respectively, of Figures 6.14 and 6.15. Clustering is achieved by the affine modelling of motion field vectors in an image block basis, using the distance J^{aff} between data sets.



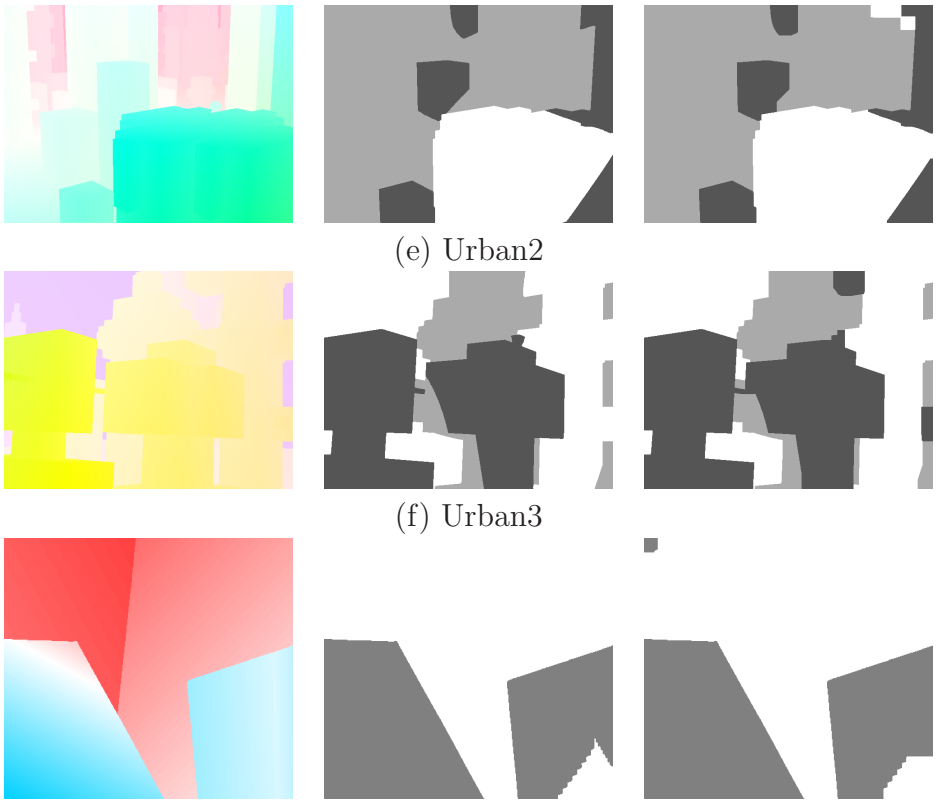
(a) RubberWhale

(b) Dimetrodon

(c) Grove2

(d) Grove3

Figure 6.14: Optical flow segmentation results for the ground truth optical flow dataset of Fig. 6.3.4 ([4]) using affine modeling. In each row: optical flow depicted by colours (left image), segmentation result using PMLFA (middle image) and result using ILFMA (right image), per sequence.



(e) Urban2

(f) Urban3

(g) Venus

Figure 6.15: Optical flow segmentation results for the dataset of [4] using affine modeling (con't). The description is similar to that of Fig. 6.14.

Chapter 7

Left Ventricle Segmentation

7.1 Introduction

In the last decades, computer vision, pattern recognition, image processing, as well as cardiac researchers, have given immense attention to cardiac image analysis and modelling. A recent review of 3D modelling methods of cardiac imagery and their applications is found in [72]. Furthermore, an analysis system based on cardiac Magnetic Resonance (MR) as well as Computed Tomography (CT) images is presented in [73].

In that framework, several approaches have been proposed for the segmentation of myocardium and endocardium of left ventricle, using low and middle level cues. Left ventricle segmentation by the combination of edge, region and shape information is presented in [74] for MR images and the method is extended to CT images in [75].

In the system presented herein, left ventricle endocardium and myocardium detection is based on a Bayesian segmentation framework. Image pixels are initially classified in three classes according to their gray value [75]. The first class corresponds to air-filled lungs and appears dark, the second is the class of myocardium with middle gray values and the last is the class of the blood-filled ventricles with brighter values. Based on the statistical description of classes by histograms of gray values, the initial regions of high confidence are determined per class. Then, the initial regions are grown by

PMLFA. After segmentation, the regions of the third class are interactively presented to the user, to decide the region of endocardium. The smoothed convex hull of the selected region is taken as the endocardium region by the system. The epicardial boundary is then computed using boundary information of the region of the segmentation map, which belongs to the class of myocardium gray values and surrounds the endocardium. Epicardium curve is computed by applying *Weighted Least Squares B-Splines minimization*, on the boundaries of myocardium region. A detailed presentation of splines and their applications is found in [76].

This chapter is organized as follows: in Section 7.2, the Bayesian segmentation framework as it is applied for the needs of left ventricle segmentation, is described. In Sections 7.3 and 7.4 the endocardium region processing and epicardium curve computation are presented respectively. Finally, experimental results are depicted and discussed in Section 7.5 for slices of both End-Diastole (ED) and End-Systole (ES) phases.

7.2 Bayesian Flooding For Left Ventricle Segmentation

7.2.1 Classification

At first a quantization step is applied, because the feature extraction is based on block clustering. As the size of block is limited, for obtaining robust statistical estimations of probability distributions, it is needed to restrict the number of intensity values. Therefore image intensity I is quantized in L gray levels using *k-means* and the $\mathcal{L}_1$ distance ("city block") between level centers and intensity values of pixels. To deal with intensity noise due to image acquisition conditions, initial centers of *k-means* are selected equidistantly in the range $[0, 1, \ldots, I_{av}]$, where I_{av} is the smoothed intensity of image.

Having available the result of intensity quantization I_q, the image is divided in non-overlapping blocks and the histogram of quantized intensity is computed as classification feature, for each block. Then, blocks are grouped

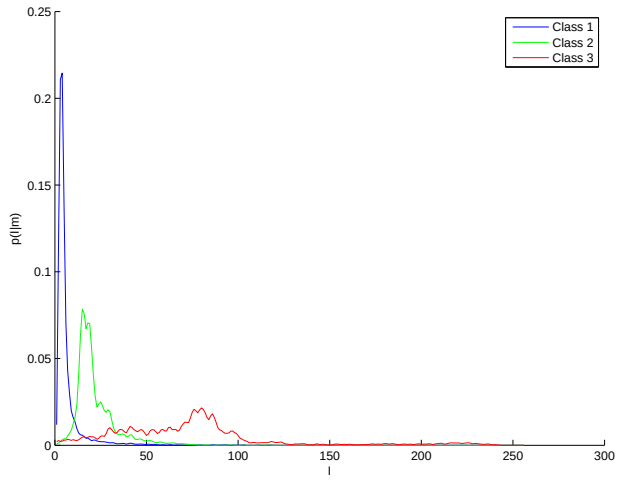


Figure 7.1: Probability distributions $p(I|m)$ $(m = 1, 2, 3)$ for image (a) of Figure 7.2.

in $K = 3$ classes, corresponding to the three classes previously described, using *k-means* and the *Bhattacharyya* distance between block and class center histograms. Since segmentation of left ventricle is crucial for our method, *k-means* is limited on the blocks of the sub-image, which includes the bright regions fairly near the center of the original image, where left ventricle resides. The upper and down block coordinates of this sub-image are determined by statistical criteria upon the pixels of brighter quantization levels.

To get the statistical description for each class m, the normalized histogram $p(I|m) = p_m(I)$ of intensity values is computed using the blocks that have been classified to this class by *k-means*. Here the initial intensity values are considered as the estimation is obtained from the groups of blocks. The probabilistic distance between each pixel s and class m is after all defined as:

$$d_m^B(s) = -\ln \Pr\{m|I(s)\} = -\ln \frac{p_m(I(s))}{\sum_{k=0}^{K-1} p_k(I(s))}$$

Probability distribution functions computed this way for the three classes, are depicted graphically in Figure 7.1 for the image of Figure 7.2(a).

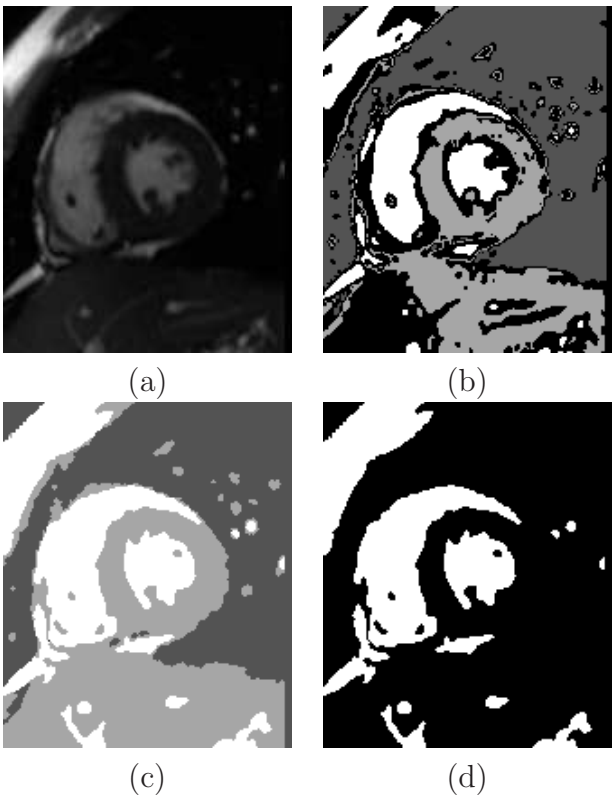


Figure 7.2: Initial regions map (b) and PMLFA segmentation map (c) for image (a).

7.2.2 Label Initialization

The output of *label initialization* is a set of spatially connected regions of pixels, which are classified to class m with high confidence, using statistical tests. These tests are based on the pixel distance d_m^B defined above, as well as on a distance computed in the neighborhood of each pixel s. For each class m, the distances in a disk Δ_r of radius r are averaged, resulting to the metric:

$$d_m^{SB}(s) = \sum_{z \in \Delta_r} g_z d_m^B(s+z),$$

where g_z are the weights of a Gaussian filter. Pixels which belong to class m with high confidence are those for whom $d_m^{SB}(s)$ is minimum compared to

the other classes and furthermore

$$d_m^B(s) < T_{m,1} \qquad \text{or} \qquad d_m^{SB}(s) < T_{m,2}$$

where

$$T_{m,1} = -\ln(\alpha_1 \tanh(0.5 \underset{s\in\Omega}{median}\{d_m^B(s)\} - \beta_1) + \gamma_1) \tag{7.1}$$

$$T_{m,2} = -\ln(\alpha_2 \tanh(0.5 \underset{s\in\Omega}{median}\{d_m^{SB}(s)\} - \beta_2) + \gamma_2) \tag{7.2}$$

and α_i, β_i and γ_i, $(i = 1, 2)$ are predefined constants, while Ω is the image domain. In the plot of Fig. 7.3, threshold $T_{m,2}$ as a function of the median distance is depicted and a similar function is obtained for $T_{m,1}$. As it is evident by the plot, $T_{m,i}$ becomes more restrictive as the median distance gets larger, since the risk of initial labelling for the class m increases with the median distance of pixels from that class.

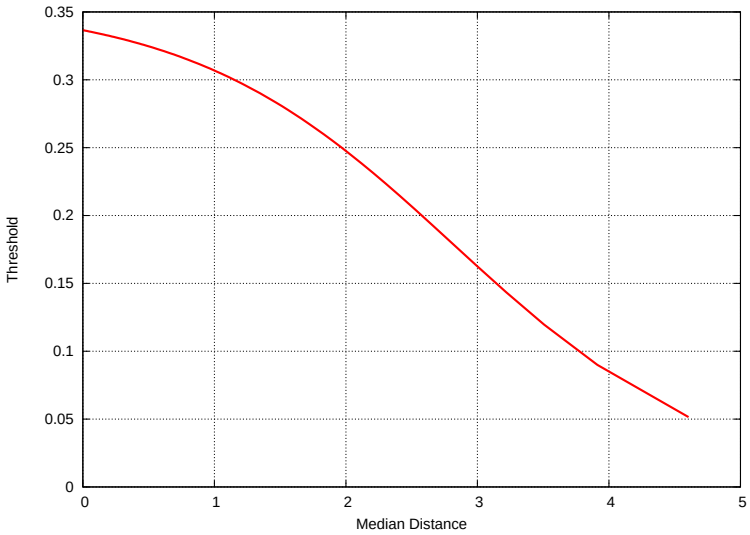


Figure 7.3: Distance threshold $T_{m,2}$ as a function of median class distance, given by Eq. (7.2), for $\alpha_2 = 0.15$, $\beta_2 = 1.5$ and $\gamma_2 = 0.85$.

In image (b) of Figure 7.2, the map of initial regions for image (a) of the figure is depicted. The dark gray regions correspond to the first class of black gray values, the middle brightness regions are those of myocardium class, while the white regions are those of the third class that includes the

bright areas of ventricles. The black regions are those of initially unlabelled pixels.

7.2.3 Priority Multi-Label Flooding

After initialization each class-label is represented by one or more spatially connected regions on image plane. These labelled regions form the initial correctly labelled pixels.

To fill the incomplete mapping of pixels to labels, the contour of each initial region is propagated towards the space of unlabelled image pixels by PMLFA. The output of PMLFA is the complete segmentation map, as it is shown in image (c) of Figure 7.2 for image (a) of the same Figure.

Based on the data set used for tests, Bayesian segmentation succeeds to extract the region of left ventricle in more than 90% of images, whenever such a region really exists. The algorithm fails in cases where the size of this region is small, or not well separated by the surrounding regions due to bad quality of image acquisition or even because of not detectable physical edges between left ventricle and the myocardium area.

To automatically detect and correct this type of failure, the intensity and segmentation map information of previous and next images of the same phase (ES/ED) with that of the problematic image is considered. We denote as

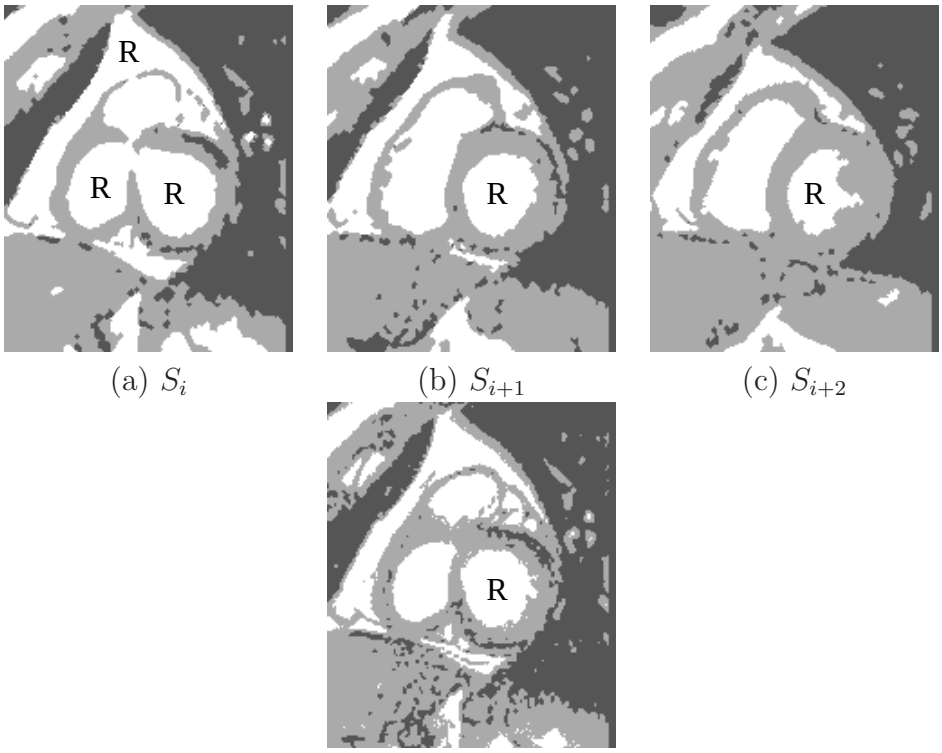
$$\{I_0, I_1, \ldots, I_{n-1}\}$$

the set of n ES or ED intensity images and as

$$\{S_0, S_1, \ldots, S_{n-1}\}$$

the corresponding segmentation maps. For each image i the region R_i of the third, brighter class with minimum centroid distance from the center of image is extracted in each map S_i. Then, the intersection between R_i and R_{i+1} is computed and the area of that intersection $A_{i,i+1}$ is measured for $i = 1, 2, n-1$, where n is the number of images. Images i with left ventricle non-detection are determined whenever $A_{i,i+1}/|R_i| < 0.5$ and $A_{i+1,i+2}/|R_{i+1}| >$

0.5, where $|\cdot|$ denotes pixel cardinality. In the first row of Fig. 7.4, three successive segmentation maps S_i, S_{i+1} and S_{i+2} for the same phase of a patient are depicted. Using the method just described, the map S_i containing the wrongly segmented endocardial region has been successfully detected.

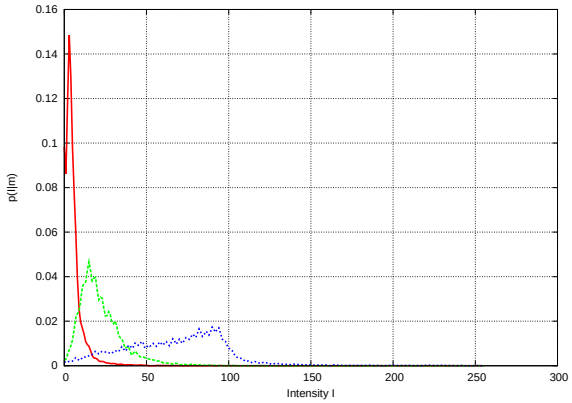


(a) S_i (b) S_{i+1} (c) S_{i+2}

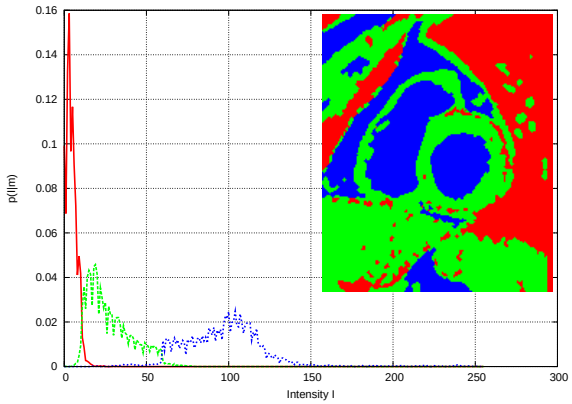
(d) Corrected map S'_i

Figure 7.4: First row: three successive segmentation results in the same phase of a patient. In S_i, the endocardial region R has not been correctly segmented. Second row: the corrected segmentation map for image i.

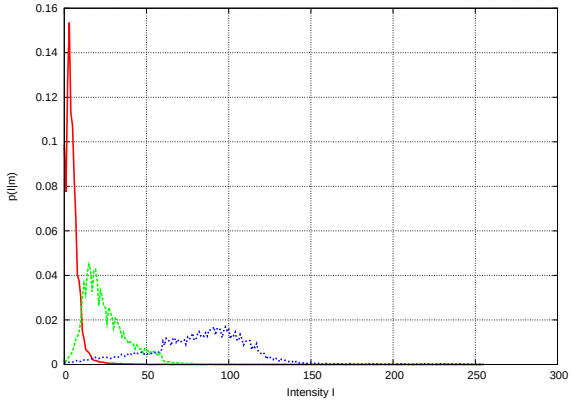
Once segmentation failure is detected for image i, the segmentation map S_i is corrected using the intensity information of images I_i and I_{i+1} together with the label information of map S_{i+1}. First, the probability distribution of each class $p_{m,i}$ is considered again for image i as it is defined in Subsection 7.2.1. Then, the distribution $p'_{m,i+1}$ is extracted for image I_{i+1} using the regions of map S_{i+1} in order to map the pixels to the corresponding class m, for distributions computation. The new corrected distribution $p'_{m,i}$ for i



(a) Distributions $p_{m,i}$

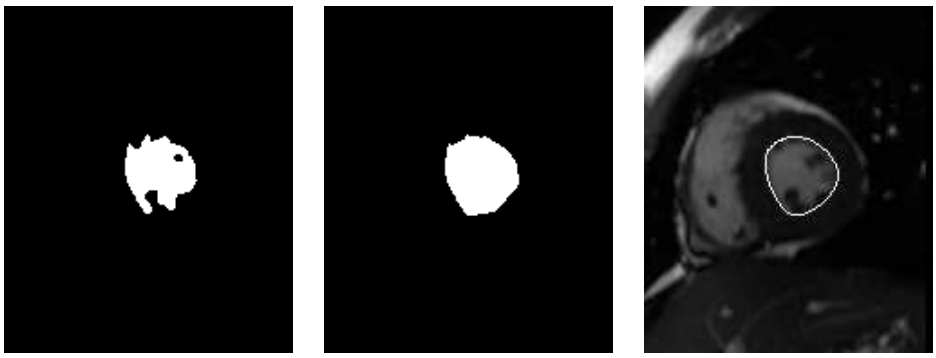


(b) Distributions $p'_{m,i+1}$ computed by the regions of each class in map S_{i+1}



(c) Corrected distributions $p'_{m,i}$

Figure 7.5: Distributions of map correction method for the i^{th} segmentation result shown in Fig. 7.4(a), for $m = 1, 2, 3$.



(a) (b) (c)

Figure 7.6: User-defined R_{endo} (a), convex hull of R_{endo} (b) and smoothed boundary of convex hull (c) for image (a) of Figure 7.2.

is defined to be the average of $p_{m,i}$ and $p'_{m,i+1}$ for each class m and the new corrected segmentation map S'_i is obtained by simply setting each image pixel s to the class-label m with maximum $p'_{m,i}(s)$. In the plots Fig. 7.5(a), (b) the two distributions used to compute the corrected one per class are shown respectively, for the wrongly segmented image of Fig. 7.4(a). The corrected distributions $p'_{m,i}$ for image i is shown in Fig. 7.5(c) and the corrected segmentation map S'_i of Fig. 7.4(a) using this method, is that of Fig. 7.4(d).

7.3 Left Ventricle Endocardial Region Detection

After segmentation, the user is faced with the segmentation results of the third class, as it is shown in image of Figure 7.2(d) and has to decide the region R_{endo} of left ventricle by a "mouse click" in it. The selected region for the image (a) of Figure 7.2 is shown in Figure 7.6(a).

Then, after removing "holes", the system computes the convex hull of R_{endo}, since in many cases the dark papillary muscles are not included in R_{endo} as it is required by the medical physicians. In image (b) of Figure 7.6, the convex hull of the region in Figure 7.6(a) is graphically depicted. The boundary of the corrected, convex region R_{endo} is then traced and smoothed.

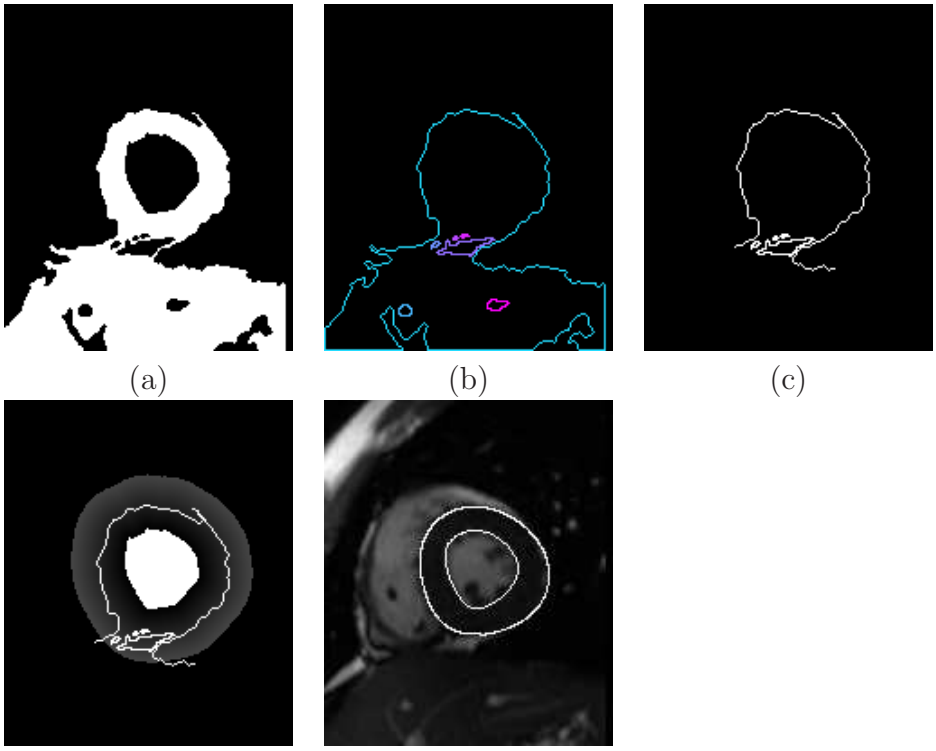


Figure 7.7: Region R_{epi} (a), set $\mathbf{B}$ of R_{epi} (b), boundary points b of $\mathbf{B}^t$, i.e. with $\tau_b < T_b$ (c), points s with $\tau_s < T_s$ (d) and extracted boundary of epicardium (e) for image (a) of Figure 7.2.

Boundary smoothing is performed by considering the boundary coordinates $b = (x, y)$ in the convex plane, after subtracting the centroid of R_{endo}, $c = (x_c, y_c)$:

$$b' = (x - x_c) + \mathbf{i}(y - y_c).$$

The Discrete Fourier Transform (DFT) of boundary points b' is computed and the high frequency content of spectrum is cut off. By adding c to the inverse DFT of the low frequency content of spectrum that remains, we get the spatial coordinates of smoothed boundary for the convex hull R_{endo}. In Figure 7.6(c), the smoothed boundary of the final segmentation result for the left ventricle of the image of Figure 7.2(a) is illustrated.

7.4 Left Ventricle Epicardial Boundary Detection

Detection of the epicardial boundary which surrounds the endocardial area involves the set of closed boundaries of region R_{epi}, which belongs to the second class of intensities and surrounds the convex hull of endocardial region R_{endo}:

$$\mathbf{B} = \{B_1, B_2, \ldots, B_N\}.$$

The boundary between R_{epi} and the convex hull of R_{endo} is not included in $\mathbf{B}$. In image Figure 7.7(a), the region R_{epi} which surrounds the convex hull of R_{endo} is shown and in image Figure 7.7(b) the boundary set $\mathbf{B}$ of R_{epi} is graphically depicted.

Segmentation is performed in two steps. The first one is to determine the set $\mathbf{B}^t$ of boundary segments that are placed in an acceptable distance from the boundary of the detected endocardium and are adequate to form the closed boundary of epicardium. This is achieved by excluding from set $\mathbf{B}$ boundary points b whose Euclidean distance τ_b from the boundary of R_{endo} exceeds a predefined threshold T_b. Definition of T_b is based on observations in the overall, given data set. The boundary set $\mathbf{B}^t$ of R_{epi} of Figure 7.7(a), is depicted in Figure 7.7(c).

Then, epicardial boundary is determined by an algorithm which fits a cubic spline of weighted minimum square error on the selected boundary parts. The computation of weights w_b for the points b in the truncated set $\mathbf{B}^t$ is based on the median distance MED_τ and variance σ_τ^2 from MED_τ. Computation of MED_τ involves the pixels s in R_{epi} whose Euclidean distance τ_s from the boundary of R_{endo} does not exceed a predefined threshold $T_s \leq T_b$. In Figure 7.7(d), the distance τ_s of pixels from the boundary of the convex hull of R_{endo} is shown for $\tau_s < T_s$. The convex hull of R_{endo} is the white region, while pixels with distance greater than T_s are depicted black. Boundary set $\mathbf{B}^t$ of points is also drawn in white. The histogram of τ_s for R_{epi} of Figure 7.7(a) is shown in Figure 7.8. The red bar indicates MED_τ.

MED_τ is a robust estimation of mean Euclidean distance of the left

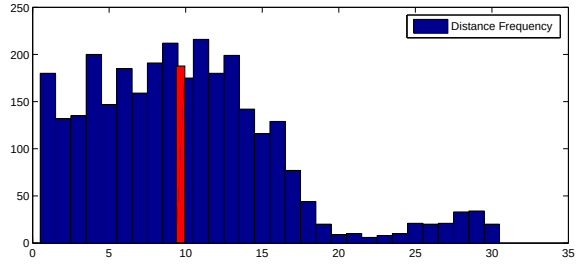


Figure 7.8: Histogram of Euclidean distance τ of pixels from the boundary of R_{endo} for region R_{epi} of Figure 7.7(a). The red bar delineates the estimated median value MED_τ.

myocardial area from R_{endo}, under the presence of segmentation errors and keeping in mind that a closed physical epicardium boundary does not really exist. Computation of σ_τ^2 is restricted to the pixels s of R_{epi} with $\tau_s \leq 2 \cdot MED_\tau$. Estimate σ_τ^2 is used to progressively factor out the impact of boundary points b as their distance τ_b from MED_τ increases. The exact formula of weights is given by equation:

$$w_b = e^{-\frac{(\tau_b - MED_\tau)^2}{\sigma_\tau^2}}. \tag{7.3}$$

Cubic spline minimization takes place in the polar (θ, ρ) space defined by the centroid $c = (x_c, y_c)$ of R_{endo}. Thus, boundary points are first transformed to polar coordinates and are sorted according to their value of θ in the range $[-\pi, \pi)$. Boundary data is reproduced in the range $[\pi, 3\pi)$, to reduce the effect of interpolation that is applied by the spline minimization algorithm on the first and last boundary points. Then, a cubic B-spline curve of t segments is fitted to the boundary points in the range $[-\pi, 3\pi)$, using the weights defined by Eq. (7.3). The resulting boundary function $\hat{\rho} = f(\theta)$, $\theta \in [0, 2\pi]$, is transformed back to Cartesian coordinates. In Figure 7.9, the red curve delineates the computed function $\hat{\rho} = f(\theta)$, for $\theta \in [0, 2\pi]$. The transformed boundary points of Figure 7.7(c) are the blue crosses, while red circles are the *knots* of cubic polynomial segments. In Figure 7.7(e) the final boundaries of left ventricle and epicardium are imposed on the image

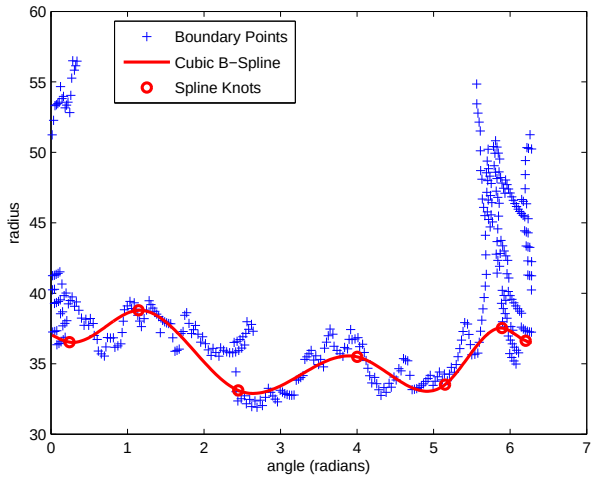


Figure 7.9: Cubic B-Spline weighted least squares minimization for the boundary points of Figure 7.7(c).

of Figure 7.2(a).

7.5 Experimental Results

We present experimental results on MR images for a number of patients and for both ED and ES phases. In the images of Figure 7.10, the resulting boundaries of epicardium and endocardium are depicted for the patient cases (a)-(e) and for ED (first row) and ES phases (second row) respectively. Although the acquisition quality and LV shape vary a lot, the LV segmentation results are stable and accurate. Furthermore, in all cases the papillary muscles are included in the endocardium region.

The not accurately placed epicardial boundary of case (b) is mainly caused by the bad slice acquisition which does not permit the correct discrimination between the first class of black and the second class of myocardium gray values, which in turn leads to the presence of black small blobs into the myocardium area. As a consequence, computation of MED_τ is inaccurate, giving miss-leading weights to the boundary points of myocardium. A situation like that appears rarely and could be avoided with better tuning of

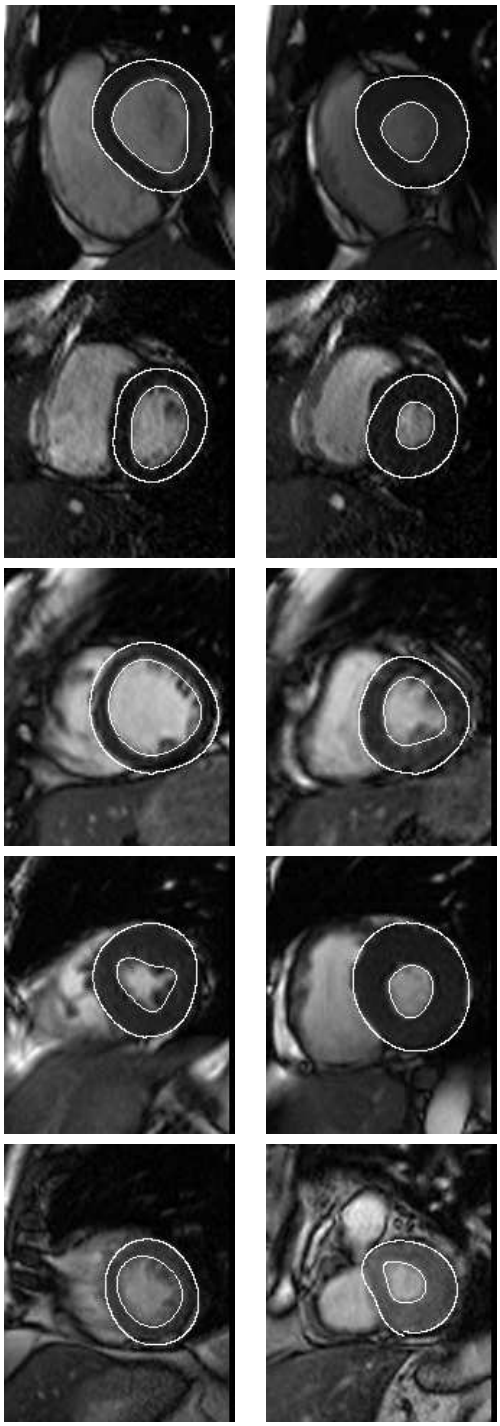


Figure 7.10: LV segmentation results for 5 patient cases, for ED (first column) and ES (second column) phases respectively.

image acquisition parameters.

LV segmentation results for the complete testing data set, are found at http://www.csd.uoc.gr/~grinias/DEMOS/lv_segm.

Appendix A

Probabilistic Distance Metrics

A.1 Probabilistic Distance Metrics on Data Sets

The concept of probabilistic distance derives from the error probability e, which for the two class case is [26]:

$$P_e = \int_{\boldsymbol{\xi}} \min\left(P_1 p_1(\boldsymbol{\xi}), P_2 p_2(\boldsymbol{\xi})\right) d\boldsymbol{\xi} \leq \sqrt{P_1 P_2} \int_{\boldsymbol{\xi}} \sqrt{p_1(\boldsymbol{\xi}) p_2(\boldsymbol{\xi})} d\boldsymbol{\xi} \qquad \text{(A.1)}$$

where $\boldsymbol{\xi}$ is the classification feature vector, $p_1(\boldsymbol{\xi})$, $p_2(\boldsymbol{\xi})$ are the conditional probability density functions $p(\boldsymbol{\xi}|1)$, $p(\boldsymbol{\xi}|2)$ of feature vector and P_1, P_2 are the a priori probabilities for the two classes, respectively. The error due to the overlapping of distributions in feature space is graphically depicted in the plot of Fig. A.1, for an 1D feature space. The area of error is shown yellow in that artificial example.

The error probability will be maximum when the integrand is maximized, that is, when density functions are completely overlapping, and will be zero when the functions $p_m(\boldsymbol{\xi})$ do not intersect at all, for $m = 1, 2$. Thus, the second integral in Eq. (A.1) indicates a measure of the *distance* between the two probability density functions. As a consequence, any "distance" measure J between the two density functions satisfying the conditions

- $J \geq 0$,

- $J = 0$, when $p_m(\boldsymbol{\xi})$ exactly overlap, and
- J is maximum, when $p_m(\boldsymbol{\xi})$ do not overlap at all

can be used as a feature distance criterion.

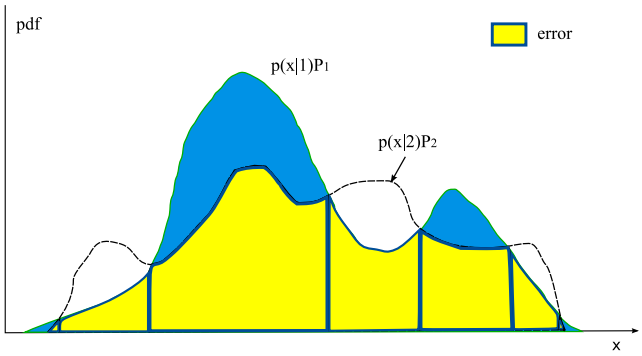


Figure A.1: Statistical error between two distributions for an 1D feature x.

Among several known distance measures between probability distributions, the Bhattacharyya distance is used herein to measure the distance between two feature classes. Bhattacharyya distance is defined as the negative logarithm of the rightmost integral in Eq. (A.1). In the case of discrete random variables the distance has the form

$$J^B(p_1, p_2) = -\ln\left(\sum_i \sqrt{p_1(i)p_2(i)}\right). \tag{A.2}$$

Furthermore, when the class-conditional probability distributions are Gaussian, *i.e.*

$$p_m(\boldsymbol{\xi}) = \left[(2\pi)^N|\boldsymbol{\Sigma}_m|\right]^{-1/2}\exp\left\{-\frac{1}{2}(\boldsymbol{\xi}-\boldsymbol{\mu}_m)^T\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{\xi}-\boldsymbol{\mu}_m)\right\}, \tag{A.3}$$

where N is the dimension of feature vector, $\boldsymbol{\mu}_m$ and $\boldsymbol{\Sigma}_m$ are the mean vector and covariance matrix of the m^{th} class distribution respectively, the distance becomes

$$J^B(p_1, p_2) = \frac{1}{4}(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)^T[\boldsymbol{\Sigma}_1+\boldsymbol{\Sigma}_2]^{-1}(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)+\frac{1}{2}\ln\left[\frac{|\frac{1}{2}(\boldsymbol{\Sigma}_1+\boldsymbol{\Sigma}_2)|}{\sqrt{|\boldsymbol{\Sigma}_1||\boldsymbol{\Sigma}_2|}}\right]. \tag{A.4}$$

Under the assumption of Gaussian distributed and zero mean, uncorrelated features, Bhattacharyya distance takes the simplified form:

$$J^B(p_1, p_2) = \frac{1}{2} \sum_{i=1}^{N} \ln \frac{\sigma_{i,1}^2 + \sigma_{i,2}^2}{2\sigma_{i,1}\sigma_{i,2}}, \tag{A.5}$$

where $\sigma_{i,m}$ is the standard deviation of the i^{th} feature of classes, for $m = 1, 2$.

A.2 Probabilistic Distance Metrics on Data Samples

Taking into consideration all parameters characterizing a feature pattern, a point belongs to a given class, if its distance from the class is minimal. The distance is defined using the likelihood function of the class label. The more probable a class l is, the less the distance is for a given point s, according to the minus logarithm of the likelihood function:

$$d_l^B(s) = -\ln p_l(\boldsymbol{\xi}(s)), \tag{A.6}$$

where $\boldsymbol{\xi}(s)$ is the feature vector of pixel s and p_l is the conditional probability density function of class l for the given features.

If the class-conditional probability distribution of features is Gaussian, Eq. (A.6) takes the form

$$d_l^B(s) = \frac{1}{2} \ln |\boldsymbol{\Sigma}_l| + \frac{1}{2} (\boldsymbol{\xi}(s) - \boldsymbol{\mu}_l)^T \boldsymbol{\Sigma}_l^{-1} (\boldsymbol{\xi}(s) - \boldsymbol{\mu}_l), \tag{A.7}$$

where $\boldsymbol{\mu}_l$ and $\boldsymbol{\Sigma}_l$ are the mean vector and covariance matrix of the class distribution. In the case of Gaussian distributed and zero-mean, uncorrelated features the distance becomes

$$d_l^B(s) = \frac{1}{2} \sum_{i=1}^{N} \left(\frac{\xi_i(s)^2}{\sigma_{i,l}^2} + \ln \sigma_{i,l}^2 \right), \tag{A.8}$$

where N is the number of features and $\sigma_{i,l}$ is the standard deviation of the

i^{th} feature of class l.

Appendix B

Performance Metrics for Segmentation

In this chapter, supervised, performance evaluation metrics are described for segmentation results. The techniques are supervised in the sense that ground truth maps have to be provided in order to measure the quality of the segmentation map derived by algorithms.

B.1 Metrics Based on Consistency Error

Consistency error metrics are region differencing measures. According to [2], given two segmentation maps S_1, S_2 for the same image, the *local refinement error* for site s of image grid Λ is defined as:

$$E(S_1, S_2, s) = \frac{|R(S_1, s) \backslash R(S_2, s)|}{|R(S_1, s)|},$$

where $\backslash$ denotes set difference, $|\cdot|$ the cardinality of its argument and $R(S, s)$ is the region to which site s belongs in segmentation map S. Based on the non symmetric pixel error E, two evaluation metrics are defined in [2]: the *Global Consistency Error*, having the analytic form

$$GCE(S_1, S_2) = \frac{1}{n} \min \left\{ \sum_{s \in \Lambda} E(S_1, S_2, s), \sum_{s \in \Lambda} E(S_2, S_1, s) \right\} \tag{B.1}$$

and the *Local Consistency Error*, given by the equation:

$$LCE(S_1, S_2) = \frac{1}{n} \sum_{s \in \Lambda} \min \{E(S_1, S_2, s), E(S_2, S_1, s)\}, \tag{B.2}$$

where $n = |\Lambda|$.

GCE metric is stronger than LCE, since always holds that $GCE > LCE$ for a pair of maps. Furthermore, GCE heavily penalizes undersegmentation, while does not penalize oversegmentation at all. GCE and LCE range from 0 (highest performance) to 1.

B.2 Probabilistic Rand Index

Denoting as $R(S, s)$ the region of s in map S, as in the previous section, then given two segmentation maps S_1 and S_2, we define the event

$$E_{=}(s_i, s_j) = [R(S_1, s_i) = R(S_1, s_j) \wedge R(S_2, s_i) = R(S_2, s_j)]$$

that sites s_i, s_j belong to the same region in each one of S_1, S_2. Similarly, we denote as

$$E_{\neq}(s_i, s_j) = [R(S_1, s_i) \neq R(S_1, s_j) \wedge R(S_2, s_i) \neq R(S_2, s_j)]$$

the event that the pair of sites belongs to different regions in each map. The Rand Index (RI) [77] between segmentation maps S_1 and S_2 is then defined as:

$$RI(S_1, S_2) = \frac{1}{\binom{n}{2}} \sum_{\substack{s_i, s_j \\ i \neq j}} [\mathbb{I}(E_{=}(s_i, s_j)) + \mathbb{I}(E_{\neq}(s_i, s_j))],$$

where $\mathbb{I}$ is the identity function. Metric RI is the ratio of number of sites having the same relationship, i.e. belonging or not belonging to the same region, in maps S_1 and S_2.

Based on RI, the Probabilistic Rand (PR) index [78] is a non parametric

measure defined as:

$$PR\left(S_{test}, \{S_k\}\right) = \frac{1}{\binom{n}{2}} \sum_{\substack{i,j \\ i<j}} \left[c_{ij}p_{ij} + (1 - c_{ij})(1 - p_{ij})\right]$$

where c_{ij} denotes the event of a pair of sites i and j belonging to the same region in the test image S_{test} and p_{ij} are the expected values of site pairs belonging to the same region. Expected values p_{ij} are computed *on average* using a set of perceptually correct segmentation maps $\{S_k\}$, provided for example by humans for S_{test}.

Advantages as well drawbacks of the index have been extensively discussed in [78]. Its main drawback is that the variation of its value does not scale well with the difficulties in segmentation, caused to algorithms by the diversity in visual image content across the images of a dataset. Thusfore, authors of [78] proposed the normalization of PR index as follows:

$$(Normalized \;\; index) = \frac{(Index) - (Expected \;\; index)}{(Maximum \;\; index) - (Expected \;\; index)},$$

where expected index is computed in the overall set of perceptually correct maps for all images and maximum index is set to the maximum value 1 of PR. Computed this way, the expected index value quantifies the difficulties of the segmentation task for the overall dataset. However, the variation of visual content per image is not captured, although often segmentation algorithms depend exclusively on it and no other knowledge is used.

Based on the same observations, a novel metric is defined using RI. The first difference is that a probabilistic Rand index is defined between a test map S_{test} and ground truth maps $\{S_k\}$, as:

$$PR\left(S_{test}, \{S_k\}\right) = \underset{k}{median}\{RI(S_{test}, S_k)\} \tag{B.3}$$

This definition replaces the mean estimator used to compute PR in [78] and it is in agreement with the fact that a small number of ground truth maps is often available per image.

Furthermore, for each image the expected index of PR for each ground truth map S_k is computed, using maps $\{S_r\}$ that are derived by the quantization of visual content, in a non systematic way. The method described below for extracting the maps, applies as is in the case of one-dimensional features. Principal Components Analysis (PCA) is performed for features in more than one dimensions, such as colour and the principal axis of maximum strength is used to quantize image data in two levels. Quantization is achieved by applying a cut-off threshold to the cumulative distribution function (cdf) of the principal axis. The result of quantization is a binary map. A set of binary maps per image is derived by incrementally varying the cut-off threshold on the cdf. The connected components of binary maps represent possible randomly generated maps, denoted as $\{S_r\}$, resulting by the visual content of images.

The method is depicted for the colour image of Fig. B.1(a). The cdf of the maximum deviation axis after PCA is given in the plot of Fig. B.1(b). The 9 cut-off thresholds that define the two levels used for the quantization of the feature axis, are also depicted as red lines in that plot. Each threshold is defined at cdf values $0.1 \times r$, for $r = 1, 2, \ldots, 9$ and for each threshold a segmentation map S_r is obtained. The set $\{S_r\}$ of segmentation maps is augmented to include the trivial map of one region consisting of the whole image. The 9 non trivial segmentation maps derived by the method are depicted in Fig. B.2, in pseudocolour.

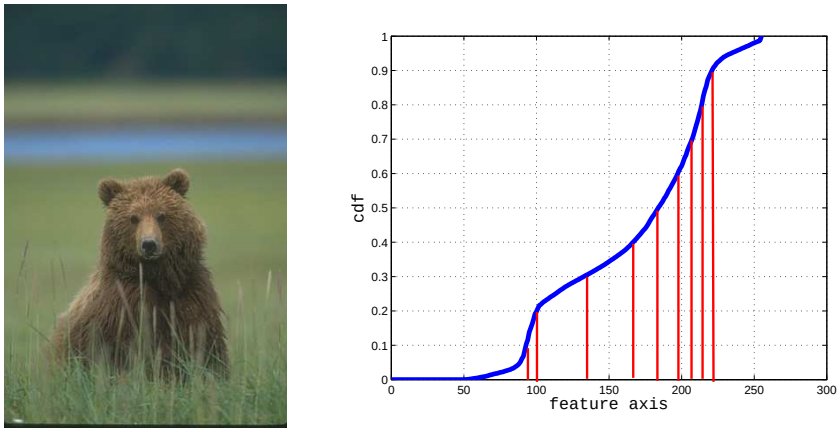
The expected index for each ground truth segmentation S_k is then computed as:

$$EPR(S_k) = PR\left(S_k, \{S_r\}\right),$$

which is a quantitative metric of how well the ground truth segmentation map reflects the visual content of the specific image. The normalized PR index for map S_{test}, compared to the reference map S_k is given by equation:

$$NPR\left(S_{test}, S_k\right) = \frac{RI(S_{test}, S_k) - EPR(S_k)}{1 - EPR(S_k)} \tag{B.4}$$

and measures how better or worse segmentation map S_{test} is, compared to



(a) (b)

Figure B.1: (a) Original colour image and (b) 9 quantization cut-off thresholds, defined on the cdf of maximum strength axis, which is derived by the PCA of colour data.

S_k according to the hardness of segmentation task of the given image.

PR value ranges from 0 to 1 (highest performance) and NPR ranges from $-\infty$ to 1 (highest performance).

Figure B.2: Visual content related random maps in pseudocolour, generated for the colour image of Fig. B.1(a)

Author's Publications

Parts of the work presented in this thesis have already been published to international scientific journals and conferences as follows:

Journal Publications

- I. Grinias, N. Komodakis and G. Tziritas, "Bayesian Flooding and MRF based Minimization for Image Segmentation," CVIU, submitted.
- P. Daras, I. Kompatsiaris, I. Grinias, G. Akrivas, G. Tziritas, S. Kolias and M. G. Strintzis, "MPEG-4 Authoring Tool using Moving Object Segmentation and Tracking in Video Shots," EURASIP Journal on Applied Signal Processing, No. 9, pp. 861-877, August 2003.
- E. Sifakis, I. Grinias, G. Tziritas, "Video Segmentation Using Fast Marching and Region Growing Algorithms," EURASIP Journal on Applied Signal Processing, No. 4, pp. 379-388, April 2002.
- I. Grinias and G. Tziritas, "A semi-automatic seeded region growing algorithm for video object localization and tracking," Signal Processing: Image Communication, Vol. 16, No. 10, pp. 977-986, August 2001.

Conference Publications

- C. Panagiotakis, I. Grinias and G. Tziritas, "MINMAX Video Summarization under Equality Principle," International Workshop on Multimedia Signal Processing (MMSP '07), Chania, Greece, October 2007.

- I. Grinias and G. Tziritas, "Foreground Object Localization Using A Flooding Algorithm Based On Inter-Frame Change And Colour," IEEE International Conference On Advanced Video and Signal Based Surveillance (AVSS '07), London, UK, September 2007.
- I. Grinias, N. Komodakis and G. Tziritas, "Bayesian Region Growing and MRF-based Minimization for Texture and Colour Segmentation," Workshop on Image Analysis For Multimedia Interactive Services (WIAMIS '07), Santorini, Greece, June 2007.
- I. Grinias and G. Tziritas, "Bayesian High Priority Region Growing for Change Detection," European Signal Processing Conference (EUSIPCO 06), Florence, Italy, September 2006.
- C. Panagiotakis, I. Grinias, and G. Tziritas, "Automatic Human Motion Analysis and Action Recognition in Athletics Videos," European Signal Processing Conference (EUSIPCO 06), Florence, Italy, September 2006.
- I. Grinias and G. Tziritas, "Region-Level Moving Object Segmentation by Graph Labeling," European Signal Processing Conference (EUSIPCO 05), Antalya, Turkey, September 2005.
- I. Grinias and G. Tziritas, "Robust Pan, Tilt and Zoom Estimation," International Conference on Digital Signal Processing (DSP2002), Santorini, Greece, July 2002.
- I. Grinias, Y. Mavrikakis and G. Tziritas, "Region growing color image segmentation applied to face detection," International Workshop on Very Low Bitrate Video Coding (VLBV 01), Athens, Greece, October 2001.
- E. Sifakis, I. Grinias and G. Tziritas, "Video Segmentation Using Fast Marching and Region Growing Algorithms," Workshop on Image Analysis For Multimedia Interactive

Services (WIAMIS 01), Tampere, Finland, May 2001 (Invited).

- I. Grinias and G. Tziritas, "Motion segmentation and tracking using a seeded region growing method," European Signal Processing Conference (EUSIPCO 98), Rhodes, Greece, September 1998.

Bibliography

[1] S. Liapis, E. Sifakis, and G. Tziritas, "Colour and texture segmentation using wavelet frame analysis, deterministic relaxation and fast marching algorithms," *Journal of Visual Communication and Image Representation*, vol. 15, no. 1, pp. 1–26, Mar. 2004. vii, 69, 71

[2] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics," in *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, July 2001, pp. 416–423. viii, 71, 75, 149

[3] E. Maggio and A. Cavallaro, "Hybrid particle filter and mean shift tracker with adaptive transition model," in *Proc. of IEEE Int. Conference on Acoustics, Speech and Signal Processing (ICASSP 2005)*, Philadelphia, Mar. 2005, pp. 221–224. ix, 103, 106

[4] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," in *Proc. Eleventh IEEE International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil, Oct. 2007. x, 124, 126, 127

[5] T. Randen and J. H. Husoy, "Filtering for texture classification: A comparative study," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 571–582, Apr. 1999. xiii, 15, 67

[6] 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, pp. 1147–1169, Dec. 1999. 1, 95

[7] Y. Boykov and G.Funka-Lea, "Graph cuts and efficient n-d image segmentation," *Intern. Journal of Computer Vision*, vol. 70, pp. 109–131, 2006. 2

[8] J. F. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, pp. 679–698, 1986. 3

[9] R. Adams and L. Bischof, "Seeded region growing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 641–647, June 1994. 3, 58, 95, 96, 110, 115

[10] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *Intern. Journal of Computer Vision*, vol. 22, pp. 61–79, 1997. 3

[11] N. Paragios, *Geodesic Active Regions and Level Set Methods: Contributions and Applications in Artificial Vision.* Phd. Thesis, Ecole Superiore en Science Informatique, Universite de Nice-Sophia Antipolis/I.N.R.I.A., 2000. 3

[12] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, pp. 888–905, Aug. 2000. 3, 77

[13] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *Intern. Journal of Computer Vision*, vol. 59, pp. 167–181, Sept. 2004. 3, 73, 77

[14] R. Zabih and V. Kolmogorov, "Spatially coherent clustering using graph cuts," in *CVPR 2004*, pp. 437–444. 3

[15] J. Ding, R. Ma, and S. Chen, "A Scale-Based Connected Coherence Tree Algorithm for Image Segmentation," *IEEE Trans. Image Processing*, vol. 17, pp. 204–216, Feb. 2008. 3, 74, 77

[16] A. Ion, W. Kropatsch, and Y. Haxhimusa, "Considerations regarding the minimum spanning tree pyramid segmentation method," in *IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition, LNCS, Vol. 4109, Springer*, 2006. 3, 53

[17] A. Falcao, J. Stolfi, and R. Lotufo, "The image foresting transform: Theory, algorithms, and applications," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, pp. 19–29, Jan. 2004. 4

[18] J. Matas and J. Kittler, "Spatial and feature space clustering," in *6th Int. Conf. on Computer Analysis of Images and Patterns*, Sept. 1997. 4

[19] G. Grimmett, *Percolation*. Springer-Verlang, 1999. 4, 7, 31

[20] J. Fan, G. Zeng, M. Body, and M.-S. Hacid, "Seeded region growing: An extensive and comparative study," *Pattern Recognition Letters*, vol. 26, no. 8, pp. 1139–1156, 2005. 4

[21] F. Kovács, C. Legány, and A. Babos, "Cluster Validity Measurement Techniques," in *Proc. 6th International Symposium of Hungarian Researchers on Computational Intelligence*, 2005. 4, 28

[22] A. Jain and F. Farrokhnia, "Unsupervised Texture Segmentation using Gabor Filters," *Pattern Recognition*, vol. 24, pp. 1167–1186, Dec. 1991. 15

[23] P. P. Rangu and B. Yegnanarayana, "Segmentation of Gabor filtered textures using deterministic relaxation," *IEEE Trans. Image Processing*, vol. 5, pp. 1625–1636, Dec. 1996. 15

[24] A. Laine and J. Fan, "Frame representation for texture segmentation," *IEEE Trans. Image Processing*, vol. 5, pp. 771–780, May 1996. 15

[25] M. Unser, "Texture classification and segmentation using wavelet frames," *IEEE Trans. Image Processing*, vol. 4, pp. 1549–1560, Nov. 1995. 15, 25, 66

[26] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. J. Wiley and sons, 2001. 16, 145

[27] E. Levina and P. Bickel, "The Earth Mover's distance is the Mallows distance: some insights from statistics," in *Proc. 8th International Conference on Computer Vision (ICCV' 01)*, vol. 2, 2001, pp. 251–256. 16

[28] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999. 20

[29] S. Salvador and P. Chan, "Determining the Number of Clusters/Segments in Hierarchical Clustering/Segmentation Algorithms," in *Proc. 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2004)*, 2004, pp. 576–584. 21, 23

[30] N. Komodakis, N. Paragios, and G. Tziritas, "Clustering via LP-based Stabilities," in *Neural Information Processing Systems Conference*, Dec. 2008. 23, 78

[31] L. Kauffman and P. Rousseeuw, *Finding Groups in Data. An Introduction to Cluster Analysis.* John Willey and Sons, 1990. 26, 27

[32] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "Cluster Validity Methods: Part I," *ACM SIGMOD Record*, vol. 31, no. 2, pp. 40–45, June 2002. 28

[33] ——, "Cluster Validity Methods: Part II," *ACM SIGMOD Record*, vol. 31, no. 3, pp. 19–27, Sept. 2002. 28

[34] D. Stauffer and A. Aharony, *Introduction to Percolation Theory.* Taylor and Francis, 2003. 32, 34, 37

[35] J. Hoshen, "On the application of the enhanced Hoshen-Kopelman algorithm for image analysis," *Pattern Recognition Letters*, vol. 19, pp. 575–584, 1998. 32

[36] M. Newman and R. Ziff, "Fast Monte Carlo Algorithm for Site or Bond Percolation," *Physical Review E*, vol. 64, no. 016706, 2001. 32, 33

[37] R. Tarjan, "Efficiency of a Good But Not Linear Set Union Algorithm," *Journal of the ACM*, vol. 22, no. 2, pp. 215–225, Apr. 32

[38] H. Stanley, "Cluster Shapes at Percolation Threshold-Effective Cluster Dimensionality and its Connection with Crirical-Point Exponents,"

Journal of Physics A, Mathematical and General, vol. 10, no. 11, pp. 211–220, 1977. 37

[39] T. H. Cormen, C. H. Leiserson, R. L. Riverst, and C. Stein, *Introduction to algorithms*, 2nd ed. The MIT Press, 2001. 57, 62

[40] D. Wilkinson and J. Willemsen, "Invasion percolation: a new form of percolation theory," *Journal of Physics A, Mathematical and General*, vol. 16, no. 14, pp. 3365–3376, Oct. 1983. 57

[41] A.-L. Barabási, "Invasion percolation and global optimization," *Physical Review Letters*, vol. 76, no. 20, pp. 3750–3753, May 1996. 57

[42] R. Glantz and M. Hilpert, "Invasion percolation through minimum-weight spanning trees," *Physical Review E*, no. 031128, 2008. 57

[43] F. Meyer, "Topographic distance and watershed lines," *Signal Processing*, vol. 38, pp. 113–125, 1994. 57

[44] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 583–598, 1991. 57

[45] S. A. Hojjattolesami and J. Kittler, "Region growing - a new approach," *IEEE Trans. Image Processing*, vol. 7, no. 7, pp. 1079–1084, July 1998. 58

[46] N. Komodakis and G. Tziritas, "Approximate labeling via graph cuts based on linear programming," *IEEE Trans. Pattern Anal. Machine Intell.*, pp. 1436–1453, Aug. 2007. 60

[47] S. Belongie, C. Carson, H. Greenspan, and J. Malik, "Color- and texture-based image segmentation using EM and its application to content based image retrieval," in *Proc. IEEE International Conference of Computer Vision (ICCV'98)*, 1998. 66

[48] K. Skretting and J. H. Husoy, "Texture classification using sparse frame-based representations," *EURASIP Journal on Applied Signal Processing*, 2006. 67, 69

[49] Y. Deng and B. S. Manjunath, "Unsupervised Segmentation of Color-Texture Regions in Images and Video," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, no. 8, Aug. 2001. 73

[50] H. Zhang, J. Fritts, and S. Goldman, "Image segmentation evaluation: A survey of unsupervised methods," *Computer Vision and Image Understanding*, vol. 110, pp. 260–280, 2008. 74

[51] T. N. Pappas, J. Chen, and D. Depalov, "Perceptually based techniques for image segmentation and semantic classification," *IEEE Communications Magazine*, Jan. 2007. 90

[52] C. Garcia and G. Tziritas, "Face detection using quantized skin color regions merging and wavelet packet analysis," *IEEE Transactions on Multimedia*, vol. 1, no. 3, pp. 264–277, 1999. 90

[53] T. Sikora, "The MPEG-4 video standard verification model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 1, pp. 19–31, 1997. 95

[54] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," vol. 13, no. 7, pp. 560–576, 2003. 95

[55] B. S. Manjunath, P. Salembier, and T. Sikora, *Introduction to MPEG-7: Multimedia Content Description Interface*. John Willey and Sons, 2002. 95

[56] T. P. Chen, H. Haussecker, A. Bovirin, R. Belenov, K. Rodyushkin, A. Kuranov, and V. Eruhimov, "Computer vision workload analysis: Case study of video surveillance systems," *Intel Technology Journal*, vol. 9, no. 2, May 2005. 95

[57] C. Stauffer and E. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 8, pp. 747–757, 2000. 95

[58] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Intern. Journal of Computer Vision*, vol. 4, pp. 7–38, 1990. 95

[59] N. Paragios and R. Deriche, "Geodesic active contours and level sets for the detection and tracking of moving objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 266–280, Mar. 2000. 95

[60] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *IEEE Trans. Image Processing*, vol. 13, no. 11, Nov. 2004. 95

[61] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov, "Bilayer segmentation of live video," in *CVPR 2006*, New York, USA. 95

[62] E. Sifakis, I. Grinias, and G. Tziritas, "Video segmentation using fast marching and region growing algorithms," *EURASIP Journal on Applied Signal Processing*, pp. 379–388, Apr. 2002. 95, 97, 98

[63] E. Sifakis and G. Tziritas, "Moving object localisation using a multi-label fast marching algorithm," *Signal Processing: Image Communication*, vol. 16, no. 10, pp. 963–976, 2001. 96

[64] E. Sifakis and G. Tziritas, "Fast marching techniques for visual grouping," in *Geometric Level Set Methods in Imaging, Vision and Graphics*, S. Osher and N. Paragios, Eds. Springer Verlag, July 2003. 96

[65] N. Paragios and G. Tziritas, "Adaptive detection and localization of moving objects in image sequences," *Signal Processing: Image Communication*, vol. 14, no. 4, pp. 277–296, 1999. 97

[66] A. A. Alatan, L. Onural, M. Wollborn, R. Mech, E. Tuncel, and T. Sikora, "Image sequence analysis for emerging interactive multimedia services-the European COST 211 framework," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 802–813, Nov. 1998. 104

[67] I. Patras, E. Hendriks, and R. Lagendijk, "Video Segmentation by Map Labeling of Watershed Segments," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 3, pp. 326–332, Mar. 2001. 104

[68] R. Fablet, P. Bouthemy, and M. Gelgon, "Moving object detection in color image sequences using region-level graph labeling," in *Proc. 6th IEEE Int. Conf. on Image Processing, ICIP' 99*, Kobe, Oct. 1999. 104

[69] E. Sifakis, C. Garcia, and G. Tziritas, "Bayesian level sets for image segmentation," *Journal of Visual Communication and Image Representation*, vol. 13, pp. 44–64, 2002. 107, 109, 110, 117

[70] E. Sifakis, "Optimization using affine models for classification and clustering," *non-archived report*, 2002. 118

[71] I. Grinias and G. Tziritas, "Robust Pan, Tilt and Zoom Estimation," in *International Conference on Digital Signal Processing (DSP2002)*, Santorini, Greece, July 2002. 125

[72] A. F. Frangi, W. J. Niessen, and M. A. Viergever, "Three-Dimensional Modeling for Functional Analysis of Cardiac Images: A Review," *IEEE Trans. Med. Imag.*, vol. 20, no. 1, pp. 2–25, Jan. 2001. 129

[73] T. O'Donnel, G. Funka-Lea, H. Tek, M.-P. Jolly, M. Rasch, and R. Setser, "Comprehensive Cardiovascular Image Analysis Using MR and CT at Siemens Corporate Research," *International Journal of Computer Vision*, vol. 70, no. 2, pp. 165–178, 2006. 129

[74] M.-P. Jolly, "Combining Edge, Region, and Shape Information to Segment the Left Ventricle in Cardiac MR Images," in *Proc. Medical Image Computing and Computer-Assisted Intervention*, Utrecht, The Netherlands, 2001, pp. 482–490. 129

[75] ——, "Automatic Segmentation of the Left Ventricle in Cardiac MR and CT Images," *International Journal of Computer Vision*, vol. 70, no. 2, pp. 151–163, 2006. 129

[76] R. Bartles, J. Beatty, and B. Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann Publishers, 1987. 130

[77] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, pp. 846–850, 1971. 150

[78] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "Toward Objective Evaluation of Image Segmentation Algorithms," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 29, no. 6, pp. 929–944, June 2007. 150, 151