University of Southampton

A Content-based Analysis of Craquelure Patterns in Paintings

by

Fazly Salleh Abas

A mini-thesis submitted
for transfer of registration from
M.Phil to Ph.D

in the
Faculty of Engineering and Applied Science
Department of Electronics and Computer Science

supervisor: Dr. Kirk Martinez

October 2002

Abstract

The advent of multimedia technology has offer new dimension to computerized applications. Art-based applications are among those which have and will continue to benefit from this advancement. With the ever growing size and variety of accessible data across museum collections, the need for flexible and efficient data retrieval is growing at an alarming rate. Content-based image retrieval (CBIR) and analysis is getting a lot of attention from museums and art institutions.

One of the image-based requirements from museums, is to automatically classify craquelure (cracks) in paintings for the purpose of aiding damage assessment. Craquelure in paintings can be an important element in judging authenticity, use of material as well as environmental and physical impact because these can lead to different craquelure patterns. Mass screening of craquelure patterns will help to establish a better platform for conservators to identify cause of damage. As a way of performing such action, a content-based approach is seen as an appropriate path.

This dissertation uncovers the main issues behind content-based craquelure analysis. The important steps namely, crack enhancement and detection are explained. We also implemented a chain-code based craquelure structuring to allow efficient feature extraction. A hierarchical craquelure feature representation is also developed in order to view features at multiple structural scale. Experiment results are presented to show the discriminating power of the features. Early strategies towards classifying the crack patterns are also described.

Acknowledgements

I would like to thank my supervisor, Dr. Kirk Martinez, for his assistance these last two years. I am also grateful to the Sultan Iskandar Foundation of Malaysia for funding me. Special thanks goes to my parents for their love and support throughout my whole university career.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The rapidly increasing size of digital storage space and the advent of the *World Wide Web* (WWW) have made storage and retrieval of digital information possible at a phenomenal rate, almost unimaginable just a decade ago. Information may range from pictorial data such as images or video sequences, as well as synthetic illustrations, diagrams, charts or computer aided graphics. In short, a huge amount of data exists with increasing diversity. These digital data are not just small chunks of several kilobytes, in fact, each constitutes a massive amount of storage space. Based on that fact, the issue of efficient information storage and retrieval is one issue that certainly needs thorough attention. This information will be less meaningful to users unless they are organized in a systematic way to allow efficient browsing, searching and retrieval.

Following the revolutionary trend, museums and art galleries are beginning to digitize their collections not just to make them available on the web for the public but also for internal use within the museums' or galleries' own environment. Digitizing the collection means providing a faster and more efficient way of recording what is available, thus giving a new dimension into methods of information retrieval within the environment itself. Instead of storing the information in a traditional manner, the ability to store them digitally opened the path for further manipulation of the technology where digital preservation and restoration can play their parts. Most collections of paintings and artefacts originated centuries ago and are in need for preservation and restoration to make sure that their physical appearances are maintained. Manual recording and detecting of aging artefacts seems far from efficient with the ever increasing collection and electronic-based approaches seems to be the best choice.

Driven by the availability of effective electronic imaging tools, image processing techniques have now been implemented for analysis, preservation and restoration of artwork. We have been witnessing significant growth in the number of research done on image processing related to arts ranging from quality evaluation of art images [1], image processing tool for art analysis [2], virtual enhancement as well as restoration [3, 4, 5], image retrieval [6, 7, 8, 9] and as an aid for conservation [10]. As tools for artwork restoration, image processing techniques generally serve two main purposes. Firstly, it can be used as a guide for actual restoration (physical restoration) of artwork or in other words, a computer-guided restoration. Secondly, it can be used as a tool to produce a digitally restored version of the original physical artwork (virtual restoration) [5].

We have witnessed rapid advancement in CBR-related research for the past few years and this is due to the rapid increase in the size of digital image collections [11]. These advancement allows efficient browsing, searching and retrieval. Since the early 1990s, content-based image retrieval (CBIR) has become a very active research area. Many retrieval systems, both commercial and research have been built. Most image retrieval systems support one or more of the following options [12]:

- random browsing
- search by example
- search by sketch
- search by text (including keyword or speech)
- navigation with customised image categories

Among the notable CBIR systems are QBIC [13], MARS [14] and NeTra [15]. The Artiste project aims to provide access across museum collections using metadata as well as CBR of image data. One of the image-based requirements, which came from the Uffizi Gallery in Florence, is to automatically classify craquelure in paintings to aid in damage assessment. Craquelure in paintings can also be used for other research [16]. It can be a very important element in judging authenticity, use of material or environmental and physical impact because these can lead to different craquelure patterns. Although most conservation of fine artwork relies on manual inspection of deterioration signs, the ability to screen the whole collection semi-automatically is believed to be a useful contribution to preservation. Crack formations are influenced by factors including aging and physical impacts which also relate to the wooden framework of the paintings. It is hoped that the mass screening of craquelure patterns will help to establish a better platform for conservators to identify the cause of damage.

In this research, we hope to integrate the functionalities of CBIR into the analysis of craquelure patterns in paintings for the purpose of aiding conservation.

1.1 Description of the Problem

Figure 1.1 shows an example of cracks in paintings as an introduction to the type of image being discussed. As can be seen the crack patterns are clearly visible and can be easily identified by the human vision system. As in the figure, the cracks are represented by dark pixels while the background is represented by the brighter ones. Obviously, as far as the human visual system (HVS) is concern, segmenting the appropriate crack affected regions does not pose a problem as long as the image is not highly distorted by noise. To automatically segment the image within a certain tolerable error range is quite problematic especially when dealing with large collection of crack images with different levels of illumination, contrast, noise and intensity. Considering the large amount of collection, it is difficult to monitor every single painting to spot cracks. Digitization of painting collection including X-radiographs of paintings serves the need for a content-based approach in analyzing the crack patterns. X-ray images in this case tend to show cracks very well as the details in the paint layer are suppressed.

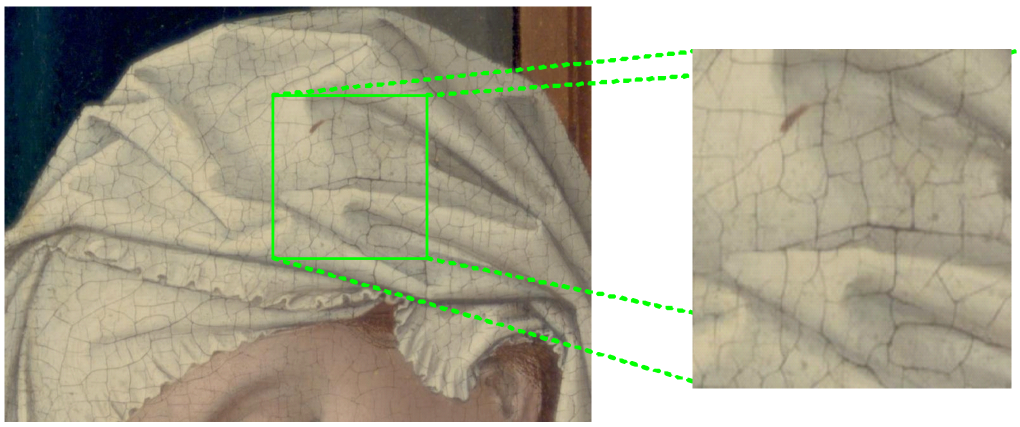


Figure 1.1: Example of a painting with cracks.

A bigger challenge after the detection stage is to extract meaningful features from the detected cracks for classification purpose. "Meaningful" in this scope refers to the capability of each feature to distinguish a particular type of crack from another. Generally speaking, cracks can be classified into many types depending on how they are observed. As mentioned before, cracks can be used to judge authenticity and that is the case in [16] where cracks are classified into 4 different categories namely, French, Flemish, Dutch and Italian. The relevant classes in which we introduce in our research takes into account the probable effect physical or environmental factors might impose upon the paint layer. Support structures such as wooden stretcher bars, nails, wedges and wooden joints are believed to introduce different pattern of cracks. The formation of cracks due to those physical factors introduce problems in a way that they remain undetected for a long period of time. Although the support structures are meant to facilitate restoration, their purpose is not fully justified since they contribute to the formation of cracks on the paint layer.

Based on observations, different crack patterns are caused by the type of support structure used by conservators. The common ones are rectangular, circular, spider-web, unidirectional and random cracks. A large portion of this project is dedicated for the highly challenging task of automatic classification of crack patterns into these 5 classes.

1.2 Overview

This dissertation will cover the initial work and literature review that have been done towards achieving the objective of the whole project. The research presented here generally falls into 3 parts - craquelure detection, high-level feature extraction and data classification. The second part, high-level feature extraction occupies the majority of the dissertation due to its complexity and importance.

Chapter 2 describes craquelure from a conservator's point of view, looking at its appearances, how it forms and its relation with art conservation. The conceptual background of the project is explained in this chapter. Questions are also asked about the possibility of implementing CBR functionalities for the purpose of assisting conservation work. A content-based approach towards analyzing craquelure is seen as the vision of the project looking into possible application scenarios.

Chapter 3 touches on the technical side of the project by first reviewing common line detectors. Further in the chapter, we explain the algorithm we used to enhance crack patterns using morphological top-hat operators. We then describe the technique we use to segment the crack patterns using variable thresholding technique which uses grid-based automatic thresholding approach. Some post-detection stages are also explained before experimental results are presented and discussed.

Chapter 4 occupies majority of the dissertation. This chapter explores the approaches that we have taken to create a structured craquelure model in which we call a *crack network*. A method based on chain-code crack following technique is explained in detail. Features are organized in a three layered hierarchical structure to allow efficient manipulation. A slightly more detail section later touches on the high-level features we have used in our experiments. Results are presented with graph plots and tables to evaluate the performance of the features.

Chapter 5 looks into a slightly more challenging issue of classification. Critical issues are outlined and possible solutions are listed. Results of chapter 4 are used to demonstrate the algorithms we have chosen, namely the k-means clustering and the fuzzy k-means clustering. Early results are presented and discussed.

Chapter 6 summarizes work that have been done and highlights the main weaknesses of the current approach. Future considerations are then listed in order to improve the performance.

Chapter 2

Content-based Analysis of Craquelure Patterns

2.1 Introduction

This chapter highlights the conceptual motivation of the project generally touching on issues from physical structure of a painting framework to the potential relationship between craquelure and support structures. The main issues behind the implementation of a CBR approach for the analysis of painting cracks are also discussed. Finally, probable application environments and scenarios are explained.

2.2 Physical Structure

In [17], Eastaugh mentioned two types of information that can be directly obtained from scientific study of paintings which are the localized analysis of composition that tells us about the materials that artist used and imaging of these materials across the painting, showing how they have been used. The author concentrated on the latter and further divides it into two more sub-sections, which are image interpretation and image formation. Image formation is how different types of radiation (i.e infra-red, X-ray, ultra-violet etc) interacts with the picture's structure. This, according to Eastaugh will enable us to determine what the picture is made of and how those materials are assembled. Basically, in most cases a picture may have 5 elements arranged in layers:
1) A support (such as canvas or wood panel).
2) A preparation layer (often referred to as ground).
3) A preparatory design, such as drawing.

4) The paint layer(s).
5) Transparent surface coating (varnish).

2.3 Crack Formation

As mentioned by Müller in [18], the paint layer is protected by the varnish and problems occur when the originally transparent varnish becomes yellowish or greenish or simply looses its transparency during ageing. As the paint layer ages the solvent is no longer capable of keeping the paint layer intact thus cracks begin to form. This is the most common reason behind the forming of cracks. Other known reasons include physical tensions within the structure of the painting and external impacts such as human faults during handling of paintings.

According to Bucklow in [19], the pattern of cracks is a visible record of the physical tensions within the structure of the painting. The ways in which tensions are generated and dissipated are dependent upon the choice of materials and methods of construction employed by the artist. He also stressed that auxiliary supports[1] contribute to patterns in the craquelure. Bucklow also mentioned in [16] that on canvas paintings, we are able to perceive the interaction between the canvas and the stretcher which creates "keying out", "tacking edge" and "stretcher bar" cracks. Interactions between the canvas and the environment are also evident in "circular" and "herring-bone" cracks, the results of mechanical impacts and scrapes respectively; it is asserted by some that circular cracks can also be caused by points of tension, such as knots in the canvas.

2.4 The Link Between Craquelure and Conservation

Ideally, cracks can serve a great deal of assistants for painting conservators in terms of providing them clues to degradation on the paint surface. Based on the proves given in the previous sub-chapter, it is clear that cracks do in some cases originate from physical impact to the structural part of a painting. As far as conservation is concern, these clues can be utilized for two main purposes; continuous paint layer degradation monitoring and a more in-depth study of how physical and environmental factors contribute to the existence of cracks so that steps can be taken to reduce them.

Due to huge amount of digitized painting related images, it is far from efficient for conservators to monitor cracks by manually searching for them. Most digitized images in

[1]The stretcher upon which the canvas is held.

well-known collections as in the National Gallery of London, The Louvre Museum in Paris and Uffizi Museum in Florence are of very high resolution. It is just impractical for a conservator to do a manual search on such large collections. Even browsing through a single painting for a particular feature can be very exhausting. As tools for artwork restoration, image processing techniques serve two purposes. A considerably huge amount of research has been done on the field of image processing related to fine arts incorporating state of the art methods.

Most of the previous researches on fine art restoration have been concentrating on the virtual restoration of digitised paintings. Giakoumis and Pitas [20] developed a method for virtual restoration of cracks on painting using morphological top-hat operator to detect crack patterns followed by an implementation of the MRBF neural network to separate brush strokes from cracks. Later they propose two crack-filling methods based on order statistics and anistropic diffusion. Barni et al. developed a virtual restoration system to remove cracks on digital images of paintings [5]. Their method is based on a semi-automatic crack detection procedure where users need to specify a point (pixel) believed to belong to a crack network. The algorithm will then track other suspected crack points based on two main features, absolute gray-level and crack uniformity. Once the algorithm has completely detected cracks, the user can decide to erase the cracks by interpolation.

2.5 CBR Implementation Issues

A design of a content-based retrieval system depends on requirements desired by users. However, in some cases, what machines are capable of doing do not fully conform to these needs. In this section, we briefly explain the main issues in designing a simple system capable of performing content-based analysis of craquelure patterns. We divide our scope of interest into 2; the first being the obstacles related to computer vision and the second as the requirements of the potential users.

From computer vision point of view, the process of identifying crack patterns is believed to be a very critical step in the whole system. High accuracy must be achieved in the detection stage to ensure that the following stages are served with valid input data. The relative difficulty in detecting cracks depends on whether their shape and typical orientation is known a priori, whether they start from the edge of the object, and on whether the texture is periodic or random. A key problem is the typically very small transverse dimensions and poor contrast of cracks. The human visual system may easily detect them, but they may actually consist of “chains” of non-adjacent single pixels in the image. In some of the worst cases, the surface is randomly textured and this will certainly pose a problem for the detection stage.

Feature selection is very important since good pattern discrimination can only be achieved if highly distinguishable features are used. Selecting relevant features is not an easy task, firstly since there is no clear grammar or language that can explain precisely how a particular crack pattern differs from another. The scenario certainly does not mimic that of the *optical character recognition* (OCR) [21, 22, 23, 24] problem where each alphabetical/numerical character has unique structural features allowing relatively easy classification assuming ideal block-based characters. This is not the case for cracks since the perception towards a particular pattern varies with respect to the observer and the scope of view (shape and scale). Content-based processing does not assume everything within an image to belong to the same "object of interest". An image contains objects with different shapes, sizes and appearances. Subjective perceptions regarding crack patterns change with varying view point, i.e the scope of view. This is seen as a very big challenge in our work. As shown in

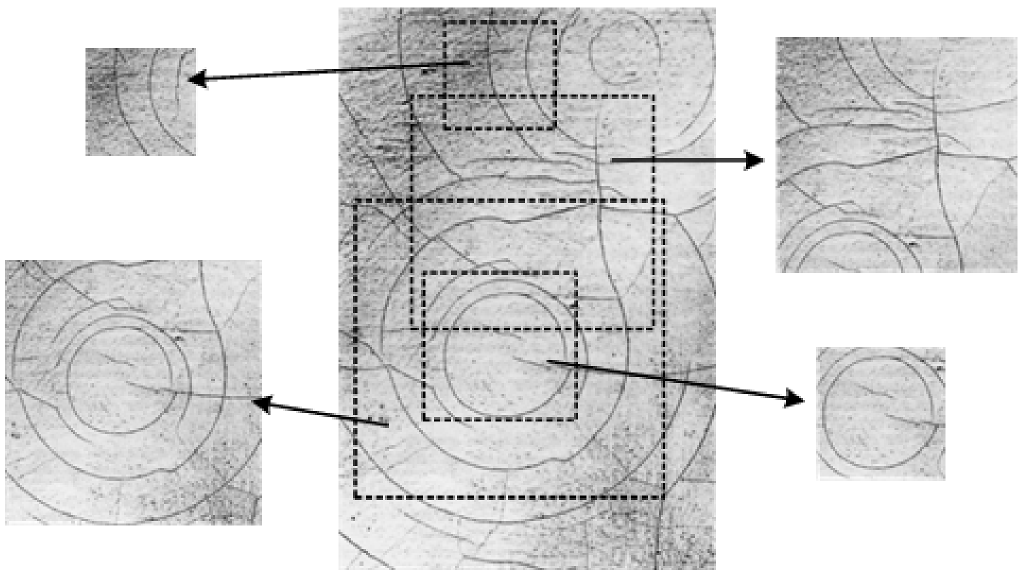


Figure 2.1: Subjective perception regarding crack patterns change with different scopes of view.

figure 2.1, the view in which crack patterns are observed affects the perceptive notion as to how they are classified. Furthermore, comparing the problem once again with the OCR problem, crack patterns consist of multiple line segments either joint or separated whereas alphabets/numbers are clearly combinations of at most seven line segments if we consider the number "8".

Another computer vision related issue concerns with the classification of crack patterns.

As mentioned earlier, there is yet to exist a standard grammar or language that describes crack patterns. It is not quite true to say that a certain crack pattern exclusively and only belong to a particular class. Unless a crack pattern totally agrees to a particular class descriptor while contradicts to descriptors of the other classes, we can assume that each crack network is a member of every crack class but with varying confidence value. We conduct experiments in the later stages of the dissertation to show the significance of the claim.

To be able to serve potential users with retrieval functionalities, the system should be able to process and access data in a considerably short time. Hence speed is a highly desirable element in the design of such system which can be a direct result from efficient handling of data query and quick database look-up. The system's approach towards the implementation of content-based functionalities also poses some questions. One of these being the type of query which depends on whether or not it is useful for the end-users bearing in mind that such application might only be useful for fine artwork conservators. Relevant query types in this context are query by example and query by type (text-based query). Examples of these queries are further elaborated in section 2.6.

2.6 Application Scenario

Utilising the capabilities of image analysis and processing in aiding conservators can prove to be very useful if the process can provide users with the functionalities that manual processes fail to serve them. Among the common problems of manual defect screening are time consumption and the risk of further damage. Required information should be retrieved correctly (i.e as close as possible to the users' individual perception) and consumes considerably less time and effort. The following sections briefly outline potential application scenarios from a user point of view.

2.6.1 Scenario 1 : Query for Similar Pattern

The main requirement to a content-based analysis of cracks will be to query for a particular region or image for similar type of crack patterns. This scenario can attract special attention from users who are interested in investigating the relation between cracks of near-similar patterns. As mentioned in the earlier chapters, crack patterns can be judged in two ways: authenticity and damage assessment. This type of scenario would definitely suit both of them given that good type descriptors are used to efficiently describe crack patterns. Scenario 1 is as follows:

A user issues a query in the possible forms/actions listed below.

(a) Uploading an example of a crack image from an external or an internal source.

(b) Selecting a particular region from a large image.

(c) Specifying a particular type/class of crack pattern from a list.

Results of the query can be of the following forms.

(a) Highlighted regions of the best matches and some associated data (i.e location of bounding box, dimension, statistical values).

(b) List of images in database containing the specified crack patterns.

2.6.2 Scenario 2 : Regional Information Retrieval

The second scenario needs less processing time compared to the first since it does not perform pattern matching. It focuses on providing users with the ability to interactively obtain information about a particular region of an image. Conservators in particular are interested in statistical values of crack patterns in a particular area including the type/class in which they fall into. This functionality can be thought as being useful for pattern learning or even for random inspection of painting surfaces. Scenario 2 is as follows:

A user selects a region from a crack image and request statistical information regarding that particular region. Results of the user request can be of the following forms.

(a) List of information (i.e most probable class membership, total crack length, crack density, average crack width, etc.).

(b) Highlighted area or feature of interest (i.e detected cracks, boundary box).

The realization of the application scenarios is not perceived as an easily achievable task as far as this M.Phil/Ph.D project is concern. Varley in his Ph.D dissertation [25], explains his approach towards classifying line structures using discriminant analysis. His thesis was totally dedicated into modelling line structures and classifying them into classes based on authenticity. He assumed cracks can be easily detected using straight-forward thresholding technique which reasonably acceptable for his work since segmented crack images are readily available. On the contrary, we will be forced to assume otherwise due to the automatic nature of a content-based retrieval application. Algorithm speed is not a major concern in his work whereas speed in our work is one of the biggest concerns.

2.7 System Overview

For the purpose of integrating CBR functionalities in the system we design an outline of the system architecture which consists of two main modules, namely the *application module* (AM) and the *processing module* (PM). AM deals with information retrieval issues such as querying, feature storage and management, pattern matching and sub-image search while PM looks into the problem through a more technical point of view by providing the core computer vision processes. Nevertheless, the prime tasks of the whole system, like all other CBR systems are to accept queries, process data, retrieve information and present them to users. Figures 2.2 and 2.3 illustrate the proposed architectures of the system corresponding to Scenario 1 and Scenario 2 respectively.

2.7.1 Application Module

The *application module* makes sure queries are processed correctly. Queries as mentioned in section 2.6 can be made in two ways; either by example or by textual description. One of the intended task of AM is to invoke the correct processes based on the type of query. This task is quite straightforward and is dealt with by the *query/result processor.* The *query/result processor* acts as a mediator between the *query/result interface*[2] and the databases. It controls the flow of information based on queries and results. For instance, a query based on scenario 1 will cause the *query/result processor* to invoke the databases while a query described by scenario 2 will involve getting information directly from PM. The three main databases are the *image database*, *features database* and *global process parameters database*[3].

The *data organizer* organizes the structured data, features and classification results before they are stored in the *feature database.* The *feature matcher* compares features extracted from queried images with those in the database and retrieves the relevant ones for display or presentation to the user.

2.7.2 Processing Module

The *processing module* executes image processing and pattern recognition algorithms when invoked by the *processing module.* Technically, this module is more difficult to design and implement compared to the *processing module.* Theoretically, the role of this module is to process inputs either off-line or on-line from sources which can be from user queries or

[2]A command line or graphical user interface to interact with users.

[3]The parameters are the updated values of certain processes such as the cluster centres of the classification sub-module. They are updated as processes commence.

images in a database. It consists of low-level and high-level computer vision algorithms which act as sub-modules. The main sub-modules are crack detection, statistical crack pattern structuring, high-level feature extraction and data classification. Vast portion of this dissertation is devoted to this module.

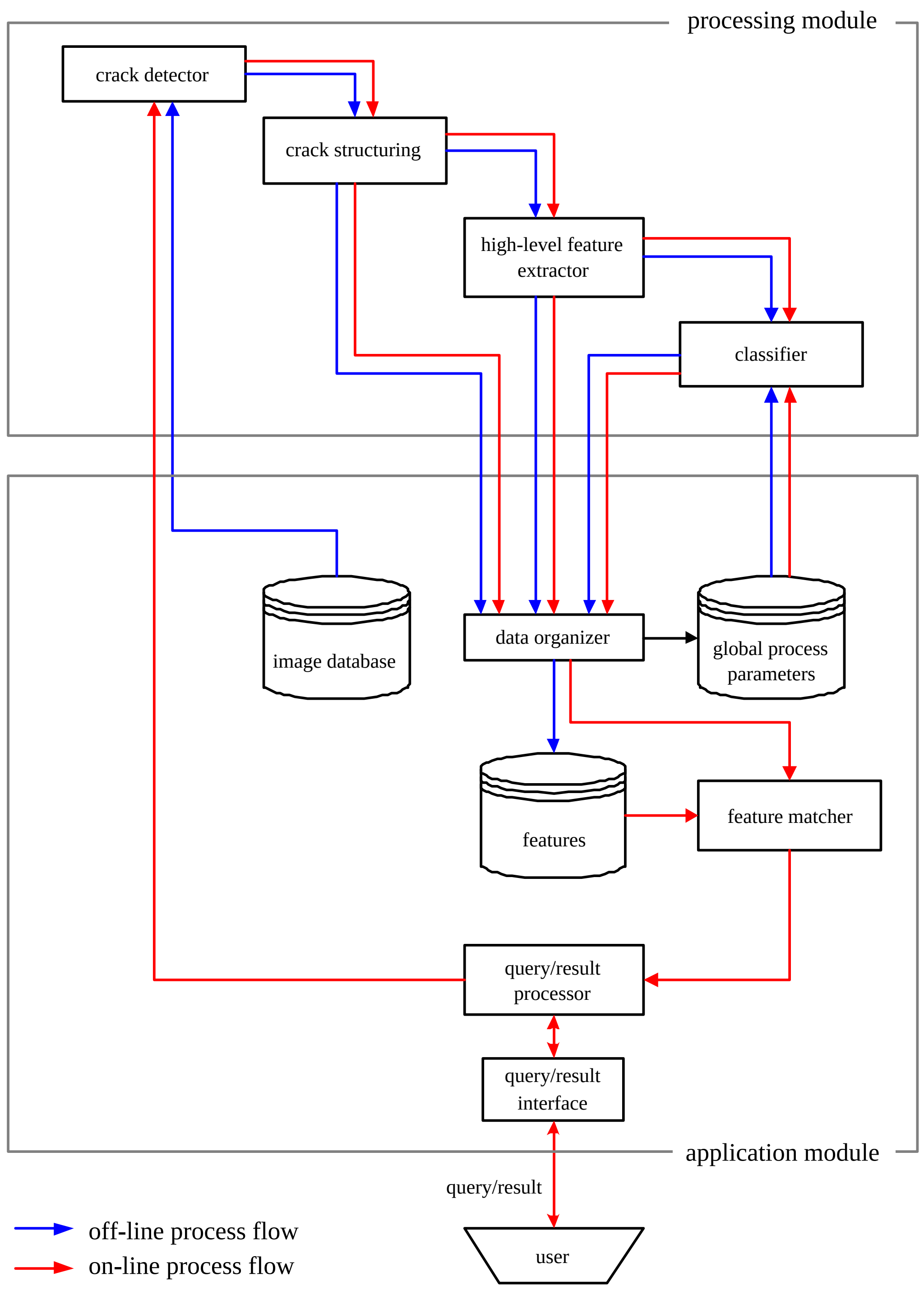


Figure 2.2: A general architecture of the system corresponding to Scenario 1.

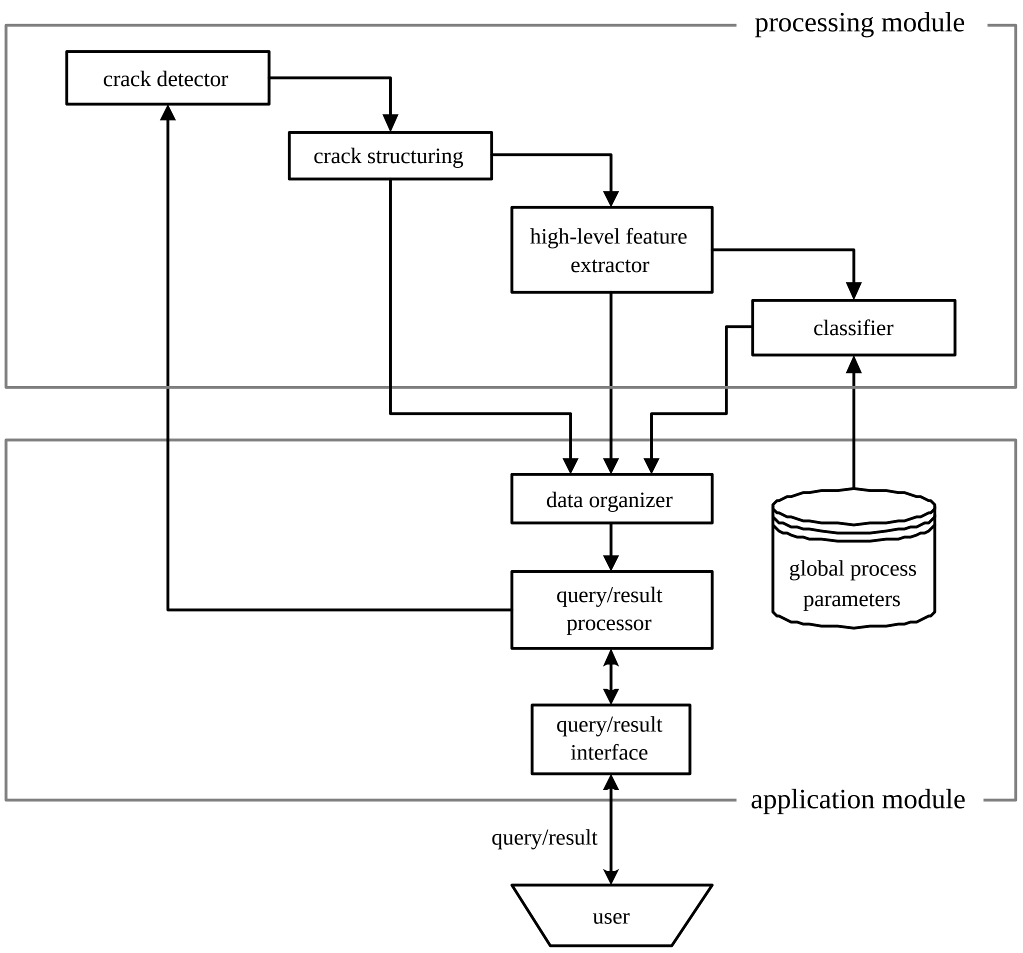


Figure 2.3: A general architecture of the system corresponding to Scenario 2.

Chapter 3

Crack Detection

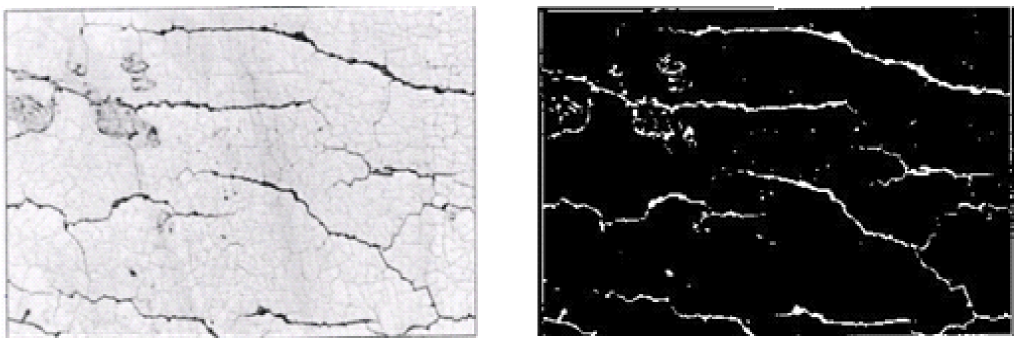
3.1 Introduction

Crack-like pattern detection or in some literature, more known as ridge-valley structure extraction [26, 27, 28, 29] have been a matter of high concern among researchers mostly for its potentially useful contribution to variety of applications. The results presented here have a much wider set of application than just the analysis of paintings, however: many images contain similar patterns - biological images of veins and tissues [30, 31, 32] , images of fingerprints [33] and multi-spectral satellite photos of rivers or roads [34, 35]. All these examples are current areas of research in modelling and classification field for which the results in this dissertation can find an application.

The aim of this stage is to extract or in other words, segment suspected cracks patterns from the background. By noting that cracks are usually considerably darker than the background and that they are characterized by a uniform gray-level which have an elongated structure, detection can be accomplished on the basis of two main features: absolute gray-level and crack uniformity [5, 20].

In [25] Varley assumes that crack patterns in paintings can be segmented by just thresholding the image by a certain manually-selected threshold level. Although the assumption is true, unfortunately the outcome is not quite encouraging if such method is applied. As shown in figure 3.1, thresholding does not always work. Factors such as inconsistent illumination, noisy presence and low contrast adds to the difficulty of obtaining a fairly accurate detection.

This chapter is dedicated to the approaches that we have taken to deal with this important early step of detecting the desired “objects of interest”, i.e the crack patterns.


(a) Sample image 1 and the result of manual thresholding

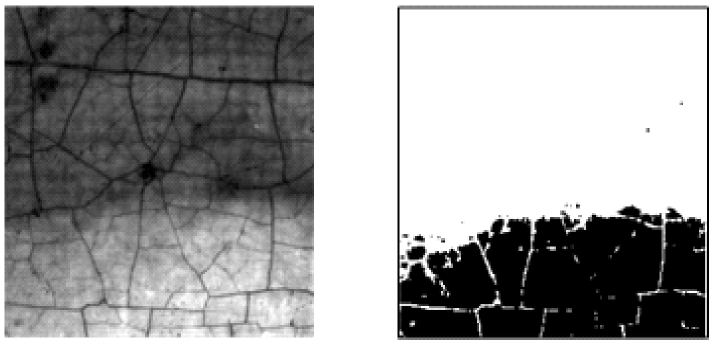

(b) Sample image 2 and the result of manual thresholding

Figure 3.1: Threshold applied on 2 sample crack images.

3.2 Common Line Detectors

A line segment on an image can be characterized as an elongated rectangular region having gray-level bounded on both its longer sides by homogeneous regions of a different gray-level. For a typical painting crack, gray-levels of the side level have higher values than the centre elongated region containing the dark line. The width along the same line segment may also vary. A general line detector should be able to detect lines in a range of widths.

Among early work on line detection is done by Vanderbrug who suggested a semi-linear line detector created by a step edge on either side of the line [36]. He also developed an algorithm which is originally designed for road detection in satellite images [37]. The method is based on the response of the image to different masks, allowing to estimate the local variations of the gray-levels. This algorithm which is also employed by Müller in [18] uses 14 masks. Good results are reported if the image is not too affected by noise. However, in the presence of noise, these techniques do not perform well. These techniques are also seen to be very poor in terms flexibility in order to cope with the unpredictable nature of crack patterns.

3.3 Mathematical Morphology

Mathematical morphology is a part of digital image processing that concerns with image filtering and geometric analysis by structuring elements. Originally, the theory and application of mathematical morphology was developed for binary images. Its main protagonists were Matheron [38] and Serra [39]. Afterwards, the theory was extended to gray-scale images by Sternberg [40] and later by Haralick, Stenberg and Zhuang [41, 42].

Since those early days, morphological operations and techniques have been applied from low-level, to intermediate, to high-level vision problems. These operations are mainly used for noise reduction and feature detection, with the objective that noise be reduced as much as possible without eliminating essential features.

3.3.1 Basic Morphology Operators

Dilation and erosion are the two fundamental operations that define the algebra of mathematical morphology. These two operations can be implemented in different combinations in order to obtain more sophisticated operations.

Binary dilation is the morphological transformation that combines two sets of pixels by using vector addition of set elements. Binary dilation was first used by Minkowski, and in the mathematics literature it is called *Minkowski addition*. If A and B are sets in N-space(E^N) with elements a and b respectively, a=$\{a_1, ..., a_N\}$ and b=$\{b_1, ..., b_N\}$ being N-tuples of element coordinates, then the dilation of A by B is the set of all possible vector sums of pairs of elements, one coming from A and one coming from B. More formally, the dilation of A by B is denoted by $A \oplus B$ and is defined by

$$A \oplus B = \{c \in E^N | c = a + b \;\; for \;\; some \;\; a \in A \;\; and \;\; b \in B\} \tag{3.1}$$

The dilation operation can also be represented as a union of translates of the structuring element:

$$A \oplus B = \bigcup_{b \in B} A_b \tag{3.2}$$

Erosion is the morphological dual of dilation. The morphological transformation combines two sets by using containment as its basis set. If A and B are sets in Euclidean N-space, then the erosion of A and B is the set of all elements x for which $x + b \in A$ for every $b \in B$. The erosion of A by B is defined by

$$A \ominus B = x \in E^N | x + b \in A \ \ for \ \ every \ \ b \in B \tag{3.3}$$

The erosion operation can also be represented as an intersection of the negative translates:

$$A \oplus B = \bigcap_{b \in B} A_{-b} \tag{3.4}$$

Dilations and erosions are usually employed in pairs; a dilation of an image is usually followed by erosion of the dilated result or vice versa. In either case, the result of successively applied dilations and erosions results in the elimination of specific image detail smaller than the structuring element without the global geometric distortion or unsuppressed features.

The *opening* of an image is obtained by first eroding the image with a structuring element and then dilating the result using the same structuring element. The *closing* of an image is obtained by first dilating the image with a structuring element and then eroding the result using the same structuring element.

The opening of A by B is denoted by $A \circ B$ and is defined as:

$$A \circ B = (A \ominus B) \oplus B \tag{3.5}$$

while the closing of A by B is denoted by $A \bullet B$ and is defined as:

$$A \bullet B = (A \oplus B) \ominus B \tag{3.6}$$

For in-depth mathematical analysis of binary morphological operators, refer [39], [43] and [44].

3.3.2 Gray-scale Morphology Operators

Although binary morphological operations serve useful analytical tool for image analysis and classification, they play only a limited role in the processing and analysis of gray-level images. To overcome this limitation, Sternberg and Serra extended binary morphology in the early 1980s to gray-scale images via the notion of an umbra [40, 39]. Similar to the binary case, dilations and erosions are the basic operations that define the algebra of gray-scale morphology. They are combined to produce the gray-scale opening and closing operations which are very useful and effective set of operations for various computer vision application. The following paragraphs brief some of the basic definitions of gray-scale morphological operators that can be useful for this research.

A discrete gray-level image, A is defined as a finite subset of Euclidean 2-dimensional (2-D) space, $\mathbb{R}^2$ whose range is $[N_{min},N_{max}]$, $A : \mathbb{R}^2 \rightarrow [N_{min},N_{max}]$ while a 2-D structuring element is defined as a function $S : \mathbb{R}^2 \rightarrow \boldsymbol{S}$ where $\boldsymbol{S}$ is the set of neighbourhoods of the origin. Gray-scale morphological dilation and erosion can be visualised as working with two images namely the image being processed I and the structuring element S. Each structuring element has a specific shape that act as a probe. The four basic gray-scale morphological operators are defined with respect to the structuring element S, scaling factor e, image A and point $M_o \in \mathbb{R}^2$. Gray-scale erosion and dilation are defined as:

$$erosion \quad : \quad \epsilon_S^e(A)(M_o) = MIN_{M \in M_o + e \cdot S(M_o)}(A(M)) \tag{3.7}$$

$$dilation \quad : \quad \delta_S^e(A)(M_o) = MAX_{M \in M_o + e \cdot S(M_o)}(A(M)) \tag{3.8}$$

Conceptually similar to the binary case, dilation followed by erosion is closing transformation, while erosion followed by dilation is opening transformation and they are defined as:

$$opening \quad : \quad \gamma_S^e(A) = \delta_S^e(\epsilon_S^e(A)) \tag{3.9}$$

$$closing \quad : \quad \varphi_S^e(A) = \epsilon_S^e(\delta_S^e(A)) \tag{3.10}$$

3.3.3 The Top-hat Transformation

Cracks can be detected with the implementation of a very useful morphological filter, known as top-hat transformation developed by Meyer [45]. These details can be lines or areas with particular sizes. Top-hat operators can function as a closing or opening operator based on the features we wish to extract from an image. Opening top-hat operators (OTH) will detect bright details in an image while closing top-hat operators (CTH) are designed to detect dark details. Formulation for OTH and CTH are shown denoted by equations (3.11) and (3.12) respectively.

$$OTH_S^e = A - \gamma_S^e(A) \tag{3.11}$$

$$CTH_S^e = \varphi_S^e(A) - A \tag{3.12}$$

The top-hat operator can be tuned for detection of specific features by modifying two important parameters [20]:

- The shape and the size of the structuring element S. A square-shaped or disk-shaped structuring element may be used. The size must be chosen carefully based on the thickness of the crack to be detected.
- The number of times in which erosion or dilation are performed.

The transformation produces a gray-scale image with the desired features enhanced to a certain level. A thresholding operation is needed to separate the features from the background.

Figure 3.3 shows some results of crack enhancement using *close top-hat operator*. The images are enhanced using a *disk* structuring element of size 5x5 as shown in figure 3.2.

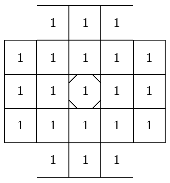


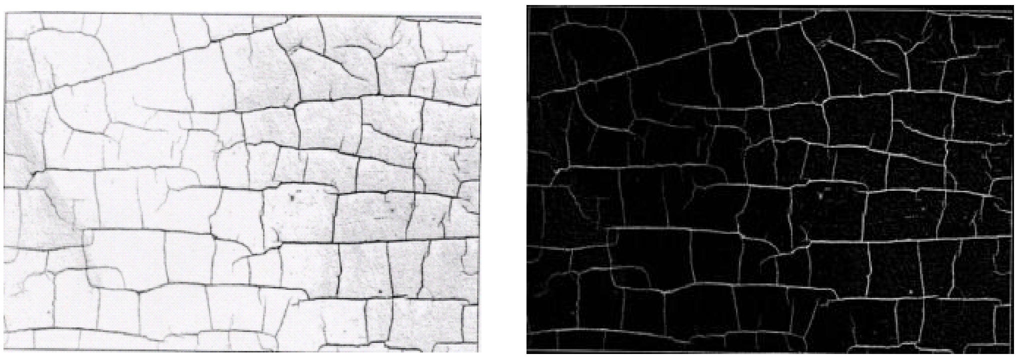
Figure 3.2: A 5x5 disk-shaped structuring element.

3.3.4 Multi-orientation Structuring Element

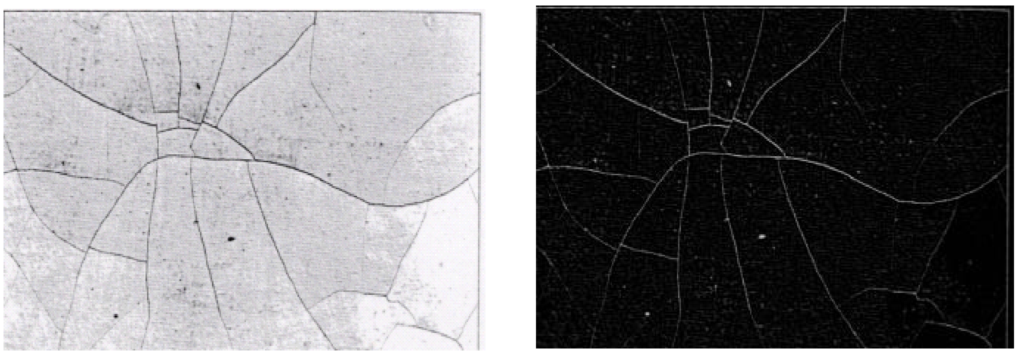
Multi-orientation filtering using directional structuring elements is another method often used [30]. Figure 3.4 shows an example of cracks enhanced using horizontal and vertical rectangular structuring element.

3.4 Automatic Thresholding

Thresholding is one of the simplest and most widely used image segmentation techniques. The goal of thresholding is to segment an image into regions of interest and to remove all other regions deemed inessential. The simplest thresholding methods use a single threshold in order to isolate objects of interest. In many cases however, especially in the case of crack images, no single threshold provides an excellent segmentation result over the entire image. In such cases, variable and multilevel threshold techniques based on various statistical measures are used. In our research we consider two most commonly used automatic threshold selection methods, namely the Otsu's technique and the *simple image statistic* technique



(a) Sample image A and its enhanced version



(b) Sample image B and its enhanced version

Figure 3.3: Cracks enhanced by a *close top-hat operator.* using a 5x5 *disk* structuring element

3.4.1 The Method of Otsu

The method of Otsu [46, 47] is based on discriminant analysis. The threshold operation is regarded as the partitioning of gray-level distribution of an image into two classes $C_0 = \{0, 1..., t\}$ and $C_1 = \{t+1, t+2..., l\}$ at gray-level $t \in G = \{0, 1, ..., l\}$. It uses the histogram information derived from the source image. The optimal threshold t can be determined by minimizing the criterion function defined on the two classes:

$$t = \min_{i \in G} \left(\frac{\sigma_b^2}{\sigma_o^2} \right) \tag{3.13}$$

where

$$\sigma_o^2 = \sum_{i=0}^{l} (i - u_o)^2 \, p_i, \tag{3.14}$$

$$u_o = \sum_{j=0}^{l} j p_j \tag{3.15}$$

and

$$t = \frac{n_i}{n}. \tag{3.16}$$

Meanwhile,

$$\sigma_b^2 = \sum_{i=0}^{t} p_i \left(1 - \sum_{i=0}^{t}\right) \left(\frac{\left(\sum_{i=0}^{l} ip_i - \sum_{i=0}^{t} ip_i\right)\left(\sum_{i=0}^{t} ip_i\right)}{\left(1 - \sum_{i=0}^{t} p_i\right)\left(\sum_{i=0}^{t} p_i\right)} \right)^2 \tag{3.17}$$

where n_i and n are the number of pixels with gray-level i and the total number of pixels in the image respectively. For more on the subject, refer [44].

3.4.2 The Simple Image Statistic Technique

The *simple image statistic* (SIS) algorithm [48] is an automatic bi-level threshold selection technique. Unlike Otsu's technique, SIS algorithm does not require computing a histogram for the image. Let $\mathbf{A} \in \mathbb{R}^{\mathbf{X}}$ be the source image over an m x n array $\mathbf{X}$. The SIS algorithm assumes that $\mathbf{A}$ is an imperfect representation of an object and its background. The ideal object is composed of pixels with gray-level a and the ideal background has gray-level b.

Let $\mathbf{e}(i, j)$ be the maximum on the absolute sense of the gradient masks $\mathbf{s}$ and $\mathbf{t}$ (see figure 3.5) applied to $\mathbf{A}$ and centred at $(i, j) \in \mathbf{X}$.

It is shown in [48] that

$$\frac{\displaystyle\sum_{i=1}^{n}\sum_{j=1}^{m} |\mathbf{a}(i,j) \cdot \mathbf{e}(i,j)|}{\displaystyle\sum_{i=1}^{n}\sum_{j=1}^{m} |\mathbf{e}(i,j)|} = \frac{a+b}{2}. \tag{3.18}$$

The fraction on the right-hand side of the equality is the midpoint between the gray-level of the object and background. Intuitively, the midpoint is an appealing level for thresholding. Thus, the threshold t set by the SIS algorithm is equal to the left-hand side of equation 3.18.

3.4.3 Performance Evaluation

We experimented with several images to demonstrate the effectiveness of these algorithms. Figure 3.10 shows some results obtained for threshold values generated by the Otsu's technique and the SIS algorithm. The enhanced image is thresholded using values from Otsu and SIS algorithm with values of 48 and 30 respectively.

Based on observations, Otsu technique is chosen due to the fact that its outcome is more consistent compared to the SIS technique. Otsu technique produces threshold values higher than the SIS method, thus there is less noise in the output image.

3.5 Variable Thresholding

Specifically, in the case of large crack images, a single threshold value over the whole image is not a good strategy to segment the cracks. It works when the background intensity level is not constant. As for the images we are working with, the main problem is due to uneven enhancement level of suspected cracks as a result of illumination inconsistency. One simple strategy to overcome this limitation is by performing a technique known as variable thresholding [49]. Variable thresholding allows different threshold levels to be applied to different regions of an image.

The image is first sub-divided into smaller pre-specified regions with similar sizes. In our current implementation, we divide the image into regions of even dimension as shown in figure 3.7, where G is the grid dimension. The image is first zero-padded if its dimension is not a multiple of the grid size. Threshold value is then established for each region separately using either the *Otsu* algorithm or the SIS algorithm and thresholding is performed locally on every sub-image. Figure 3.8 shows threshold values generated using Otsu and SIS over an image.

The exact methodology is as follows. Let $\mathbf{a} \in \mathbb{R}^{\mathbf{X}}$ be the source image, and let image $\mathbf{d} \in \mathbb{R}^{\mathbf{X}}$ denote the region threshold value associated with each point in $\mathbf{X}$, that is, $\mathbf{d}(\mathbf{x})$ is the threshold value associated with the region in which point $\mathbf{x}$ lies. The thresholded image $\mathbf{b} \in \{0, 1\}^{X}$ is defined by

$$\mathbf{b}(\mathbf{x}) = \begin{cases} 1 & \text{if} \quad \mathbf{a}(\mathbf{x}) \geq \mathbf{d}(\mathbf{x}) \\ 0 & \text{if} \quad \mathbf{a}(\mathbf{x}) < \mathbf{d}(\mathbf{x}) \end{cases} \tag{3.19}$$

Instead of globally thresholded, the enhanced crack image is locally processed, thus weak cracks are better detected. However, we have to be extra careful in choosing the dimension of the grids. Very small grids will result in the emergence of unwanted noise in the thresholded image since the assumption made prior to the process is every grid should contain cracks (the object and the background). Even if a region in reality does not contain cracks, the algorithm will “force” cracks to appear since the threshold will emerge to be extremely low in these regions.

3.6 Crack Thinning

For the later stages, it is more convenient and less time consuming to work with one-pixel wide cracks rather than those of variable width. Although width is seen as an important element to characterize a crack pattern [16], we ignore it for the time being to concentrate on other characteristics.

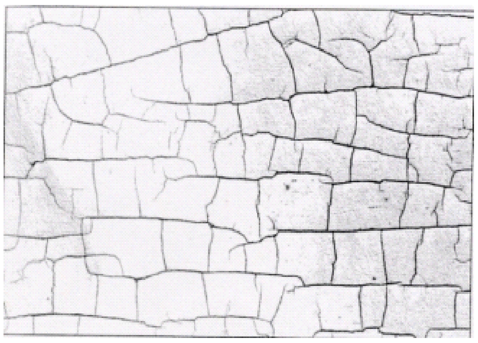
We thin the crack patterns using the *hit-and-miss* algorithm [39]. To make sure the cracks have been thinned to one-pixel wide, we perform 10 iterations. A *hit-and-miss* cleaning algorithm is then applied to remove isolated pixels. A “thinned” and “cleaned” image (see figure 3.9) is the final outcome of the *crack detection* process, which will be the input to the next stage.

3.7 Results and Discussion

The close top-hat operator is a good tool to enhance cracks. However in reality, cracks in paintings exist in multiple widths and the use of a single structuring element could not deal with this condition. A set of multi-size structuring element might be able to deal with this problem. Using multi-orientation structuring elements is also a potentially useful step to concentrate on the elongated structures alone since fully symmetrical structuring elements in most cases resemble noise and in this situation noise is not suppressed.

The ability of the algorithm to detect cracks accurately is highly desired. The grid-based variable thresholding is seen as a good early step to deal with illumination inconsistency. We demonstrate in figure 3.10 the effectiveness of the technique using Otsu technique with grid size G of 64.

It is extremely difficult to achieve high accuracy in this detection stage. However, due to the large portion of cracks detected over the portion of noise, we are quite optimistic that the outcome of the algorithm is acceptable and can be used for the later stages. Nevertheless, improving the current detection algorithm is still a part of the main priority of the research.

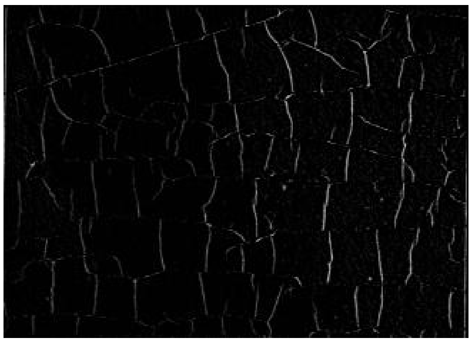

(a) original image

$$\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & \boxed{1} & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

(b) diagonally oriented structuring element

$$\begin{array}{|c|c|c|c|c|c|c|}\hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & \boxed{1} & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

(c) horizontally oriented structuring element


(d) diagonally enhanced cracks

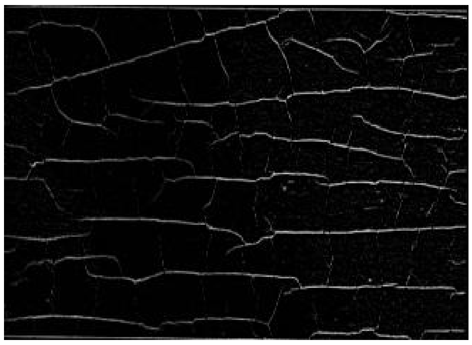

(e) horizontally enhanced cracks

Figure 3.4: Results of using multi-orientation structuring elements.

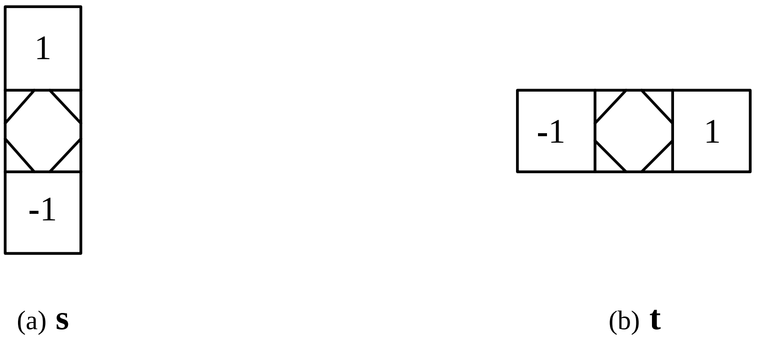


Figure 3.5: Gradient masks **s** and **t** used in SIS technique.

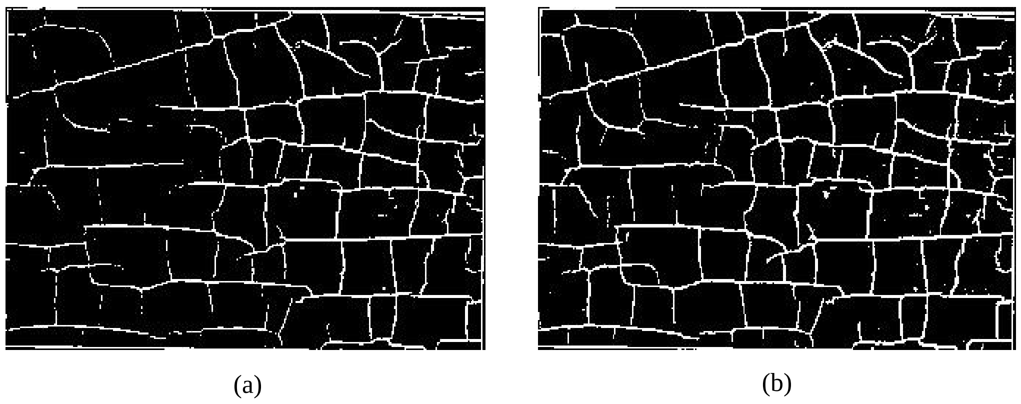


Figure 3.6: Enhanced image segmented using automatically determined threshold values.

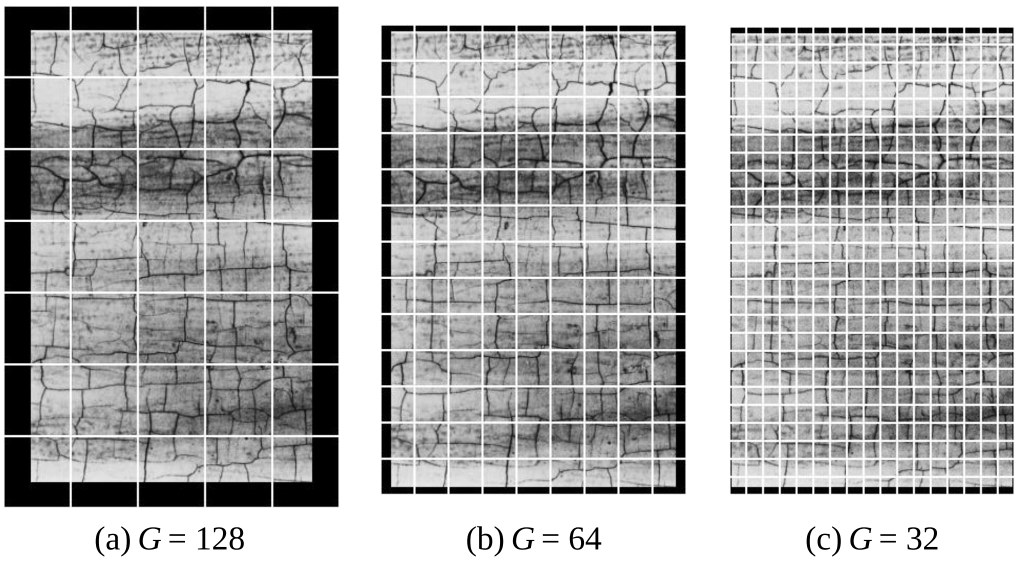


Figure 3.7: Grids overlaid on image.

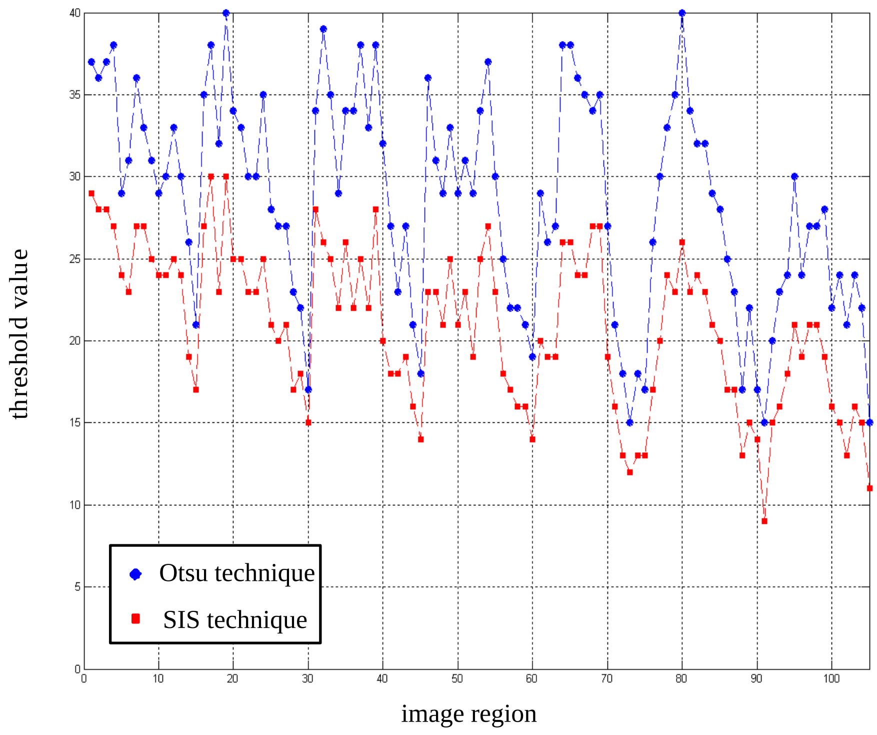


Figure 3.8: Variable thresholding: comparison between values generated by Otsu and SIS techniques.

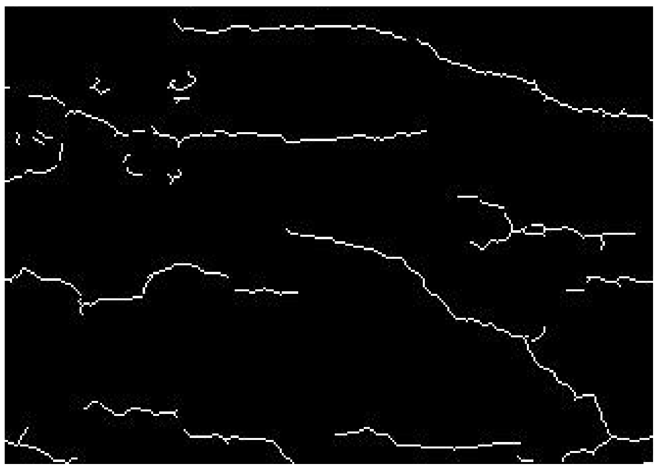
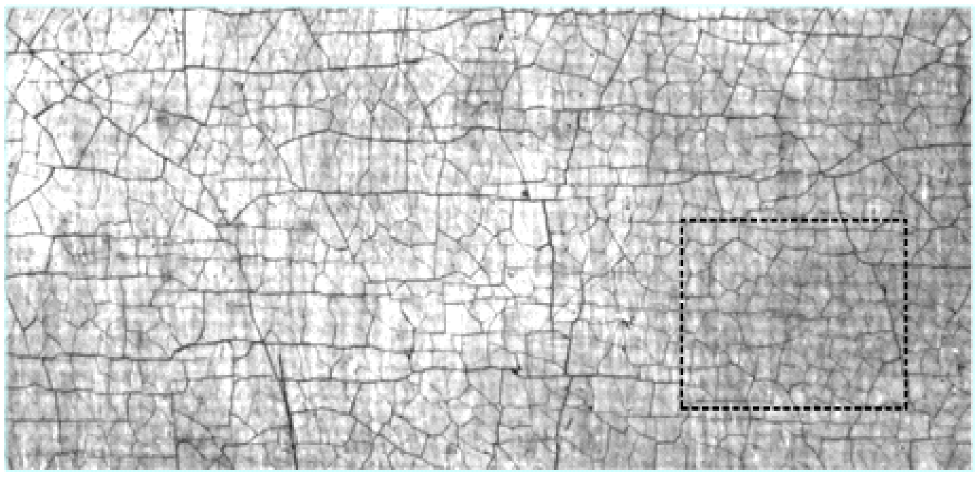
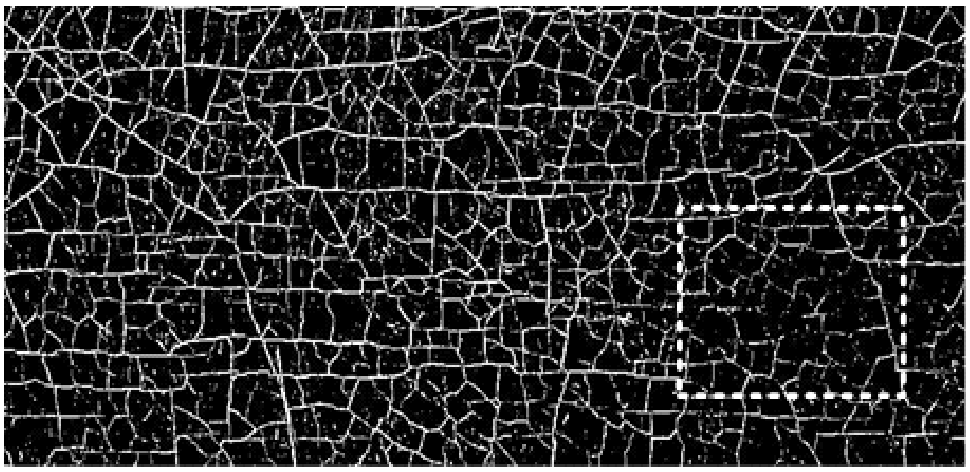


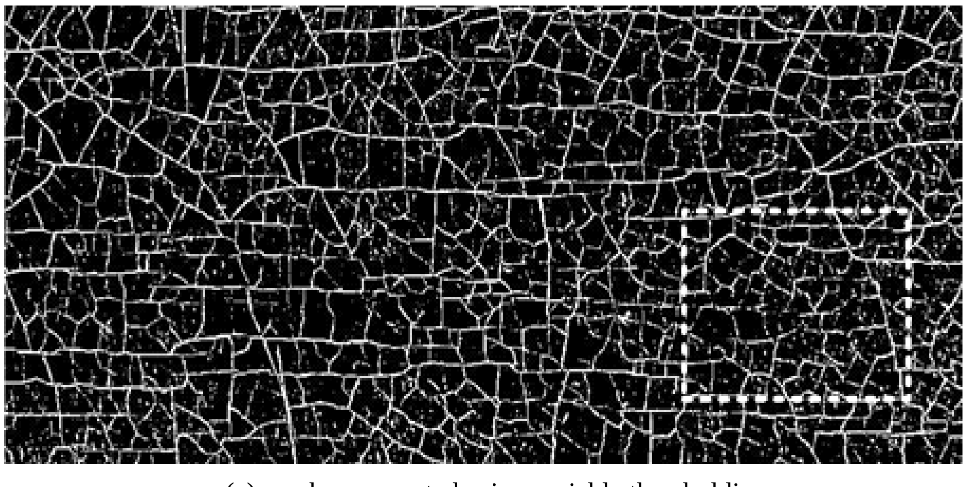
Figure 3.9: Thinned and cleaned cracks.



(a) original image



(b) cracks segmented using normal thresholding



(a) cracks segmented using variable thresholding

Figure 3.10: Dealing with illumination inconsistency using variable thresholding.

Chapter 4

High-level Feature Extraction

4.1 Introduction

Descriptions and features cannot be considered pure knowledge representations. Nevertheless, they can be used for representing knowledge as a part of a more complex representation structure. Descriptions usually represent some scalar properties of objects, and are called features [50]. Typically, a single description is insufficient for object representation; therefore, multiple descriptions are combined into what is called a *feature vector*.

Literally, the patterns of crack have been described using structural descriptions such as in [16], [19] and [51] where Bucklow defined a descriptive framework of cracks based upon the following features:

1. Predominant direction and orientation of cracks.
    - NO DIRECTION or DIRECTION; isotropy or anisotropy?
    - if anisotropic, then PARALLEL or PERPENDICULAR to grain?
2. Changes in direction of cracks.
    - locally - SMOOTH or JAGGED
    - globally - STRAIGHT or CURVED
3. Relationship between crack directions.
    - paint islands - SQUARE or NOT SQUARE; is there an orthogonal relationship?
4. Distance between cracks.
    - spatial frequency - are the paint islands SMALL or LARGE?

5. Thickness of cracks.
    - are all cracks of UNIFORM thickness or are SECONDARY cracks present?
6. Junctions or termination of crack.
    - is crack CONNECTED or BROKEN?
7. Organization of cracks.
    - is crack network ORDERED or RANDOM?

The elaboration of the descriptions can be found in [16] where Bucklow also stressed that fewer than half of them proved to be really necessary for a very high level of discrimination between categories used in his demonstrations. The other remaining terms however enable finer discrimination within the categories. The discriminatory power of features varies and is dependent upon the characteristics of classes sought. The characteristics of a pattern are determined not so much by individual features but by their relationships. The best combination of features that best describes a crack pattern is a task that is difficult to accomplish.

This chapter explains issues related to the classes of crack pattern. Next it touches on the current methodology we are employing and finally experiments conducted with their respective results.

4.2 Description of Crack Patterns

The main aim of this stage of the research is to find a way of describing crack patterns numerically. Prior to that, clear distinctions must be outlined among classes mentioned in section 1.1 generally stating the criteria and characteristics of each one. Figure 4.1 shows some typical examples of these 5 crack classes.

Throughout most of the remaining chapters we will use the images in figure 4.1 as sample images to represent the 5 crack classes. Through observations based on the structural appearances of these images, it is clear that line direction is one of the most discriminating features. Crack patterns in figures 4.1(a) and 4.1(c) can easily be distinguished from those in figures 4.1(b), 4.1(d) and 4.1(e) when the shape of the line segments are used as a feature. Line segments in figures 4.1(a) and 4.1(c) are collectively more curvy than those in the other images. However, globally there is no clear distinction between figures 4.1(b), 4.1(d) and 4.1(e) in terms of the structural shape of each of the line segment. Other features should be used to distinguish them.

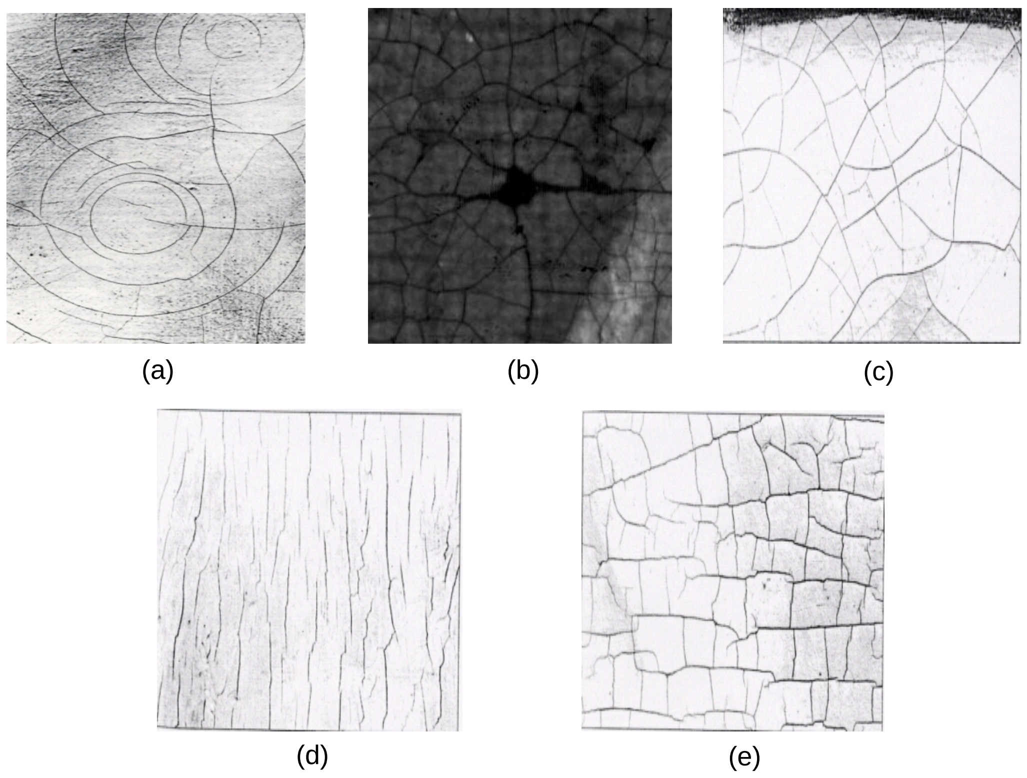


Figure 4.1: Typical classes of cracks related to support structures and physical impact.

Another feature that seems to be quite useful is the length of the line segments. The lengths of line segments in figures 4.1(b), 4.1(d) and 4.1(e) are more likely to vary slightly if compared to those of figures 4.1(a) and 4.1(c). From a statistical view point, the standard deviation of the line segment lengths might be a potentially beneficial measure to classify crack patterns.

Given the general observations of the differences, how do we describe the crack numerically?

Varley, in his work [25] modelled cracks using unknown number of Bézier curves [52]. He described how the parameters of the Bézier curve model are sampled using the *Reversible Markov Chain Monte Carlo* (MCMC) technique [53, 54]. He also explained in brief the methodology in which the parameters of the Bézier curve are used as features for classification. In total, 28 features are extracted from the Bézier curve.

The fact that the MCMC technique is a global process makes it very inefficient in terms of its computational cost. Varley reported in [54], a total iteration of 10^7 to model a simple crack image which is extremely time consuming. Being a global process makes it less suitable for a content-based analysis since processes are made on an image basis instead of a regional or pixel basis.

We employed a simple technique in which we base our analysis on; a chain-code based crack network structuring approach. The main reason the technique is used is based on the fact that it is a pixel-based process which allows analysis down to the pixel level. It also allows analysis at multiple structural levels of a crack pattern. The next section explains the approaches that we have taken.

4.3 Structural Representation of Crack Patterns

A *crack following* algorithm is applied on a crack detected image such as the one shown in figure 3.9. Statistical data are collected as it “runs” along the lines. This feature extraction approach collates statistical information while marking important points such as junctions and end points.

The *Freeman chain-code* [55, 56] has been used for various image processing including for finding features of curves/lines [57, 58]. We employed a similar scheme to record the direction of the crack pixels. The 8-connected Freeman chain coding uses a 3-bit code $0 \leq c \leq 7$ for each boundary point. The integer c indicates the direction in which the next crack pixel is located. The 8-connected scheme is as shown in figure 4.2. Boundary chain-codes can be determined using contour following [59], which is a traversing process to identify the boundary of a binary object. The algorithm requires operations of $O(N)$. We employ a conceptually similar method to that of [59] except that we implement it on open boundary line segments instead of close boundary objects.

Its implementation in our approach serves two main purposes, namely post-detection filtering/pruning and high-level feature extraction. However, unlike most of the applications which implement chain-codes, the approach we take does not use the 8-connected direction as means of storing and reconstructing a crack structure. It is utilized just for the sole purpose of building a structured representation of statistical data to allow efficient data access and manipulation. Having that, high-level feature extraction and crack pruning can be done easily. The extracted high-level features are the actual information stored as metadata describing the nature of an analyzed crack pattern.

4.3.1 Hierarchical Structuring of a Crack Network

The *crack following* routine collates information on the basis that pixels are 8-connected. In other words, the process starts from one point and “follows” a connected route until it

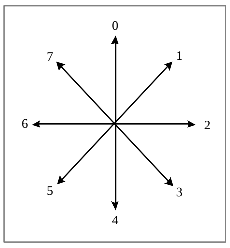


Figure 4.2: The 8-connected Freeman chain-code.

detects a discontinuity and along the "way" it marks all important points. Figure 4.3 shows 1000 chain-codes for the first 1001 8-connected pixels. Statistical information are stored temporarily in a structured manner where all 8-connected pixels are regarded as a single entity which we named a *crack network*. The structure of a *crack network* is as shown in figure 4.4 where N, M and P represent number of crack network, number of joints and number of lines respectively .

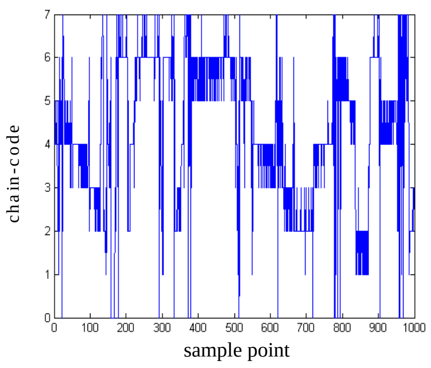
Each *crack network* has its own sub-structures that hold information related to junctions and line segments. The number of sub-structures depends on how many of these detected. The location of junctions is seen as a potentially good feature although the best way of manipulating it has not yet been looked into. The line segments can also act as locally useful descriptors on the basis that a single crack network consists of multiple line segments connected in a random or regular pattern.

With reference to figure 4.4, the structured networks are divided into 2 parts; global features and local features. In simple definition, the former refers to statistical values which represent a single network while the later numerically represent local entities such as junctions and line segments.

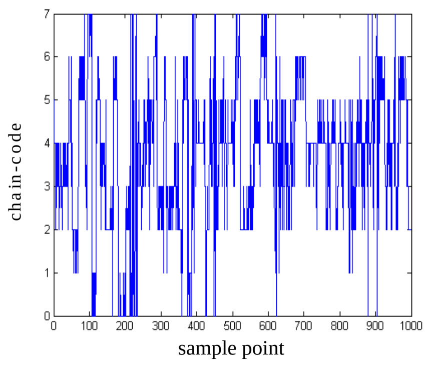
Based on what have been said, a question that can be asked is; how do we define a crack pattern? Is a crack pattern regarded as a single crack network or is it regarded as cracks bounded in a particular image or region. This is the issue in which we would to refer in CBR terms as "content of interest" of "object of interest". This question will be further looked into in section 5.2. For simplicity however, it is assume at this point that a single crack network is defined as a crack pattern.

4.3.2 The Basic Features

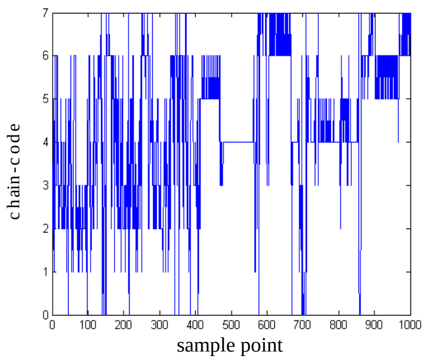
From a global point of view, a single crack network holds information on the number of local entities detected by the *crack following* routine where these include the number of joints and



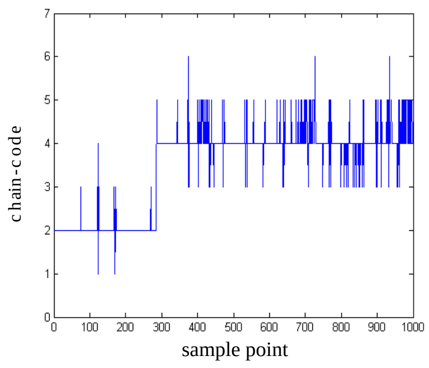
a) Circular



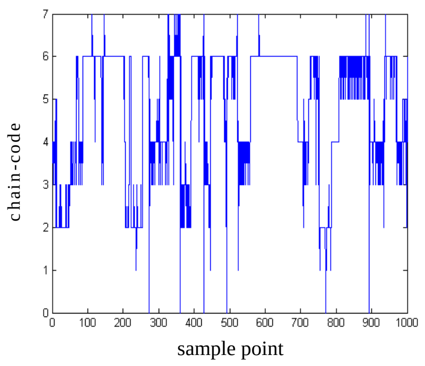
b) Spider-web



c) Random



d) Uni-directional



e) Rectangular

Figure 4.3: Chain-codes for the first 1001 pixels representing images of figures 4.1(a), 4.1(b), 4.1(c), 4.1(d) and 4.1(e) respectively.

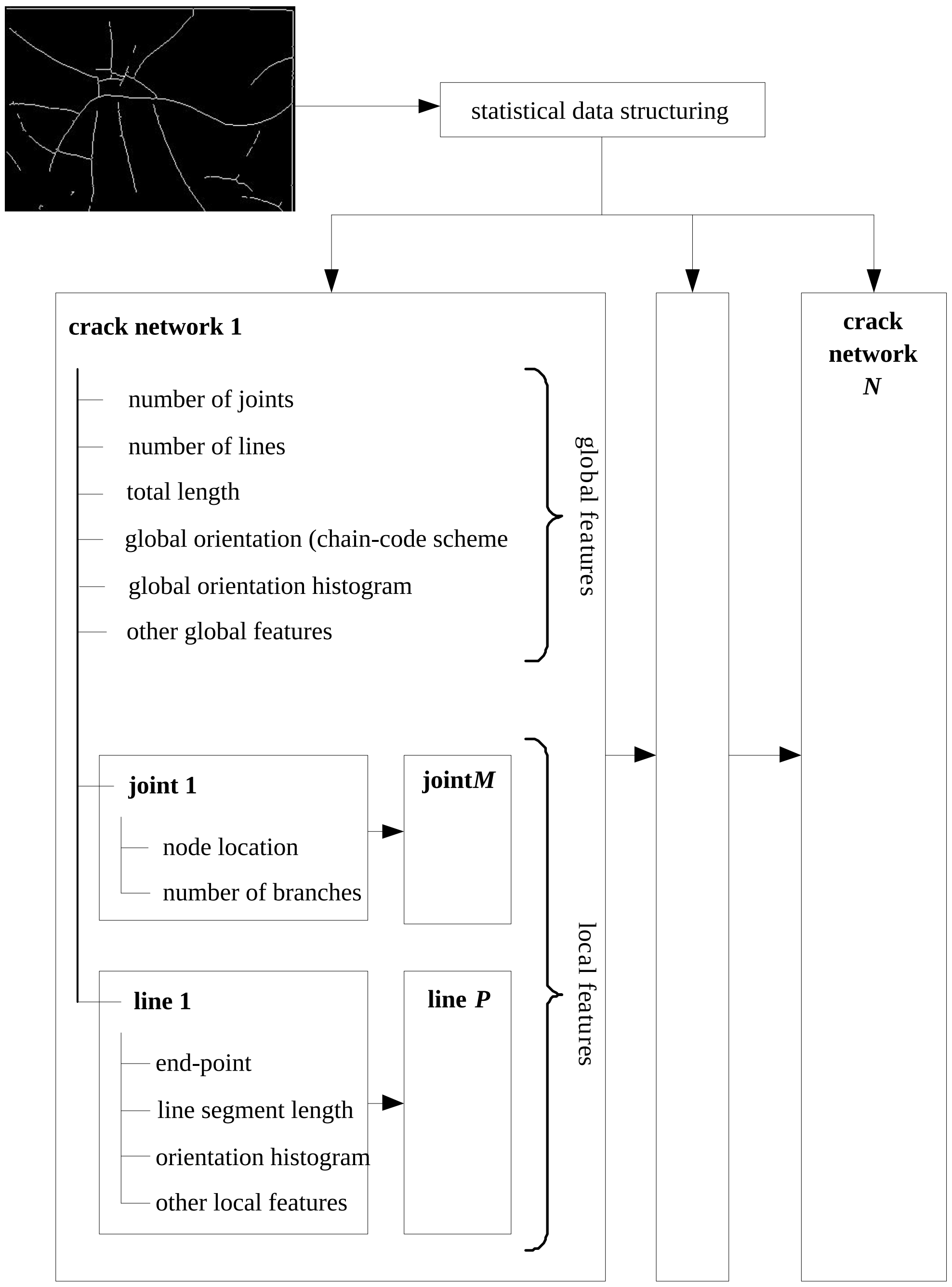


Figure 4.4: Statistically structured crack network.

line segments. These entities are accumulated as the cracks are "followed". Basic features are also computed for use in higher level feature extraction processes.

The perimeter is computed as the length of a chain [55, 60]. The formula for the perimeter is

$$P = n_e + \sqrt{2}n_o \tag{4.1}$$

where n_e is the number of even chain elements and n_o the number of odd chain elements. The total length of a crack network length is calculated using equation 4.1 and its significance in our approach will be explained in section 4.6.

One of the most important information that need to be captured by the *crack following* routine is line orientation. We accumulate the chain-codes in a histogram with 8 bins where each bin represents a chain-code. In global perspective, we call this histogram a *global orientation histogram*. It roughly indicates the orientation spread of a particular *crack network*, thus globally we can tell whether there is any dominant direction or are the directions equally spread.

Each crack network consist of line segments and these line segments are connected by junctions. These two components (line segments and junctions) are considered as local entities of a crack network. The important feature of a junction is its location in a crack network. As a whole, this generally tells how concentrated or sparsely distributed junctions are in a crack network. Unfortunately, the best way of representing this feature numerically is still unknown.

Line segments keep almost same type of information compared to the global entity except that it captures statistical data on a line-to-line basis. This means that each line segment have their own features. For every line segments, the end points are marked and the length is recorded using equation 4.1. Orientation histograms are also constructed for each line segment but this time the term used is *local orientation histogram* due to the local nature of the lines.

A complete crack network structure allows straight forward manipulation of global and local network entities. This simple framework is useful for the later stages of extracting high-level features.

4.4 Crack Network Pruning

One available option once network of cracks have been structurally represented, is to perform pruning in order to eliminate any unwanted network. "Unwanted" in this scope refers to the significance of a crack network or the level in which it influence the outcome of a certain process.

Among the criteria that can be used as an indication of the significance level is total length, number of junctions and number of line segments. Among these, total length is the most reasonable criteria. We can always assume that a "valid" crack network in some cases can appear without any junctions or with only a single line segment. On the contrary, short cracks will certainly pose minimal significance if rest of the crack network are lengthy.

In the implementation, we traverse the *network tree* and search for networks with total length less than a predetermined value, L_{min}. Once these networks have been detected, they are deleted from the *network tree* leaving only "significant" networks behind.

Figure 4.5 and 4.6 show cracks for several level of pruning using network length, L as a pruning criteria. The cracks are reconstructed after pruning for display purposes.

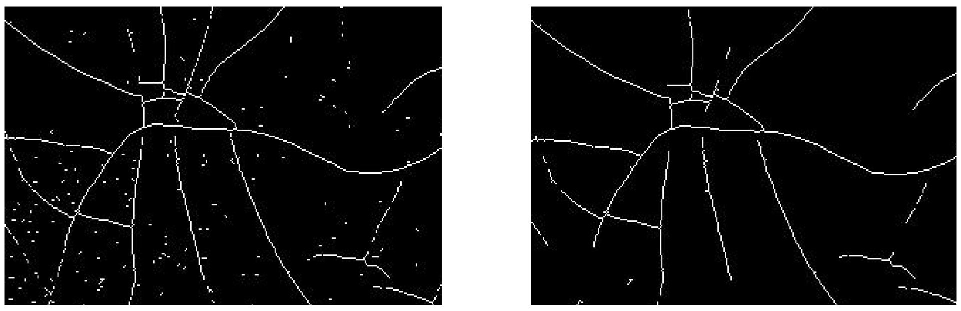


Figure 4.5: Crack pruning as a tool to eliminate noise.

This action behaves as a filtering mechanism to eliminate any element of a crack network which does not fulfil a certain criteria. This functionality can be extended not only to prune a whole crack network, but instead it can also be tuned to eliminate short line segments. However, this mechanism has not been implemented and will remain as a consideration for future work.

4.5 Hierarchical Representation of Features

Features are hierarchically divided into three layers, namely *local layer*, *global layer* and *image layer*. This is graphically explained in figure 4.7. Conceptually, the approach we take can be described as a structural scaling approach that views objects of interest at

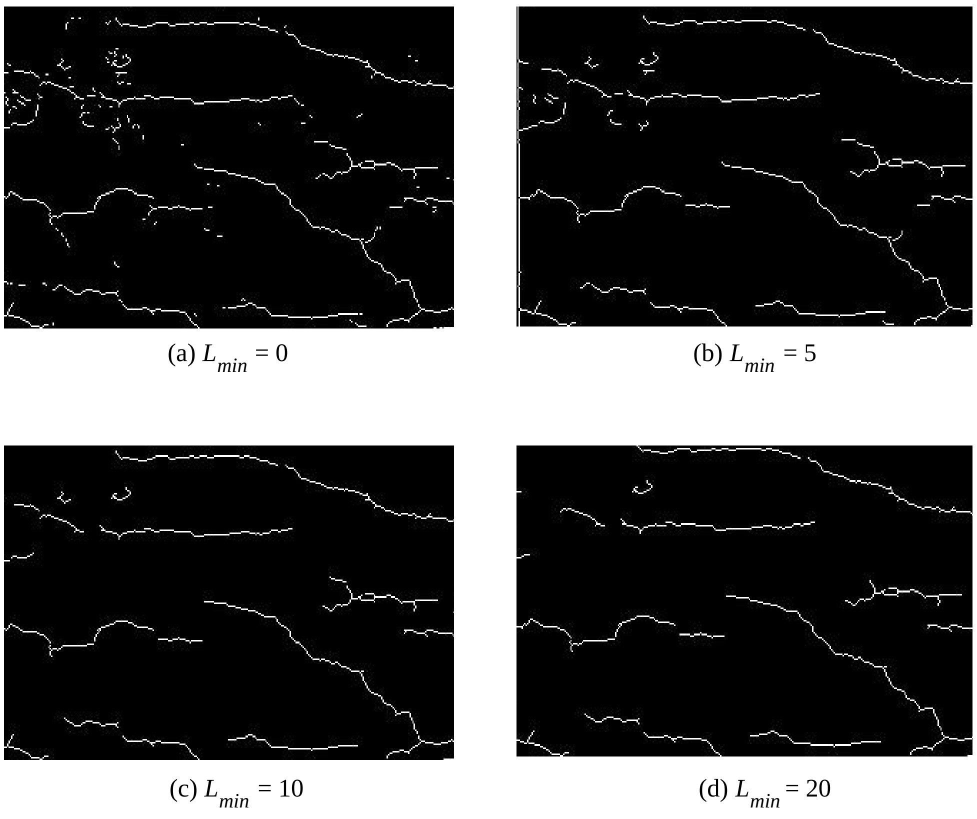

(a) $L_{min} = 0$ (b) $L_{min} = 5$ (c) $L_{min} = 10$ (d) $L_{min} = 20$

Figure 4.6: Eliminating insignificant cracks.

hierarchically-separated yet related structural levels. The following sub-sections demonstrate the relation between levels while presenting preliminary experimental results.

The first layer (local) is more like a “hidden layer” since it is not used explicitly as an object descriptor. However, its importance is unquestionable due to the fact that it functions as a “root” for the higher level features to “grow”. This first layer of interest concentrates on fine entities which involves line segments. The primary assumption is that a crack network consists of multiple line segments. However this assumption can be relaxed when we deal with crack networks that do not contain more than a single line. The second layer (global) is used basically as means of numerically describing a crack network. It is useful in the sense that each crack network can be initially assumed to be our “object of interest”. Each “object of interest” must be associated with a descriptor thus explains the need for the global layer features. The top layer which is the image layer, is most useful when we assume that an image contains just one particular type of crack pattern. In other words, there is only one “object of interest” in an image.

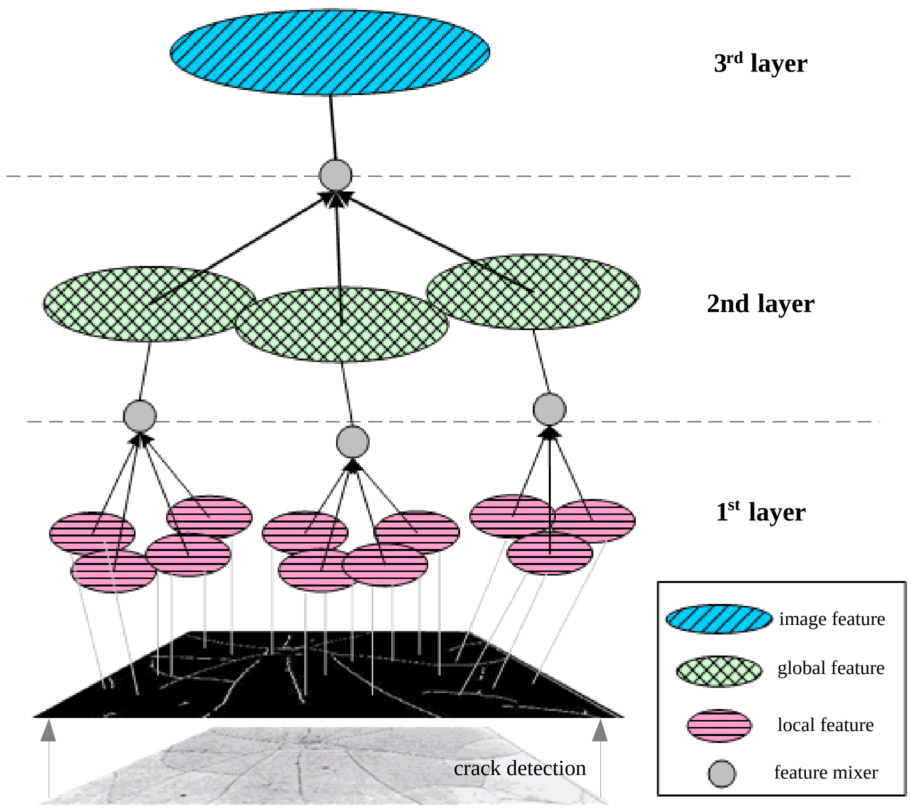


Figure 4.7: Hierarchy of features.

The next section describes in a sequential manner, the methodology in which the multi-layered features are computed.

4.6 Extracting High-level Features

The next approach involves generating meaningful features to numerically describe crack patterns. We take into account the fact that cracks are represented by line segments. The arrangements of these line segments are the main factor that separate cracks into different pattern classes. Since in most cases crack patterns are formed through combination of line segments, the approaches we undertake consider generating meaningful local features as an important starting step. These local features are then exploited to form global descriptors. For analytical purposes, we generate higher level features based on the global features which we called image feature.

For the following sections, we address local, global and image layer parameters with subscripts a, b and c respectively. The following are common terminologies and notations needed for the ensuing discussions:

(a) Local layer :

- n_a : total number of pixels in a line segment
- $C_a = \{c_{(1)}, ..., c_{(n_a-1)}\}$: a set of line segment chain-code
- ℓ_a : length of a line segment
- s_a : significance measure of a line segment
- r_a : straight line to actual length ratio (LTLR) of a line segment
- d_a : directionality of a line segment

(b) Global layer :

- n_b : total number of line segments in a single crack network
- $L_b = \{\ell_{a_{(1)}}, ..., \ell_{a_{(n_b)}}\}$: a set of line segment length
- ℓ_b : total length of a crack network
- $S_b = \{s_{a_{(1)}}, ..., s_{a_{(n_b)}}\}$: a set of line segment significance measure
- s_b : significance measure of a crack network
- r_b : straight line to actual length ratio (LTLR) of a crack network
- d_b : directionality of a crack network
- μ_b : mean line segment length percentage
- σ_b : standard deviation of line segment length percentage

(c) Image layer :

- n_c : total number of crack networks in an image
- $L_c = \{\ell_{b_{(1)}}, ..., \ell_{b_{(nc)}}\}$: a set of crack network length
- ℓ_c : total length of cracks in an image
- $S_c = \{s_{b_{(1)}}, ..., s_{b_{(nc)}}\}$: a set of crack network significance measure
- r_b : straight line to actual length ratio (LRTLR) of cracks in an image
- d_b : directionality of cracks in an image
- μ_c : mean crack network length percentage
- σ_c : standard deviation of crack network length percentage

4.6.1 The Significance Measure

For each of the line segment, we compute a significance value that roughly tells numerically how significant a line segment is in global terms. In other words, it indicates how much a line segment can influence or contribute to the computation of a global feature. In doing this, we use the set line segment length L_b as an indication of their influence to the whole crack network. The formulation of the *significance measure*, s_a is

$$s_{a_{(i)}} = \frac{\ell_{a_{(i)}}}{\ell_b} \quad \forall \quad i = 1...n_b \tag{4.2}$$

where $S_b = \{s_{a_{(1)}}, ..., s_{a_{(n_b)}}\}$ is a set of significance value representing each line segment in a crack network. For the image level, the computation of S_c is

$$s_{b_{(i)}} = \frac{\ell_{b_{(i)}}}{\ell_c} \quad \forall \quad i = 1...n_c \tag{4.3}$$

where $S_c = \{s_{b_{(1)}}, ..., s_{b_{(n_c)}}\}$ is a set of significance value representing each crack network in an image.

4.6.2 Line Segment Length as a Feature

The length of the line segments can be used as a feature since the distribution of lengths is observed to vary over different crack types. The temporarily stored lengths of the line segments are utilized by first converting them into a measure of percentage with respect to the total length of their corresponding crack network ℓ_b which we denote as a set of values, $\overline{S_b} = \{\overline{s_a}_{(1)}, ..., \overline{s_a}_{(n_b)}\}$. The mean and standard deviation of the percentage length distribution is then computed for a single crack network denoted as μ_b and σ_b respectively.

The same methodology is applied to calculate mean and standard deviation of crack network length percentage distribution, μ_c and σ_c. $\overline{S_c} = \{\overline{s_b}_{(1)}, ..., \overline{s_b}_{(n_b)}\}$ is determined where $\overline{S_c}$ represents a set of the crack network length percentage.

The value of σ_b and σ_c, varies between 0 for a single line crack network up to an unknown maxima. For classification purposes, the values will have to be normalized into the interval [0,1].

4.6.3 Directionality Measure

Through rough observations, circular cracks and rectangular cracks differ significantly in terms of their orientation spread which is visible from their orientation histograms [10]. Our approach considers this factor by extracting local orientation features that are then combined to produce a global feature representing a particular crack network. The *directionality value* indicates the straightness measure of a line segment.

Firstly, we compute local directionality value for each line segment in a crack network using the steps below.

1. Generate orientation histograms for each line segment in a single crack network and normalize the histograms such that the values are within the interval [0,1] as shown by the equations below:

$$H = \{h_{a_{(0)}}, ..., h_{a_{(7)}}\}, \tag{4.4}$$

$$\overline{h_{a(j)}} = \frac{h_{a_{(j)}}}{\sum_{i=0}^{7} h_{a_{(i)}}} \qquad \forall \quad j = 0...7, \tag{4.5}$$

   where H is the orientation histogram while $\overline{H}$ is the normalized orientation histogram defined as a set of accumulation values $\{\overline{h_{a(i)}}, ..., \overline{h_{a(7)}}\}$.

2. We define 4 simple directionality histogram models where each bin represents a chain-code index of 0 to 7 respectively:

$$\begin{aligned} m_1 &= 0.125\{1,1,1,1,1,1,1,1\}, & (4.6)\\ m_2 &= 0.25\{1,1,1,1,0,0,0,0\}, & (4.7)\\ m_3 &= 0.5\{1,1,0,0,0,0,0,0\}, & (4.8)\\ m_4 &= \{1,0,0,0,0,0,0,0\}. & (4.9) \end{aligned}$$

   m_1, m_2, m_3 and m_4 represent ideal histograms as far as orientation spread is concerned where they roughly indicate circular, semi-circular, bi-directional and uni-directional spread respectively.

3. Measure dissimilarity between each normalized histogram and all 4 ideal histograms using Euclidean distance measure. Dissimilarity corresponding to m_1 is as shown below:

$$e_1 = \sqrt{\sum_{j=0}^{7} (\overline{h_{a(j)}} - m_{1_{(j)}})^2} \tag{4.10}$$

   where e_1 represents error for a single histogram dissimilarity measure. A different approach is taken to compute dissimilarity for m_2, m_3 and m_4 where each of them are rotated and dissimilarity, e_2, e_3 and e_4 are calculated for each rotation.

4. Construct a similarity histogram, $W_a = \{(1-e_1),(1-e_2),(1-e_3),(1-e_4)\}$. Let $W_a = \{f_1, f_2, f_3, f_4\}$, the histogram indicates the score obtained by a particular line corresponding to each class defined in step 2.

5. Compute the directionality value by using the following equation:

$$d_a = \begin{cases} 0.25 - \frac{f_1}{4} & \text{if } max\{W_a\} = f_1 \\ 0.5 - \frac{f_2}{4} & \text{if } max\{W_a\} = f_2 \\ 0.5 + \frac{f_3}{4} & \text{if } max\{W_a\} = f_3 \\ 0.75 + \frac{f_4}{4} & \text{if } max\{W_a\} = f_4 \end{cases} \tag{4.11}$$

where directionality, d_a is used as one of the orientation-based local features. Each line in a particular network has now been assigned a directionality value which is normalized to the interval [0,1].

The global directionality histogram, W_b is then generated taking into account the significance of each line in the crack network, S_b.

Global directionality d_b is computed using the algorithm below:

1. For all existing lines in the crack network, find the index, k (where $k \in$[0,3]) of the maximum similarity, $max\{W_b\}$.

2. Multiply $max\{W_b\}$, with the significance measure, S_a.

3. Accumulate the resultant values in a global directionality histogram, W_b at the appropriate bin, k.

4. Compute the directionality value, d_b similar to the one explained in equation 4.11

A slightly different approach is implemented in order to compute the image directionality, d_c. Image directionality is computed by taking the sum of weighted d_b over the number of crack network as shown in equation 4.12.

$$d_c = \sum_{i=1}^{n_c} s_{b_{(i)}} d_b \tag{4.12}$$

4.6.4 Straight Line to Actual Length Ratio

Another feature which we find useful is the straight line to actual length ratio (LTLR). Prior to the process, we construct a straight line model of the crack patterns. Straight line length, $\hat{\delta}$ is defined as Euclidean distance between two points and the points can be as the following:

- junction to junction
- junction to end point
- end point to end point.

Actual length is the distance as given by equation 4.1. Local LTLR is computed by taking the ratio between $\hat{\delta}$ and actual line segment length, ℓ_a for all existing line segments.

Each crack network is then assigned a LTLR value r_b based on the set of values obtained in the local layer. The same goes for the image layer where a set of r_b is used to determine r_c. The computations are as shown in equations 4.13 and 4.14.

$$r_b = \sum_{i=1}^{n_b} s_{b_{(i)}} r_a \tag{4.13}$$

$$r_c = \sum_{i=1}^{n_c} s_{c_{(i)}} r_b \tag{4.14}$$

The ratio between the direct distance and the actual distance between two points of a line segment gives a rough measure of "straightness". Locally it tells how straight the lines are. A low value generally means that a line is either circular or jagged.

4.7 Experimental Setup

For our experiments, we aim to show the level in which each of the chosen features succeeded in separating crack patterns according to their appropriate classes. We also hope to identify the weaknesses of the approach we used for further improvements and additions in the future.

To make analytical comparisons simple and comprehensive, we used the images in figure 4.1 as our test images for majority of our experiments. On subjective terms, these are the images which we think can represent the five crack classes we mentioned in section 4.2. It is important to note that these images are not deemed the best representations of the crack classes. We purposely set this condition because we believe intermediate representations will draw our experiments closer to practical reality. Let us label the images before we commence. The labels are shown in table 4.1.

Source	Dimension	Type/Class	Label
Figure 4.1(a)	398 x 268	Circular	A
Figure 4.1(b)	256 x 256	Spider-web	B
Figure 4.1(c)	216 x 326	Random	C
Figure 4.1(d)	218 x 326	Uni-directional	D
Figure 4.1(e)	218 x 326	Random	E

Table 4.1: Images labelled according to classes.

The fact that these images are not of the same size is not a worrying factor since the features are not sensitive to different image dimension since they depend on the detected cracks as means of existence. Furthermore, the features are normalized so the analysis is invariant to scaling.

4.8 Results and Discussion

This section is divided into three parts, each corresponding to experiments performed on a particular feature level. We demonstrate the flow of features from the local level up to the image level. The aim of the experiments is to show the separability of features corresponding to different crack classes. The results are deemed crucial and important since it tells the effectiveness of the features used.

Figure 4.8 shows plots of the sorted local features (ascending) for each line in a crack network. The plots are shown for all five class representatives. The y-axis represents the feature value while the x-axis represents the n^{th} line segment. In the figure, $S_b = \{s_{a_{(1)}}, ..., s_{a_{(n_b)}}\}$, a set of line segment significance measure, $R_b = \{r_{a_{(1)}}, ..., r_{a_{(n_b)}}\}$, a set of line segment LTLR and $D_b = \{d_{a_{(1)}}, ..., d_{a_{(n_b)}}\}$, a set of line segment directionality. To simplify the analysis, we combine the local features of all crack network and assume they belong to the same network. This step is valid since the initial assumption is these five images represent each of the five classes.

As can be seen, S_b for figures 4.8(d) and 4.8(e) increases quite rapidly compared to the others. Most of the lines in class D have maximum significance value since they are mostly single line networks. On the contrary, line segments in class E are mostly classified as insignificant due to the fact that they have very low values. There is no significant difference between plots in figures 4.8(a), 4.8(b) and 4.8(c) apart from the range between the minimum and maximum values of the features. Table 4.2 summarizes the results.

Class	S_b			D_b			R_b		
	min	max	dev.	min	max	dev.	min	max	dev.
A	0	1	1	0	1	1	0.5	1	0.5
B	0	1	1	0.1771	1	0.8229	0.3705	1	0.6295
C	0	1	1	0.1671	1	0.8329	0.5270	1	0.4730
D	0.0018	1	0.9982	0.5	1	0.5	0.7071	1	0.2929
E	0	1	1	0.4271	1	0.5729	0.7071	1	0.2929

Table 4.2: Feature deviation for the five classes.

Figures 4.9, 4.10, 4.11, 4.12 and 4.13 show scatter plots showing features distributed in 2-dimensional plots for the five classes. The y-axis and x-axis represent directionality d and LTLR r respectively. It demonstrates the difference of using the *significance measure* S as a weighting for calculation of a global feature. For all the plots, points marked by a "*" sign show the location of global feature in the feature space. These global features are then combined to form the image feature marked by a "+" sign.

Table 4.3 summarizes the results while figure 4.14 visualizes the position of extracted features in a scatter plot.

Image	Directionality	LTLR
A	0.5777	0.6768
B	0.5530	0.6907
C	0.6975	0.8201
D	0.8432	0.8131
E	0.8666	0.8708

Table 4.3: Directionality and LTLR for crack classes A, B, C, D and E.

As can be seen, the image samples are quite distance apart from each other except classes A and B. Class D and E are also quite close together. This is expected since uni-directional and rectangular cracks are observed to consist of mainly straight line segments. Class C however resides in the middle of the two clusters.

One interesting observation is the location of the clusters, which only populate one quarter of the scatter plot. The reason for this lies in the fact that the images used are not "extreme cases". They are just intermediate representations of the five crack classes, which in this experiment shows considerably good results.

In practical, the probability of an image consisting of only one crack pattern is very low.

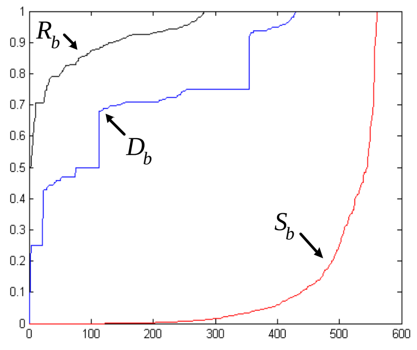
The fact that actual painting images available in our collection are in many cases several thousands in dimension strengthens the claim. In this case, we can relax the assumption that an image consist of only one crack pattern thus making image feature unimportant. We assume that each crack network represents a single crack pattern or in other words, our "object of interest". Our next experiment demonstrates the pattern in which local and global features possess for a considerably large image as in figure 3.10. We extract all the local features and later generate the global features. Although image features are not crucially important in this case, we also calculate them for analytical reason. Important statistical values and relevant features associated with the image layers are as shown in table 4.4

feature	value
number of crack networks	202
number of lines	5833
number of junctions	1358
total length	27780.93
d_c	0.7428
r_c	0.8005
μ_c	0.3366
σ_c	4.5049

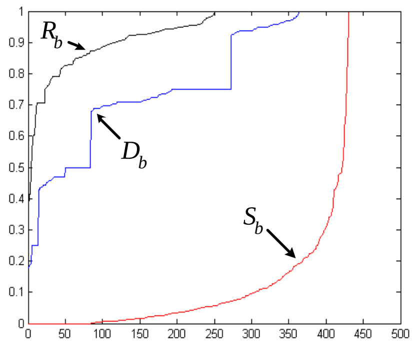
Table 4.4: Statistical information and image layer features.

The extracted local and global features are marked on the same plot to better visualize the relations between them (see figure 4.15). The y and x axes represent directionality, (d_a and d_b) and LTLR, (r_a and r_b) respectively.

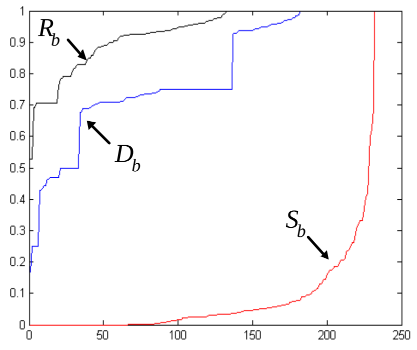
The data will be used for the next stage - data classification.



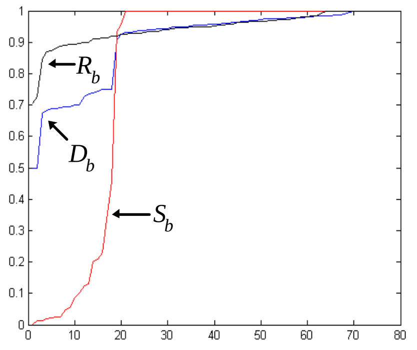
a) Circular



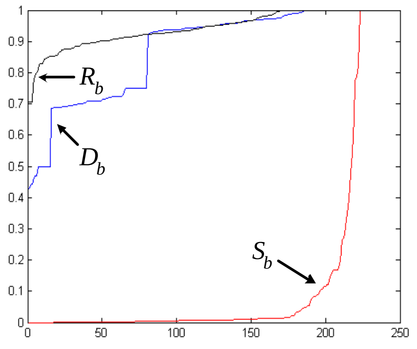
b) Spider-web



c) Random



d) Uni-directional



e) Rectangular

Figure 4.8: Sorted local features for 5 crack classes.

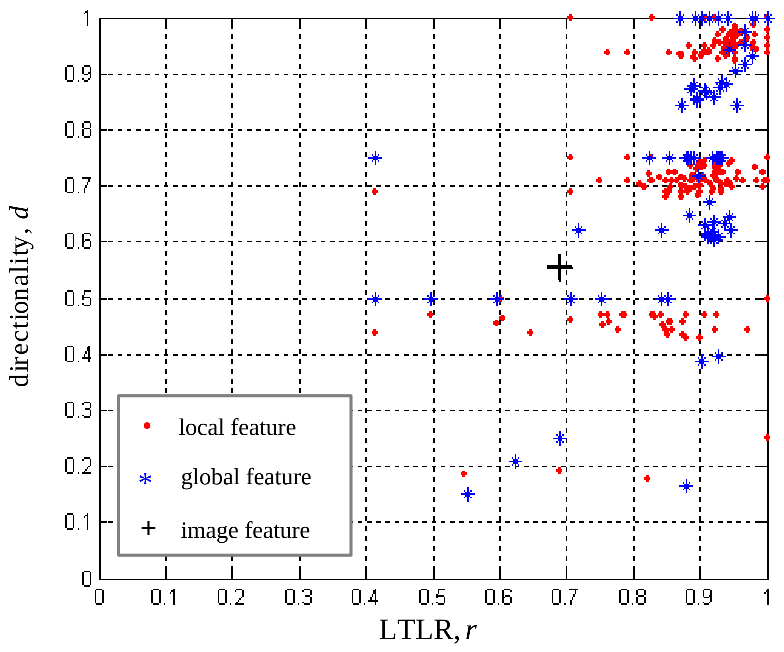


Figure 4.9: Features of the three layers corresponding to class A (circular).

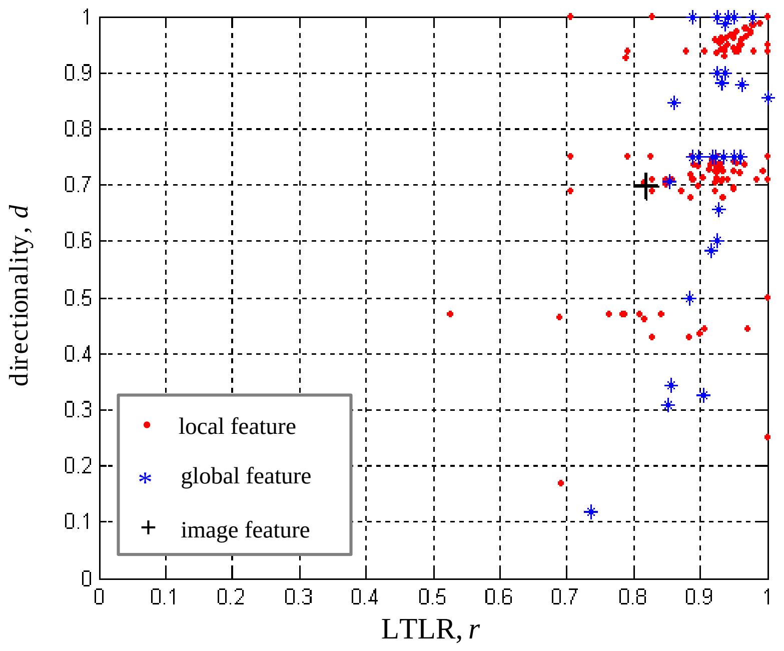


Figure 4.10: Features of the three layers corresponding to class B (spider-web).

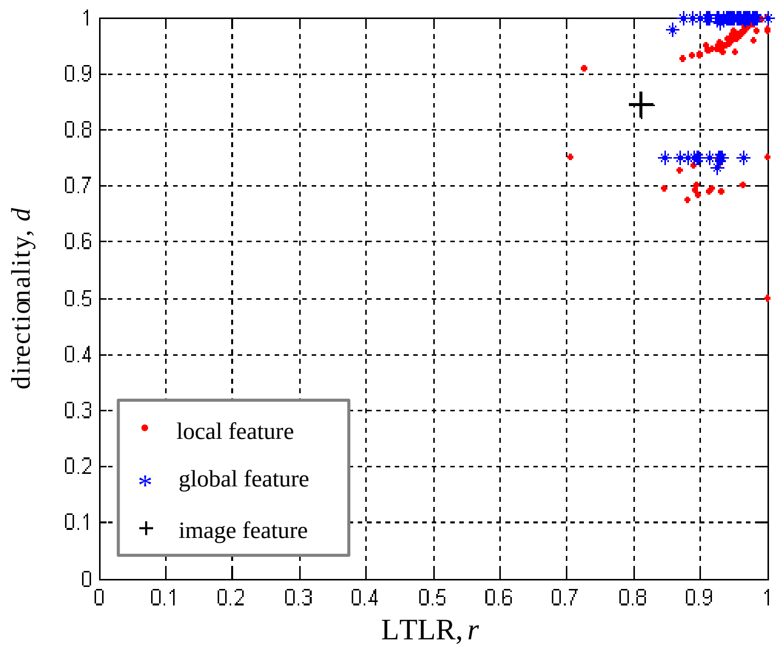


Figure 4.11: Features of the three layers corresponding to class C (random).

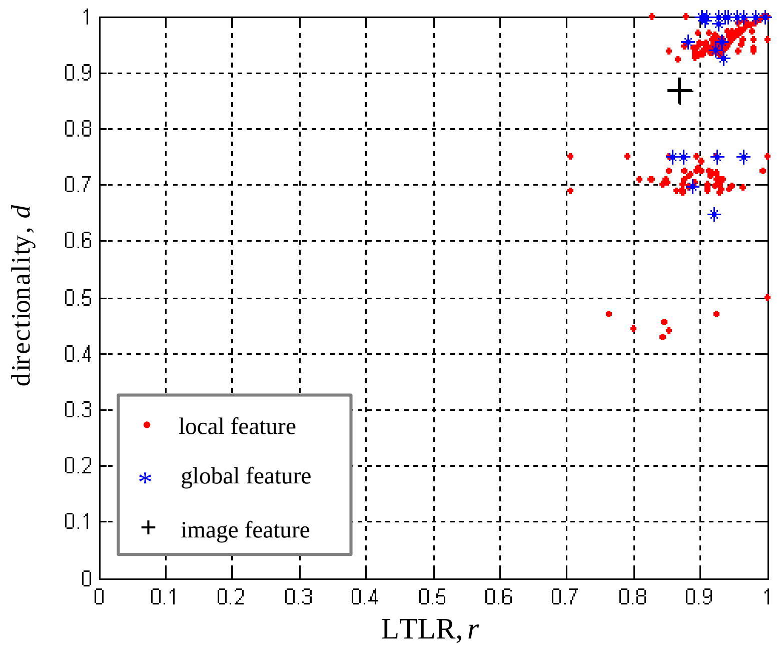


Figure 4.12: Features of the three layers corresponding to class D (uni-directional).

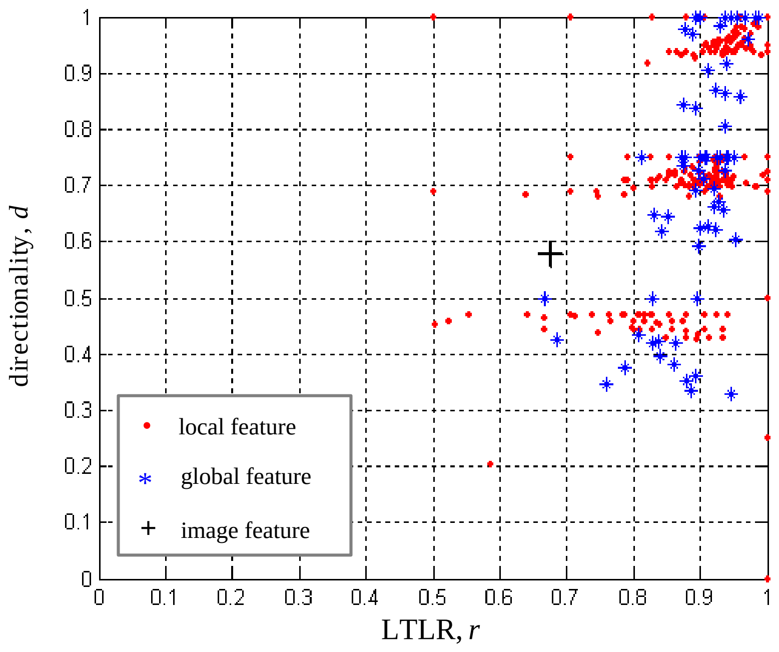


Figure 4.13: Features of the three layers corresponding to class E (rectangular).

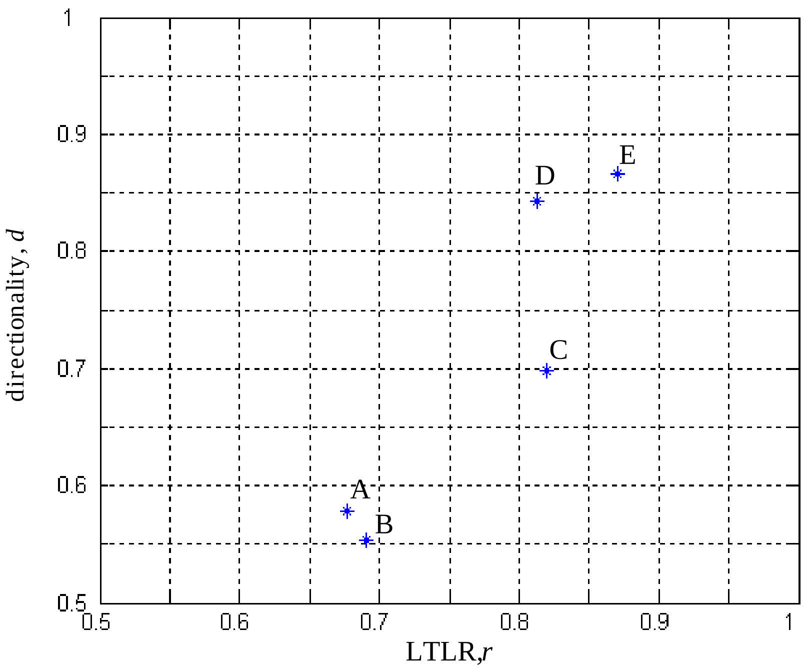


Figure 4.14: Image features in scatter plot corresponding to classes A, B, C, D and E.

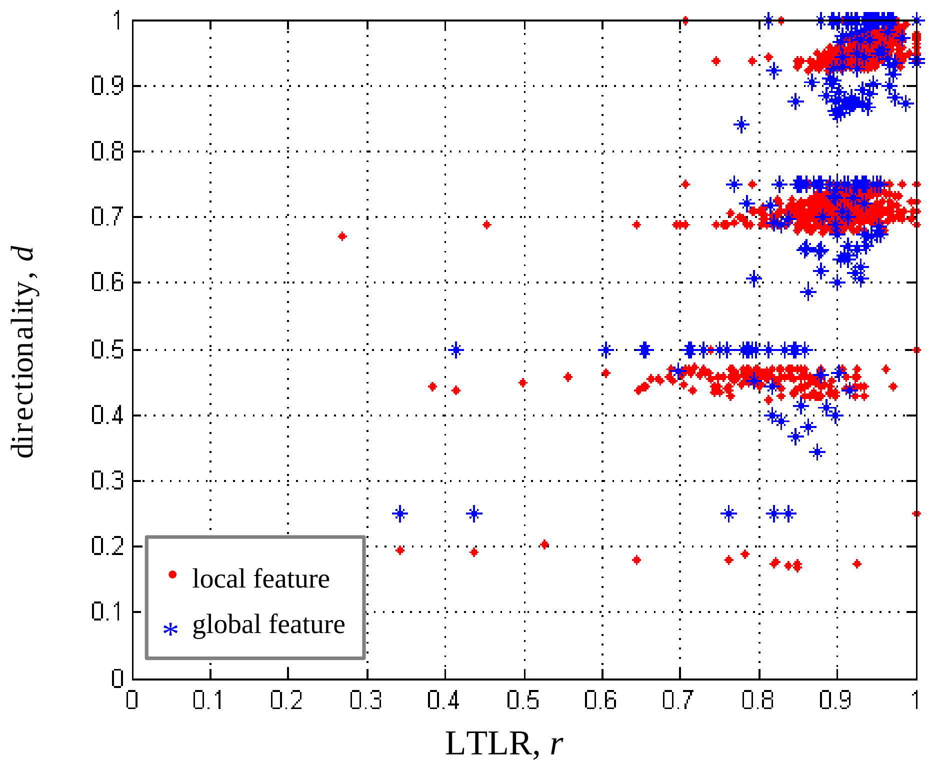


Figure 4.15: Local and global features for an image assumed to contain multiple crack patterns.

Chapter 5

Classification

5.1 Introduction

In this chapter the techniques and results of the previous chapters are brought together by applying them to a specific problem - that of classifying different patterns of cracks on the surface of paintings. Issues surrounding the problems are discussed and probable solutions are outlined. The techniques that have been used are described and results obtained are finally presented.

5.2 Classification Issues

In [25], Varley performed classification of a set of 40 images of the paint layer of old oil paintings showing the patterns of cracks formed as the painting ages. His approach is motivated by the work done by Bucklow in [61] where classification of crack images was based on subjective scores given by a panel of human subjects. The classification accuracy obtained in that study was deemed to be excellent by experts in the art history field.

5.2.1 Class Labelling

As explained briefly in section 4.2, Varley modelled cracks using Bézier curves and the parameters of the Bézier curves were used as features. He performed a two-stage classification technique using linear discriminant classifier [62] and EM classifier [63] to train and classify data. The training stage used 200 objects for a 4-class problem.

The main issue that we would like to raise is the existence of priori classified data in Varley's work which are taken from [61]. On the contrary, data labels in our work are mostly unknown due to the subjective nature of the crack patterns. To-date, there is no manual classification performed on our sample images based on the fact that such action needs high level of expertise to get a reliable classification result. Furthermore, the number of images that need classification is extremely high, thus making the task a lot harder.

Having these questions in mind, as an initial classification step, we purposely work on unlabelled images and we adopt a classification approach that will conform to this situation.

5.2.2 The Interpretation of "Content"

Crack patterns can be interpreted in several ways. From a content-based view-point, the question to be asked is how we define a single crack pattern. Based on Varley's work [25], classification is made based on image-to-image basis. In other words, all crack patterns in a single image was assumed to belong to the same crack class. This is fully understood since his work was not at all concerned with content-based issues. However, in our case, this undefined interpretation of our "content of interest" is an important matter. Real application of content-based crack analysis needs processes on very large images with hundreds of crack networks and thus the interpretation of "content" must be addressed.

For content extraction, processing and storage this definition is seen as an important element. For instance, a single crack network can be acknowledged to represent a "content". However, this assumption might be less reasonable the shorter the length of a crack network. In another case, a priori determined square grid can be assumed to contain a single "content" but yet again this assumption is not particularly true since cracks are thin structures and they tend to "overflow" into neighbouring grids. Furthermore, patterns of cracks are dependent on the region size and shape in which we view them. The approach explained in section 4.3 and later visualized in section 4.4 processes cracks assuming that each pair of adjacent pixels belong to the same entity which is a crack network. Post-processing must be done to further elaborate the definition of "content" using more complex criterion.

To-date, we define a crack pattern as a single crack network. Determination of alternative definitions for crack pattern is still an on-going work.

5.2.3 Supervised or Unsupervised Learning?

The main difference between supervised and unsupervised learning is in terms of prior knowledge of the data samples used for training [62]. In the former, data samples are labelled, which means that their classes are priori known. However, it is not the same for unsupervised learning where class labels do not exist. Putting it in a different form, supervised learning benefits from the guidance of a “teacher” while unsupervised learning gain knowledge without any assistance from a “teacher”.

Our early approach towards classifying crack data takes into account the unavailability of labelled samples so unsupervised learning techniques are more reasonable options.

5.3 Unsupervised Crack Classification

In many applications of pattern recognition, it is extremely difficult or expensive, or even impossible, to reliably label a training sample with its true category. Unsupervised classification refers to situations where the objective is to construct decision boundaries based on unlabelled training data. Unsupervised classification is also known as data clustering which is a generic label for a variety of procedures designed to find natural groupings, or clusters, in multidimensional data, based on measured or perceived similarities among the patterns [64]. Category labels and other information about the source of the data influence the interpretation of the clustering, not the formation of the clusters. Unsupervised classification or clustering is a very difficult problem because data can reveal clusters with different shapes and sizes. However, it is a very important and useful technique in the sense that it is fast, reliable and consistent in organizing large amount of data [65].

Suppose a problem is given where the training set, H_u are unlabelled. For each sample, $\underline{x} \in H_u$, the class origin or label is unknown. Three desirable attributes of H_u are [62] :

- the cardinality of H_u is large
- all classes are represented in H_u
- subsets of H_u may be formed into natural groupings or clusters, where each cluster most likely corresponds to an underlying pattern class. This attribute of H_u is strongly influenced by our choice of features

Although we cannot guarantee that our training set will satisfy the three conditions, we will perform experiments using two clustering techniques namely, the *MacQueen k-means clustering* [66, 67] and *fuzzy k-means clustering* [68, 67].

Cluster analysis methods divide the set of processed patterns into subsets which is also known as clusters, based on the mutual similarity of subset elements. Each cluster contains patterns representing objects that are similar according to the selected object description and similarity criteria. Objects that are not similar reside in different clusters.

To make things simpler let us define some commonly used notations:

- n - number of data points/samples
- d - number of features
- k - number of classes

Let us define a d-dimensional set of n data points $X = \{x_1, ..., x_n\}$ as the data to be clustered and a d-dimensional set of k centres $C = \{c_1, ...c_k\}$ as the clustering solution that an iterative algorithm refines.

A membership function $m(c_j|x_i)$ defines the proportion of data points x_i that belongs to centre c_j with constraints $m(c_j|x_i) \geq 0$ and $\sum_{j=1}^{k} m(c_j|x_i) = 1$. Some algorithm uses a *hard* membership function, meaning $m(c_j|x_i) \in 0, 1$, while others use a *soft* membership function, meaning $0 \leq m(c_j|x_i) \leq 1$.

5.3.1 K-means Clustering

One of the earliest and simplest cluster analysis method is the *MacQueen k-means* (KM) technique [66]. It partitions data into k sets where k in this case refers to the number of classes. The solution is then a set of k centres, each of which is located at the centroid of data for which it is the closest centre. For the membership function, each data point belongs to its nearest centre, forming a Voronoi partition of the data. The objective function that the KM algorithm optimizes is

$$KM(X, C) = \sum_{i=1}^{n} \min_{j \in \{1...k\}} \parallel x_i - c_j \parallel^2 \tag{5.1}$$

This objective function gives an algorithm which minimizes the within-cluster variance (the squared distance between each centre and its assigned data points).

The membership function function for KM is:

$$m_{KM}(c_l|x_i) = \begin{cases} 1 & \text{if } l = \arg\min_j \parallel x_i - c_j \parallel^2 \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

For implementation, the centre points (C) have to be initialized prior to the process. The choice of C is still a challenging issue [62].

5.3.2 Fuzzy K-means Clustering

The *Fuzzy k-means* algorithm (FKM) (which is also called *Fuzzy c-means*) [68] is an adaptation of the *k-means* algorithm that uses *soft* membership function. Unlike KM which assigns each data point to its closest centre, the FKM algorithm allows a data point to belong partly to all centres.

The objective function for FKM is:

$$FKM(X,C) \quad = \quad \sum_{i=1}^{n}\sum_{j=1}^{n} u_{ij}^{r} \parallel x_i - c_j \parallel^2 \tag{5.3}$$

where the parameter u_{ij} denotes the proportion of data point x_i that is assigned to centre c_j, and is under the constraint $\sum_{j=1}^{k} u_{ij} = 1$ for all i and $u_{ij} \geq 0$. The parameter $r \geq 1$. A larger value for r makes the method "more fuzzy".

Bezdek presented separate update function for u_{ij} and c_j. The u_{ij} update equation depends only on C and X, so we incorporate its update into the update for c_j. The membership function for FKM is:

$$m_{FKM}(c_l|x_i) = \frac{\parallel x_i - c_j \parallel^{\frac{-2}{r-1}}}{\sum_{j=1}^{k} \parallel x_i - c_j \parallel^{\frac{-2}{r-1}}}. \tag{5.4}$$

As r tends to approach 1 from above, "fuzziness" decreases and the algorithm behaves more like standard *k-means*. The centres share the data points less in this condition.

As for the *fuzzy k-means* technique, the centre points and the *fuzzy membership matrix* are the parameters which have to be initialized prior to the clustering process. The *soft* membership scheme introduced by the algorithm suits the nature of crack patterns which are quite subjective in terms of their class origin.

5.4 Results and Discussion

In our approach, classification can be done in two ways: the first being intra-image classification and the other option is inter-image classification. Recalling the experiments done in the previous chapter, the first assumes an image consists of just a single crack pattern whereas the second assumes an image consists of multiple patterns. We follow the same approach in this chapter.

The data shown in figure 4.15 are used for our first experiment, which corresponds to image of figure 3.10. We initialize our centres by using centres A and E (see table 4.3).

We cluster the data using the k-means clustering algorithm. Figure 5.1 visualizes the results. As can be seen, the data are clustered into 2 separate clusters after 10 iterations.

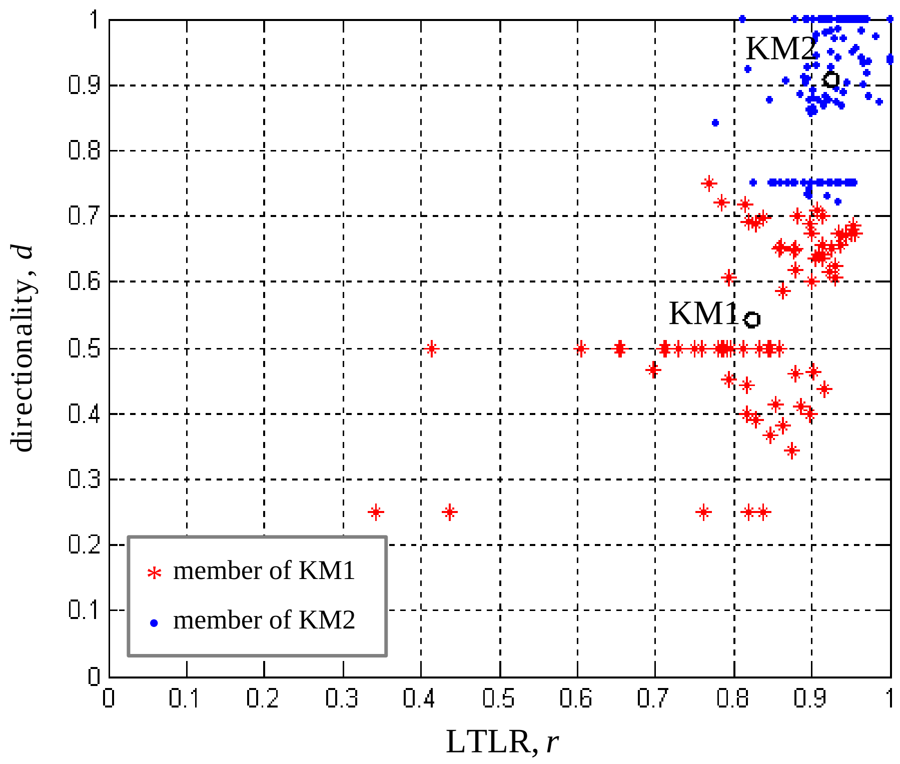


Figure 5.1: Classification using the k-means clustering technique.

The final centres are marked by circles with labels KM1 and KM2. Table 5.1 summarizes the results.

	KM1	KM2
location	(0.5429, 8248)	(0.9065, 0.9259)
total membership	72	130
percent membership	36.14 %	64.36 %

Table 5.1: Summarized k-means clustering results.

The same set of data is then clustered using the fuzzy k-means technique. As mentioned previously, this technique employs a *soft* membership strategy where each data is assigned a membership to each centre but with different values. For our experiment, we set our "fuzziness" at 1.25 and use the same centres, A and E. The results are as shown in figure 5.2.

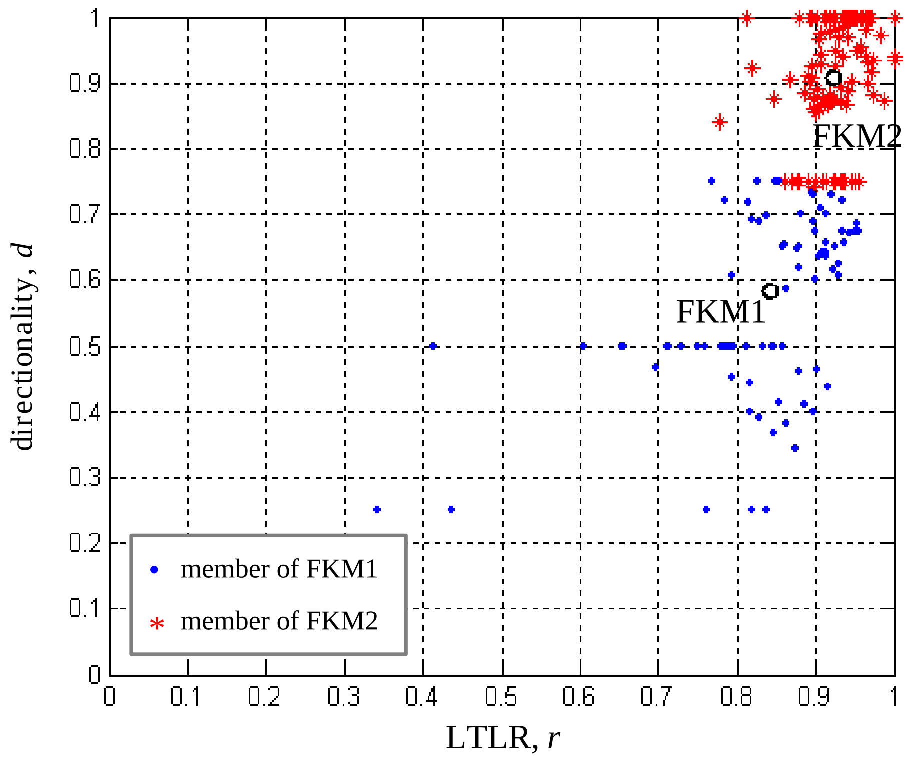


Figure 5.2: Classification using the fuzzy k-means clustering technique.

Based on the scatter plot, 2 clusters are clearly visible where each corresponds to the final centres which are marked by circles labelled FKM1 and FKM2 respectively. After 17 iterations, the clustered data owns memberships to both FKM1 (figure 5.3(a)) and FKM2 (figure 5.3(b)). We select the highest between the two memberships as an absolute cluster membership. The results are as summarized in table 5.2.

The *soft* membership strategy offers a different path into how we perceive class memberships in data classification. In the case of classifying crack patterns, this is seen as a potentially useful strategy since the perception of whether a particular crack pattern falls into a certain class can be thought to be a subjective matter. By introducing fuzzy membership, a notion of subjectivity can be integrated into the classification methodology.

Real implementation concerns will be those related to parameter initializations. Traditional clustering techniques such as the ones presented in this dissertation assume the

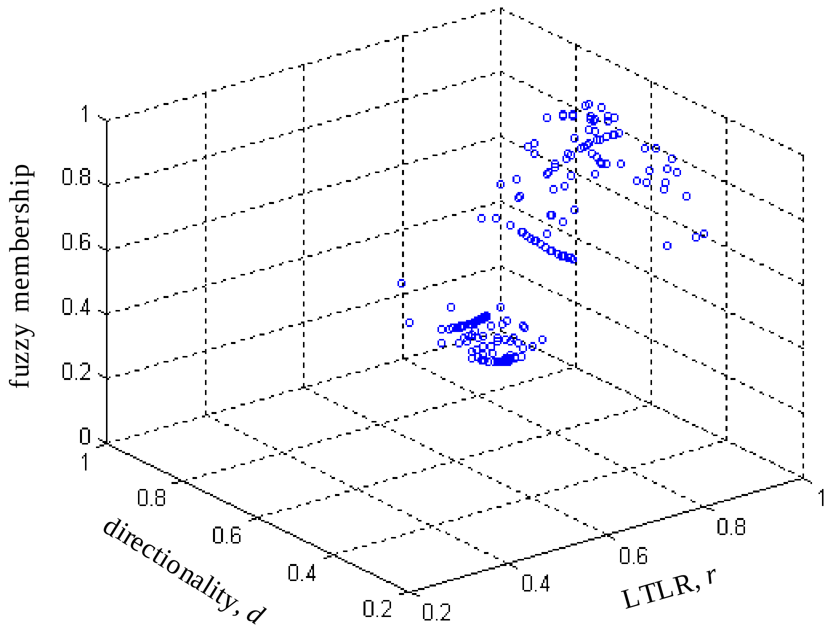
	FKM1	FKM2
location	(0.5827, 0.8416)	(0.9065, 0.9222)
highest fuzzy membership	0.9954	0.9969
lowest fuzzy membership	0.0031	0.0046
total membership	80	122
percent membership	39.60 %	60.40 %

Table 5.2: Summarized fuzzy k-means clustering results.

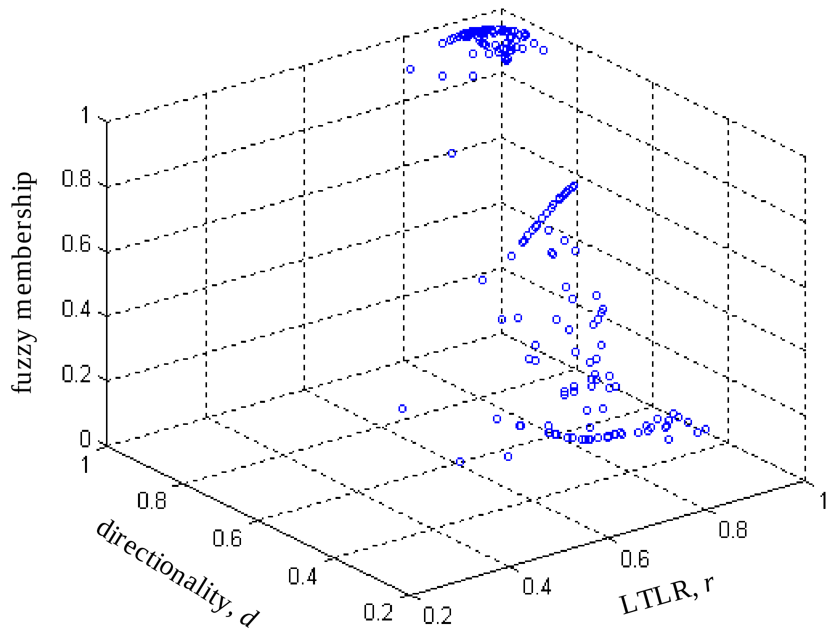
number of class k and the location of the initial centres are known prior to a process. Bearing in mind the end objective of the research, an automatic technique is highly desirable to adapt to the randomly changing nature of the data. Researches have been done [69, 70] and still on-going to find the appropriate number of clusters and analyzing the difference between the cluster solution and the data-set. This is useful when the appropriate number of centres are unknown, or the algorithm is stuck at a sub-optimal solution.

Recently, many wrapper methods have been proposed to improve clustering solutions. A wrapper method is one that transforms the input or output of the clustering algorithm, and/or uses the algorithm multiple times. One wrapper technique is simply running the clustering algorithm several times from different starting points and take the best solution [67]. Methods such as the one used by Likas [71] push this technique to its extreme, at the cost of computation. Another wrapper method to find quality clustering is by using the best initializations possible; this has been looked into in [72, 73].

Further work on classification will consider the optimal technique that compensates between classification accuracy and computational cost.



a) FKM1



b) FKM2

Figure 5.3: Fuzzy memberships visualized in 3-D plots.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

We have explained in quite a detail fashion work that have been done. The approaches taken have been explained theoretically and practically. The remaining sections concentrate on future work by first summarizing the completed tasks and then planning the way towards improving the current algorithm and adding more functionalities to the modules.

6.1.1 Summary of Completed Work

In summary, the following tasks have been accomplished:

- A survey of relevant literature has been performed.
- A brief description of the potential application scenarios.
- An explanation of a probable system implementation of a content-based craquelure retrieval system.
- Critical issues and obstacles have been been identified.
- Crack detection algorithm using morphological top-hat operator and grid-based variable thresholding has been implemented.
- A statistical crack pattern structuring methodology has been coded and implemented.
- A 3-layer hierarchy of features has been experimentally demonstrated.
- Several high-level features have been used as crack descriptors.

- An initial approach towards classification of craquelure patterns using common clustering techniques has been demonstrated.
- An implementation of the whole algorithm in C programming language.

6.2 Algorithm Evaluation

The current system has been programmed in both MATLAB and C programming language. To-date, the algorithm as a whole does not resemble a content-based retrieval system since it still lacks some of the important functionalities. In other words, it does not offer content query and retrieval. The completed work have been concentrating on the internal content extraction modules (*Processing Module*) instead of the user-side interface (*Application Module*). Whilst the *Application Module* need more attention, there are still some weaknesses related to the *Processing Module* that still need improvement. The following possibilities have been identified:

6.2.1 More Reliable Crack Detection Algorithm

The use of fully symmetrical structuring element as means of suppressing noise while retaining crack patterns is not considered as the best strategy. A combination of multi-orientation and multi-size set of structuring element might perform the task more accurately since they are better tuned to the structural appearances of crack patterns [30].

As explained in chapter 4 the main weakness of the crack detection stage is the assumption that every region contain cracks. This is not true for every case since in some images, cracks populate just a small portion of an area. In this case, false cracks will be “forced” to appear due to the very low threshold obtained from the automatic thresholding technique. Some of the alternatives to counter this drawback are as follows:

(a) *Conditional region labelling* where a minimum secondary threshold is set for decision on the presence of cracks in a particular region.

(b) The implementation of *multi-size region* thresholding to cater with varying size of “meaningful” regions and to determine the “regions of interest” (ROI).

6.2.2 Definition of “Content”

The algorithm still lacks in terms of how “content” is defined in the scope of craquelure. This is a very important concept since it effects the way metadata are stored, the way classification is performed and the manner in which data are queried as well as presented. The following are some alternatives to our current assumption:

(a) Each processed image is subdivided into square grids of a particular size where each grid is assumed to represent a "content", i.e a crack pattern as used in [9].

(b) Crack network boundaries are merged if they fulfill a particular condition and the crack network structure is modified.

6.2.3 Feature Organization and Storage

High-level features extracted from cracks should be stored as feature vectors and the way in which the feature vectors are stored should correspond to how "content" is perceived. This is an important step to ensure efficient database search and retrieval.

6.2.4 Rich Combination of Meaningful Features

Current features used to represent the crack patterns are not sufficient to distinguish all 5 classes. More features are needed besides those explained in chapter 4. Features that represent structural, spatial and frequency properties of the crack patterns can be considered.

6.2.5 Flexible Classification Techniques

The classification scheme utilized does not possess the level of flexibility needed to classify the patterns. The initial values of the centre points are critical values in the clustering process. If not selected properly, the algorithm can converge to bad local optima. Good convergence is related to sensitivity to initialization, and is a primary problem in data clustering. Another problem regarding the data set we are dealing with is the fact that in a single process or image, the number of relevant classes is unknown. For instance, one image is not guaranteed to contain all five classes. There might only be three, two or even one class representative in the whole image. A more adaptive classification technique must be employed to cater with these uncertainties. Hybrid or semi-supervised classification techniques [74] that combines supervised and unsupervised approaches can be a good option and will be considered in the future.

6.3 Future Work

The implementation of a good content-based retrieval system tuned for craquelure analysis is a highly challenging task. Realization of such objective depends on how well the computer vision modules can perform their jobs. To-date, a general framework of the system has already been explained in this dissertation. The remaining work involves improving the remaining algorithms and integrating CBR functionalities in the system. A time planner for the remainder of this work is as listed below:

- Oct - Dec '02: Improvements to the existing *processing module* by integrating more reliable processes with special attention to the feature extraction and classification sub-modules.
- Jan - Mar '03: Construction of the *application module* to integrate CBR functionalities in the system.
- Apr - June '03: Development of a graphical user interface for the purpose of performance evaluation most probably a web-based application.
- July - Sep '03: Algorithm evaluation with user feedback to validate the performance.
- Oct '03: Completion of full Ph.D thesis.

Bibliography

[1] M. Barni, F. Bartolini, A. De Rosa, "HVS Modelling for Quality Evaluation of Art Images," in *14th International Conference on Digital Signal Processing*, Santorini, Greece, July 2002.

[2] A.M. Bonacchi, V. Cappellini, M. Corsini, A. De Rosa, "Artshop: A Tool for Art Image Processing," in *14th International Conference on Digital Signal Processing*, Santorini, Greece, July 2002.

[3] B. Smolka, M. Szezapanski, "New Technique for the Restoration of Noisy Color Images," in *14th International Conference on Digital Signal Processing*, Santorini, Greece, July 2002.

[4] A. De Polo, F. Alinari, "Digital Picture Restoration and Enhancement for Quality Archiving," in *14th International Conference on Digital Signal Processing*, Santorini, Greece, July 2002.

[5] M. Barni, F. Bartolini, V. Cappellini, "Image Processing for Virtual Restoration of Artworks," *IEEE Multimedia*, vol. 7, no. 2, pp. 34–37, April-June 2000.

[6] J.M. Corridoni, A. Del Bimbo, S. De Magistris, E. Vicario, "A Visual Language for Color-Based Painting Retrieval," in *IEEE Symposium on Visual Languages*, 3-6 September 1996, pp. 68–75.

[7] Y. Isomoto, K. Yoshine, H. Yamasaki, N. Ishii, "Color, Shape and Impression Keyword as Attributes of Paintings for Information Retrieval," in *IEEE International Conference on Systems, Man, and Cybernatics*, October 1999, pp. 12–15.

[8] M. Westmacott, P. Lewis, K. Martinez, "Using Colour Pair Patches for Image Retrieval," in *First European Conference on Colour in Graphics, Imaging and Vision*, 2002, pp. 245–247.

[9] S. Chan, K. Martinez, P. Lewis, C. Lahanier, J. Stevenson, "Handling of Sub-Image Queries in Content-Based Retrieval of High Resolution Art Images," in *International Cultural Heritage Informatics Meeting 2*, 2002, pp. 245–247.

[10] F.S. Abas, K. Martinez, "Craquelure Analysis for Content-based Retrieval," in *14th International Conference on Digital Signal Processing*, Santorini, Greece, July 2002.

[11] Y. Rui, T.S. Huang, "Image Retrieval: Current Techniques, Promising Directions, and Open Issues," *Journal of Visual Communications and Image Representation*, vol. 10, no. 1, pp. 39–62, March 1999.

[12] S. Chang, A. Eleftheriadis, R. McClintock, "Next-Generation Content Representation, Creation and Searching for New-Media Applications in Education," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 884–904, May 1997.

[13] M. Flickner et al., "Query by Image and Video Content: The QBIC System," *IEEE Computer Magazine*, vol. 28, no. 9, pp. 23–32, September 1995.

[14] S. Mehrotra, Y. Rui, M. Ortega-Binderberger, T.S. Huang, "Supporting Content-based Queries over Images in MARS," in *IEEE International Conference on Multimedia Computing and Systems*, 3-6 June 1997, pp. 632–633.

[15] W.Y. Ma, B.S. Manjunath, "NeTra: A Toolbox for Navigating Large Image Databases," in *Proceedings of the International Conference on Image Processing.*

[16] S. Bucklow, "The Description of Craquelure Patterns," *Studies in Conservation*, vol. 42, 1997.

[17] N. Eastaugh, "Examination of Paintings by Infra-red and Other Techniques," in *IEE Colloquium on NDT in Archaelogy and Art*, 25 May 1995, p. 6/1.

[18] M. Müller, *Mise en Œuvre des Techniques de Traitement des Images pour I'Interprètation des Peintures*, Telecom Paris, de I'Ecole Nationale Supèrieure des Tèlècommunications, 1994.

[19] S. Bucklow, "A Stylometric Analysis of Craquelure," *Computers and Humanities*, vol. 31, 1998.

[20] I. Giakoumis, I. Pitas, "Digital Restoration of Painting Cracks," in *ISCAS '98, Proceedings of the IEEE International Symposium on Circuits and Signals*, 31 May-3 June 1998, pp. 269–272.

[21] L. Vuurpijl, L. Schomaker, “Two-stage Character Classification: A Combined Approach of Clustering and Support Vector Classifiers,” in *7th International Workshop on Frontiers in Handwriting Recognition*, Amsterdam, Netherlands, September 2000.

[22] X. Li, D.Y. Yeung, “On-Line Handwritten Alphanumeric Character Recognition Using Feature Sequences,” *Pattern Recogntion*, vol. 30, no. 1, pp. 31–44, 1995.

[23] Ø.D. Trier, A.K. Jain, T. Taxt, “Feature Extraction Methods for Character Recognition - A Survey,” *Pattern Recognition*, vol. 29, no. 4, pp. 641–662, 1996.

[24] V. Wu, R. Manmatha, E.M. Riseman, “Finding Text in Images,” in *2nd International Conference on Digital Libraries*, July 1997, pp. 3–12.

[25] A.J. Varley, *Statistical Image Analysis Methods for Line Detection*, PhD thesis, Cambridge University, 1999.

[26] A.M. López, F. Lumbreras, “Evaluation of Methods for Ridge and Valley Detection,” *IEEE Transactions on Pattern and Machine Intelligence*, vol. 21, no. 4, pp. 327–335, April 1999.

[27] D. Elberly, R. Gardner, B. Morse, S. Pizer, C. Scharlach, “Ridges for Image Analysis,” *Journal of Mathematical Imaging and Vision*, vol. 4, no. 4, pp. 353–373, December 1994.

[28] J. Gaugh, S. Pizer, “Multiresolution Analysis of Ridges and Valleys in Grey-Scale Images,” *IEEE Transactions of Pattern Analysis and Machine Intelligence*, vol. 15, no. 6, pp. 635–646, June 1993.

[29] R. Haralick, “Ridges and Valleys on Digital Images,” *Computer Vision, Graphics and Image Processing*, vol. 22, no. 10, pp. 28–38, April 1983.

[30] F. Zana, J.C. Klein, “Segmentation of Vessel-Like Patterns Using Mathematical Morphology and Curvature Evaluation,” *IEEE Transactions on Image Processing*, vol. 10, no. 7, pp. 1010–1019, July 2001.

[31] F. Zana, J.C Klein, “Robust Segmentation of Vessels From Retinal Angiography,” in *International Conference on Digital Signal Processing*, Santorini, Greece, July 1997.

[32] P. van del Elsen, J. Maintz, E-J. Pol, M. Viergever, “Automatic Registration of CT and MR Brain Images Using Correlation of Geometrical Features,” *IEEE Transactions on Medical Imaging*, vol. 14, no. 2, pp. 384–396, June 1995.

[33] A.K. Jain, S. Prabhakar, L. Hong, "A Multichannel Approach to Fingerprint Classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 348–359, April 1999.

[34] N. Merlet, J. Zerubia, "A Curvature-Dependent Energy Function for Detecting Lines in Satellite Images," in *SCIA*, Tromso, Norway, 1993.

[35] F. Tupin, H. Maitre, J. Mangin, J. Nicholas, E. Pechersky, "Detection of Linear Features in SAR Images: Application to Road Network Extraction," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, no. 2, pp. 434–453, 1998.

[36] G.J. Vanderbrug, "Semilinear Line Detectors," *Computer Graphics and Image Processing*, vol. 4, 1975.

[37] G.J Vanderbrug, "Line Detection in Satellite Imagery," *IEEE Transactions on Geoscience Electronics*, 1976.

[38] G. Matheron, "Elements pour une Theorie del Milieux Poreux," in *Masson*, Paris, 1967.

[39] J. Serra, "Image Analysis and Mathematical Morphology," in *Academic Press*, London, 1982.

[40] S.R. Sternberg, "Grayscale Morphology," *Computer Vision, Graphics and Image Processing*, vol. 35, 1985.

[41] R.M. Haralick, S.R. Sternberg, X. Zhuang, "Grayscale Morphology," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1986, pp. 523–550.

[42] R.M. Haralick, S.R. Sternberg, X. Zhuang, "Image Analysis Using Mathematical Morphology," *IEEE Transactions on Pattern Analysis and Machine Vision*, 1987.

[43] R.M. Haralick, L.G. Shapiro, *Computer and Robot Vision*, Addison Wesley, Washington, 1992.

[44] G.H. Ritter, J.N. Wilson, *Handbook of Computer Vision Algorithms in Image Algebra*, CRC Press, Florida, 1996.

[45] F. Meyer, "Iterative Image Transformations for an Automatic Screening of Cervical Smears," *J.Histoch.Cytochem*, vol. 27, 1979.

[46] N. Otsu, "A Threshold Selection Method From Gray-level Histogram," *IEEE Transactions on Systems, Man, Cybernatics*, vol. 8, 1978.

[47] S.Y. Chen, W.C Lin, C.T. Chen, "Split-and-Merge Image Segmentation Based on Localized Feature Analysis and Statistical Tests," *Graphical Models and Image Processing*, vol. 53, no. 5, pp. 457–475, September 1991.

[48] J. Kittler, J. Illingworth, J. Foglein, "Threshold Selection Based on a Simple Image Statistics," *Computer Vision, Graphics and Image Processing*, vol. 30, 1985.

[49] A. Rosenfeld, A. Kak, *Digital Picture Processing*, Academic Press, New York, 1982.

[50] M. Sonka, V. Hlavac, R. Boyle, *Image Processing, Analysis and Machine Vision*, Chapman and Hall Computing, London,UK, 1993.

[51] S. Bucklow, "Consensus in the Classification of Craquelure," *Hamilton Kerr Institute Bulletin, Fitzwilliam Museum, Cambridge*, vol. 3, 2001.

[52] A. Varley, P. Rayner, "Bézier Modelling of Cracks," in *International Conference in Image Analysis and Processing*, 1997, pp. 551–558.

[53] P.J. Green, "Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination," *Biometrika*, vol. 82, no. 4, pp. 711–732, 1996.

[54] A.J. Varley, P.J.W. Rayner, "Analysis of Crack Patterns Using MCMC Sampling," in *International Conference on Mathematics in Signal Processing*, June 1996.

[55] H. Freeman, "Boundary Encoding and Processing," *Picture Processing and Psychopictories, Academic Press*, 1970.

[56] H. Freeman, "Computer Processing of Line Drawing Images," *Computer Surveys*, vol. 6, no. 1, pp. 57–98, 1974.

[57] F. Arebola, P. Camacho, A. Bandera, F. Sandoval, "Corner Detection and Curve Representation by Circular Histograms of Contour Chain Code," *IEEE Electronics Letter*, vol. 35, no. 13, pp. 1065–1067, June 1999.

[58] F. Arrebola, A. Bandera, P. Camacho, F. Sandoval, "Corner Detection by Local Histograms of Contour Chain Code," *IEEE Electronics Letter*, vol. 33, no. 21, pp. 1769–1771, October 1997.

[59] W.K. Pratt, *Digital Image Processing*, Wiley Interscience, 2nd edition, 1991.

[60] A.M. Vossepoel, A.W.M. Smeulders, "Vector Code Probabilty and Metrication Error in the Representation of Straight Lines of Finite Length," *Computer Graphics and Image Processing*, vol. 20, 1982.

[61] S. Bucklow, *Formal Connoisseurship and the Characterisation of Craquelure*, PhD thesis, 1996.

[62] R.J. Schalkoff, *Pattern Recognition: Statistical, Structural and Neural Network*, John Wiley & Sons. Inc., New York, 1991.

[63] A.P. Dempster, N.M. Laird, D.B. Rubin, "Maximum Likelihood From Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society - B*, vol. 39, 1977.

[64] A.K. Jain, R.C. Dubes, *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs, N.J, 1988.

[65] A.K. Jain, R.P.W. Duin, J. Mao, "Statistical Pattern Recognition: A Review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, January 2000.

[66] J. MacQueen, "Some methods for Classification and Analysis of Multivariate Observations," in *Proceedings of the 5th Berkeley Symposium - 1*, 1967, pp. 281–297.

[67] A.K. Jain, M.N. Murty, P.J. Flynn, "Data Clustering: A Review," *ACM Computing Surveys*, vol. 31, 1999.

[68] J.C. Bezdek, "Pattern Recognition With Fuzzy Objective Function Algorithms," *New York: Plenum Press*, 1981.

[69] D. Pelleg, A. Moore, "X-means: Extending K-means With Efficient Estimation of the Number of Clusters," in *17th International Conference on Machine Learning*, San Francisco, CA, 2000, pp. 727–734.

[70] P. Sand, A. Moore, "Repairing Faulty Mixture Models Using Density Estimation," in *18th International Conference on Machine Learning*, San Francisco, CA, 2001.

[71] A. Likas, N. Vlassis, J. Verbeek, *The Global K-means Clustering Algorithm*, Technical Report, Computer Science Institute, University of Amsterdam, 2001.

[72] J. Pẽna, J. Lozano, P. Larrañaga, "An Empirical Comparison of Four Initialization Methods for the K-means Algorithm," *Pattern Recognition Letters*, vol. 20, 1999.

[73] M. Meǐla, D. Heckermann, "An Experimental Comparison of Model-based Clustering Methods," *Machine Learning*, vol. 42, 2001.

[74] G. Fung, O.L. Mangasarian, "Semi-supervised Support Vector Machines for Unlabeled Data Classification," *Optimization Methods and Software*, vol. 15, no. 1, April 2001.