3.4 Automatic Detection & Inpainting of Defaced Regions and Cracks in Heritage Monuments

Milind G. Padalkar and Manjunath V. Joshi

Abstract Historical monuments are considered as one of the key aspects of our cultural heritage. Unfortunately, due to a variety of factors the monuments get damaged. The need for preservation of cultural heritage has desiderated research on digitally repairing the photographs of damaged monuments. One may think of digitally undoing the damage to the monuments by inpainting, a process to fill-in missing regions in an image. For images of historic monuments in particular, there is a consensus to fill-up the defaced regions and cracks so that one can view these in their undamaged form. Thus, we are not talking about image restoration, but about object completion by digitally repairing defaced regions / cracks that the physical objects have. In this chapter, we discuss techniques for automatically detecting the damaged facial regions and cracks in photographs of monuments. Unlike the usual practice of manually selecting the mask for inpainting, the regions to be inpainted are automatically selected and inpainting is done using the existing algorithm. Thus, the process of digital repair using inpainting is completely automated. We also extend our work on crack detection to perform auto-inpainting in videos by making use of scale invariant feature transform (SIFT) and homography. Finally, we provide a temporal consistency measure to quantify the quality of the inpainted video.

1 Introduction

Heritage sites are essential sources of precious historical information. These are not only an inherent part of our cultural identity, but also valuable assets of archaeological,

Milind G. Padalkar
Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, India, e-mail: milind_padalkar@daiict.ac.in

Manjunath V. Joshi
Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, India, e-mail: mv_joshi@daiict.ac.in

architectural and cultural significance. However, due to factors such as weathering, misanthropy, etc., the monuments get ruined and defaced. Renovating such sites is a very sensitive activity and requires great expertise. The process of renovation not only poses danger to the undamaged monuments but may also introduce notable changes from the monument's historic existence. Also, the access to many heritage sites is restricted fearing the risk of further damage by visitors. Hence, an obvious solution that avoids the physical contact is to digitally renovate these monuments by repairing the damaged regions in a plausible manner. This task can be performed by means of inpainting [4, 25], which is a process to fill the missing regions in images. In fact, the term inpainting originates from the art of restoring damaged images in museums by professional restorers [2].

In general, inpainting is used to restore or modify the contents of an image, imperceptibly. This is done by propagating information into the missing region from its neighborhood or a similar region from a different image. Research performed in this area was initially reported in the work by Masnou and Morel [20]. Their technique connected contours of constant intensity arriving at the boundary of the region to be inpainted. On similar lines, Bertalmio et al. [2] proposed a method that not only connected the contours of constant intensity, but also enabled their plausible curving inside the inpainted region. These methods successfully propagate structure, but are unable to perform plausible propagation of texture for large missing regions. For propagation of texture, Criminisi et al. [4] proposed a method that performs patch replication using exemplars. Likewise, Pérez et al. [25] proposed a technique for filling missing regions in one image by considering gradients from a different image as source for inpainting. We consider the use of techniques proposed in [4, 25] for inpainting automatically detected regions, which we discuss later in this chapter. A comparative survey of inpainting techniques can be found in the work by Guillemot and Meur [10]. However, one may note that these techniques do not perform an automatic selection of regions that need to be inpainted. An advantage with automatic detection of a region in an image or a video along with its inpainting is that, it can be used to perform on-the-fly inpainting of heritage scenes without requiring any human intervention. This will not only create excitement among the visitors, but will also provide them with an exhilarating experience to visualize the digitally reconstructed heritage sites in parallel with its existing damaged form.

Generally, the assessment of regions to be inpainted are subjective as provided by the users, i.e., the users manually select certain regions which they feel are to be inpainted in the given image. In other words, the process of selecting the inpainting region is subjective as this depends on the users' choice. However, when looking at heritage monuments, we human beings have a consensus about the desire to view these in their undamaged form. Importantly, the facial regions like eyes, nose and lips in statues are visually the most dominant regions and a damage to them is clearly noticeable. Likewise, cracks also diminish the attractiveness of monuments. Moreover, such a damage may provide an incomplete or even wrong information about the artistic work that was carried out at heritage site. In other words, the artistic intent is lost which needs to be recovered through the process of digital inpainting. In this chapter, we discuss how such damaged region can be automatically detected in the acquired images so that their digi-

tal repair using existing inpainting techniques can be completely automated. Moreover, we extend our work on crack detection in images to automatically perform inpainting in videos and provide a method to quantify the quality of the inpainted video.

It may be argued that the identification of damaged regions in heritage images should be done under the supervision of an expert because (a) an expert could provide a better judgment about the extent and context of damage and whether it indeed needs to be inpainted and (b) an automated method may not correctly identify the damaged regions if one does not take into account the artistic intent and visual quality of the inpainted picture. However, one may note that this is an enervating and tedious task because the selection of the region to be inpainted needs to be done by careful observation in a pixel-by-pixel manner. Such a time-consuming and tiresome process may introduce human errors. Moreover, a great amount of domain knowledge is required for performing an accurate selection and inpainting of damaged regions, for which an expert may not always be available for assessing every heritage monument. This can happen especially when new historic monuments are discovered / excavated. These factors provide the motivation to completely automate the process of detection and inpainting of damaged regions. Such a process can be used to both (a) assist an expert whenever available, who may then provide inputs for performing refinement if required, (b) provide an estimate of the historic view of the monument in the absence of an expert or lack of domain knowledge. It can also be used by a heritage site surveillance system that raises an alarm whenever someone tries to deface the monuments or cracks are being developed, so that a timely corrective action can be taken.

Whenever the digital repair is being done under the supervision of an expert, the acquisition of images and videos can be performed in a controlled environment wherein the illumination changes, camera motion or movement of people in the scene can be minimal. However, it is often the case that enthusiasts, hobbyists and tourists visiting a heritage site wish to have personal memories and use their own handheld devices for acquisition of the images and videos. Providing a reconstructed view of the damaged monuments in such a scenario makes it difficult to have any control over the acquisition environment. Here, it is also not possible to have an expert's opinion for guiding the process of detecting and inpainting the damaged regions. While images contain stationary scenes, the videos typically captured by visitors at heritage sites have a moving camera. We therefore consider such videos in our work.

The contents of this chapter are based on our works in [22, 21, 36]. We discuss the detection and inpainting of the damaged eye, nose and lip regions in statues in section 2. A technique for detecting the cracked regions in is discussed in section 3 and its extension to perform automatic inpainting in videos in section 4. Our method to quantify the the inpainted video is discussed in section 5 followed by the conclusion in section 6.

2 Detecting and inpainting dominant facial regions

In this section, we discuss our method that automates the process of identifying damage to the visually dominant regions viz. eyes, nose and lips in facial images of statues and inpainting them. Here, the bilateral symmetry of face is used as a cue to detect the eye, nose and lip regions. Textons features [33] are then extracted from each of these regions in a multi-resolution framework to characterize their textures. These textons are matched with those extracted from a training set consisting of true damaged and non-damaged regions in order to perform the classification. The repair of the identified damaged regions is then performed using the Poisson image editing method [25] by considering the best matching non-damaged region from the training set. Fig. 1 illustrates our proposed approach, the details of which are discussed below.

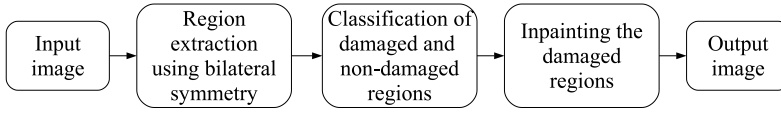


Fig. 1: Our approach to automate the repair of damaged eyes, nose and lips in statues.

The input is assumed to be a frontal face image. Here, the visually dominant regions viz. eyes, nose and lips have a common property of being bilaterally symmetrical. Motived by the work in [13], our approach uses this property as a cue for detecting the eye, nose and lip regions. Our region extraction process involves making the input image illumination invariant using the single scale retinex (SSR) algorithm [12, 34]. We then perform the edge preserving smoothing operation [26] so as to detect the regions with better accuracy. Following this, the edges are extracted to get an edge image I_e and using the same, symmetry measures $b_h(x,y)$ and $b_v(x,y)$ around each pixel location (x,y) are calculated in the horizontal and vertical directions, respectively, as follows,

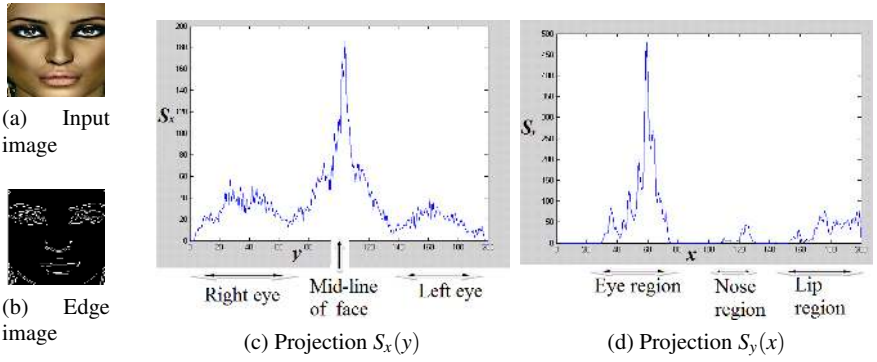
$$\begin{aligned} b_h(x,y) &= \sum_{j=1}^{\min(y,y-N)} [1(I_e(x,y-j) = I_e(x,y+j))] \text{ and} \\ b_v(x,y) &= \sum_{i=1}^{\min(x,x-M)} [1(I_e(x-i,y) = I_e(x+i,y))], \end{aligned} \tag{1}$$

where, $M \times N$ represents the size of input image and $1(condition)$ is an indicator function that outputs the value of 1 if *condition* is true, else outputs 0. The calculated symmetry measures are then used to obtain the projections S_x and S_y as follows:

$$S_x(y) = \sum_{i=1}^{M} b_h(i,y), \quad \text{and} \quad S_y(x) = \sum_{j=1}^{N} b_v(x,j), \tag{2}$$

where, y and x respectively denote the column and row being projected. The peak in projection S_x provides the mid-line about which the face is nearly symmetric, while

the peaks in projection S_y help in identifying vertical locations of the eye, nose and lip regions. This is illustrated using the example shown in Fig. 2. The regions of interest can then be extracted using appropriately sized windows around the locations of the peaks detected in projections S_x and S_y.



(a) Input image

(b) Edge image

(c) Projection $S_x(y)$

(d) Projection $S_y(x)$

Fig. 2: Extraction of the potential regions of interest using bilateral symmetry. (Reproduced from [36])

For classifying the detected regions as damaged or non-damaged we use texture as a cue. A method for modeling different texture classes having uniformity within each class has been proposed in [33]. Our work, however, deals with the images of statues at historic monuments that have natural textures with no uniformity. In such cases, it is difficult to extract any repetitive pattern at a single scale. However, irregular patterns and structures in nature have been successfully represented using fractals [5, 16]. The fractals are geometric patterns that repeat at smaller scales to produce irregular shapes and surfaces that cannot be represented by classical geometry. This motivated us to make use of a multi-resolution framework to address the issue of irregularities in natural texture at different resolutions; a property characterized by stone-work and monument surfaces. Moreover, our method automatically calculates the number of clusters required to represent the two classes that correspond to damaged and non-damaged regions, as opposed to the approach in [33] which uses fixed number of clusters for representing several texture classes.

The texture features are extracted in the form of textons, which are cluster centers in the filter response space. These textons are obtained in a multi-resolution framework by convolving the detected potential region of interest and its two coarser resolution versions with the maximum-response-8 (MR8) filter bank [33]. In order to obtain the coarser versions of the detected region, it is low-pass filtered using Gaussian filter before downsampling. The MR8 filter bank consists of 38 filters viz. edge and bar filters with 6 orientations at 3 scales along with a Gaussian and a Laplacian of Gaussian filter. Each pixel of the input region is now transformed into a vector of size 8 by considering 8 maximum responses out of the 38 filters. In other words, the maximum response for orientation of the edge and bar filters at each scale along with the response for the

Gaussian and Laplacian of Gaussian filters are recorded to obtain a vector of size 8. The K-means algorithm is then applied on these vectors to obtain the K cluster centers i.e. textons. We illustrate the process of extracting the textons for a detected nose region, with the help of Fig. 3. A similar process is independently applied to extract the textons features from the eye and lip regions.

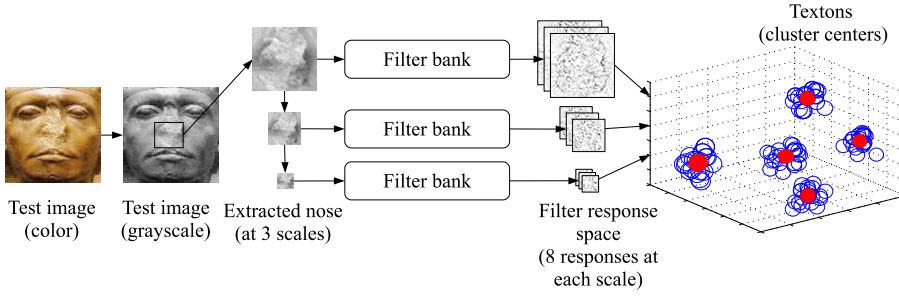


Fig. 3: Extraction of textons using the MR8 filter bank for a detected nose region.

One may note that the method proposed in [33] requires the number of clusters (K) to be known in advance. However, it may not be possible to pre-determine the number of clusters K as this is a data dependent term. In our work, we use a simple approach to estimate the optimal number of clusters. Here, we plot a two dimensional evaluation graph, where X-axis shows number of clusters (K) and Y-axis shows the pooled within cluster sum of squares around the cluster means (W_k) calculated as follows [31]:

$$W_K = \sum_{r=1}^{K} \left(\sum_{\forall i,i' \in C_r} d_{i,i'} \right), \tag{3}$$

where $d_{i,i'}$ is the squared Euclidean distance between members (i, i') of cluster C_r. Here, [31] have shown that the point at which the monotonic decrease flattens markedly provides the optimal value of K. However, if the curve is smooth, it is difficult to determine where exactly this decrease flattens. We then have a challenging task to obtain the optimal value of K. To overcome this difficulty, we attempted to best fit two straight lines to the curve using expectation-maximization (EM) algorithm. The point of intersection of the two best fit lines then gives the approximate point at which the curve starts to flatten. The projected point on the axis of number of clusters is then considered as the optimal value of K as illustrated in Fig. 4.

A process as described above is used offline for extracting the textons from a training set consisting of true damaged and non-damaged regions. Here, the textons representing a damaged eye, nose or lip region are extracted using all the training images containing the corresponding true damaged region. Likewise, textons representing the non-damaged regions are extracted using the true non-damaged regions from all the training images. We now compute the Euclidean distance between textons of the detected re-

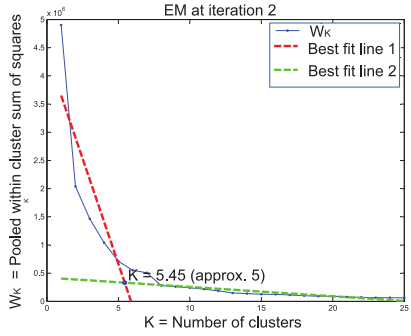


Fig. 4 Auto-selection of number of clusters K by fitting two straight lines to the data. (Reproduced from [36])

gion (viz. eye, nose or lip) in the test image and those from the corresponding true damaged and non-damaged regions of training images, to perform the classification. Here, the minimum distance criteria is used to classify the region as either damaged or non-damaged. It may be noted that for each extracted region, viz. eyes, nose and lips, the classification is performed independently. This enables the simultaneous detection of multiple damaged regions in the test image.

Once a region is identified to be damaged, we make use of the Poisson image editing method [25] to inpaint the damaged region by considering a suitable non-damaged source region. Here, if one eye is damaged we use the flipped version of the other eye (detected automatically) from the same image as the source. However, if both eyes or the nose or lip regions are damaged, we make use of the images from the training set as the source for inpainting. Here, the source selection criteria is the extent of similarity in the Euclidean space, of the undamaged region in the image containing the detected damaged region, with the true undamaged regions in the training set images. However, if all the detected regions in an image are damaged, then the source regions need to be provided manually. Our method is summarized in Table 1.

We now discuss the results of our experiments conducted on a database consisting of 40 facial images of Egyptian statues having damaged and non-damaged regions, downloaded from the Internet [9]. The spatial resolution of the images is adjusted such that all images are of the same size. A mean correction is applied to the images so that, they have the same average brightness. Training for the eye, nose and lip regions was done independently. For training we have used 10 images each for damaged and non-damaged regions. Testing was carried out on all the images from the database including those used for training.

The results using our approach are shown in Figs. 5–8. The detection and inpainting of a damaged nose is shown in Fig. 5, where the source used for inpainting is an image from the training set containing an undamaged nose. In Fig. 6, the reflected version of the non-damaged left eye has been used to inpaint damaged right eye. However, in Fig. 7 since both eyes are damaged, an image from the training set containing non-damaged eyes is used as the source for inpainting. Note that the criteria used for selecting the source is similarly of the non-damaged regions in the test image with the corresponding regions in the images from the training set. In Fig. 8, we show a result where our

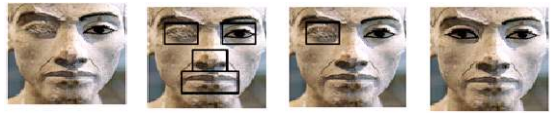
Table 1: Summary of our approach to automate the repair of damaged eyes, nose and lips in statues.

1: Make the input image illumination invariant using SSR algorithm [13] and perform edge preserving smoothing [26].
2: Extract the edges to get image I_e and calculate the symmetry measures $b_h(x,y)$ and $b_v(x,y)$ using Eq. (1).
3: Calculate the projections S_x and S_y using Eq. (2) to extract the eye, nose and lip regions.
4: Consider one detected region at a time and extract the corresponding texton features by: (a) obtaining the MR8 filter responses [33] using the detected region along with its two coarser resolutions and (b) clustering the filter responses into K clusters by auto-selecting K as shown in Fig. 4.
5: Textons extracted offline using steps 1–4 from the known damaged and non-damaged regions of images in a training set are now compared with the extracted textons from the detected region of the given test image.
6: The detected region is identified as either damaged or non-damaged based on the nearest neighbor criteria of the compared textons.
7: Repeat steps 4–6 for each detected region independently.
8: Detected damaged regions are inpainted using the method in [25] by considering a suitable source region as follows: (a) if only one eye is damaged, use the other eye as the source. (b) if both eyes or other regions are damaged, use a corresponding non-damaged region from the training set image as the source. The training set image is selected based on its similarity with the non-damaged regions in the given image. (c) if all the detected regions are damaged then manually provide the source regions.

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

Fig. 5: Detecting and inpainting a damaged nose; (a) input image, (b) extracted potential regions of interest, (c) detected damaged nose, (d) inpainted nose using the source image (e). (Reproduced from [22])

method fails to detect the damaged nose. Here, the input image contains the nose region



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

Fig. 6: Detecting and inpainting a damaged eye; (a) input image, (b) extracted potential regions of interest, (c) detected damaged eye, (d) inpainted eye. (Reproduced from [22])

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

Fig. 7: Detecting and inpainting damaged eyes; (a) input image, (b) extracted potential regions of interest, (c) detected damaged eyes (d) inpainted eyes using the source image (e). (Reproduced from [22])

(a) (b) (c)

Fig. 8: Failure case; (a) input image, (b) extracted potential regions of interest, (c) damaged nose is incorrectly classified as undamaged. (Reproduced from [22])

having a small amount of damage, due to which the corresponding textons match those of the non-damaged nose regions from the training set. This is caused by the extracted statistics of the damaged and non-damaged regions. Thus, among the extracted potential regions of interest shown in Fig. 8b, the damaged nose is incorrectly classified as undamaged and is therefore undetected in Fig. 8c.

We now discuss the performance of our method of automatic detection of facial regions and inpainting by considering the ground truth from the inputs provided by the volunteers. Performance evaluation is done in terms of the standard recall and precision metrics defined as: Recall= $\frac{|Ref \bigcap Dect|}{|Ref|}$ and Precision= $\frac{|Ref \bigcap Dect|}{|Dect|}$. Here, *Ref* are the regions declared to be damaged or undamaged by volunteers and *Dect* are the regions detected as damaged or undamaged by the proposed technique. From a set of 40 images, 50 regions were found to be damaged, while 50 were undamaged. Out of 50 damaged regions 47 were correctly classified, while all 50 undamaged regions were correctly classified. For source region selection, 49 out of 50 regions were correctly selected. The performance in terms of the recall and precision metrics is summarized in Table 2 which shows the effectiveness of our method.

Table 2: Performance in terms of recall and precision. ([22])

Region type	**# regions**	**Recall**	**Precision**
Damaged	50	0.9400	1.0000
Undamaged	50	1.0000	1.0000

Note that the source selection method used in our approach is not comparable with content based image retrieval (CBIR) techniques. This is because for a large damaged region, a CBIR system may not find adequate amount of non-damaged content to retrieve a good match relevant for inpainting. Although the proposed method is developed for images of statues, it can be effective for facial regions in natural images as both have same the facial characteristics. Thus, we have presented a texture based approach to automatically detect the damaged regions in facial images of statues for performing their digital repair using an existing inpainting technique. The results show that these regions can be effectively repaired. Here, we have addressed the repair of specific regions viz. the facial regions of statues in heritage monuments. However, damage like cracks in the non-facial regions of the monuments also diminishes their attractiveness. We address this in the following section 3 wherein we present a technique for automating the digital repair of cracks in heritage monuments.

3 Cracks detection and inpainting

Cracks are typically characterized by dark areas in an image. These can be easily identified by human beings but pose difficulty to computers. In trivial cases, simple thresholding is sufficient for detecting the cracks. However, in general, the subtle variations in pixel intensities make the detection of cracks a challenging task. Our crack detection approach is shown in Fig. 9 the details of which are discussed below.

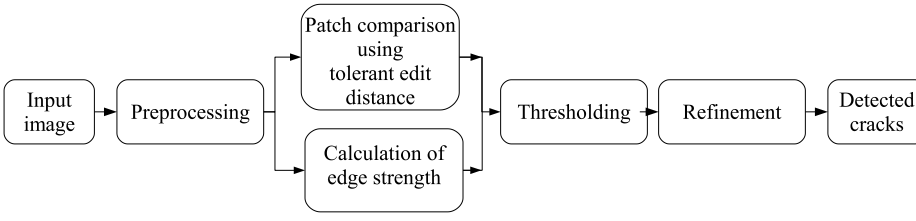


Fig. 9: Our approach for crack detection.

Preprocessing
For a given input image of size $M \times N$ we perform a pre-processing step by considering its intensity normalized version I_0. Since the cracked regions are dark, the low intensity pixels are more likely to be part of a crack. We construct a weight matrix I_w from I_0 such that dark pixels have higher weights given by,

$$I_w(x,y) = \exp(-I_0(x,y)), \tag{4}$$

where (x,y) denote the pixel coordinates. The weights in I_w are multiplied to the corresponding pixels in I_0 and the resulting image is eroded to obtain I_v. The erosion oper-

ation is performed so that the narrow dark regions grow sizeable for proper detection, which may otherwise remain undetected during further processing.

Patch comparison using tolerant edit distance

After the preprocessing step, we consider patches of size $m \times n$ in I_v and its right and bottom neighbors, and determine their similarity. Here, a patch Φ_p at pixel p with coordinates (x,y) in the image I_v consists of pixels with coordinates (X,Y), such that $X = x,\ldots,x+m-1$ and $Y = y,\ldots,y+n-1$ as shown in Fig. 10. For patch Φ_p, the right and bottom non-overlapping adjacent patches are Φ_r and Φ_s at pixels $r = (x,y+n)$ and $s = (x+m,y)$, respectively. Let the pixels of patches Φ_p, Φ_r and Φ_s be rearranged using lexicographical ordering to form vectors v_p, v_r and v_s, respectively. We then mea-

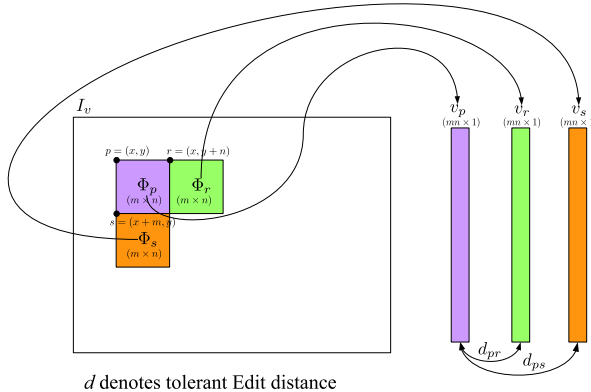


Fig. 10 Patch comparison.

sure the patch-similarity by calculating the tolerant edit distance (tED) d_{pr} and d_{ps}, respectively, between the pairs v_p, v_r and v_p, v_s, the average of which is assigned to the pixel p, i.e., $I_{tED}(p) = \frac{d_{pr}+d_{ps}}{2}$. The tED is calculated using the edit distance calculation method described in [35] and considering pixel values within a tolerance value δ_t to be equivalent. It is calculated for all the patches for which there exist both left and bottom non-overlapping adjacent patches. The calculated tED values are used to form an image I_{tED}, which, when multiplied with an edge strength image makes it easier to detect the cracked regions.

Calculation of edge strength

Since the cracked regions are distinct from their neighboring regions, these exhibit higher edge strengths. In order to give preference to patches having higher edge strengths, we generate an image I_g consisting of normalized gradient magnitudes from the preprocessed image I_v. Now, by convolving the image I_g with horizontal, vertical, diagonal and anti-diagonal line filters of size 3×3, the maximum response at each pixel is recorded to create an image I_m. The image I_m is then updated by discarding the low responses followed by morphological closing to detect the connected components. The gradient magnitude image I_g is now updated using the updated image I_m, such that, the highest gradient magnitude within each connected component is assigned to all the pix-

els within the respective component. Updating I_g in this manner enables us to assign a unique edge strength value to distinct components. The edge strength image I_e is now constructed by taking the normalized sum of I_g and I_w. We now multiply these edge strengths with the corresponding tED, to get the weighted tED image I_{tw}.

In order to fill the gap between the boundaries, a morphological closing operation is applied on I_{tw}, with the size of the structuring element depending on the size of the preprocessed image I_v. The morphologically closed image I_{tw} is now multiplied with the resized version of the weight matrix I_w to obtain an intermediate image I_{wc}. In order to assign unique values to different objects for segmentation in image I_{wc}, we employ the method used earlier for updating the gradient magnitude image I_g described in the previous paragraph. Thus, by convolving the intermediate image I_{wc} with the 3×3 line filters, thresholding the maximum response image and applying the morphological closing operation, we obtain the image I_c of size $(\frac{M}{m} - 1) \times (\frac{N}{n} - 1)$, in which the connected components have unique values.

Thresholding

Higher the value of a region in I_c, more likely it is to be a crack. Thus, the regions with values lower than a threshold T need to be discarded. Let V denote the array consisting of k unique values in I_c arranged in ascending order. Then, inspired by the threshold selection method for matching features of the scale invariant feature transform (SIFT) given in [17], we estimate the threshold $T := V[i]$ such that $(\frac{V[i]}{V[i+1]}) \geq (\frac{V[i-1]}{V[i]}), i = 1, \ldots, k$. The image I_c is now updated by setting values less than T to zero. Each pixel in I_c corresponds to an $m \times n$ overlapping patch in I_v. We obtain an initial detection image I_1 which is of the same size as that of I_v by copying pixels values from I_c to corresponding patches in I_1. A second morphological closing operation is now applied on the binary image I_1 in order to avoid splitting of the detected region. Note that the image I_1 gives a good estimate of the cracked regions, however, few pixels of the cracked regions which are similar to the surroundings may still remain undetected. Therefore, a refinement step is required to achieve a more accurate detection.

Refinement

Interactive image segmentation techniques based on curve evolution [3], graph-cut optimization [27] have been widely used for accurately detecting roughly marked objects. For refining I_1, we use the method based on active contours proposed in [3], to obtain the final crack detected binary image I_f, an example of which is shown in Fig. 11b. In order to justify the suitability of the proposed method for inpainting, we also show the inpainted result in Fig. 11c obtained using the method proposed in [4]. The steps involved in this approach are given in Table 3.

In our experiments for crack detection, we show the results for three input images of size 684×912 captured from the world heritage site at Hampi, Karnataka in India. We considered patches Φ_p of size 3×3 and tolerance value for tED calculation as $\delta_t = 10$ in all our experiments. In Fig. 12 we show a comparison our results with those obtained using the techniques in [1, 32, 23]. It may be noted that the results for the technique in [1] are obtained after fine-tuning the parameters. Here, we also show the regions

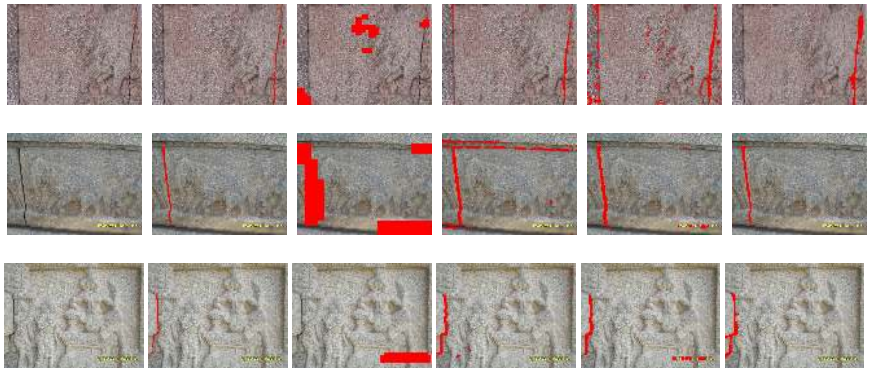
(a) Input image (b) Detected crack (I_f) (c) Inpainted crack

Fig. 11: Automatic detection and inpainting of cracks. (Reproduced from [21])

Table 3: Steps of our approach for crack detection.

1:	Obtain the weight image I_w from the intensity normalized input image I_0 using Eq. (4).
2:	Use I_0 and I_w to obtain the preprocessed image I_v.
3:	Compute the tolerant Edit Distance image I_{tED} by comparing all the non-overlapping adjacent patches in I_v for similarity as shown in Fig. 10.
4.	Generate an image I_g consisting of the normalized gradient magnitudes from I_v.
5.	Convolve I_g with horizontal, vertical, diagonal and anti-diagonal line filters of size 3×3 and record the maximum response at each pixel to create an image I_m.
6.	Discard the low responses in I_m and perform morphological closing to detect the connected components.
7.	Update I_g using I_m such that the highest gradient magnitude within each connected component is assigned to all the pixels within the respective component.
8.	Take the normalized sum of I_g and I_w to construct image I_e.
9.	Multiply I_e with I_{tED} to get the weighted tED image I_{tw} and apply morphological closing to fill gaps.
10.	Multiply I_{tw} with the resized version of the weight matrix I_w to obtain an intermediate image I_{wc}.
11.	Perform steps 5–7 considering I_{wc} in place of I_g to obtain the image I_c having unique values for the connected components.
12.	Update I_c by setting values less than an automatically obtained threshold T to zero.
13.	Obtain the initial detection image I_1 using I_c.
14.	Refine I_1 using the method in [3] to obtain the final crack detected image I_f. The regions detected in I_f are inpainted using the technique in [4].

marked as cracks by volunteers in Fig. 12b. These are used as the ground truth for an objective comparison shown in Table 4 by considering the standard recall and precision metrics. A higher value of precision indicates that a large number of detected pixels indeed belong to the cracked regions, while a higher value of recall indicates that a large number of cracked pixels have been detected. From Table 4 we observe that our proposed method performs better crack detection. In section 4 we extend this approach to perform automatic inpainting in the videos captured at heritage sites.



(a) (b) (c) (d) (e) (f)

Fig. 12: Comparison of crack detection techniques: (a) input image; (b) manual selection by volunteers; detection results – (c) Amano [1], Turakhia et al. (d) [32], (e) Padalkar et al. [23], (f) proposed method. (Reproduced from [21])

Table 4: Comparison in terms of recall and precision for images shown in Fig. 12. ([21])

Input	Amano [1]		Turakhia et al. [32]		Padalkar et al. [23]		Proposed approach	
	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision
Image 1	0.046	0.068	0.749	0.678	0.863	0.392	0.840	0.997
Image 2	1.000	0.579	0.974	0.974	0.987	0.857	0.985	0.996
Image 3	0.000	0.000	0.988	0.887	0.953	0.743	0.990	1.000

4 Automatic inpainting in videos

To extend the approach of crack detection and inpainting to videos, it would be intuitive to think of performing frame-by-frame detection and inpainting. This abstraction, however, in practice is a long-drawn-out process as it does not exploit the inter-frame redundancy. In the proposed video inpainting method, we consider pairs of temporally adjacent frames and use the homography [11] to track the cracked regions from one frame to another.

The first video frame is initially considered as the reference frame, which is later updated based on the camera movement. The cracked regions are detected in reference frames using the method described in section 3 and then tracked to subsequent frames. Once again, the detected cracks are inpainted in the reference frames using the technique proposed in [4] and then mapped to the tracked regions in the subsequent frames. Since videos typically captured by tourists at heritage sites have a moving camera, we consider such videos in our work. Note that here the inpainting of video frames cannot be done by simply copying objects visible in other frames, as done in [24]. This is because, an object to be inpainted in one frame also needs to be inpainted in other frames as well, which mandates the use of a hole filling technique. Fig. 13 shows our proposed approach for detecting and inpainting the cracked regions in videos. In what follows,

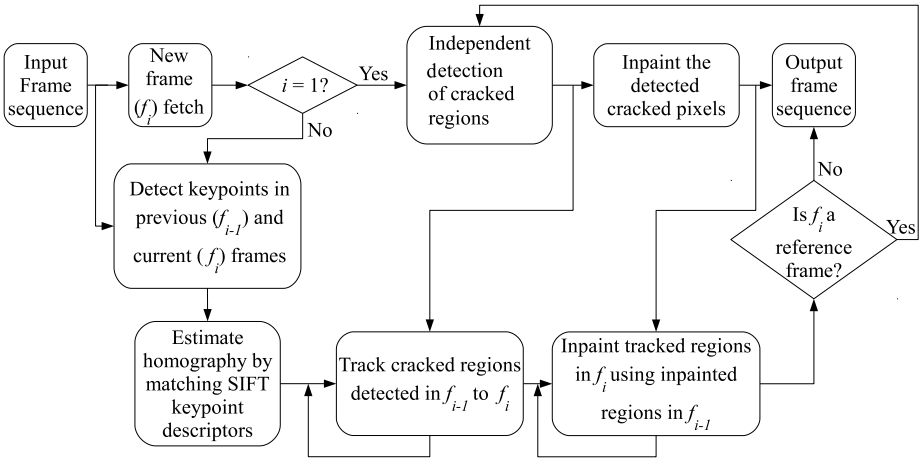


Fig. 13: A complete framework for crack detection and inpainting in videos.

we briefly describe the important stages involved in automating the inpainting in videos viz. (1) estimation of homography, (2) reference frame detection and (3) tracking and inpainting the cracked regions across frames.

Since the videos captured at heritage sites usually contain nearly planar rigid objects / scene with a moving camera, we can consider the video frames to be images captured from different viewpoints. Hence, the transformation between these frames can be represented by a homography [11, 15], which we estimate by extracting keypoints and matching the scale invariant feature transform (SIFT) descriptors [17]. The estimation of homography is done by a random sampling consensus (RANSAC) [8] of the matching keypoints at locations (x_1, y_1) and (x_2, y_2) in the two frames, obeying the relationship $[x_2, y_2, 1]^T = H[x_1, y_1, 1]^T$, where H is a 3×3 non-singular matrix representing the homography.

While capturing the video with a moving camera, new cracked regions may appear. Therefore, an independent crack detection needs to be performed quasi-periodically depending the camera movement. An intuitive way to quantify the camera motion is to calculate the magnitude of translation. We use the method proposed in [7, 19] to calculate the magnitude of translation from the estimated homography matrix. If the magnitude of translation is above a threshold δ_r, then the incoming frame f_i is declared to be a reference frame. For a pair of temporally adjacent frames f_{i-1} and f_i, the locations (x_i, y_i) of cracked pixels in f_i can be tracked from the frame f_{i-1} using the corresponding locations (x_{i-1}, y_{i-1}) as follows,

$$\begin{bmatrix} x_i' \\ y_i' \\ z_i' \end{bmatrix} = H_i \begin{bmatrix} x_{i-1} \\ y_{i-1} \\ 1 \end{bmatrix}, \tag{5}$$

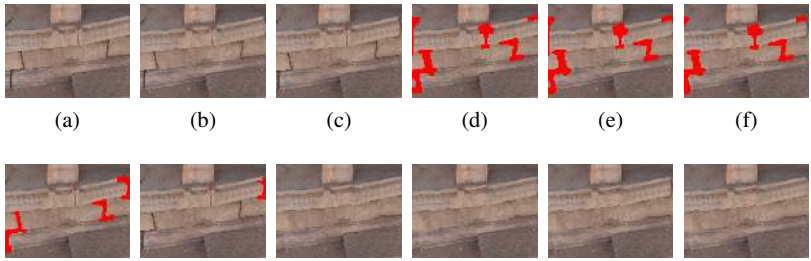


Fig. 14: Inpainting a newly appearing reference frame f_i. (a),(b),(c) show frames f_{i-2}, f_{i-1} and f_i, respectively; the cracked regions corresponding to (a),(b),(c) tracked from detected cracks in previous frames are shown in (d),(e),(f); independent crack detection in f_i is shown in (g), while the newly appearing cracked pixels in (g) with respect to (f) are displayed in (h); the inpainted versions of f_{i-2}, f_{i-1}, f_i obtained by copying pixels from respective previous inpainted frames are shown in (i),(j),(k); final inpainted version of f_i obtained after inpainting the newly detected pixels is shown in (l). Note that the crack visible near the right side in (k) is filled in (l) by independently inpainting pixels shown in (h). (Reproduced from [36])

where $(x_i^{'}, y_i^{'}, z_i^{'})$ are the homogeneous coordinates for the point (x_i, y_i) such that $x_i = \frac{x_i^{'}}{z_i^{'}}$, $y_i = \frac{y_i^{'}}{z_i^{'}}$ and H_i denotes the homography between frames f_{i-1} and f_i. In subsequent frames, the cracks detected in the reference frame are tracked using the estimated homography between these frames. Also, the inpainted pixels from the reference frame are copied to the tracked regions. Here, any newly appearing regions are independently inpainted. The tracking of cracked regions detected in a reference frame to subsequent frames, along with their inpainting is illustrated in Fig. 14, where we choose the threshold for translation δ_r as 5. In the following section 5 we discuss a novel measure to quantify the quality of the inpainted video.

5 Measuring the temporal consistency of the inpainted video

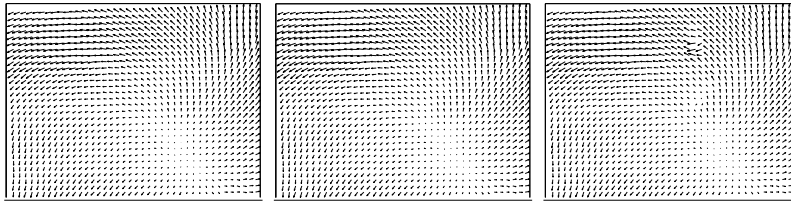
The quality of a processed image / video can be quantified in terms of a metric by comparing the image / video available from an undistorted source. However, in some applications the original source or reference is not available for comparison. Video inpainting is one such application in which missing regions in frames need to be filled up and hence a reference i.e., the original video for comparison is not available. In such a case, the objective quantification of the video quality is based on no-reference video quality assessment (NR VQA) metrics viz. blockiness, bluriness and sudden local changes [6, 29, 28].

In an application like video inpainting, the information between temporally adjacent frames in the unprocessed video can be extracted. In our method, we make use of this temporal information to quantify the quality of the processed i.e. inpainted video. The temporal consistency measure between two videos that we introduce here indicates similarity between two videos in terms of the optical flow. Intuitively, to obtain a temporally plausible inpainted video, the optical flow of the input video should be maintained after inpainting, provided the objects to be inpainted are stationary. In other words, the optical flow between every pair of temporally adjacent frame in input and corresponding pair of frames in the inpainted video should be similar. The inpainting of only the stationary object is a valid assumption for inpainting videos of heritage monuments. With this cue, the optical flow between every pair of adjacent frames in both input as well as inpainted video can be estimated and used to quantify the quality of the inpainted video. An example of temporal consistency in terms of optical flow is shown in Fig. 15. The optical flow can be estimated by using the classic method proposed in [18].

Let $L_0(i)$ and $D_0(i)$ be the magnitude and direction, respectively, of the optical flow between the i^{th} and $i+1^{th}$ frames in the input video. Similarly, let $L_1(i)$ and $D_1(i)$ be the magnitude and direction, respectively, of the optical flow between the i^{th} and $i+1^{th}$ frames in the inpainted video. Both L and D are vectorized using lexicographical ordering. Then, the temporal consistency between i^{th} and $i+1^{th}$ frames is given by the Pearson's correlation coefficient $r(i)$ as follows [14].

$$r(i) = \frac{1}{l-1} \sum_{j=1}^{l} \frac{(K_0^j(i) - \bar{K}_0)(K_1^j(i) - \bar{K}_1)}{\sigma_0(i)\sigma_1(i)}, \tag{6}$$

where K can be the vector of magnitude (L) or direction (D), $\bar{K}$ and σ are mean and standard deviation of K respectively, and l represents the length of K. The value $r(i) = +1$ indicates perfect positive correlation, $r(i) = -1$ indicates perfect negative correlation while $r(i) = 0$ for un-correlated data. The average value of r for all the pairs



(a) (b) (c)

Fig. 15: Optical flow between a pair of temporally adjacent frames in (a) input video, (b) auto-inpainted video using proposed method, (c) video generated by auto-inpainting every frame independently. The optical flow in (a) and (b) appear to be similar while some haphazard orientations in the optical flow are observed in (c). (Reproduced from [21])

of adjacent frames then gives the temporal consistency between the input and the inpainted videos. A higher average value of r indicates higher temporal consistency.

To demonstrate the effectiveness of our method to perform automatic inpainting in videos, we now present an objective comparison with frame-by-frame inpainting in Table 5. Here, "Video1"–"Video4" represent the videos captured by us from the world heritage site at Hampi, Karnataka in India, while "Video5" denotes a video of the McConkie Ranch Petroglyphs near Vernal, Utah in USA, uploaded by an enthusiast on the popular streaming site YouTube [30]. All the videos have a spatial resolution of 270×360.

Table 5: Comparison of proposed for inpainting in videos with frame-by-frame inpainting, in terms of temporal consistency in optical flow's direction (**A**) and magnitude (**B**) and the NR VQA measures viz. blockiness (**C**) & bluriness (**D**) [6] and sudden local change (**E**) [29, 28]. ([21])

	Proposed method					**Frame-by-frame auto-inpainting**				
Video	**A**	**B**	**C**	**D**	**E**	**A**	**B**	**C**	**D**	**E**
Video1	0.9529	0.7501	0.1125	5.1020	1.0737	0.5064	0.2496	0.1296	5.1073	1.3126
Video2	0.6671	0.9604	0.1034	4.1261	1.5459	0.1978	0.4148	0.1270	4.2057	1.9463
Video3	0.9979	0.5424	0.2975	4.3382	1.2908	0.1862	0.6134	0.2292	4.3666	1.5322
Video4	0.8173	0.9678	0.1453	4.6306	1.8454	0.2009	0.8946	0.1473	4.7223	2.0858
Video5	0.5821	0.9654	0.1582	3.1264	2.0559	0.2301	0.9381	0.1662	3.1586	2.7768

A video with blocking artefacts and blur has higher value for the blockiness and bluriness measures [6]. For a temporally plausible video, the sudden local change [29, 28] is less, while the temporal consistency measure has a higher value. From the objective comparison shown in Table 5, we observe that the proposed method performs better in terms of blockiness, sudden local change and temporal consistency.

6 Conclusion

In this chapter, we have presented techniques for automatic detection of damaged dominant facial regions in statues and cracks in heritage monuments for their digital repair. In our first approach, a bilateral symmetry based method is used to identify the eyes, nose and lips. Here, texton features are extracted from each of these regions in a multi-resolution framework to characterize the textures of damaged and non-damaged regions. These textons are matched with those extracted from a training set of true damaged and non-damaged regions for detecting the damaged ones which are then inpainted with the help of suitable source regions. To automate the digital repair of non-facial regions we have also presented a technique for crack detection. Here, by comparing non-overlapping patches using the tolerant edit distance measure, our method initially localizes the cracks. Further, using an active contour based segmentation, the results are refined to accurately detect the cracks. Based upon this, we build up a method that automatically detects and inpaints cracked regions in videos captured at heritage sites. For

an incoming video frame, the homography estimated with respect to its previous frame is used to track and inpaint the cracked regions while the newly appearing crack pixels are independently inpainted. Different measures are used in quantifying the quality of inpainted videos.

Acknowledgements This work is a part of project sponsored by Department of Science and Technology (DST), Govt. of India (Grant No: NRDMS/11/1586/2009/Phase-II). The authors would like to thank the co-authors of [22] for their help in developing the contents of section 2. The authors are also grateful to Prof. Toshiyuki Amano, Faculty of Systems Engineering, Wakayama University, for his valuable inputs and sharing the code of his work in [1].

References

1. Correlation based image defect detection. In: Proceedings of the 18th International Conference on Pattern Recognition, IEEE Computer Society, Washington, DC, USA, ICPR '06, vol 01, pp 163–166
2. Bertalmio M, Sapiro G, Caselles V, Ballester C (2000) : Image inpainting. In Proc. 27th Annual Conf. Computer Graphics and Interactive Techniques, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, pp 417–424
3. Chan T, Vese L (2001) : Active contours without edges. IEEE Trans Image Processing 10(2):266–277
4. Criminisi A, Pérez P, Toyama K (2004) : Region filling and object removal by exemplar-based image inpainting. IEEE Trans Image Processing 13:1200–1212
5. Emerson C, Lam N, Quattrochi D (1999) : Multi-scale fractal analysis of image texture and patterns. Photogrammetric Engg and Remote Sensing 65(1):51–62
6. Farias M, Mitra S (2005) : No-reference video quality metric based on artifact measurements. In: Proc. Int. Conf. Image Processing, vol 3, pp III–141–4
7. Faugeras O, Lustman F (1988) : Motion and structure from motion in a piecewise planar environment. Tech. Rep. RR-0856, INRIA
8. Fischler MA, Bolles RC (1981) : Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 24(6):381–395
9. Google Images, [Online] Available: *http* : *//www.images.google.com* (2012)
10. Guillemot C, Meur OL (2014) : Image inpainting : Overview and recent advances. IEEE Signal Processing Magazine 31(1):127–144
11. Hartley R, Zisserman A (2003) : Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, New York, NY, USA
12. Jobson DJ, Rahman Z, Woodell GA (1997) : Properties and performance of a center/surround retinex. IEEE Trans Image Processing 6(3):451–462
13. Katahara S, Aoki M (1999) : Face parts extraction windows based on bilateral symmetry of gradient direction. In: Computer Analysis of Images and Patterns, vol 1689, Springer Berlin Heidelberg, pp 834–834
14. Kenney JF (1954) : Mathematicals of Statistics. Van Nostrand
15. Kovesi PD (2005) MATLAB and Octave : functions for computer vision and image processing. Centre for Exploration Targeting, School of Earth and Environment, The University of Western Australia, available from: <http://www.csse.uwa.edu.au/∼pk/research/matlabfns /robust/ransacfithomography.m>
16. Legrand P (2009) : Local regularity and multifractal methods for image and signal analysis. In: Scaling, Fractals and Wavelets, Wiley
17. Lowe DG (2004) : Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2):91–110

18. Lucas BD, Kanade T (1981) : An iterative image registration technique with an application to stereo vision. In: Proc. 7th Int. Joint Conf. on AI, pp 674–679
19. Ma Y, Soatto S, Kosecka J, Sastry SS (2003) : An Invitation to 3-D Vision: From Images to Geometric Models. SpringerVerlag
20. Masnou S, Morel JM (1998) : Level lines based disocclusion. In: Proc. Int. Conf. Image Processing, vol 3, pp 259–263
21. Padalkar MG, Joshi MV (2015) : Auto-inpainting heritage scenes: a complete framework for detecting and infilling cracks in images and videos with quantitative assessment. Machine Vision and Applications 26(2):317–337
22. Padalkar MG, Vora MV, Joshi MV, Zaveri MA, Raval MS (2013a) : Identifying vandalized regions in facial images of statues for inpainting. In: New Trends in Image Analysis and Processing–ICIAP 2013, Lecture Notes in Computer Science, vol 8158, Springer Berlin Heidelberg, pp 208–217
23. Padalkar MG, Zaveri MA, Joshi MV (2013b) : SVD based automatic detection of target regions for image inpainting. In: Park JI, Kim J (eds) Computer Vision - ACCV 2012 Workshops, Springer Berlin Heidelberg, Lecture Notes in Computer Science, vol 7729, pp 61–71
24. Patwardhan KA, Sapiro G, Bertalmío M (2007) : Video inpainting under constrained camera motion. IEEE Trans Image Processing 16(2):545–553
25. Pérez P, Gangnet M, Blake A (2003) : Poisson image editing. ACM Trans Graphics 22(3):313–318
26. Perona P, Malik J (1990) : Scale-space and edge detection using anisotropic diffusion. IEEE Trans Pattern Analy Machine Intell 12:629–639
27. Rother C, Kolmogorov V, Blake A (2004) : "grabcut": Interactive foreground extraction using iterated graph cuts. ACM Trans Graphics 23(3):309–314
28. Saad M, Bovik A, Charrier C (2014) : Blind prediction of natural video quality. IEEE Trans Image Processing 23(3):1352–1365
29. Saad MA, Bovik AC (2012) : Blind quality assessment of videos using a model of natural scene statistics and motion coherency. In: Asilomar Conference on Signals, Systems, and Computers, pp 332–336
30. Scholes S (2011) : Mcconkie ranch petroglyphs near vernal, utah. URL `https://www.youtube.com/watch?v=jmewuqEXTK8`, [Accessed 01 Sept. 2014]
31. Tibshirani R, Walther G, Hastie T (2001) : Estimating the number of clusters in a data set via the gap statistic. Journal of Royal Stat Soc, B 63(2):411–423
32. Turakhia N, Shah R, Joshi M (2012) : Automatic crack detection in heritage site images for image inpainting. In: Eighth Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP), p 68
33. Varma M, Zisserman A (2002) : Classifying images of materials: Achieving viewpoint and illumination independence. In: 7th European Conference on Computer Vision (ECCV 2002), pp 255–271
34. Štruc V, Pavešić N (2011) : Photometric normalization techniques for illumination invariance, IGI-Global, pp 279–300
35. Wagner RA, Fischer MJ (1974) : The string-to-string correction problem. J ACM 21(1):168–173
36. Padalkar MG, Joshi MV, Khatri NL (2016) : Digital Heritage Reconstruction Using Super-resolution and Inpainting, Synthesis Lectures on Visual Computing, Morgan & Claypool Publishers