Classification of painting cracks for content-based analysis

F.S. Abas and K. Martinez

Intelligence, Agents, Multimedia Group, Department of Electronics and Computer Science,
University of Southampton, United Kingdom, S017 1BJ.

ABSTRACT

In this paper we present steps taken to implement a content-based analysis of crack patterns in paintings. Cracks are first detected using a morphological *top-hat* operator and grid-based automatic thresholding. From a 1-pixel wide representation of crack patterns, we generate a statistical structure of global and local features from a *chain-code* based representation. A well structured model of the crack patterns allows post-processing to be performed such as pruning and high-level feature extraction. High-level features are extracted from the structured model utilising information mainly based on orientation and length of line segments. Our strategy for classifying the crack patterns makes use of an unsupervised approach which incorporates fuzzy clustering of the patterns. We present results using the fuzzy k-means technique.

Keywords: Feature extraction, morphological filters, crack detection, clustering

1. INTRODUCTION

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 volume 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] image retrieval[4] and as an aid for conservation.[5] 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).[3]

The aim of the Artiste project was to provide access across museum collections using metadata as well as content-based retrieval of image data. One of the image-based requirements, which came from the Uffizi Gallery in Florence, is to automatically classify the *craquelure* (cracks) in paintings to aid in damage assessment. Craquelure in paintings can also be used for other research.[6] 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.

Figure 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 (HVS). As in the figure, the cracks are represented by dark pixels while the background is represented by the brighter ones. As far as the 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 a large collection of crack images with different levels of illumination, contrast, noise and intensity. Considering the large size of collections, it is difficult to monitor every single painting to spot cracks. The digitization of painting collections includes X-radiographs of paintings, which are better for a content-based approach in analyzing the crack patterns. This is because they tend to show cracks very well as the details in the paint layer are suppressed. A bigger challenge

Send correspondence to Fazly S. Abas.
Fazly S. Abas: E-mail: fsa00r@ecs.soton.ac.uk, Telephone: +44 (0) 23 8059 5415.
Kirk Martinez: E-mail: km@ecs.soton.ac.uk, Telephone: +44 (0) 23 8059 4491.

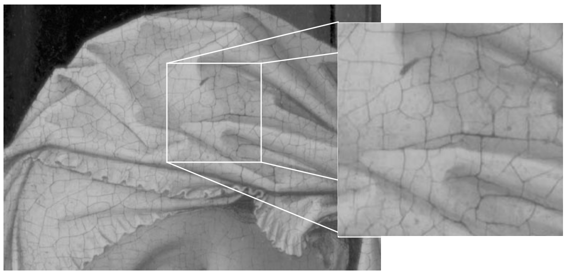


Figure 1. Example of a painting with cracks.

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[6] where cracks are classified into 4 different categories namely, French, Flemish, Dutch and Italian. The relevant classes which we introduce in our research take into account the probable effect of 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, uni-directional and random cracks. The scope of this paper generally falls into 3 parts - crack detection, feature extraction and data classification.

2. APPLICATION SCENARIO

The capabilities of image analysis and processing can aid conservators if the process can provide them with the functionalities that manual processes fail to do. 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.

2.1. Scenario 1 : Query for similar pattern

The main requirement of a content-based analysis of cracks will be to query for a particular region or image for similar type of crack patterns. This scenario is of particular interest to users who are interested in investigating the relationship 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.

2.2. 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).

These content-based functionalities relies on a good mixture of image processing, pattern recognition and data handling approaches. Although there have been work done on crack detection and classification, the intentions of our work extends the traditional image-based approach to a content-based nature.

3. CRACK DETECTION

Crack-like pattern detection is also known in some literature as ridge-valley structure extraction[7,8] have been a matter of high concern among researchers mostly for its useful contribution to a 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, images of fingerprints and multi-spectral satellite photos of rivers or roads.

3.1. The top-hat transformation

Mathematical morphology is a part of digital image processing that concerns image filtering and geometric analysis by structuring elements. Originally, the theory and application of mathematical morphology was developed for binary images.[9] Afterwards, the theory was extended to gray-scale images. Although binary morphological operations serve as useful analytical tools for image analysis and classification, they play only a limited role in the processing and analysis of gray-level images. 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.

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 visualized as working with two images namely the image being processed A 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[10]:

$$\begin{aligned} \textit{erosion} &: \quad \epsilon^e_S(A)(M_o) = MIN_{M \in M_o + e \cdot S(M_o)}(A(M)), & (1)\\ \textit{dilation} &: \quad \delta^e_S(A)(M_o) = MAX_{M \in M_o + e \cdot S(M_o)}(A(M)). & (2) \end{aligned}$$

Conceptually similar to the binary case, *dilation* followed by *erosion* is the *closing* transformation, while *erosion* followed by *dilation* is the *opening* transformation and they are defined as[10]:

$$\begin{aligned} \textit{opening} &: \quad \gamma^e_S(A) = \delta^e_S(\epsilon^e_S(A)), & (3)\\ \textit{closing} &: \quad \varphi^e_S(A) = \epsilon^e_S(\delta^e_S(A)). & (4) \end{aligned}$$

Cracks can be detected with the implementation of a very useful morphological filter, known as *top-hat* transformation developed by Meyer.[11] 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 5 and 6 respectively.[10]

$$OTH_S^e = A - \gamma_S^e(A) \quad (5)$$
$$CTH_S^e = \varphi_S^e(A) - A \quad (6)$$

The top-hat operator can be tuned for detection of specific features by modifying two important parameters[12]:

(a) 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.

(b) 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.

3.2. Grid-based automatic thresholding

A single threshold value over the whole image is not a good strategy to segment the cracks, especially in the case of large crack images. It works when the background intensity level is not constant. As for the images we are working with, the main problem is due to the uneven enhancement 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.[13] 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 dimensions. The image is first zero-padded if its dimension is not a multiple of the grid size. The threshold value is then established for each region separately using the Otsu algorithm.[14, 15] The Otsu technique is an automatic threshold determination technique which is based on discriminant analysis. It minimizes the between-class variance ratio. Thresholding is performed locally on every sub-image using different values. In another term, this technique can be described as a grid-based automatic thresholding. Instead of using a global threshold, 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 produce spurious features to appear since the threshold will emerge to be extremely low in these regions.

The prime objective of tessellating the images into square grids is to deal with illumination inconsistency. Even after enhancement, the desired output of the CTH technique does not totally solve this problem. Assuming consistent illumination within the square regions, the grid-based automatic thresholding ensures that the threshold values are automatically determined for each square area.

4. HIERARCHICAL CRACK NETWORK STRUCTURING AND REPRESENTATION

A *crack following* algorithm is applied on a thinned (down to 1-pixel wide) crack detected image. 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*[16] has been used for various image processing algorithms including finding features of curves/lines. 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. 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 a means of storing and reconstructing a crack structure. It is utilized solely for 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.

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 detects a discontinuity and along the "way" it marks all important points. Statistical information are stored temporarily in a structured manner where all 8-connected pixels are regarded as a single entity which we name a *crack network*. The structure of a *crack network* is as shown in Figure 2 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 2, 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.

4.1. 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 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.[16] The formula for the perimeter is

$$P = n_e + \sqrt{2}n_o \tag{7}$$

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 7 and its significance in our approach will be explained in Section 5.

We accumulate the chain-codes in a histogram with 8 bins where each bin represents a *chain-code*. We call this histogram the 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 7. 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.

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

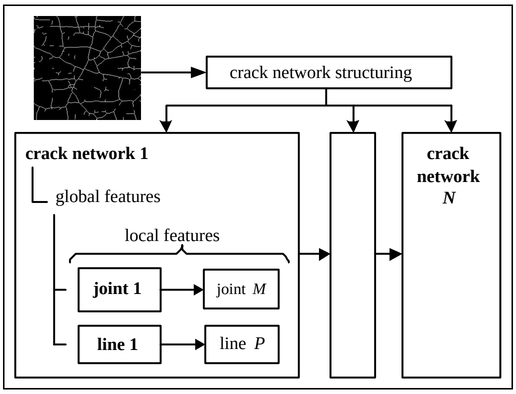


Figure 2. Statistically structured *crack network*.

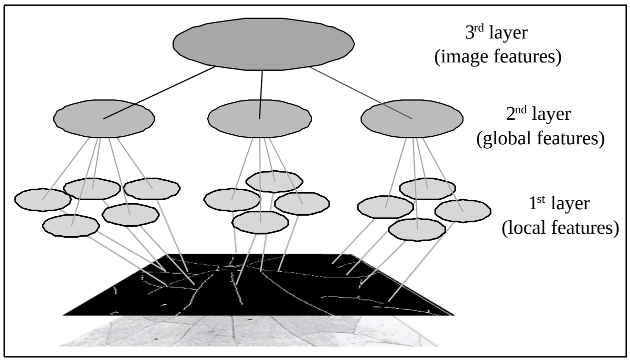


Figure 3. Hierarchy of features.

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 networks* 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.

5. HIGH-LEVEL FEATURE EXTRACTION

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 a 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 call image features. This is graphically illustrated in Figure 3.

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 :

- ℓ_a : length of a line segment
- s_a : significance measure of a line segment
- r_a : straight line to actual line segment 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 line segment 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

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

5.1. 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 of 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

$$s_{a_{(i)}} \quad = \quad \frac{\ell_{a_{(i)}}}{\ell_b} \qquad \forall \quad i = 1...n_b \tag{8}$$

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

$$s_{b_{(i)}} \quad = \quad \frac{\ell_{b_{(i)}}}{\ell_c} \qquad \forall \quad i = 1...n_c \tag{9}$$

where $S_c = \{s_{b_{(1)}}, ..., s_{b_{(n_c)}}\}$ is a set of significance value representing each *crack network* in an image. The significance values are used frequently in combining features at a low layer to obtain a representation of the same feature at a higher layer.

5.2. 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 length percentage distribution is then computed for a single *crack network* denoted as μ_b and σ_b respectively. In short we name these 2 features MLP and SLP.

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].

5.3. Straight line to actual line segment length ratio

Another feature which we find useful is the straight line to actual line segment 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 7. 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 10 and 11.

$$r_b = \sum_{i=1}^{n_b} s_{b_{(i)}} r_a \tag{10}$$

$$r_c = \sum_{i=1}^{n_c} s_{c_{(i)}} r_b \tag{11}$$

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.

5.4. Directionality

Through rough observations, circular cracks and rectangular cracks differ significantly in terms of their orientation spread which is visible from their orientation histograms.[5] 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* indicates the straightness measure of a line segment.

Local directionality is computed by first generating *directionality* for every line segment in *crack network.* Four histogram models, $m_1 = 0.125\{1,1,1,1,1,1,1,1\}$, $m_2 = 0.25\{1,1,1,1,0,0,0,0\}$, $m_3 = 0.5\{1,1,0,0,0,0,0,0\}$ and $m_4 = \{1,0,0,0,0,0,0,0\}$ are first generated. They represent ideal histograms as far as orientation spread is concerned where they roughly indicate circular, semi-circular, bi-directional and uni-directional spread respectively. Orientation histogram for every line segment is compared with these for ideal histograms and sum of errors are computed. Let e_1, e_2, e_3 and e_4 represent sum of errors corresponding to m_1, m_2, m_3 and m_4, similarity histogram, $W_a = \{(1-e_1), (1-e_2), (1-e_3), (1-e_4)\}$ is computed. Score is then determined for each line segment based on the similarity histogram.[5] From the *local directionality* values, *global directionality* is established for a single *crack network* using significance measure s_a as weighting.[5]

6. CLASSIFICATION

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.

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[17] work, classification is made 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.

To-date, we define a crack pattern as a single *crack network.* Every *crack network* is bounded by a rectangular box and each of these network is treated as a *content.* Determination of alternative definitions for crack pattern is still an on-going work.

6.1. 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.[18] 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.[19]

6.1.1. Fuzzy k-means clustering

The fuzzy k-means algorithm (FKM) (which is also called the fuzzy c-means)[20] is an adaptation of the k-means (KM)[21] algorithm that uses *soft* membership function. Unlike KM which assigns each data point to its closest center, the FKM algorithm allows a data point to belong partly to all centers.

The objective function for FKM is:

$$FKM(X,C) = \sum_{i=1}^{n}\sum_{j=1}^{n} u_{ij}^{r} \parallel x_i - c_j \parallel^2 \tag{12}$$

where the *fuzzy membership matrix* u_{ij} denotes the proportion of data point x_i that is assigned to center 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[20] 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:

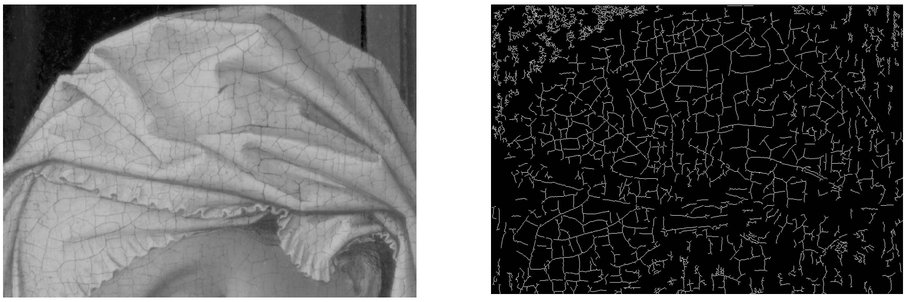
$$m_{FKM}(c_j|x_i) = \frac{\parallel x_i - c_j \parallel^{\frac{-2}{r-1}}}{\sum_{j=1}^{k} \parallel x_i - c_j \parallel^{\frac{-2}{r-1}}}. \tag{13}$$

As r tends to approach 1 from above, *fuzziness* decreases and the algorithm behaves more like standard KM. The centers share the data points less in this condition. As for the FKM 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. *Fuzziness* serves the application well since we would like to expect perceptions towards crack pattern labelling to be described as a confidence measure instead of being an absolutely certain decision. A human observer for instance, might have divided perceptions about a particular pattern and so does a fuzzy classifier system. We intend to incorporate *fuzziness* in our classification approach and FKM is seen as a reasonable approach.

We conduct 2 experiments using 2-class and 3-class problems to observe the effectiveness of the features and potential improvements that can be made to the current technique.

6.2. Results and discussion

Figure 4(a) shows an image of a painting while Figure 4(b) highlights the detected cracks. Cracks are enhanced using the CTH technique with a disk-shaped structuring element of size 5x5 before they are segmented using the grid-based automatic thresholding method. The thinned cracks are then "followed" and high-level features are extracted. We perform FKM clustering in 3-dimensional space using *directionality*, LTLR and SMP as our features. Initial center points are chosen based on some preliminary processes while the *fuzzy membership matrix* u_{ij} is initialized randomly.



(a) (b)

Figure 4. Cracks enhanced using the CTH technique with a disk-shaped structuring element of size 5x5, segmented using the grid-based automatic thresholding method and later thinned to 1-pixel wide. a) Painting with cracks, and b) detected cracks.

Crack networks as well as rectangular bounding boxes are labelled and reconstructed. Rectangular bounding boxes are used to indicate the extent to which a *crack network* occupies an image. The results are shown based on the number of pre-specified classes. Figure 5 and Figure 6 show results for a 2-class and a 3-class problems respectively.

From the results, several observations can be made. The number of pre-specified clusters/classes is a very crucial parameter. There is no way of knowing how many patterns there are in a single image. As can be seen from the results, there are no distinct differences between crack patterns of the different classes. A human observer might classify all the *crack network* into the same class whereas the algorithm will be forced to classify according to the manually pre-specified number of classes no matter how large the number is.

Another issue which we are looking seriously into is the definition of *object of interest* or *content of interest*. The experiments are done on the assumption that a single *crack network* represents a *content of interest*. This is not necessarily the case since crack patterns should be thought of as combinations of crack networks rather than a single *crack network*. A combined representation in the form of merged networks is believed to improve the algorithm since feature variances will also increase.

7. CONCLUSIONS AND FUTURE WORK

Analysis of painting cracks has been a subject of interest for decades particularly for fine artwork conservators. It is believed that the existence of cracks on a painting does in a way relate to the structural support framework and physical impacts. In most cases, analysis are done manually by experts. A truly useful analysis is the classification of painting cracks into distinct patterns which can be used as a clue as to what really cause the cracks to form.

A novel idea of applying content-based functionalities to support the analysis of painting cracks has been briefly explained. We are still investigating many aspects and issues related to the implementation of a content-based approach in the scope of crack analysis. A simple yet effective crack detection strategy has been implemented as a preliminary stage to segment the suspected cracks from the background. Several improvements can be made to this stage such as using multi-orientation structuring elements but to the expense of processing speed.

The structured representation of features allows flexible and useful operations to be performed as witnessed from the crack pruning process and the multi-layer feature representation. It is believed that hierarchical representation allows cracks to be analyzed both locally and globally, a desired criteria for a content-based analysis. Intermediate representation of crack patterns or in other words, combination or merging of crack networks are also desirable since patterns vary according to scope of view. We plan on testing several "crack merging" strategies to merge features belonging to separate crack networks based on proximity rules and pattern similarities.

Uncertainties and lack of prior knowledge are the main problems in the classification stage. These originated from the lack of labelled data and random presence of cracks in a certain image. A straightforward clustering algorithm alone can not predict the number of center points one can expect from a clustering process. Number of classes or clusters in a process should be automatically determined. Our next challenge is to resolve this problematic situation.

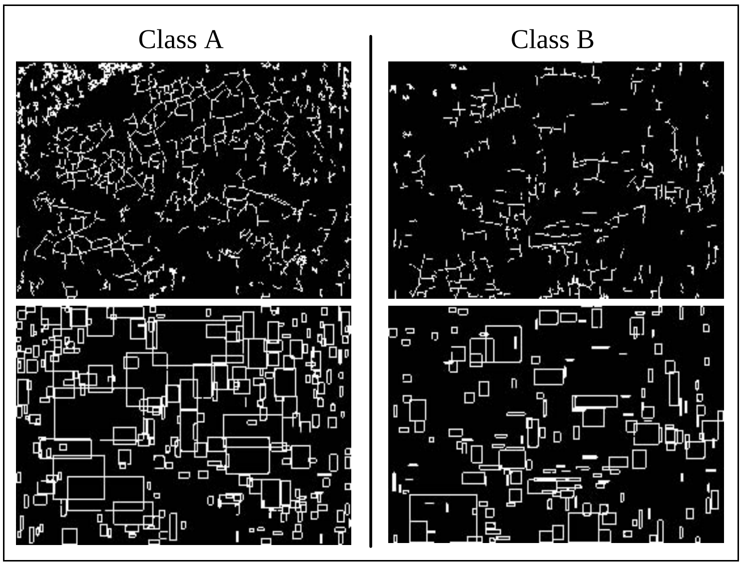


Figure 5. Results of a 2-class problem. Images of cracks (upper half) and their respective rectangular bounding box (lower half) are shown where each column represents a class.

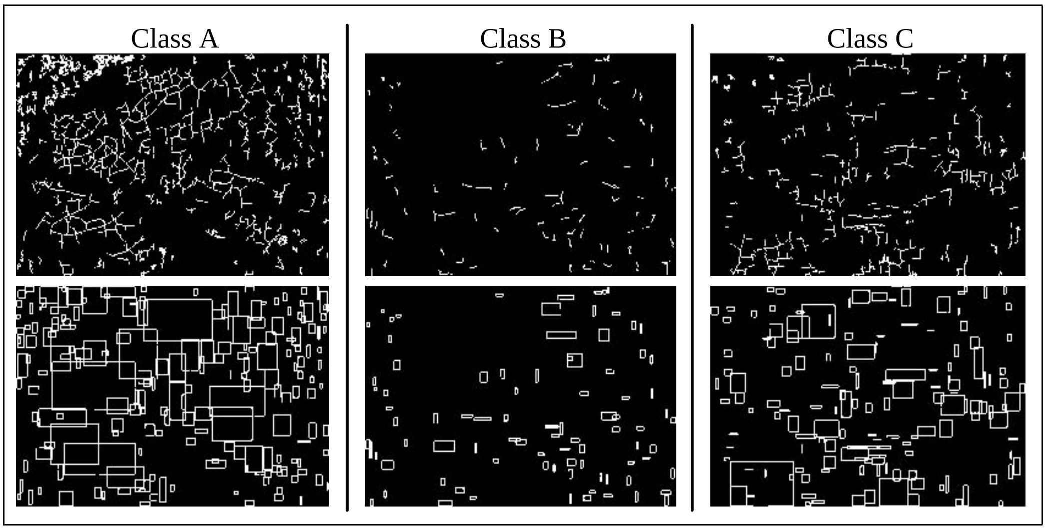


Figure 6. Results of a 3-class problem. Images of cracks (upper half) and their respective rectangular bounding box (lower half) are shown where each column represents a class.

ACKNOWLEDGMENTS

We would like to thank Centre of Research and Restoration of the museums of France (C2RMF), Victoria and Albert Museum, National Gallery (London), Uffizi Gallery (Florence), NCR and Giunti Interactive Labs as well as Dr. Spike Bucklow (Hamilton Kerr Institute, Cambridge) and Dr. Paul H. Lewis (University of Southampton) for their support.

REFERENCES

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. M. Barni, F. Bartolini, V. Cappellini, "Image Processing for Virtual Restoration of Artworks," *IEEE Multimedia* **7**, pp. 34–37, April-June 2000.
4. 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*, pp. 68–75, 3-6 September 1996.
5. F.S. Abas, K. Martinez, "Craquelure Analysis for Content-based Retrieval," in *14th International Conference on Digital Signal Processing*, (Santorini, Greece), July 2002.
6. S. Bucklow, "The Description of Craquelure Patterns," *Studies in Conservation* **42**, 1997.
7. A.M. López, F. Lumbreras, "Evaluation of Methods for Ridge and Valley Detection," *IEEE Transactions on Pattern and Machine Intelligence* **21**, pp. 327–335, April 1999.
8. D. Elberly, R. Gardner, B. Morse, S. Pizer, C. Scharlach, "Ridges for Image Analysis," *Journal of Mathematical Imaging and Vision* **4**, pp. 353–373, December 1994.
9. G. Matheron, "Elements pour une Theorie del Milieux Poreux," in *Masson*, (Paris), 1967.
10. F. Zana, J.C. Klein, "Segmentation of Vessel-Like Patterns Using Mathematical Morphology and Curvature Evaluation," *IEEE Transactions on Image Processing* **10**, pp. 1010–1019, July 2001.
11. F. Meyer, "Iterative Image Transformations for an Automatic Screening of Cervical Smears," *J.Histoch.Cytochem* **27**, 1979.
12. I. Giakoumis, I. Pitas, "Digital Restoration of Painting Cracks," in *ISCAS '98, Proceedings of the IEEE International Symposium on Circuits and Signals*, pp. 269–272, 31 May-3 June 1998.
13. A. Rosenfeld, A. Kak, *Digital Picture Processing*, Academic Press, New York, 1982.
14. N. Otsu, "A Threshold Selection Method From Gray-level Histogram," *IEEE Transactions on Systems, Man, Cybernatics* **8**, 1978.
15. 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* **53**, pp. 457–475, September 1991.
16. H. Freeman, "Boundary Encoding and Processing," *Picture Processing and Psychopictories, Academic Press* , 1970.
17. A.J. Varley, *Statistical Image Analysis Methods for Line Detection*, PhD thesis, Cambridge University, 1999.
18. A.K. Jain, R.C. Dubes, *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs, N.J, 1988.
19. A.K. Jain, R.P.W. Duin, J. Mao, "Statistical Pattern Recognition: A Review," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**, pp. 4–37, January 2000.
20. J.C. Bezdek, "Pattern Recognition With Fuzzy Objective Function Algorithms," *New York: Plenum Press* , 1981.
21. J. MacQueen, "Some methods for Classification and Analysis of Multivariate Observations," in *Proceedings of the 5th Berkeley Symposium - 1*, pp. 281–297, 1967.