

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/236117103>

# SVD Based Automatic Detection of Target Regions for Image Inpainting

Conference Paper · January 2013

DOI: 10.1007/978-3-642-37484-5\_6

---

CITATIONS

4

READS

302

3 authors:



Milind Padalkar  
Istituto Italiano di Tecnologia

12 PUBLICATIONS 20 CITATIONS

[SEE PROFILE](#)



Mukesh Zaveri  
Sardar Vallabhbhai National Institute of Technology

179 PUBLICATIONS 1,022 CITATIONS

[SEE PROFILE](#)



Manjunath V. Joshi  
Dhirubhai Ambani Institute of Information and Communication Technology

94 PUBLICATIONS 943 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



M.Tech Dissertation [View project](#)



Point Target Detection and Tracking (Sponsored by IRDE, DRDO @IIT Bombay) [View project](#)

# SVD Based Automatic Detection of Target Regions for Image Inpainting

Milind G. Padalkar<sup>1</sup>, Mukesh A. Zaveri<sup>2</sup>, and Manjunath V. Joshi<sup>1</sup>

<sup>1</sup> Dhirubhai Ambani Institute of Information and Communication Technology, India

[{milind.padalkar,mv.joshi}@daiict.ac.in](mailto:{milind.padalkar,mv.joshi}@daiict.ac.in)

<sup>2</sup> Sardar Vallabhbhai National Institute of Technology, Surat, India

[mazaveri@coed.svnit.ac.in](mailto:mazaveri@coed.svnit.ac.in)

**Abstract.** We are often required to retouch images in order to improve their visual appearance, by removing the visual discontinuities like breaks and damaged regions. Such retouching of images may be achieved by inpainting. Current techniques for image inpainting require the user to manually select the target regions to be inpainted. Very few techniques for automatically detecting the target regions for inpainting are reported in the literature, which are suitable to detect an actual damage or alteration to the given photograph. In this paper, we propose a Singular Value Decomposition (SVD) based novel technique for automatic detection of the damaged regions in the photographed object / scene, for the purpose of digitally restoring them to their entirety using inpainting. Results on an exhaustive set of images suggest that the mask generated using the proposed technique can be suitably used for inpainting purpose to digitally restore the given images.

## 1 Introduction

Many times we need to improve the visual appearance of a given image by identifying and retouching the visual discontinuities in it. Such a task of digital restoration may be achieved using image inpainting [1–3]. Given an image and a region of interest in it, the task of an inpainting algorithm is to fill up the pixels in that region in a visually plausible manner. Digital restoration of the given images thus consists of two steps viz. (a) selection of the regions to be modified (target regions) and (b) applying a suitable inpainting algorithm on these regions .

The present techniques for inpainting based on propagation of structure [1, 4, 5] and texture [2, 3, 6], require the user to manually select the target regions. Since the manual selection of target regions may vary for different user, the inpainting results being dependent on the target region selection may also vary. By automating the target region detection process, human intervention for inpainting can be avoided. Automatic detection is also useful for reconstruction and repair of digitized 3D models that may be used for creating walk-through applications [7] and on-the-fly inpainting cameras for creating efficient immersive navigation / digital walk-through systems.

The literature reports only a few inpainting techniques that also facilitate the automatic detection of target regions [8–11]. Chang et al. [8] proposed a method to detect damage in images due to color ink spray and scratch drawing. Based on the use of several filters and structural information of damages, their method is limited to detection of color ink spray and scratch drawings. Tamaki and Suzuki [9] address the detection of visually less important string-like objects that block user’s view of a discernible scene. Their method however is restricted to the detection of only those occluding objects that are long and narrow, and contrasted in intensity with respect to the background. Amano [10] present a correlation based method for detecting defects in images. This method relies on correlation between adjacent patches for detection of defects i.e. small number of regions disobeying an “image description rule”, complied by most local regions. The method works well for detecting computer generated superimposed characters having uniform pattern.

All the above mentioned techniques are suitable for detecting actual damaged or alteration caused to a photograph. These techniques do not address the identification of damage in the objects or scenes that are photographed. To the best of our knowledge, the only method that also addresses this issue is proposed by Parmar et al. [11]. Their technique uses matching of edge based features with pre-existing templates to distinguish **vandalized** and non-vandalized regions in frontal face images of **monuments at heritage sites**. However, their inpainting results are highly dependent on the selected templates and their method is restricted to frontal face images of monuments. The template creation of both vandalized and non-vandalized regions may not be practically realizable for such images and therefore the detection process may lead to undesired results.

On a similar line, techniques for micro-crack detection in concrete can be found in [12, 13], but one may note that they require special imaging conditions. Recent attempt for **crack detection using tensor voting in pavement images** can be found in the work by Zou et al. [14]. The performance of their technique is heavily dependent on the accuracy of generation of crack-pixel binary map, that acts as an input to the tensor voting framework.

In this paper we propose a novel technique based on the use of singular value decomposition (SVD) [15], to automatically detect the target regions that are required to be inpainted in the given image. Unlike the techniques reported in the literature that detect an actual external damage or defect due to alteration of a photograph, the proposed method aims to detect damage to the photographed scenes / objects. The damaged areas appear like breaks splitting the objects, developed over a period of time due to environmental effects or due to manual destruction. Such detection followed by inpainting shall enable us to digitally restore the photographed scenes / objects in their integrality. By comparing the similarity of adjacent overlapping patches in the given image, the proposed approach generates a binary map that consists the target region and can be used as an input mask for inpainting.

## 2 Proposed Approach

Visual discontinuities like cracks / damaged regions in a photographed scene / object attract attention of the human visual system. The damaged areas appear like breaks splitting the objects, developed over a period of time due to natural calamities or due to manual destruction. The visual appearance can be improved by inpainting the given images in which the cracks / damaged regions are the targets to be filled. Such digital restoration shall enable one to view the photographed scene / object in an undamaged form. Figure 1 shows our proposed technique to automatically detect the target regions for inpainting.

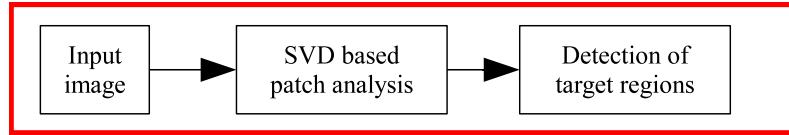


Fig. 1. Proposed approach.

(1) A natural image if split into a number of smaller non-overlapping patches, the adjacent patches may exhibit a drastic variations in intensity of corresponding pixels. Thus, even if the adjacent patches contain a single object or scene, measuring similarity between these patches would incorrectly suggest presence of visual discontinuity. On the other hand, if overlapping patches in a sliding window fashion are compared then a high amount of similarity is revealed due to the overlap of most pixels. Here, if a drastic change is encountered between the adjacent overlapping patches then comparison reveals dissimilarity. Patches exhibiting higher amount of dissimilarity are the ones that constitute the visual discontinuity. Thus, the main idea of the proposed method is to compare the overlapping adjacent patches for similarity. The average dissimilarity of the row and column adjacent patches with respect to a patch under consideration helps to reveal the amount of visual discontinuity between the patches. By using a threshold, the damaged areas can then be identified as the ones having higher dissimilarity value.

(2) Given an input image  $I$  in the RGB color space, we first transform it into HSV color space and extract the grayscale image  $I_V$  that corresponds to the intensity image. Now consider a patch  $\Phi_p$  of size  $m \times n$  at pixel  $p \in I_V$  with coordinates  $(x, y)$ . Here,  $x = 1, \dots, M - m$  and  $y = 1, \dots, N - n$ , such that  $M \times N$  represents the size of the image  $I_V$ . The elements of patch  $\Phi_p$  are rearranged to form a column vector  $v_p$  of length  $L = mn$  by using lexicographical ordering of pixels.

Now for any two adjacent pixels  $r$  and  $s$  with respective patches  $\Phi_r$  and  $\Phi_s$ , the corresponding vectors  $v_r$  and  $v_s$  may be compared for similarity. We find the similarity between the vectors  $v_r$  and  $v_s$  using the geometric interpretation of the SVD model [15] on the matrix having these vectors as its columns. By calculating the similarity between vectors of adjacent patches, we create a similarity matrix

$S$  whose elements are then compared with a threshold  $\delta$ , to detect patches having discontinuities. In the following subsection we discuss patch analysis in the SVD domain.

## 2.1 Singular value decomposition and patch analysis

We form a matrix  $A$  with the columns as  $v_r$  and  $v_s$  corresponding to patches  $\Phi_r$  and  $\Phi_s$ , and decompose it using SVD such that  $A = U\Sigma V^T$ . Here  $U$  is a  $L \times L$  matrix, the columns of which are the eigenvectors of  $AA^T$ ,  $V$  is a  $2 \times 2$  matrix consisting of eigenvectors of  $A^TA$ , and  $\Sigma$  is  $L \times 2$  matrix of singular values ( $\sigma_1 \geq \sigma_2 \geq 0$ ) at diagonals. We now reduce the size of matrices  $U$  to  $L \times 2$  and  $\Sigma$  to  $2 \times 2$ , which however does not affect the reconstruction of  $A = U\Sigma V^T$ .

Now, the rows  $w_1$  and  $w_2$  of matrix  $V\Sigma$  reflect the extent to which pixels in the two vectors  $v_r$  and  $v_s$  have a similar pattern of occurrence [15]. The similarity between columns of  $A$  corresponding to patches  $\Phi_r$  and  $\Phi_s$ , is therefore given by the cosine of angle between corresponding rows of the matrix  $V\Sigma$  as follows.

$$\cos(\theta_{rs}) = \frac{w_1 \cdot w_2}{\|w_1\| \|w_2\|} \quad (1)$$

If the vectors  $v_r$  and  $v_s$  are similar then the angle between them is small, whereas it is large when these two vectors are dissimilar. Therefore, from equation (1) it is clear that  $\cos(\theta_{rs})$  is nearer to 1 when  $v_r$  and  $v_s$  are similar, whereas it is nearer to 0 when they are dissimilar. However, here it is observed that since the complete reconstruction of  $A$  is possible using the reduced matrices  $U$ ,  $\Sigma$  and  $V$ , the angle obtained by directly considering the vectors  $v_r$  and  $v_s$  is the same as that obtained between the rows of matrix  $V\Sigma$ .

Now we consider one more vector  $v_t$  as a column of matrix  $A$ , where  $v_t$  corresponds to a patch  $\Phi_t$  which is also adjacent to patch  $\Phi_r$ . If we now decompose  $A$  using SVD, we have  $A = U\Sigma V^T$  with matrices  $U$ ,  $\Sigma$  and  $V$  of sizes  $L \times L$ ,  $L \times 3$  and  $3 \times 3$ , respectively. By discarding the smallest eigenvalue, we reduce the size of matrices  $U$  to  $L \times 2$ ,  $\Sigma$  to  $2 \times 2$  and  $V$  to  $3 \times 2$ , which now leads to an approximate reconstruction of matrix  $A$ . Such a method is widely used for image compression and noise reduction [16]. If we now consider the angle between rows of the matrix  $V\Sigma$  as in equation (1), the true extent of similarity between the corresponding columns of matrix  $A$  is still maintained. This is because, we are discarding the eigenvectors corresponding to the smallest eigenvalue. This in turn helps to calculate the true similarity even when the patches are noisy.

One may easily verify this from [the following example](#). Consider the vectors  $v_r = [1, 2, 3, 4, 3, 2, 1]^T$  and  $v_s = [4, 5, 6, 7, 6, 5, 4]^T$ ,  $v_t = [4, 4, 6, 8, 6, 4, 4]^T$ . In SVD domain representation of a matrix having these vectors as its columns, the rows of matrix  $V\Sigma$  are obtained to be  $w_1 = [6.52, 1.05]$ ,  $w_2 = [14.21, -0.89]$ , while  $\cos(\theta_{rs}) = 0.9756$  and  $\cos(\theta_{rt}) = 0.9916$ . On other hand, if we directly use the vectors  $v_r, v_s, v_t$  instead of  $w_1, w_2, w_3$ , respectively, we get  $\cos(\theta_{rs}) = 0.9734$  and  $\cos(\theta_{rt}) = 0.9807$ .

It may be noted that unlike calculating the correlation directly between the actual patches, our method performs the similarity comparison in the SVD domain. By discarding the smallest eigenvalue and the associated eigenvector, the obtained similarity values are robust to noisy patches. We now compare the overlapping patches for similarity by first creating a set  $E_p$  corresponding to pixel  $p \in I_V$  with coordinates  $(x, y)$ . The set  $E_p$  consists of pixels in the neighbourhood of  $p$  having coordinates  $(x+1, y)$  and  $(x, y+1)$ . Every patch  $\Phi_q$  at pixel  $q \in E_p$ , overlaps with the patch  $\Phi_p$ , such that  $\Phi_p$  is a  $m \times n$  patch at pixel  $p \in I_V$ . The size of patch  $\Phi_q$  is same as that of patch  $\Phi_p$ . The comparison of patches  $\Phi_q$  at pixels  $q \in E_p$ , which overlap with and are row, column adjacent to the patch  $\Phi_p$ , enables one to simultaneously capture horizontal, vertical and diagonal discontinuities. Thus, pixels having coordinates  $(x+1, y)$  and  $(x, y+1)$  are sufficient to form the set  $E_p$  in order to capture visual discontinuities. Including any more pixels in set  $E_p$  will induce redundancy leading to processing overhead.

The similarity of all the patches  $\Phi_q$  corresponding to every pixel  $q \in E_p$  with the patch  $\Phi_p$  is calculated by first arranging the vector  $v_p$  and all the vectors  $v_q$  corresponding to patches  $\Phi_q$  as columns of matrix  $A$ . After applying SVD on  $A$  and reducing the sizes of matrices  $U$ ,  $\Sigma$  and  $V$  as explained earlier, the similarity between columns of  $A$  is calculated by using the corresponding rows of matrix  $V\Sigma$  as  $w_1$  and  $w_2$  in equation (1). We now create a similarity matrix  $S$  such that its element  $S(p)$  represents the average similarity value of patch  $\Phi_p$  with overlapping patches  $\Phi_q \forall q \in E_p$  and is calculated as follows.

$$S(p) = \frac{1}{|E_p|} \sum_{q \in E_p} \cos(\theta_{pq}), \quad \forall p \equiv (x, y) \in I_V \quad (2)$$

Here  $|E_p|$  is the number of pixels in the set  $E_p$ .

Once the similarity matrix  $S$  is obtained, we use it to binarize  $I_V$  for detection of the target regions. If  $S(p) < \delta$ , we declare the corresponding patches  $\Phi_p$  and  $\Phi_q, \forall q \in E_p$  to be significantly dissimilar. In this manner, all the elements of  $S$  are compared with threshold  $\delta$  to detect dissimilar patches, using which a binary map  $B$  is constructed as follows.

$$\begin{aligned} B(\Phi_p) &= \begin{cases} 1, & \text{if } S(p) \geq \delta, \forall p \equiv (x, y) \in I_V, \\ 0, & \text{otherwise,} \end{cases} \quad \text{and} \\ B(\Phi_q) &= B(\Phi_p), \forall q \in E_p \end{aligned} \quad (3)$$

The binary map  $B$  generated in this way has the target regions represented by value 1. The elements in  $S$  may have different values for different input images. It may be noted that if the overlapping patches in an input image have very high similarity, then the corresponding matrix  $S$  may have many elements with values nearer to 1, and therefore a high value of threshold  $\delta$  could be required for correct detection of patches having discontinuities. Also, variation in threshold value  $\delta$ , significantly changes the resulting binary map  $B$ . Therefore, **selection of the threshold  $\delta$  based on the input image** is required for correct detection of target regions, which we describe in the following subsection.

## 2.2 Selection of threshold value $\delta$

In order to select the threshold value  $\delta$  dynamically for a given image, we consider three quantities derived from the similarity matrix  $S$ , viz. the average value  $avg(S)$ , minimum value  $min(S)$  and the maximum value  $max(S)$ . Since the compared patches are adjacent and also overlap each other, they show high content similarity. Therefore, it is reasonable to assume that the values in  $S$  less than the average value  $avg(S)$  would definitely correspond to the patches having discontinuities. Thus, the lowest value that  $\delta$  may take is  $avg(S)$ .

If the difference between lowest and highest values of  $S$  is high, it would mean that the values corresponding to patches with discontinuities are spread over a wider range, while the spread is over a narrow range when the difference is small. If the values in the similarity matrix  $S$  vary in a narrow range, then the threshold value  $\delta$  that detects the patches with discontinuities, would be nearer to  $avg(S)$ . Thus, we infer that the threshold value is higher than the average value  $avg(S)$  and also depends on the minimum  $min(S)$  and maximum  $max(S)$  values of the similarity matrix  $S$ . We set an initial threshold  $\alpha$  to be an average of these three terms as given in the following equation.

$$\alpha = \frac{min(S) + max(S) + avg(S)}{3} \quad (4)$$

However, experimentally we found that a correction factor depending on the value  $\alpha$  is required for correct detection. Based on our experimentation, we arrive at the following equation that incorporates suitable correction factors to determine the threshold  $\delta$ .

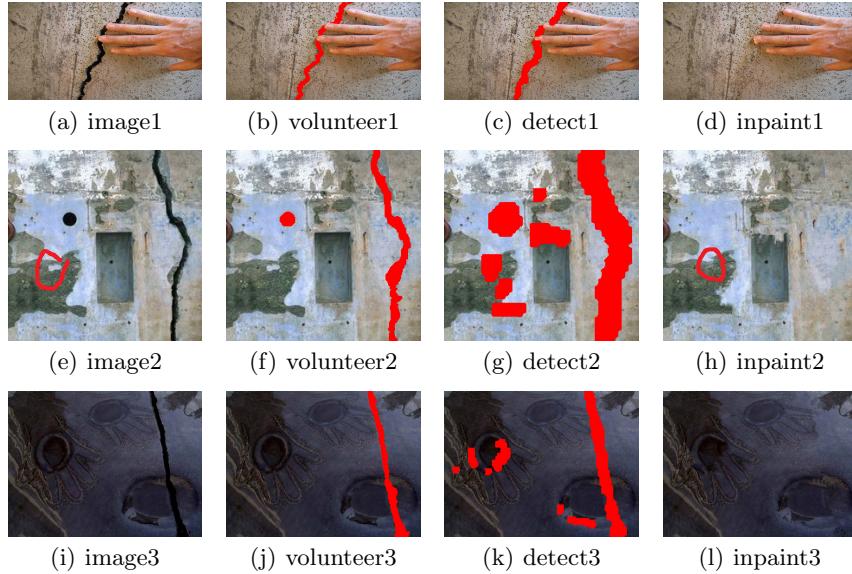
$$\delta = \begin{cases} \alpha + 0.10, & \text{if } 0 \leq \alpha < 0.90, \\ \alpha + 0.05, & \text{if } 0.90 \leq \alpha < 0.95, \\ \alpha + 0.01, & \text{if } 0.95 \leq \alpha < 0.99, \\ \alpha, & \text{if } \alpha \geq 0.99 \end{cases} \quad (5)$$

In this way, the initial threshold  $\alpha$  is calculated automatically, based on which an appropriate correction factor is added, to dynamically set the threshold  $\delta$  depending on the input image.

## 3 Experimental Results

In this section, we present the results of our proposed technique for automatic region detection, on images downloaded from the Internet [17], as well as on those captured by us. These images contain regions that appear damaged. With our proposed technique, we intend to detect such regions and generate a mask that can be suitably used to inpaint them. Inpainted results using technique proposed in [2] show the suitability of our proposed method to auto-detect target regions for inpainting.

In all our experiments we have considered patches  $\Phi_p$  of size  $3 \times 3$ . We present the results of our experiments on wall & ceiling images in figure 2, pavement

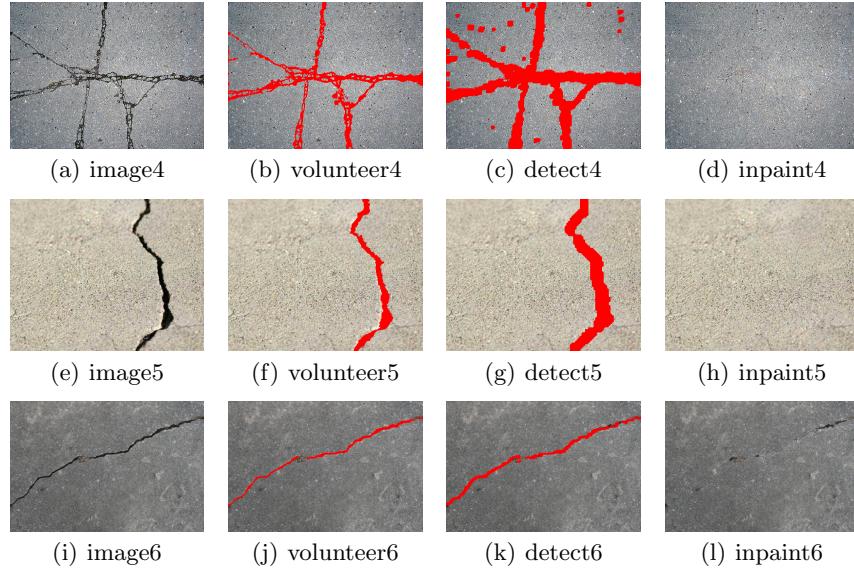


**Fig. 2.** Wall and ceiling images. (a),(e),(i) input images, (b),(f),(j) target regions selected by volunteers in red color, (c),(g),(k) detected target regions in red color, (d),(h),(l) inpainted results using technique in [2] for regions detected in (c),(g),(k).

images in figure 3. The input images for wall, ceiling and pavement were all downloaded from the Internet [17]. In order to determine the suitability of the resulting masks for the use by inpainting algorithms, we consider the popularly used recall and precision metrics [14] defined as follows.

$$\begin{aligned} \underline{\text{Recall}} &= \frac{|Ref \cap Dect|}{|Ref|}, \\ \underline{\text{Precision}} &= \frac{|Ref \cap Dect|}{|Dect|} \end{aligned} \quad (6)$$

*Ref* are the pixels declared to be in the target regions by volunteers and *Dect* are the pixels detected by the algorithm to be in the target regions. Higher value of *Precision* indicates that a large number of detected pixels indeed belong to the target region, while a higher value of *Recall* indicates that a large number of target pixels have been detected. **For a mask to be suitable for use to an inpainting algorithm, it is therefore desired to have the *Recall* value nearer to 1. On the other hand, a low value for *Precision* indicates that more pixels than desired have been detected, which only increase the area to be inpainted and is therefore acceptable.** However, if a mask with low *Recall* value is used for inpainting, information from the undetected target regions may propagate inside the detected region, leading to poor inpainting results.



**Fig. 3.** Pavement images. (a),(e),(i) input images, (b),(f),(j) target regions selected by volunteers in red color, (c),(g),(k) detected target regions in red color, (d),(h),(l) inpainted results using technique in [2] for regions detected in (c),(g),(k).

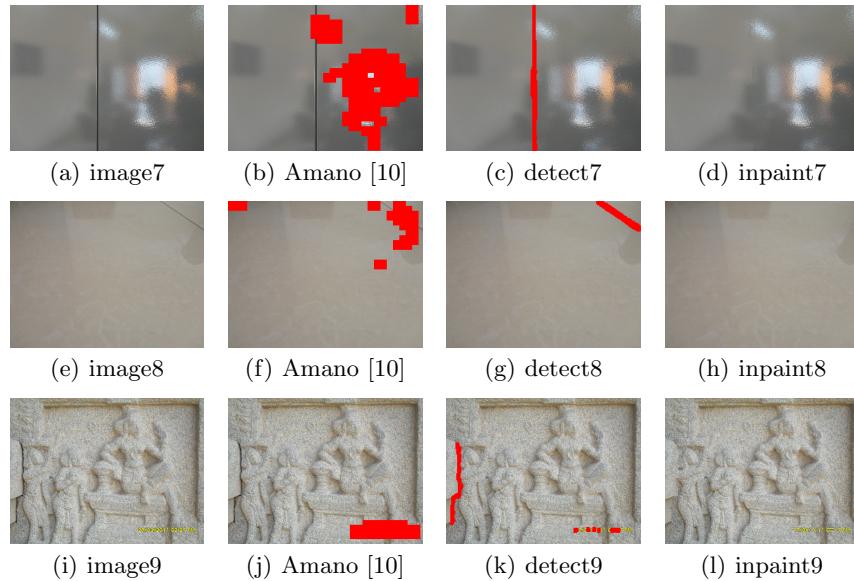
The performance in terms of *Recall* and *Precision* values for input images in figures 2 and 3 is given in table 1. The results in figures 2 and 3 show that the damaged areas with non-uniform pattern or complex texture have been successfully detected. The detection results obtained using our proposed technique are remarkably similar to the detection performed manually by volunteers and is evident from the performance table 1. We observe that *Recall* value for all the detected target regions in these images is nearer to 1. This clearly indicates that the desired target pixels have been detected. Low *Precision* values indicate that more pixels than desired have been detected. However, from the results we observe that all the desired target regions are covered by the generated mask and its use for inpainting generates visually plausible images.

To the best of our knowledge, no techniques for detection of damaged areas in the photographed scene / object, have been reported in the literature. The nearest technique with which our proposed method can be compared is [10]. The comparative results on images captured by us are shown in figure 4. It may be noted that the results for technique [10] are the best possible, obtained after fine-tuning the parameters. Whereas, the parameter  $\alpha$  in our proposed technique is dynamically calculated, depending on the input image.

From the results shown in figure 4 and performance comparison in table 2, it is clear that the desired regions are successfully detected by the proposed method. Although the technique in [10] is good for detection for an alteration to the photograph (like overlay text), our proposed method is comparatively fast

**Table 1.** Performance of the proposed technique in terms of Recall and Precision.

Input	#Target Pixels	Recall	Precision
image1	05414	0.9540	0.6702
image2	02513	0.9988	0.2290
image3	05431	0.9742	0.3708
image4	40741	0.9914	0.4445
image5	05613	0.9984	0.4772
image6	29333	0.8919	0.5781

**Fig. 4.** Images captured by us. (a),(e),(i) input images, (b),(f),(j) target regions detected using technique in [10], (c),(g),(k) detected target regions in red color, (d),(h),(l) inpainted results using technique in [2] for regions detected in (c),(g),(k).**Table 2.** Performance comparison in terms of Recall and Precision.

Input	#Target Pixels	Proposed Tech.			Tech. in [10]		
		Recall	Precision	Time (sec)	Recall	Precision	Time (sec)
image7	8217	1.0000	0.5372	4.63	0.1503	0.0093	512
image8	1353	1.0000	0.1749	4.41	0.6438	0.0260	093
image9	3494	0.9531	0.2971	4.51	0.0000	0.0000	109

and more suitable when it comes to detection of damage in the photographed scene / object.

## 4 Conclusion

In this paper we have presented a technique that can be used to generate an input mask for inpainting algorithms. By comparing overlapping patches in the SVD domain, we form a similarity matrix. The dissimilar patches detected using an image adaptive threshold are used to construct a binary map having the target regions and can therefore be suitably used as an input mask for image inpainting. The obtained results on an exhaustive set of images show that the generated masks can be indeed used to inpaint the input images. In future, we aim to extend this detection method to perform simultaneous on-the-fly detection and inpainting, which can be used to build an immersive walk-through system.

**Acknowledgement.** This work is a part of the project *Indian Digital Heritage (IDH) - Hampi* sponsored by Department of Science and Technology (DST), Govt. of India (Grant No: NRDMS/11/1586/2009/Phase-II). The authors are also grateful to Prof. Toshiyuki Amano, Department of Systems Innovation, Yamagata University, for his valuable inputs and sharing the code of his work in [10].

## References

1. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques. SIGGRAPH '00, New York, NY, USA, ACM Press/Addison-Wesley Publishing Co. (2000) 417–424
2. Criminisi, A., Pérez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing* **13** (2004) 1200–1212
3. Pérez, P., Gangnet, M., Blake, A.: Poisson image editing. In: ACM SIGGRAPH 2003 Papers. SIGGRAPH '03, New York, NY, USA, ACM (2003) 313–318
4. Grossauer, H.: A combined pde and texture synthesis approach to inpainting. *Euro. Conf. Comp. Vision* **4** (2004) 214–224
5. Shibata, T., Iketani, A., Senda, S.: Image inpainting based on probabilistic structure estimation. In: Proceedings of the 10th Asian conference on Computer vision - Volume Part III. ACCV'10, Berlin, Heidelberg, Springer-Verlag (2011) 109–120
6. Padalkar, M.G., Joshi, M.V., Zaveri, M.A., Parmar, C.M.: Exemplar based inpainting using autoregressive parameter estimation. In: Proceedings of the International Conference on Signal, Image and Video Processing. ICSIVP'12, IIT Patna, India (2012) 154–160
7. Shih, T.K., Tang, N.C., Yeh, W.S., Chen, T.J., Lee, W.: Video inpainting and implant via diversified temporal continuations. In: Proceedings of the 14th annual ACM international conference on Multimedia. MULTIMEDIA '06, New York, NY, USA, ACM (2006) 133–136

8. Chang, R.C., Sie, Y.L., Chou, S.M., Shih, T.K.: Photo defect detection for image inpainting. In: Proceedings of the Seventh IEEE International Symposium on Multimedia. ISM '05, Washington, DC, USA, IEEE Computer Society (2005) 403–407
9. Tamaki, T., Suzuki, H., Yamamoto, M.: String-like occluding region extraction for background restoration. Pattern Recognition, International Conference on **3** (2006) 615–618
10. Amano, T.: Correlation based image defect detection. In: Proceedings of the 18th International Conference on Pattern Recognition. ICPR '06, Washington, DC, USA, IEEE Computer Society (2006) 163–166
11. Parmar, C.M., Joshi, M.V., Raval, M.S., Zaveri, M.A.: Automatic image inpainting for the facial images of monuments. In: Proceedings of Electrical Engineering Centenary Conference 2011, IISc Bangalore, India (2011) 415–420
12. Ammouche, A., Riss, J., Breysse, D., Marchand, J.: Image analysis for the automated study of microcracks in concrete. Cement and Concrete Composites **23** (2001) 267 – 278 Special Theme Issue on Image Analysis.
13. Ringot, E., Bascul, A.: About the analysis of microcracking in concrete. Cement and Concrete Composites **23** (2001) 261 – 266 Special Theme Issue on Image Analysis.
14. Zou, Q., Cao, Y., Li, Q., Mao, Q., Wang, S.: Cracktree: Automatic crack detection from pavement images. Pattern Recognition Letters **33** (2012) 227 – 238
15. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science **41** (1990) 391–407
16. Wall, M.E., Rechtsteiner, A., Rocha, L.M.: Singular value decomposition and principal component analysis. In: Singular value decomposition and principal component analysis. Kluwer: Norwell (2003) 91–109
17. Google Images: <http://www.images.google.com> (2012)