



Crack detection and inpainting for virtual restoration of paintings: The case of the Ghent Altarpiece[☆]

B. Cornelis^{a,c,*}, T. Ružič^{b,c}, E. Gezels^d, A. Dooms^{a,c}, A. Pižurica^{b,c}, L. Platiša^{b,c}, J. Cornelis^a,
M. Martens^d, M. De Mey^e, I. Daubechies^f

^a Department of Electronics and Informatics (ETRO), Vrije Universiteit Brussel (VUB), Pleinlaan 2, B-1050 Brussels, Belgium

^b Department for Telecommunications and Information Processing (TELIN), Ghent University (UGent), Sint-Pietersnieuwstraat 4, B-9000 Ghent, Belgium

^c Interdisciplinary Institute for Broadband Technology (IBBT), Gaston Crommenlaan 8 (box 102), B-9050 Ghent, Belgium

^d Department of Art, Music and Theatre Sciences, Ghent University (UGent), Blandijnberg 2, B-9000 Ghent, Belgium

^e The Flemish Academic Centre for Science and the Arts (VLAC), Hertogsstraat 1, B-1000 Brussels, Belgium

^f Mathematics Department, Duke University, Box 90320, Durham, NC, USA

ARTICLE INFO

Article history:

Received 23 August 2011

Received in revised form

19 March 2012

Accepted 10 July 2012

Available online 25 July 2012

Keywords:

Crack detection

Patch-based inpainting

Digital restoration

Ghent Altarpiece

Adoration of the Mystic Lamb

ABSTRACT

Digital image processing is proving to be of great help in the analysis and documentation of our vast cultural heritage. In this paper, we present a new method for the virtual restoration of digitized paintings with special attention for the Ghent Altarpiece (1432), a large polyptych panel painting of which very few digital reproductions exist. We achieve our objective by detecting and digitally removing cracks. The detection of cracks is particularly difficult because of the varying content features in different parts of the polyptych. Three new detection methods are proposed and combined in order to detect cracks of different sizes as well as varying brightness. Semi-supervised clustering based post-processing is used to remove objects falsely labelled as cracks. For the subsequent inpainting stage, a patch-based technique is applied to handle the noisy nature of the images and to increase the performance for crack removal. We demonstrate the usefulness of our method by means of a case study where the goal is to improve readability of the depiction of text in a book, present in one of the panels, in order to assist paleographers in its deciphering.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

In this work, we focus on crack detection and inpainting in the *Ghent Altarpiece* and extend upon the work introduced in [1]. The polyptych, dated by inscription 1432, was painted by Jan and Hubert van Eyck and is considered as one of their most important masterpieces

known all over the world. It is still located in the Saint Bavo Cathedral in Ghent, its original destination.

Cracking is one of the most common deteriorations found in old masters paintings; it is a sign of the inevitable aging of materials and constitutes a record of their degradation. Generally speaking, a crack (or *craquelure*) appears in paint layers when pressures develop within or on it through the action of various factors and cause the material to break [2]. The state of preservation of a painting is mainly influenced by climate changes such as variations in temperature, relative humidity or pressurization (e.g. during transport via air) [3]. As for most 15th century Flemish paintings on Baltic oak, fluctuations in relative humidity, causing the wooden support to shrink or expand, are the main causes for age crack

[☆] This research was supported by the Fund for Scientific Research Flanders (FWO) (PhD fellowship of Bruno Cornelis and Projects G.0206.08 and G021311N).

* Corresponding author at: Department of Electronics and Informatics (ETRO), Vrije Universiteit Brussel (VUB), Pleinlaan 2, B-1050 Brussels, Belgium. Tel.: +32 26291671.

E-mail address: bcorneli@etro.vub.ac.be (B. Cornelis).

formation. Age related or mechanical cracks can affect the entire paint layer structure, including both the preparation and the paint layers on it. Age cracks are to be distinguished from premature cracking [2]. The latter are more dull-edged than those formed when aging and originate in only one of the layers of paint. They generally reveal a defective technical execution at the painting stage, such as not leaving enough time for a layer to dry, or applying a layer that dries faster than the underlying one. A third type is varnish cracks, formed only in the varnish layer, when it becomes brittle through oxidation.

Cracks form an undesired pattern that can be either rectangular, circular, web-shaped, unidirectional, tree-shaped or even completely random. The way in which they manifest themselves and spread partly depends upon the choice of materials and methods used by the artist. This makes cracks useful for judging authenticity, as is proposed in [4]. Cracks can also assist conservators by providing clues to the causes of degradation of the paint surface. This can be used for degradation monitoring of the paint layer or a more in-depth study on which factors contribute to the formation of cracks, so that steps can be taken to reduce them [5]. The potential of using cracks as a *non-invasive* means of identifying the structural components of paintings is highlighted in [6]. The correlation between the network of cracks on the surface and the structure of the panel below is also investigated in [2] by using multi-layered X-ray radiography. An area which is thought to be of great interest to art conservation is content-based analysis where cracks are used for content-based retrieval of information from image databases [3].

Digital image processing can automatically detect crack-like patterns. In the literature, this process is often referred to as *ridge-valley structure extraction* [7]. Many types of images contain similar elongated structures (e.g. medical images of veins and vessels [8], images of fingerprints and satellite imagery of rivers and roads) and the common goal is to extract or detect these crack-like patterns in order to separate them from the rest of the image. An overview of different crack detection techniques can be found in [5]. These include different types of thresholding, the use of multi-oriented filters (such as Gabor filters) and a plethora of morphological transforms.

In the context of the virtual restoration of digitized paintings, crack detection is often treated side-by-side with crack removal. Inpainting, which is the image restoration task of filling in missing parts of the image, is used for this purpose. The literature contains a vast number of general inpainting methods which can be roughly separated into two groups: pixel-based and patch-based methods. Pixel-based methods aim at replacing one missing pixel at the time [9,10] by specially focusing on structure propagation, i.e. propagation of lines and object contours, from the boundaries of the missing region to its center. Patch-based methods [11–14], on the other hand, fill in the missing region patch-by-patch ensuring in that way better texture propagation. They also consider structural propagation by introducing a certain priority in which the patches are visited.

A virtual restoration system to remove cracks on digital images of paintings was developed in [15]. Their method is based on a semi-automatic crack detection procedure, where users need to specify a location 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, they can be removed by interpolation. In [16–18] crack patterns are detected by thresholding the output of a morphological top-hat transform. Cracks are subsequently separated from brushstrokes (i) by using the hue and saturation information in the HSV or HSI colour space and feeding it to a neural network or (ii) by letting a user manually select seed points. Finally, the cracks are inpainted using order statistics filtering for interpolation [17], controlled anisotropic diffusion [16] or patch-based texture synthesis [18].

The cracks considered here are particular in a number of ways. Their width ranges from very narrow and barely visible to larger areas of missing paint. Furthermore, depending on the painting's content, they appear as dark thin lines on a bright background or vice versa, bright thin lines on a darker background. In Section 2 we introduce three different detection schemes, each able to detect bright and dark cracks and each having its own strength. Since this masterpiece contains many details and some of the brushstrokes are of similar colour and structure as the cracks, we introduce a semi-supervised clustering based post-processing step to remove false positives. The detection is finalized by combining the cleaned crack maps of each of the three methods using a simple voting scheme.

Additionally, the bright borders that are present around some of the cracks (and are accentuated by the way the data set was acquired) cause incorrect and visually disturbing inpainting results. These borders are the result of two factors: where a crack is forming, the paint is pushed upwards and forms a small inclination. Light gets reflected on the ridges caused by this cracking and makes them appear brighter than their immediate surroundings. Also, during previous cleaning, the surface paint on these elevated ridges may have been accidentally removed, revealing parts of the underlying white preparation layer. Section 3 elaborates on the improved patch-based inpainting for the digital filling of cracks and the inpainting results.

In Section 4 the practicability of our technique is confirmed by means of a case study which consists of improving the readability of depicted text in a very small detail in one of the panels. We end the paper with concluding remarks, presented in Section 5.

2. Crack detection

Cracks can visually be categorized into two classes, bright cracks on a dark background or dark cracks on a bright background (see Fig. 1). Mainly dark cracks are treated in the literature, where they are typically considered as having low luminance and being local (grayscale) intensity minima with elongated structure [19]. Different crack

detection techniques include simple thresholding, line detectors and various morphological filters (see [5] for an overview). Thresholding does generally not work well due to the noisy nature of the images and the presence of other “crack-like” structures in the image. The varying quality of the images and the difficulty of detecting cracks in low-contrast zones justifies several pre-processing steps. We introduce three different crack detection techniques that can be applied for the detection of both dark and bright cracks, each having its own strength. A semi-automatic clustering based post-processing step is applied to reduce the number of false detections. We subsequently combine the results of each technique and hence put to use their respective strengths. The procedure for the entire crack detection is shown in Fig. 2.

Furthermore, we identified the need to detect bright borders that surround some of the cracks (as depicted in detail in the upper right corner of Fig. 1 and in Fig. 4) because they cause incorrect and visually displeasing inpainting results. We will show in Section 3 that their detection and treatment as missing regions lead to improved inpainting.

2.1. Pre-detection processing

The detection of cracks in low contrast areas is a particularly difficult task. To deal with this problem, we introduce a local contrast enhancement step for darker zones in the image prior to crack detection. This



Fig. 1. Crack types: dark cracks on bright background (top), and bright cracks on dark background (bottom). The width of the mouth is 2.86 cm (approximately 1.13 in) on the panel.

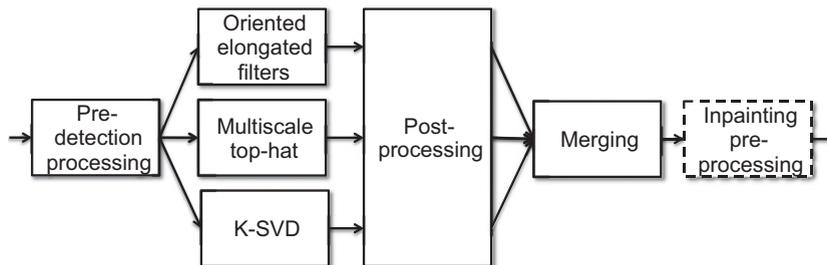


Fig. 2. Workflow for the entire crack detection procedure. The optional inpainting preprocessing (see Section 2.5) is applied whenever bright borders around the crack are prominently present (e.g. in Fig. 21).

pre-processing step consists of taking the weighted average of the original grayscale image (I_{orig}) and its locally contrast enhanced version (I_{CLAHE}) such that only darker areas are replaced with a contrast enhanced version. The resulting image I is constructed as follows:

$$I = (1-w)I_{orig} + wI_{CLAHE}, \tag{1}$$

where the high contrast image I_{CLAHE} is constructed by using *contrast limited adaptive histogram equalization* (CLAHE) [20] and the weights w for each pixel are determined by blurring (Gaussian kernel with $\sigma = 15$ and mask size 45×45) the inverted image I_{orig} (see Fig. 3).

Due to the noisy nature of the images we perform anisotropic diffusion [21] on image I as isotropic blurring would remove edges too much, which is in fact what we want to detect. We work on high resolution scans of original photographic negatives (Kodak Safety Film 13×18 cm) taken by a professional photographer, the late Rev. Alfons Dierick, whose material is currently preserved in the Alfons Dierickfonds archive of the Ghent University. The capturing process (i.e. field of view, lighting, etc.) is undocumented and the images are noisy (an example is given in Fig. 4). Not only were different chemical processes used to develop the negatives, but some of them were also acquired at different resolutions and with different scanning hardware. In strong contrast to the medical world, where standardization made common image processing tools possible, it is very challenging to find a global parameter set for all images under investigation. The data set being heterogeneous in nature, the amount of anisotropic diffusion is determined heuristically: the higher the resolution of the images, the more iterations are chosen for the diffusion.

2.2. Detection methods

Three novel crack detection methods are applied for the detection of both dark and bright cracks. These techniques have complementary strengths, which are explained in Section 2.4.

2.2.1. Filtering with oriented elongated filters

Oriented elongated filters (see Fig. 5) were originally introduced to detect and enhance blood vessels of different thicknesses and orientations in medical images [22]. They are used here for the detection of cracks and are obtained

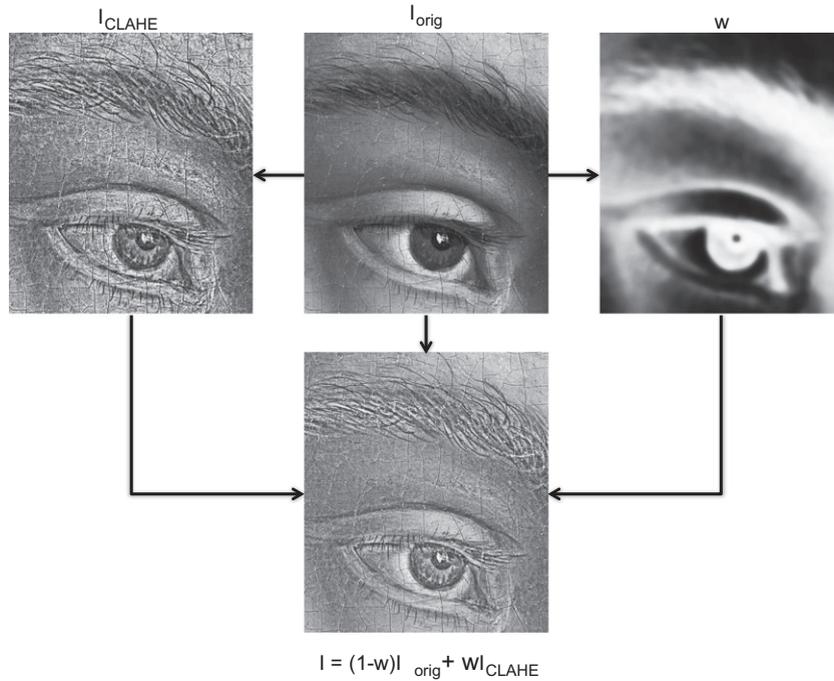


Fig. 3. Local contrast enhancement workflow.

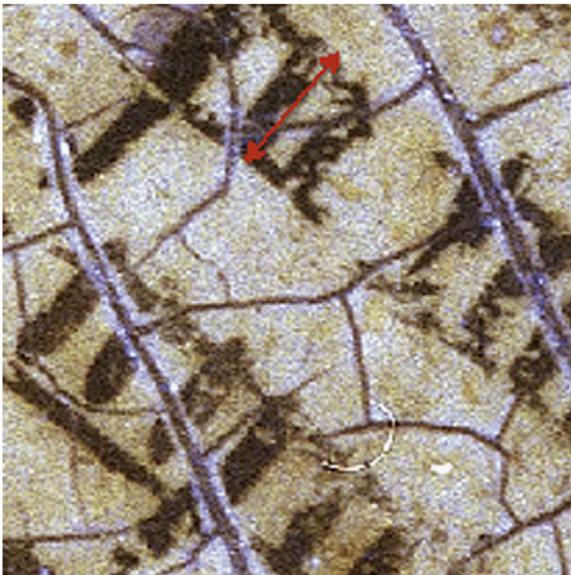


Fig. 4. Close-up of Kodak negative scan (showing bright borders around the cracks and noise). The height of one letter (marked in red) is approximately 0.17 cm (or 0.07 in) on the panel. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this article.)

by performing linear combinations of 2D Gaussian kernels:

$$G(x,y) = \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right). \quad (2)$$

In order to construct a filter that emphasizes edges orthogonal to a given unit vector $\mathbf{n} = (n_x, n_y)$, the directional

derivative of the Gaussian kernel G is calculated as

$$G_n(x,y) = n_x \frac{\partial G(x,y)}{\partial x} + n_y \frac{\partial G(x,y)}{\partial y}. \quad (3)$$

The first order partial derivatives of G can be approximated with sufficient accuracy by adequately shifting the Gaussian kernels. For example:

$$\frac{\partial G(x,y)}{\partial x} \approx \frac{1}{\sigma} \left[G\left(x + \left(\frac{\sigma}{2}\right), y\right) - G\left(x - \left(\frac{\sigma}{2}\right), y\right) \right]. \quad (4)$$

For the filter to be sensitive to vessel- or crack-like structures the kernel obtained above is combined in the following way:

$$G'_n(x,y,w) = G_n(x+wn_x, y+wn_y) - G_n(x-wn_x, y-wn_y), \quad (5)$$

where w controls the half-width of the filter. To make the filter more directionally sensitive we combine shifted versions of the kernel G'_n in a slightly different fashion as what is proposed in [22]: a finer sampling grid is used in horizontal and vertical direction which induces a denser succession of kernels and results in a smoother filter,

$$\overline{G}_n(x,y,w,l) = \frac{1}{2lN+1} \sum_{k=-lN}^{lN} G'_n\left(x + \frac{k}{N}t_1, y + \frac{k}{N}t_2, w\right), \quad (6)$$

where $\mathbf{t} = (t_1, t_2)$ is a unit vector orthogonal to \mathbf{n} and l controls the length of the resulting kernel. These filters can directly be applied to detect cracks brighter than their background. To detect dark cracks it is sufficient to invert the sign of all filter coefficients and apply this new set of filters on the image I . Examples of oriented elongated filters for the detection of dark cracks are shown in Fig. 5.

As the filters tend to respond to step-like edges as well, the filtered images are validated by comparing gray values of pixels on both sides of the edge pixel at a



Fig. 5. Directional kernels \overline{G}_n of size 51×51 : $\sigma = 1.4$, $w = \sigma$, $N=1$ and $l = 10\sigma$.

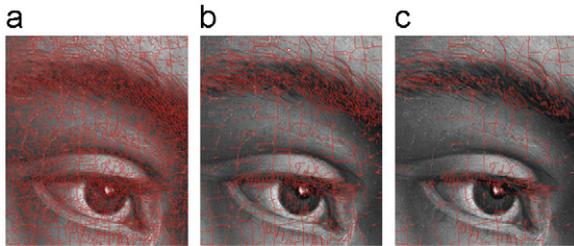


Fig. 6. Influence of the high threshold (fixed lower threshold): (a) threshold too low, (b) selected threshold, and (c) threshold too high.

certain distance and angle (provided by \mathbf{n}) in the blurred version ($B = I * G$) of the image I as defined in Fig. 3. For a dark crack, the pixels on each side of the edge pixel need to have a higher grayscale value than the crack pixel and vice versa for bright cracks.

The resulting filtered and validated images (one for each directional filter) are hysteresis thresholded. First, the image is thresholded with a high and a lower threshold. All the high thresholded edges are retained together with low thresholded edges that are connected to them. The low thresholded edges that are not connected to high thresholded edges are discarded. Both thresholds are image dependent, which forces us to manually determine them through visual inspection of the results. Note that we use a single threshold value for all directionally filtered images. The final binary image, marking the location of cracks, is formed by combining the thresholded images with a logical OR operation and is referred to as *crack map*. We found experimentally that especially the value of the high threshold is critical, Fig. 6 shows its effect on the resulting crack map. The entire workflow is depicted in Fig. 7.

2.2.2. Multiscale morphological top-hat

A popular technique to detect details with particular sizes is the use of a morphological filter known as the *top-hat transformation* [23] which was already successfully applied in crack detection [3,17,19]. For the detection of dark cracks on a lighter background we use the *black top-hat* (or *closing top-hat*) transform $TH_b(I)$ which is defined as the difference between the morphological closing $\varphi_b(I)$ of a grayscale image I using a structuring

element b and the input image I and results in a grayscale image with enhanced details. Clearly, the structuring element should be chosen according to the size and nature of the cracks to be detected. However, this process is still an open problem [5]. Recall that the morphological closing operation is defined as *dilation* followed by *erosion*. By thresholding $TH_b(I)$, which in our case is performed automatically by using Otsu's method [24], we create a binary image of details which are most likely to be dark cracks. A similar technique can be employed for the detection of bright cracks by replacing the black top-hat transform with a *white top-hat* (or *opening top-hat*) transform, which is defined as the difference between the input image I and its *opening* $\gamma_b(I)$ by a chosen structuring element b .

Instead of using the classical top-hat transforms we use a *multiscale* morphological approach, introduced in [1] and depicted in the workflow of Fig. 8. The noisy nature of the images as well as the multitude of structures having crack-like properties can cause many misdetections (undesired for the subsequent inpainting). Both the reduction of the undesired false positives and the detection of cracks of varying size (ranging from very small hairline cracks to larger areas of missing paint) benefit from a multiscale detection scheme. We perform the top-hat operation described above with square shaped¹ structuring elements b of varying sizes (ranging from 3×3 to $n \times n$ pixels, where n depends on the width of the crack to be detected²). By choosing a small structuring element, we extract hairlike cracks but also a lot of other fine scale structures that do not correspond to cracks. When using a large structuring element, on the other hand, we detect cracks but coarser structures as well. Next, we automatically threshold the results of the successive morphological filterings using Otsu's method to obtain different *crack maps*, which is an improvement on the detection method of [1] because manual selection of thresholds is avoided. All the crack maps are subsequently further processed, by bridging pixel gaps and

¹ Using other shapes of similar dimensions did not really affect the detection process.

² $n=8$ for all images except for the special case of the book of which a detail is shown in Fig. 4, where $n=10$.

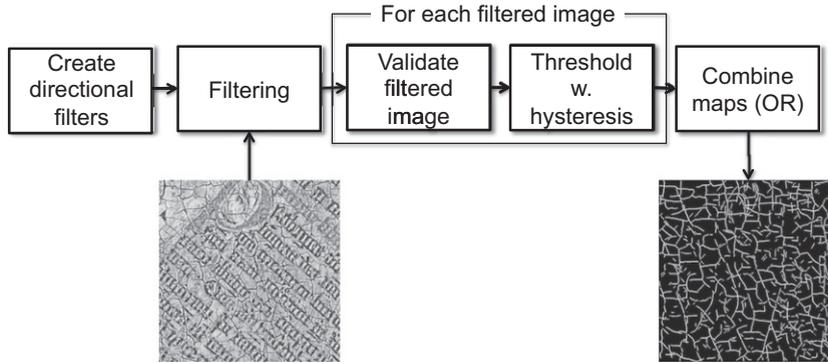


Fig. 7. Filtering with elongated directional filters workflow.

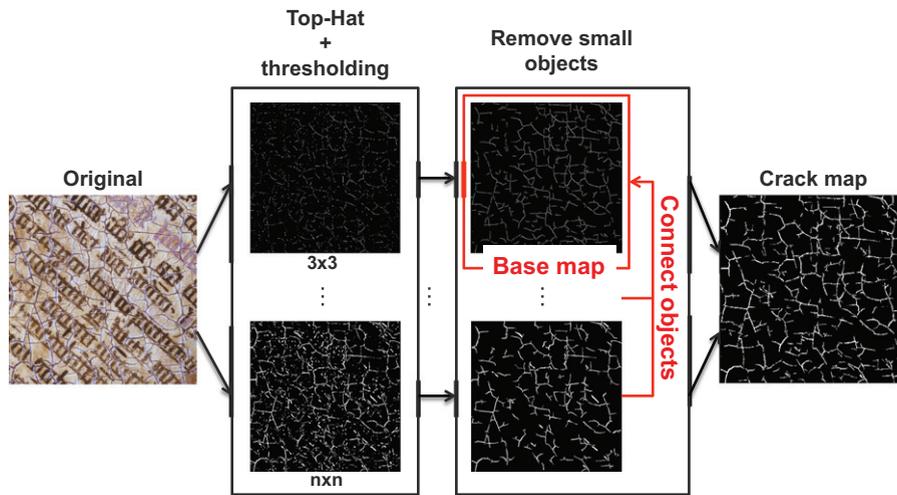


Fig. 8. Multiscale morphological top-hat workflow.

removing isolated or small groups of pixels as much as possible.³

We now combine the resulting crack maps from successive scales in a novel way. The crack maps of the three smallest scales are added to form our *base map* which is used as a reference for selecting cracks in crack maps corresponding to coarser scales. The final crack map results from the base map by adding only objects (i.e. groups of connected pixels) from maps at coarser scales that are connected to the base map. This way of working has the major advantage that unwanted larger structures such as the mustache of Adam in Fig. 1 or the letters in the book image depicted in the example of Fig. 8, more often detected at larger scales, will not be included in the final map while cracks are still allowed to grow through successive scales.

2.2.3. K-SVD for crack enhancement

The third technique proposed in this paper relies on the use of the K-SVD algorithm [25], which is used for constructing and adapting dictionaries in order to achieve sparse signal representations and which was never applied before in a crack detection context. It is an iterative method that alternates between sparse coding of the data based on the current dictionary and the updating of dictionary atoms to better fit the data.

The K-SVD algorithm works on small (usually overlapping) patches $x \in \mathbb{R}^n$ of size $\sqrt{n} \times \sqrt{n}$. The learning of the over-complete dictionary $D \in \mathbb{R}^{n \times k}$ with a fixed number of atoms k and sparsity constraint L (i.e. each patch is composed of no more than L dictionary atoms) can be described by the following optimization problem:

$$\min_{\alpha, D} \|X - D\alpha\|_2^2 \quad \text{s.t.} \quad \|\alpha_i\|_0 \leq L, \quad (7)$$

where the column vectors (x_i) of X are M image patches on which the dictionary D is learned. Each dictionary atom $d_i \in D$, for $i = 1 \dots k$, is a unit vector in the ℓ_2 -norm. The column vector α_i of α is the sparse coefficient matrix vector corresponding to the patch x_i and $\|\alpha_i\|_0$ denotes the number

³ Standard parameterless Matlab functions are used in the following sequence: *bridge*, *clean* and *fill*. These are followed by an additional cleaning step, removing objects of size smaller than 20, 30, 40, ..., 70 pixels for the six successive morphological filtered images.

of non-zero elements in the vector. K-SVD is an iterative method specifically designed to minimize the energy of (7) and where each iteration consists of two steps. The first one is the sparse coding step which addresses the problem of finding the best decomposition $D\alpha_i$ for each patch x_i . To this end, a greedy *orthogonal matching pursuit* (OMP) is used. The second step, called the dictionary update step, optimizes the least-square problem (i.e. the ℓ_2 -norm in (7)) for each atom individually while keeping the remaining atoms fixed and updates the corresponding non-zero coefficients in α . K-SVD is used in this context as a method for enhancing cracks within the image by training an optimal dictionary D for each image and altering the coefficient matrix α during reconstruction in an appropriate way.

A dictionary ($k=128$) is trained on overlapping patches of the image I , containing cracks, hence producing an over-complete dictionary tweaked to that image. Patches must be sufficiently large to capture crack-like edges ($\sqrt{n}=16$) and the sparsity constraint is very strict (L equal to 1 is typically chosen). After training (using 20 iterations, and on a sufficiently large number of patches), some dictionary elements will contain information for the “optimal” representation of cracks due to the strong sparsity constraint. We use the trained dictionary to reconstruct an image patch I_c containing *only* cracks, which obviously favours the usage of crack-like dictionary elements. This can be observed in the histogram of dictionary usage depicted in Fig. 9. Choosing an appropriate I_c is straightforward: examples are given in Fig. 10. When reconstructing the image I , the number of elements in the coefficient matrix α , corresponding to less used dictionary elements for I_c are put to zero. Note that the dictionary itself remains unaltered. The rule for zeroing out coefficients is as follows:

$$\forall l, \alpha_{li} = \begin{cases} \alpha_{li} & \text{if } \|\alpha'_i\|_0 > \beta \max\|\alpha'_i\|_0, \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

where α' is the coefficient matrix when reconstructing the *crack-only* image patch I_c , $\|\alpha'_i\|_0$ the corresponding total usage

of dictionary element d_i , l the index of the patch extracted from I and β a parameter controlling the amount of coefficients to put to zero. The default value for β is chosen as 0.5 and its value is lowered when it is observed that an insufficient number of cracks are reconstructed, but this happens rarely and hence user interaction is kept to a minimum. Through this procedure, non-crack dictionary elements are omitted during the reconstruction of I .

Note that the DC component of each patch is subtracted before training and reconstruction, which results in a reconstructed image that can contain negative values as well. As a final step the image is hysteresis thresholded in the same fashion as described in Section 2.2.1 for the detection of bright cracks while for the detection of dark cracks the reconstructed image needs to be inverted first (results are shown in Fig. 11).

2.3. *k*-Means clustering based semi-automatic post-processing

In the three proposed methods some objects, having a similar structure as cracks (such as eyebrows, eyelashes and the mustache in the Panel depicting *Adam* or the letters of the book in the *Annunciation to Mary* panel) are

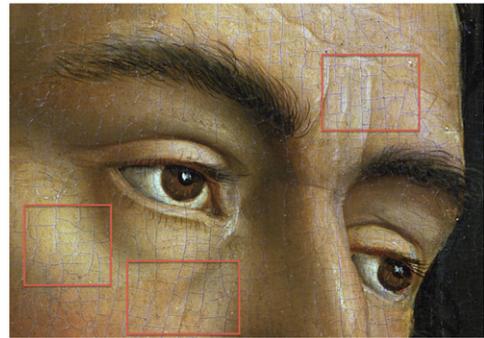


Fig. 10. Candidate image patches, I_c , for training of dictionary.

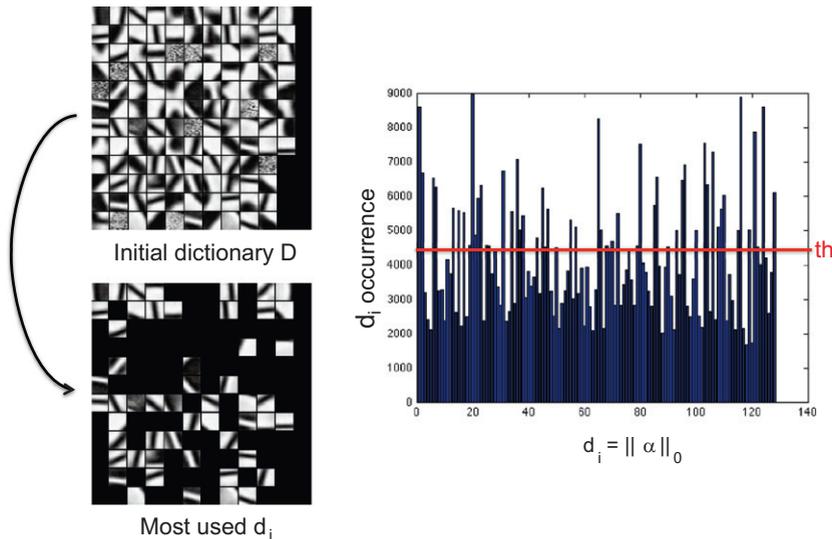


Fig. 9. Dictionary usage histogram and most used d_i ($th = \beta \max\|\alpha'_i\|_0$).

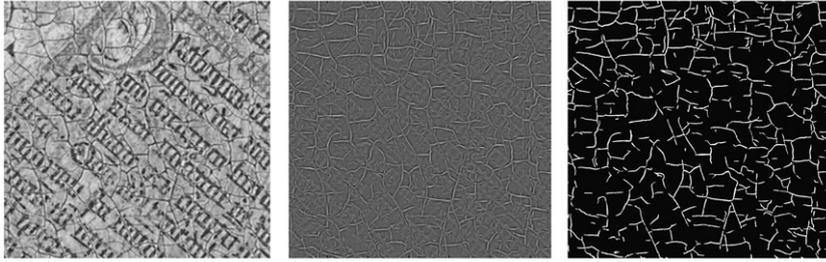


Fig. 11. K-SVD results (left: original; middle: reconstructed (inverted); right: thresholded).

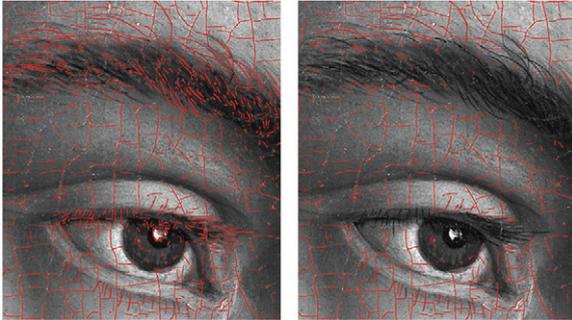


Fig. 12. Results of detection with K-SVD (left: original detection; right: after post-processing with k -means clustering).

sometimes falsely labelled as cracks. We propose a clustering based semi-automatic method to assist in filtering out those structures.

First the binary map, obtained by applying one of the methods described above, is thinned to obtain one-pixel wide cracks [26]. Crack pixels are linked together into lists of coordinate pairs,⁴ where a crack junction is encountered. The list is terminated and a separate list is generated for each of the branches. In this manner, each crack is broken down into segments which can be treated as separate objects. For each of these objects, a number of features are calculated: colour (the object's mean colour values in RGB and HSV colour space) and physical properties such as length, orientation and eccentricity. Next to these, the colour values of the immediate region surrounding the crack are computed as well. Recall that cracks can sometimes be surrounded by a bright border (as explained in the beginning of Section 2) which can be a crucial feature for discerning cracks from non-cracks. As the last feature, a spatial density value is assigned to each object, which will prove to be a very significant feature (e.g. eyebrow edges typically belong to more dense regions than crack edges in the image as can be observed in Fig. 12). All these features are combined into a feature vector and serve as input to a k -means clustering. It is now a matter of manually removing undesired clusters which correspond to falsely labelled objects, by simply specifying one or more cluster numbers. Results

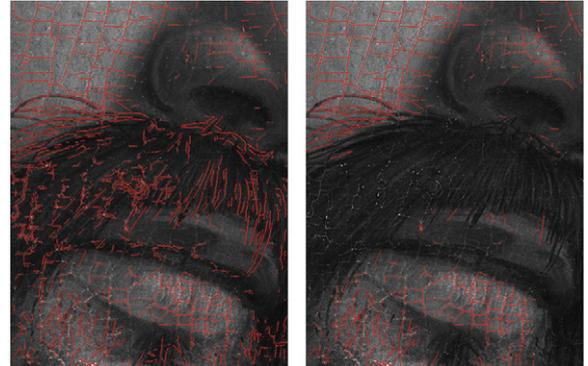


Fig. 13. Results of detection with elongated filters (left: original detection; right: after post-processing with k -means clustering).

are shown in Figs. 12 and 13 for k equal to 3 and after the removal of one cluster.

2.4. Combining the results of each method

During experimentation, it became clear that each method has its own strength and weakness. Filtering with the elongated oriented filters usually detects most of the cracks since a single value for σ (which controls the width of the filter, see Section 2.2.1) will generate a high response from elongated structures of various widths. This is one of the major advantages of this technique, but also its weakness since other elongated structures tend to have a high response after filtering as well. The multiscale morphological approach described in Section 2.2.2 works well on images containing letters and significantly reduces the number of mislabelings compared to the classical top-hat transform. Moreover, thanks to the introduced notion of scale when constructing the final crack map, a distinction between fine and coarser cracks is possible. However, on images such as parts of the face of Adam, the method can miss some of the very small cracks during the construction of the base map, resulting in a number of undetected cracks when gradually building the final crack map through scale. The K-SVD approach works generally well and provides a very smooth crack map but results will depend on how well the dictionary reflects different crack widths and orientations.

⁴ Matlab functions are available on: <http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/index.html#edgelinek>.

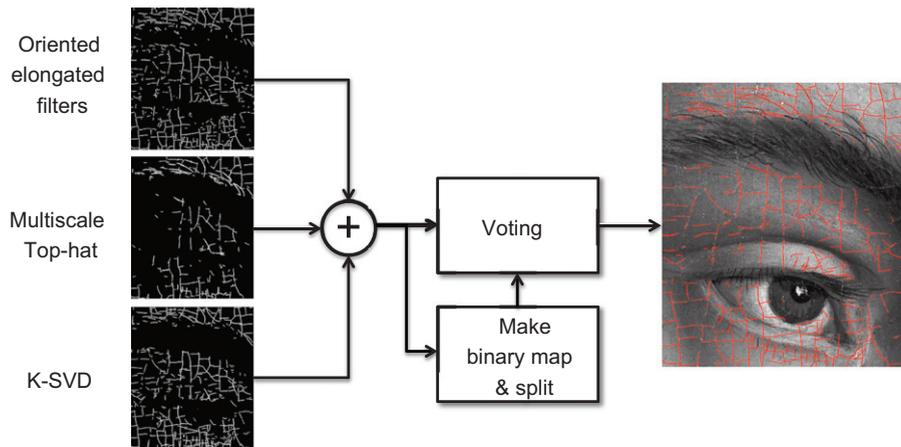


Fig. 14. Crack merging workflow.

In summary:

- We use elongated filters as a sensitive crack detector with low selectivity for crack width.
- The multiscale morphological approach adds selectivity with respect to the crack width.
- K-SVD adds smoothness to the crack delineation.

We propose a method for combining the above-mentioned techniques to exploit their respective strengths. First we add all crack maps to obtain one “image” where the value of each pixel ranges from 0 (no crack was detected at that location) to 3 (all schemes detected a crack at that location). From that image we generate a binary map which we split in the same fashion as mentioned in Section 2.3. For each of the objects a ratio of “pixels detected by at least two methods” to “the total number of pixels” is calculated. When that ratio falls under a chosen threshold, the crack segment is discarded from the final map (see Fig. 14 for an overview). In all our examples a fixed threshold of 0.3 gave the best results.

2.5. Detecting crack borders-inpainting preprocessing

Some of the images suffer from an additional artefact that proves to be very bothersome when inpainting, i.e. the presence of whitish/bright borders along some of the cracks. Most inpainting algorithms fill in gaps based on pixel values from their immediate surroundings, in this case the whitish borders around a crack. Hence the missing regions will likely be filled with incorrect content and the positions of cracks remain visible after inpainting. When detecting bright cracks most of the thin bright borders are captured, which on its own, proves to be insufficient for acceptable inpainting results, since white borders can be much wider than cracks. To solve this problem, we extend the crack map with the corresponding bright regions by using their high response in the blue plane of the RGB representation of the image (Fig. 15 shows the blue plane for a window in the original). A flood filling algorithm, using the maxima in the blue plane as seeds and RGB values as filling condition, extends



Fig. 15. Detecting white borders (left: original; right: detection results). (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this article.)

the crack map so that the extension contains most whitish borders. To avoid excessive extension of the borders too far from the crack, we restrict the flood filling by an Euclidean distance transform. Fig. 15 shows an example where the detected bright borders are marked in yellow (bright borders cannot extend further than 6 pixels from a crack).

2.6. Two extra examples of crack detection results

Figs. 16 and 17 contain some extra examples of detection results for dark and bright cracks, obtained with the method described above (see Fig. 2). The test images contain typical objects found in the different panels from the Ghent Altarpiece.

3. Inpainting

3.1. Existing crack inpainting methods

In the process of virtual restoration of digitized paintings, cracks, once detected, can be treated as missing regions that need to be filled in. Therefore, removal of cracks falls into the category of image inpainting. Crack inpainting methods considered in the literature so far are mostly pixel-based and include order statistics filtering [16,17], controlled

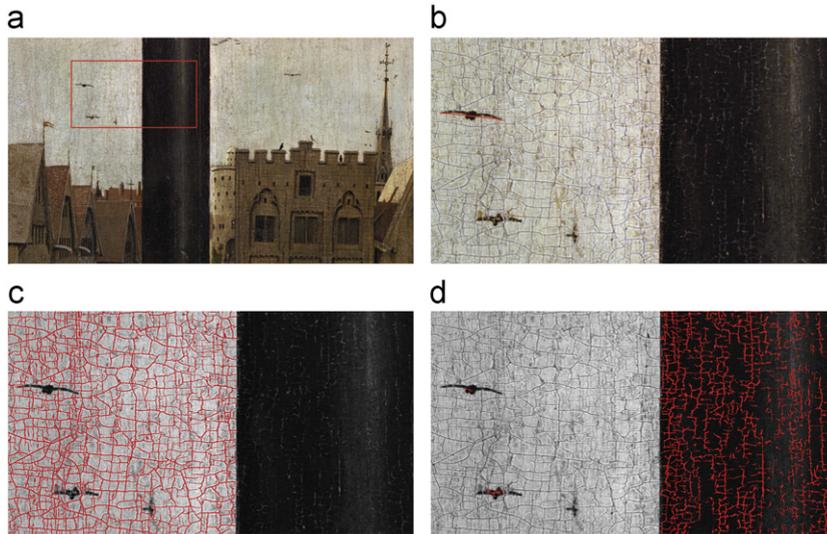


Fig. 16. Detection results: (a) original, (b) detail, wing span of bird (marked in red) is 1.2 cm (approximately 0.47 in) on the panel (c) overlap with dark crack map, and (d) overlap with bright crack map. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this article.)

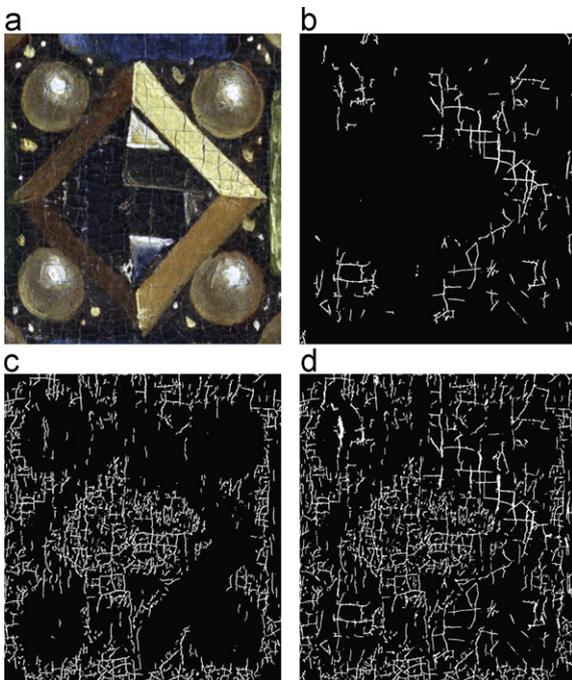


Fig. 17. Detection results: (a) original, (b) dark crack map, (c) bright crack map, and (d) combined dark and bright crack maps.

anisotropic diffusion [16] and interpolation [15]. In [18] a patch-based texture synthesis method was used.

We shall compare our approach to the best performing crack inpainting methods among aforementioned ones. In particular, we use as a reference controlled anisotropic diffusion, which was reported in [16] to outperform other pixel-based crack inpainting methods, including order statistics filtering. Our second reference method for

comparison is a “greedy” patch-based method of [11], which was employed for crack inpainting (in a slightly simplified form) in [18]. In addition, we tested a global patch-based method of [13], which gives state-of-the-art results among general inpainting techniques even though it was not, to our knowledge, used for crack inpainting before. For all of these methods, only crack pixels are substituted, leaving the rest of the image intact.

Anisotropic diffusion [21] belongs to the pixel-based inpainting methods where crack pixels are updated iteratively as a result of the diffusion process within their neighbourhoods. It combines smoothing of slowly varying intensity regions and edge enhancement. Controlled anisotropic diffusion [16] takes into account crack orientation, i.e. the operation is performed only in the direction perpendicular to the crack direction. In our experiments, this pixel-wise method does not always perform sufficiently well (see Figs. 18 and 19) because of its inability to reproduce texture and to fill in larger holes. Cracks should normally be represented by thin lines, but in our case they are quite wide because of the high resolution at which the negatives were scanned. The lack of performance is aggravated when the cracks suffer from whitish borders which need to be treated as missing regions too (see Section 2.5). Furthermore, the quality of the scans, i.e. the presence of noise and scanning artefacts, raises the need for better texture replication because diffusion-based methods produce blurry results.

Patch-based methods fill in the missing (*target*) region patch-by-patch by searching for similar patches in the known (*source*) region and placing them at corresponding positions. The basic idea of the “greedy” method [11] is the following: for each patch at the border of the missing region (*target* patch), find *only* the best matching patch from the source region (*source* patch) and replace the missing pixels with corresponding pixels from that match,

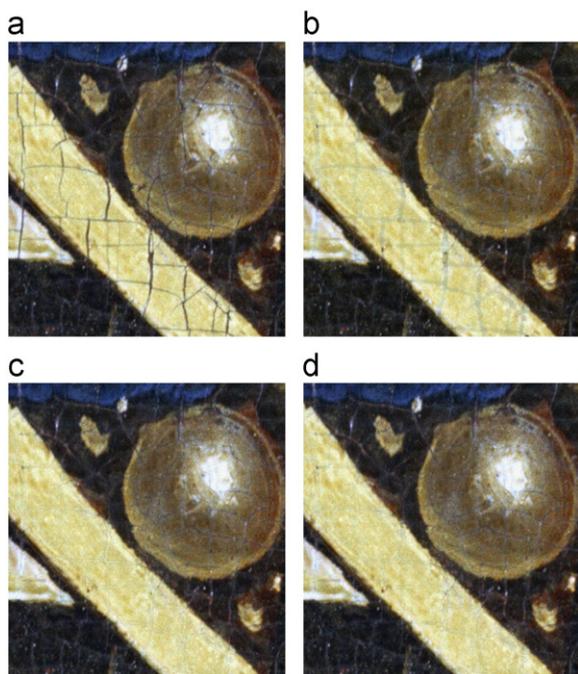


Fig. 18. Inpainting results for the central part of Fig. 17(a): (a) original, (b) with controlled anisotropic diffusion, (c) with “greedy” patch-based method, and (d) with global patch-based method.

until there are no more missing pixels (Fig. 20). The matching criterion is usually the sum of squared differences between the known pixels in the target patch and the corresponding pixels in the source patch. In this way, both texture and structure are replicated. Preserving structures is achieved by defining the *filling order*. Priority should be given to the target patches that contain object boundaries and less missing pixels. In the case of digitized paintings, the object boundaries are usually difficult to determine due to painting technique (incomplete brushstrokes), scanning artefacts, etc. Therefore, we define priority based only on the relative number of existing pixels within the target patch.

The *global patch-based method* [13] poses inpainting as a global optimization problem. For each target patch in the missing region several candidate patches are found based on the known pixels and/or neighbouring context. Again, the target patches are visited in a certain order that favours patches containing object boundaries and less missing pixels. Then one of the candidates is chosen for each position so that the whole set of patches (at all positions) minimizes the global optimization function.

The performance of the aforementioned methods is evaluated only by visual inspection, as in other papers on this application [16–18]. Quantitative comparison is infeasible due to the unavailability of the ground truth data, i.e. we have no information on how the painting looked like in its original state, before the deterioration of wooden panels. On the other hand, the nature of the painting itself and the influences of the acquisition process of the digitized version (such as noise and

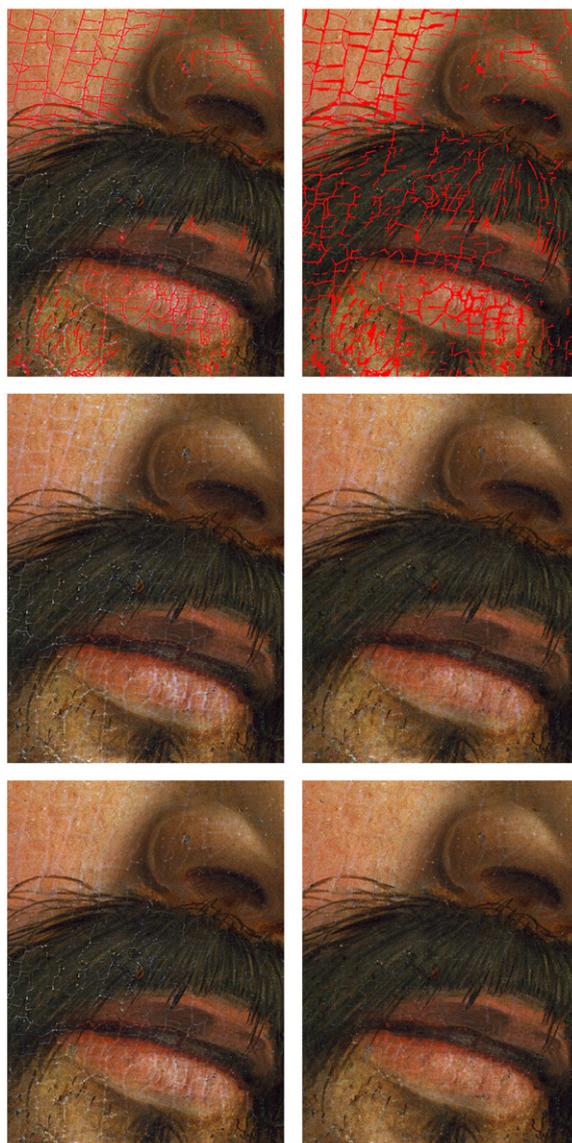


Fig. 19. Influence of bright borders on inpainting. Left: Dark crack map as input. Right: Combined dark and bright crack map as input. First row shows the original image overlapped with crack maps. Second and third rows show inpainting results with controlled anisotropic diffusion and the patch-based method, respectively.

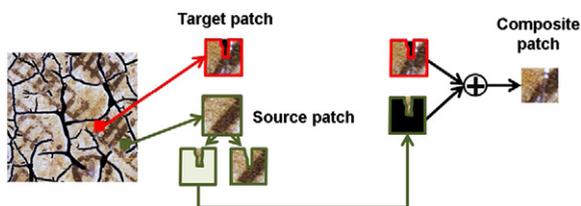


Fig. 20. Schematic representation of a patch-based inpainting algorithm.

scanning artefacts) make it very difficult to replicate the problem in a form of a suitable toy example on which the objective measurements could be performed.

By visually comparing the results of these three methods (see Fig. 18), we can see that both patch-based methods outperform the pixel-based one, due to the aforementioned reasons. However, they both still leave much room for improvement when crack inpainting is considered. The complex global method performs similar to the simpler greedy one, but it results in a very high computational load due to the high resolution of the scans (e.g. for an image of 660×700 pixels it is around 100 times slower than the “greedy” method), making it impractical for processing of larger areas. On the other hand, limiting the method to small areas can jeopardize finding the right match. Therefore, we adopt the “greedy” patch-based method and improve it for crack inpainting.

3.2. Open problems and proposed solutions for crack inpainting

To improve the inpainting performance, some specifics of the problem need to be tackled. In some cases the presence of bright borders around the cracks (see beginning of Section 2) causes the missing crack regions to be filled with incorrect content and the positions of cracks to remain visible after inpainting (see the results on the left of Fig. 19). Often, this problem is partially solved by using the bright crack map, which extends the dark crack map with the corresponding bright regions. Because this bright crack map also marks some of the bright borders, the benefit of using this map is evident in all cases: in the results on the right of Fig. 19 more cracks are detected and inpainted, causing a more pleasing visual appearance. However, for some images this procedure might not be sufficient due to the width of the borders. In those cases, the inpainting preprocessing from Section 2.5 is used to obtain the map of crack border locations (see Fig. 15). The improvement of the inpainting result is shown in Fig. 21c, in comparison with the result obtained using just the dark crack map shown in Fig. 21b. If the image also contains bright cracks, all three maps (dark crack map, bright crack map and border crack map) are combined together.

The standard patch-based method gives reasonably good visual results for most parts of the panels. However, the book of the *Annunciation to Mary* panel is exceptionally difficult to process due to the width of the cracks, prominent scanning artefacts and imperfect brushstrokes (see Fig. 4 for a detail and Fig. 24 for the whole book). This causes some cracks to remain undetected and misguides the inpainting during the patch matching process. A first consequence is that we can get an inpainted image where small parts of letters appear erroneously in the background and the other way around, parts of letters get “deleted”, i.e. replaced by background. A second consequence is that positions of cracks remain visible. Exactly in the part of the panel containing the book, accurate inpainting is very important because of the case study on paleographical deciphering explained in Section 4. To further improve the crack inpainting results, we introduce a novel method that involves two contributions: an approach to patch candidate selection and an approach to patch size adaptation. This method, that we call *constrained candidate selection*, aims at performing context-aware

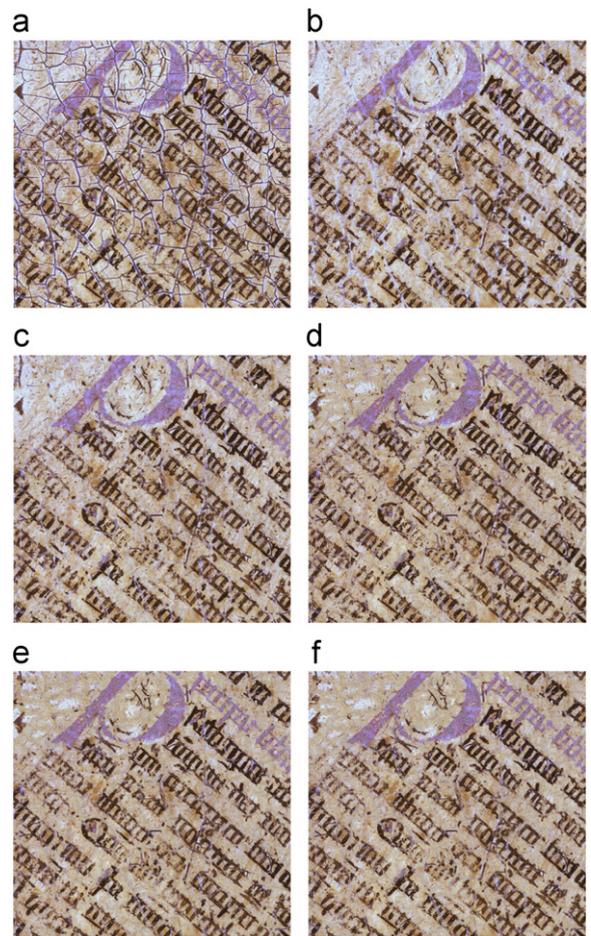


Fig. 21. Inpainting with patch-based methods on part of the book: (a) original image, (b) method [11] with dark crack map, (c) method [11] with combined dark and border crack map, (d) label constrained method, k -means segmentation [1], (e) label constrained method, MRF segmentation, fixed patch size, and (f) label constrained method, MRF segmentation, adaptive patch size.

inpainting by constraining the search to certain parts of the image, depending on the content of the current target patch. Our method consists of three main steps:

1. *Exclusion of damaged pixels*: Although we use the bright crack map and/or border crack map (see earlier in Section 3.2) to deal with the problem of whitish borders around the cracks, some damaged pixels still remain. These pixels are either too distant from the crack, belong to the non-detected cracks or appear in the source region not related to the cracks. We detect these pixels based on their high values in the blue plane and we treat them as missing ones. We do not use the patches from the source region containing damaged pixels as possible matches. The threshold applied to the blue plane is chosen high enough to allow sufficient number of candidate source patches, while still detecting the artefacts around cracks. In particular, we chose a fixed threshold equal to 220 by inspecting the histogram of manually marked damaged regions.

2. *Label constrained matching*: In the results from Fig. 21c it can be seen that patch-based inpainting occasionally introduces some artefacts. This can happen because the known part of the target patch is not distinctive enough to find the right source patch. Another reason is that undetected cracks can be present in the known part of the target patch so that the matched source patch will probably contain a letter, since cracks and letters often have similar properties. To minimize these errors we first segment the image into two classes: foreground (the letters and undetected cracks) and background (the page of the book). In [1], we used the *k*-means segmentation algorithm on the values of the red plane because the difference between the two classes is most visible there. Here, we improve the results by using Markov Random Field (MRF) based segmentation [27], which is briefly explained below. Based on this segmentation, we constrain the search for candidate patches. When inpainting a part of the background, i.e. when all the known pixels in the target patch are labelled as background, we only accept source patches that belong completely to the background as candidate patches. We could perform a similar procedure for the target patches belonging completely to the foreground. However, some cracks that remain undetected by using the detection methods of Section 2 are also identified as foreground. This can result in the unjustified insertion of letters and/or cracks (foreground) in the background. Therefore, if the target patch is not entirely in the background, we search through all possible candidates.
3. *Adaptive patch size*: Instead of using a fixed patch size, as most inpainting methods do, we make the patch size adaptive to the local context. We start from the maximal patch size and check if the target patch completely belongs to the background. If this is the case, we constrain the search to the background, as in the previous step. If not, we reduce the patch size by half and repeat the same procedure. Finally, if even this smaller patch only partially belongs to the

background, we search for the match of the target patch of the maximal size at all possible locations.

As can be seen in Fig. 22a, *k*-means (as used in [1]) yields a noisy segmentation result with a lot of misclassified isolated dots in the background. Better context awareness would be beneficial to circumvent the incompleteness of letters. For these reasons, we use MRF based segmentation. To define the MRF, we need to specify *local evidence*, i.e. the relationship between the measured (pixel) value and the segment label, and a *pairwise potential function*, i.e. the dependency between two neighbouring segment labels (see [27] for more details). The former is defined as the Gaussian function with mean value equal to the value of the cluster center obtained with the *k*-means algorithm, and standard deviation computed within each cluster. The pairwise potential is determined by the discontinuity preserving Potts model $V_{i,j}(x_i, x_j) = KT(x_i \neq x_j)$, where T is one if its argument is true and zero otherwise, K is a positive constant and x_i and x_j are neighbouring segment labels. For inference, i.e. to find the maximum a posteriori estimate of the global optimization function, we use the method from [28] previously developed by some of the authors of this paper. The segmentation yielded by this method is shown in Fig. 22b, and we can see that the isolated dots have been removed and that the letters are more compact.

Fig. 21d contains the result of the constrained candidate selection from [1] that uses *k*-means segmentation. The effects of the proposed constrained candidate selection, using the MRF based segmentation, are illustrated in Fig. 21e for the constant patch size, and in Fig. 21f for the adaptive patch size. The results in Fig. 21e and f are perceptually better than those in Fig. 21d. Compared to Fig. 21e, Fig. 21f has less artefacts in the background meaning that the adaptive patch size approach can better locate target and source patches belonging to the background. Also, some letters are better inpainted. In comparison with the results of the method from [11] in

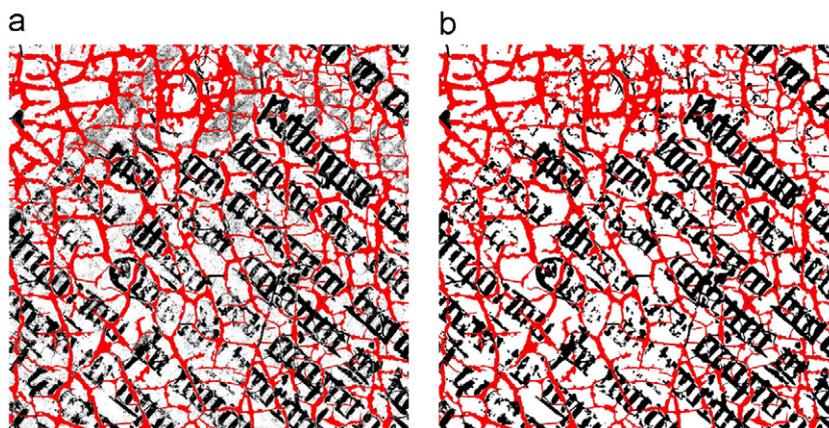


Fig. 22. Segmentation results for part of the book (cracks detected as in Section 2 in red, letters and undetected cracks in black, background in white): (a) result of *k*-means [1] and (b) result of MRF based segmentation. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this article.)

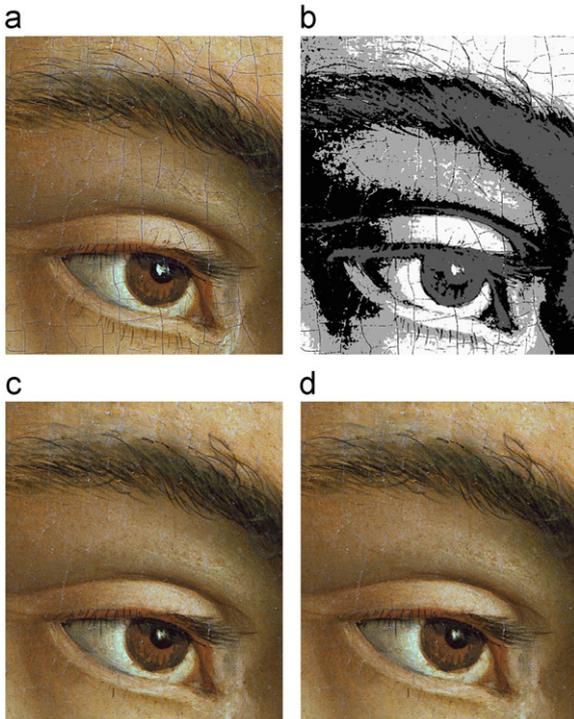


Fig. 23. Label constrained matching for image containing four segments: (a) original, (b) MRF segmentation, (c) result of patch-based method [11], and (d) result with label constrained matching.

Fig. 21c, the letters are better inpainted and the whole image contains less visually disturbing bright borders.

Note that our label constrained matching, i.e. the second step of the complete method, can in principle also be used on more complex images, containing more than two segments, to limit the search to specific areas so that the computation time is reduced. However, in general, the improvement on the quality of the inpainting result is minimal. Results are shown in Fig. 23: (c) shows results without label constrained matching and (d) with label constrained matching, based on the segmentation shown in (b).

4. A case study—De Visione Dei

In this section we focus on a detail that has puzzled art historians for years, namely the text in a book represented on the panel of the *Annunciation to Mary* (see Fig. 24). The text can only be studied from high resolution photographs as access to the altarpiece (which is being kept inside a vault behind glass) is difficult. Furthermore, the height at which the text is situated on the panel makes a direct reading for paleographers impossible. The text in the Virgin Annunciate's book is written in a so-called *littera formata* (see Fig. 4 for a detail), which always poses difficulties for paleographers to decipher. Sequences of vertical stripes, often without ligatures, as e.g. |||||, can be read as combinations of i, u, n, m, v, w, etc. Moreover, horizontal abbreviation marks above the text lines replace

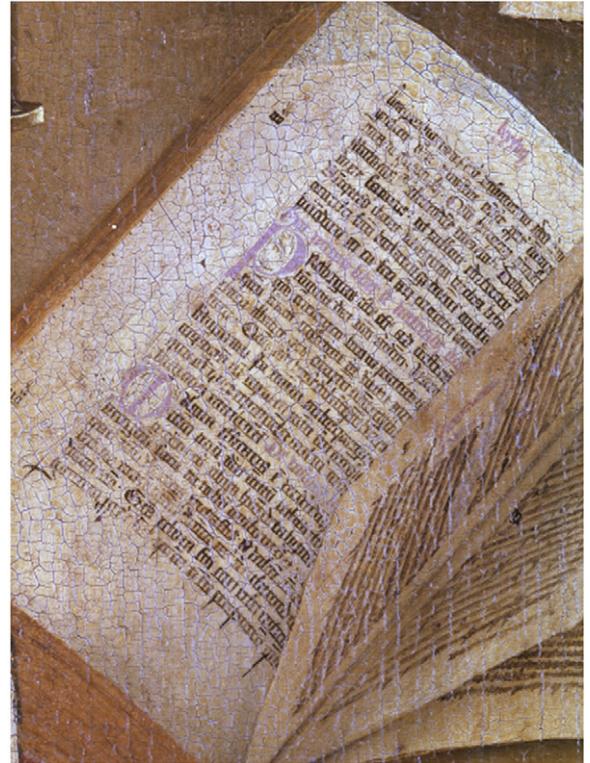


Fig. 24. Book in the panel of the *Annunciation to Mary*. Its actual size on the panel is approximately 12 cm (or 4.7 in) in length and a page is approximately 7 cm (2.8 in) wide.

letters 'n' or 'm'. In this particular text, the crack pattern made deciphering even more problematic. Therefore, only two significant word groups have been read: *de visione dei* (on the vision of God), and *dicit sapiens: ut possim edificare* (says the wise man: that I may build) [29]. This provides not enough information to identify the text; it allows only to speculate that it could be one of the numerous medieval commentaries on the Bible.

As amply demonstrated in the abundant art historical literature on the Ghent Altarpiece, each one of the many inscriptions or texts on this painting bares important clues to the original complex iconographical and theological meaning of this work of art. Therefore any attempt to improve the legibility of the text next to the Virgin Annunciate is of great relevance to its art historical interpretation.

The crack detection and inpainting process described above has yielded a better legibility indeed. Although the text cannot be read entirely, some additional word groups can be deciphered now as: *hio dicta significata* (telling the message with mouth wide open), *de virtutibus d[ei]* (on the virtues of God), *in videndo* (the appearance of God). The former reading of *Prologus iste est ad* can be completed with the words *differentiam cognite dei*. Moreover, the paragraph mark on the upper left of the page should be read as LXII (62) rather than VII (7).

All deciphered text fragments are related to the Annunciation, and can be found in Thomas of Aquino's

Summa Theologica (written between 1266 and 1273). This prominent medieval theologian commented in his book on the Annunciation, on the vision of God, and in the 62nd paragraph of the *Summa*, on the cardinal and theological virtues (*virtutibus*). These first results provide a basis for further research into the iconographical implications of this text.

5. Conclusions

In this paper, we introduce a novel way of virtually restoring paintings by detecting and removing cracks. The practicability of the proposed methods is demonstrated on images taken from the Ghent Altarpiece. Due to the heterogeneous nature of these images, existing solutions found in the literature have proven to be insufficient. For the detection of cracks, the outputs of three new detection schemes are combined, hence capitalizing on each method's strength as much as possible. The paper describes in detail all problems encountered and proposes solutions for each of them. Due to the complexity of the problem to separate the semantic content of the painting from the crack pattern, several case dependent parameters have to be introduced, either automatically determined or manually tuned. For each of them we describe how the values can be found and we motivate the choices made. Furthermore, we explored the use of patch-based inpainting for the removal of the detected cracks and highlighted specific issues when inpainting the Ghent Altarpiece. Improvements to existing patch-based inpainting are proposed and we demonstrate the gained performance. The inpainted images show less artefacts and the results are overall visually more pleasing. To demonstrate the applicability and the use of the proposed techniques, we presented the initial findings of a case study involving the deciphering of text in the panel of the *Annunciation to Mary*.

Acknowledgements

The Van Eyck images are based on photographic negatives (b45, b24, g09, 39-19) from the Dierickfonds made available to the Ghent University by the family of the late Alfons Dierick. We thank Saint Bavo cathedral, Lukas Art in Flanders and the Dierickfonds for permission to use these materials in this research article.

References

- [1] T. Ruzic, B. Cornelis, L. Platasa, A. Pizurica, A. Dooms, W. Philips, M. Martens, M. De Mey, I. Daubechies, Virtual restoration of the Ghent Altarpiece using crack detection and inpainting, in: *Advanced Concepts for Intelligent Vision Systems (ACIVS 2011)*, Ghent, Belgium.
- [2] J. Mohen, M. Menu, B. Mottin, *Mona Lisa: Inside the Painting*, Harry N. Abrams, New York, NY, USA, 2006.
- [3] F.S. Abas, K. Martinez, Classification of painting cracks for content-based analysis, in: *IS&T/SPIEs 15th Annual Symposium on Electronic Imaging 2003: Machine Vision Applications in Industrial Inspection XI*, 2003.
- [4] S. Bucklow, The description of craquelure patterns, *Studies in Conservation* 42 (1997).
- [5] F.S. Abas, *Analysis of Craquelure Patterns for Content-Based Retrieval*, Ph.D. Thesis, University of Southampton, 2004.
- [6] S. Bucklow, A stylometric analysis of craquelure, *Computers and Humanities* 31 (1998).
- [7] A.M. López, F. Lumbreras, J. Serrat, J.J. Villanueva, Evaluation of methods for ridge and valley detection, *IEEE Transactions on Pattern and Machine Intelligence* 21 (1999) 327–335.
- [8] F. Zana, J.C. Klein, Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation, *IEEE Transactions on Image Processing* 10 (2001) 1010–1019.
- [9] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, *Image inpainting*, in: *SIGGRAPH*, New Orleans, USA, 2000.
- [10] T.F. Chan, J. Shen, Non-texture inpainting by curvature-driven diffusions (cdd), *Journal of Visual Communication and Image Representation* 12 (2001) 436–449.
- [11] A. Criminisi, P. Pérez, K. Toyama, Region filling and object removal by exemplar-based image inpainting, *IEEE Transactions on Image Processing* 13 (2004) 1200–1212.
- [12] J. Sun, L. Yuan, J. Jia, H.-Y. Shum, Image completion with structure propagation, *ACM Transactions on Graphics* 24 (2005) 861–868.
- [13] N. Komodakis, G. Tziritis, Image completion using efficient belief propagation via priority scheduling and dynamic pruning, *IEEE Transactions on Image Processing* 16 (2007) 2649–2661.
- [14] Z. Xu, J. Sun, Image inpainting by patch propagation using patch sparsity, *IEEE Transactions on Image Processing* 19 (2010) 1153–1165.
- [15] M. Barni, F. Bartolini, V. Cappellini, Image processing for virtual restoration of artworks, *IEEE Multimedia* 7 (2000) 34–37.
- [16] I. Giakoumis, N. Nikolaidis, I. Pitas, Digital image processing techniques for the detection and removal of cracks in digitized paintings, *IEEE Transactions on Image Processing* 15 (2006) 178–188.
- [17] S.V. Solanki, A.R. Mahajan, Cracks inspection and interpolation in digitized artistic picture using image processing approach, *International Journal of Recent Trends in Engineering (IJRTE)* 1 (2009) 97–99.
- [18] G.S. Spagnolo, F. Somma, Virtual restoration of cracks in digitized image of paintings, *Journal of Physics: Conference Series* 249 (2010) 012059.
- [19] I. Giakoumis, I. Pitas, Digital restoration of painting cracks, in: *ISCAS 98, Proceedings of the IEEE International Symposium on Circuits and Signals*, 1998, pp. 269–272.
- [20] K. Zuiderveld, *Contrast Limited Adaptive Histogram Equalization*, Academic Press Professional, Inc., San Diego, CA, USA, 1994, pp. 474–485.
- [21] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12 (1990) 629–639.
- [22] R. Poli, G. Valli, An algorithm for real-time vessel enhancement and detection, *Computer Methods and Programs in Biomedicine* 52 (1997) 1–22.
- [23] F. Meyer, *Cytologie quantitative et morphologie mathématique*, Ph.D. Thesis, Ecole des Mines, 1979.
- [24] N. Otsu, A threshold selection method from Gray-level Histograms, *IEEE Transactions on Systems, Man and Cybernetics* 9 (1979) 62–66.
- [25] M. Elad, M. Aharon, A. Bruckstein, The k-svd: an algorithm for designing of overcomplete dictionaries for sparse representation, *IEEE Transactions on Signal Processing* 54 (2006) 4311–4322.
- [26] L. Lam, S. Lee, C. Suen, Thinning methodologies—a comprehensive survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14 (1992) 869–885.
- [27] S.Z. Li, *Markov Random Field Modeling in Computer Vision*, Springer-Verlag, 1995.
- [28] T. Ruzic, A. Pizurica, W. Philips, Neighbourhood-consensus message passing and its potentials in image processing applications, in: *SPIE Electronic Imaging*, 2011.
- [29] J.D. Baets, *De gewijde teksten van het lam gods kritisch onderzocht, Verslagen en Mededelingen van de Koninklijke Academie voor Taal-en Letterkunde* (1961) 560–561.