

# Exemplar based Image Inpainting with Reduced Search Region

Jayesh Patel

Student,

Computer Engineering Department,  
Thadomal Shahani Engineering College,  
Mumbai University, India

Tanuja K. Sarode, Ph. D

Associate Professor,

Computer Engineering Department,  
Thadomal Shahani Engineering College,  
Mumbai University, India

## ABSTRACT

Image inpainting is a technique to recover lost data or manually remove data from an image. This process is carried out in such a way that it seems reasonable to human eyes. The lost part of the image is filled with the remaining available data of the image. Inpainting can be done using various techniques, mainly variation based, exemplar based and wavelet based. In this paper, exemplar based image inpainting techniques is used and an algorithm which improves and extends the previously proposed algorithm is presented. Instead of searching the entire image, the proposed approach has reduced the search region by searching the nearby pixels since most of the relevant information lies in neighbouring pixels which reduces time to complete inpainting. This technique is mainly used for removing objects and cracks from the image.

## General Terms

Image processing, Inpainting, Object removal.

## Keywords

Image inpainting, Exemplar, Object removal, Search region

## 1. INTRODUCTION

Digital image inpainting is a technique used for image recovery in which the missing region is filled or replaced by the available information. The main goal of image inpainting is to reconstruct an image after object removal in such a way that it seems reasonable and changes are undetectable to the human eye. Using this technique one can recover small portions such as cracks and noise from an image. Large portions can also be recovered from the image such as removing large objects or an unwanted person.

Image inpainting is mainly used for image restoration and object removal. Image restoration includes restoring old photographs and damaged film by removal of speckles, scratches, superimposed text like dates, subtitles, or publicity. Object removal includes removing unwanted objects from an image such as removing an unwanted person, tree etc. Other applications of image inpainting are object completion and video inpainting. Object completion includes completing the incomplete image or half taken image of an object and video inpainting is to apply image inpainting on each frame of the video.

In the past, this problem has been addressed by two classes of algorithms: “texture synthesis” algorithms for generating large image regions from sample textures and “inpainting” techniques for filling in small image gaps [1]. Texture synthesis focuses on the texture part of image while inpainting focuses on the structural part of image. Exemplar based image

inpainting is a combination of both of the above techniques, hence this technique can reconstruct structural as well as texture part of an image. However, structural propagation of the image is highly dependent on the filling order. The filling order is decided based on the pixel’s priority.

In image inpainting approach user selects the target region which is to be restored then algorithm automatically starts filling in that region using the available information from the source region. The algorithm starts filling automatically because in this approach user does not have to specify where to search for that information. Filling proceeds based on the pixel’s priority, higher the priority sooner the pixel gets filled. Priority of pixel is high if pixel lies on isophote lines arriving at the regions’ boundaries.

The rest of the paper is organized as follows. Section 2 presents the related work. The details of the proposed methodology are described in section 3. Finally, an inpainting results on digital images is presented in section 4.

## 2. RELATED WORK

Currently lot of researches are being explored in the field of image inpainting. The following are some technologies that are currently available and play major role in the proposed technique. Criminisi et al [1] have proposed the exemplar based inpainting approach for removing large objects from an image by combining texture synthesis and inpainting approach for removing small cracks. Anupam et al [2] have improved the traditional algorithm such that inpainting completes in lesser amount of time compared to traditional approach. The major modification includes changes in the priority function and inclusion of mean and variance in case of two patch with same MSE and reduced the search region. Cao et al [4] have proposed an algorithm which is capable of recovering complex geometric structures using Variational model and creating sketch of an image. Bertalmio et al [5] have surveyed about Image inpainting, applications and various techniques. Li et al [6] have proposed a Curvature Driven Diffusion model which improves effectiveness and linear structure propagation by rational confidence and data computing methods. Qin et al [7] have proposed an algorithm to improve the structural quality of image and recovered curve edges properly and optimized strategy to find similar patch. Sarawut et al [8] have proposed patch shifting scheme, if particular patch don’t have sufficient source region pixels or known pixels then patch is shifted such that more source region pixels come into the patch.

### 3. METHODOLOGY

In this paper, improved version of exemplar based image inpainting is proposed. The algorithm search for a matching patch of pixel  $p$  in a particular area/areas instead of searching the entire image, which will reduce computation time to great extent. The area or region selected should be near to the pixel  $p$  because most of the relevant and useful information for any pixel is located near to that pixel.

Example: suppose consider an image of size 200x200 (total 40000 pixels) and target region of 4000 pixels which is 10% of the entire image. For each pixel in the target region, MSE should be calculated for all the pixels in the source region, which is of size 36000 pixels (40000-4000) initially and increases gradually as filling proceeds. Hence the computation is more than 150 million MSE for the entire image. If this is the case for small image, then the approximate estimation of number of computations for larger images will be in billions and trillions.

Hence, by reducing the size of the search region, the number of computations can be reduced, thereby reducing computation time.

#### 3.1. Algorithm:

- Set the patch size, the default patch size used is 9x9.
- Maintain Location matrix  $L_{h \times w}$  where  $h$  and  $w$  are height and width of the image.

A location matrix  $L_{h \times w}$  of size  $h \times w$  is maintained to store the location of pixel from where it is copied into the target region. For example if the target region pixel at location  $(x_1, y_1)$  is filled with a source region pixel  $(x_2, y_2)$  then  $L(x_1, y_1) = x_2, y_2$ .

Location of the target region pixels in location matrix will change from empty to some location as filling proceeds. But the location of the source region pixels in location matrix will remain empty.

- Set the window size, the default window size used is 3x3.

#### 1. Select an Image for Inpainting.

The image can be RGB image or Grey scale image.

#### 2. Initialize the target region ( $\Omega$ ) [2].

User will manually select the region to be inpainted. Region can be an object to be removed, cracks in the image etc.

#### 3. Find the mask of the image.

If pixel  $(i, j) \in \Phi$  it implies that the pixel belongs to source region ( $\Phi$ ) and mask  $(i, j)$  is set to 1.

If pixel  $(i, j) \in \Omega$  it implies that the pixel belongs to target region ( $\Omega$ ) and mask  $(i, j)$  is set to 0.

#### 4. Find boundary pixel of a target region ( $\Omega$ ).

Finding boundary of target region is necessary because further processing is done on the boundary pixels only.

#### 5. Find the priority $P(p)$ for all pixel $p \in$ boundary.

The formula for finding priority is given in equation (1)

$$P(p) = C(p) * D(p) \quad (1)$$

Where  $C(p)$  is the confidence term and  $D(p)$  is the data term, which is calculated as per equation (2) and (3):

$$C(p) = \frac{\sum_{q \in (\Psi_p \cap \Phi)} C(q)}{|\Psi_p|} \quad (2)$$

$$D(p) = \frac{|\nabla I_p^\perp \cdot n_p|}{\alpha} \quad (3)$$

The confidence  $C(p)$  term may be thought of as a measure of the amount of reliable information surrounding the pixel  $p$  [1]. The intention is to fill first those patches which have more of their pixels already filled [1].

The data term  $D(p)$  is a function of the strength of isophotes hitting the front  $\delta\Omega$  at each iteration [1]. This term boosts the priority of a patch that an isophote “flows” into [1].

Where  $|\Psi_p|$  is area of patch centered at pixel  $p$ .  $n_p$  is unit vector orthogonal to front  $\delta\Omega$  at pixel  $p$ .  $\nabla I_p^\perp$  represents the perpendicular isophote at pixel  $p$ .  $\alpha$  is normalization factor ( $\alpha = 255$  for grey scale image).

#### 6. Select the pixel $p$ with the highest priority having patch $\Psi_p$ .

After finding priority of all the pixels on the boundary, select the pixel with the maximum priority having patch  $\Psi_p$  for further processing.

#### 7. Find the search region/regions.

Current patch ( $\Psi_p$ ) is searched for a matching patch ( $\Psi_q$ ) in the search region. The search region/regions can be calculated as:

##### 7.1. Find Window $W_p$ of the current pixel $p$ .

From the location matrix  $L$ , find a window  $W_p$  of the pixel  $p$  which is shown in fig. 1(a) and fig. 1(d).

##### 7.2. If $W_p$ does not contain any location, then a search region of size $\left[ \begin{matrix} h & w \\ n_2 & n_2 \end{matrix} \right]$ centered at pixel $p$ is selected as shown in fig. 1(b).

##### 7.3. Else if $W_p$ contain any location or multiple locations then one of the location should be selected (say $L_p$ ) and proceeded with the following steps.

i. A search region of size  $\left[ \begin{matrix} h & w \\ n_1 & n_1 \end{matrix} \right]$  centered at pixel  $p$  is selected as shown in fig. 1(e).

ii. A search region of size  $\left[ \begin{matrix} h & w \\ n_1 & n_1 \end{matrix} \right]$  centered at a pixel in the location  $L_p$  as shown in fig. 1(e).

##### 7.4. Search region should be shifted if either entire row/rows or column/columns of the search region is

empty or having a number of source region  $\Phi$  pixels less than the threshold  $\lambda$ .

Where  $h$  = height of image.

$w$  = width of image.

$n_1$  and  $n_2$  = constants,  $n_1 > n_2$  ( $n_1 \approx 2 * n_2$ ).

For example, as shown in fig. 1(c) and fig. 1(f), the left few columns of search region centered at pixel  $p$  does not contain any source region pixels, then search region is shifted, such that every rows or columns at least have few pixels or pixels greater than the threshold  $\lambda$  from the source region.

- Calculate the MSE for all the pixel in search region with respect to patch  $\Psi_p$ .

MSE is calculated as per following equation (4).

$$MSE = \sum \frac{(f_{(x,y)} - g_{(x,y)})^2}{|\Psi_p|} \quad (4)$$

Where  $f_{(x,y)}$  represents the element of the patch  $\Psi_p$  and  $g_{(x,y)}$  represents the elements of the patch  $\Psi_q$  for which MSE is to be calculated.  $|\Psi_p|$  is area of patch centered at pixel  $p$ .

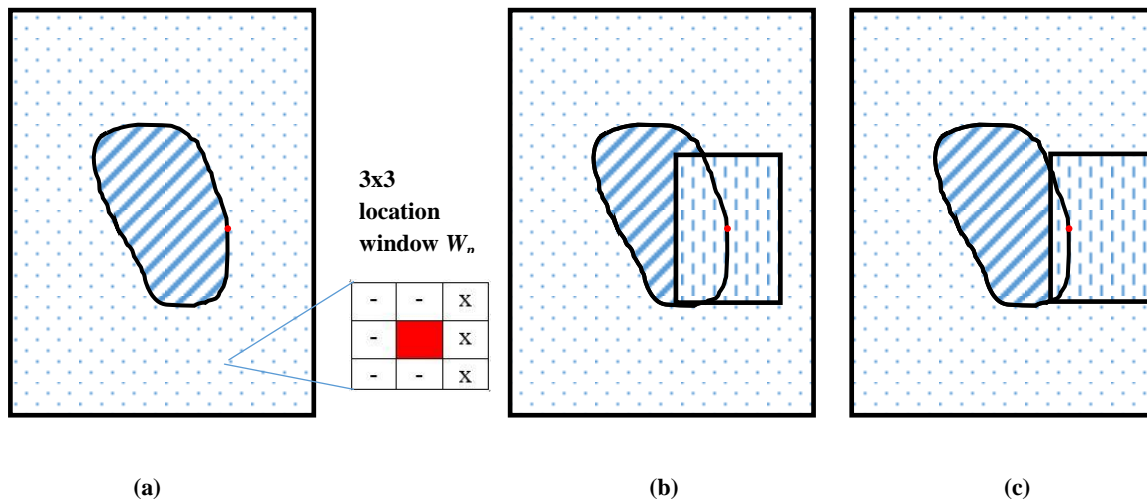
- Find the patch  $\Psi_q$  with least MSE.

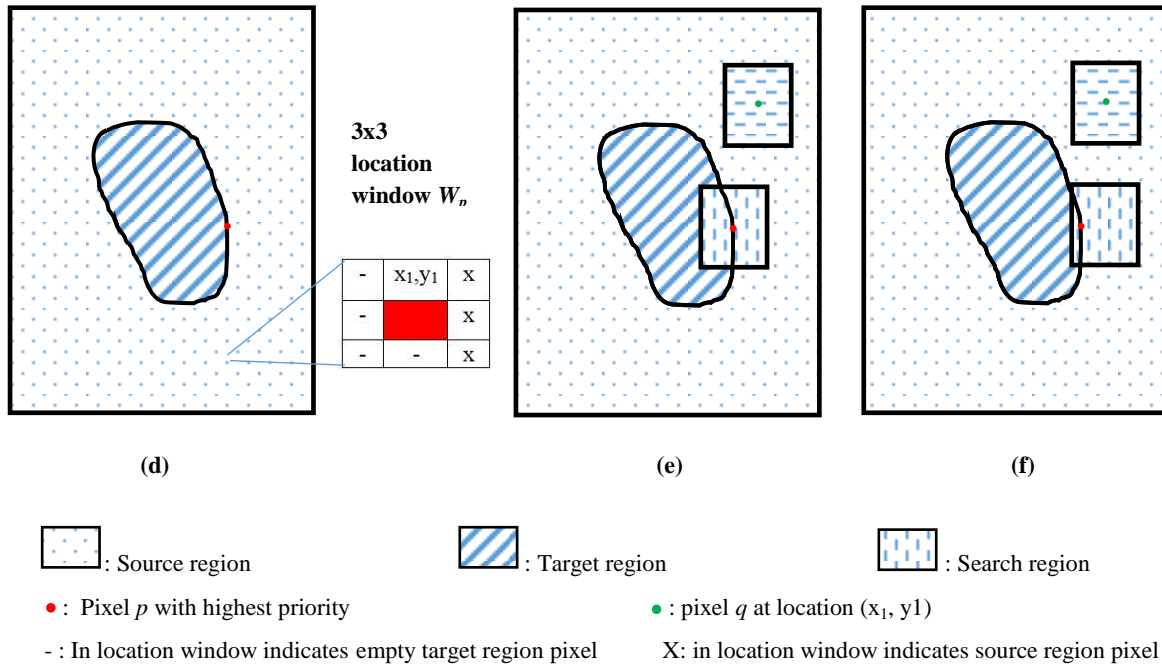
After calculating MSE of all pixels in search region, find a pixel with least MSE. If there are more than one pixel with least MSE then find mean and variance with respect to patch  $\Psi_p$  as per the equation (5) and (6), and select the patch  $\Psi_q$  with least mean and variance

$$M = \frac{\sum f_{p \in \phi \cap \Psi}}{\#\{p \mid p \in \phi \cap \Psi\}} \quad (5)$$

$$V = \frac{\sum (f_{p \in \phi \cap \Psi} - M)^2}{\#\{p \mid p \in \phi \cap \Psi\}} \quad (6)$$

Where 'f' denotes the pixel value of the element,  $\#\{..\}$  represents the cardinality of the set [2].





**Fig. 1: (a) an image with pixel  $p$  in red having empty location matrix. (b) Search region centered at pixel  $p$ . (c) Shifted search region. (d) An image with pixel  $p$  in red having non-empty location matrix. (e) Search regions centered at pixel  $p$  and pixel  $q$ . (f) Shifted search region of pixel  $p$ .**

10. Replace all the pixel from  $\Psi_p \in \Omega$  with corresponding pixels from  $\Psi_q$ .

Replace all the target region pixels of patch  $\Psi_p$  with the corresponding pixels from patch  $\Psi_q$ .

11. Update the mask, Confidence values and location matrix of corresponding filled pixel.

After replacing, update the mask as  $\text{mask}(x_p, y_p) = 1$  for all the pixels in  $\Psi_p \in \Omega$  and update the confidence value as per updated mask.

Update location matrix as  $L(x_p, y_p) = x_q, y_q$  for all the pixels  $\Psi_p \in \Omega$  with corresponding pixels from patch  $\Psi_q$

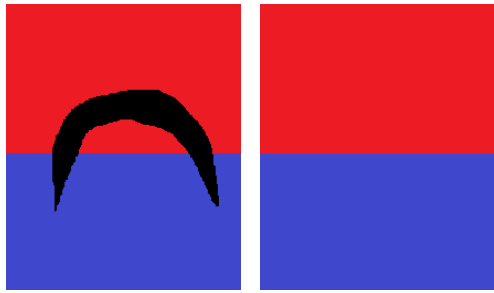
12. Repeat from step 3 to 11 until the target region  $\Phi$  become empty.

#### 4. RESULTS

Comparison of computation time between Criminisi et al [1], Anupam et al [2] and proposed approach is given in the table 1. Computation time of proposed approach depends on the parameter  $n_1$  and  $n_2$ . Higher the values of  $n_1$  and  $n_2$ , lower the computation time but resultant image will not be proper and vice versa. Comparing with the results of previous approaches, the proposed approach reduces the computation time to greater extent. The corresponding output images are given below (fig. 2 to fig. 8). The obtained resultant images does not differ much from the results of other approaches.

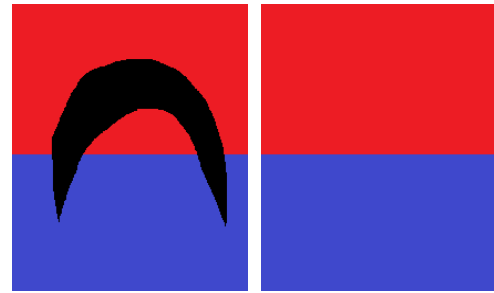
**Table 1: Comparison of computation time between Criminisi [1], Anupam [2] and proposed approach.**

Image	Total Number of pixels	Fig. No.	Target region pixel	% area	Criminisi [1]	Anupam [2]	Proposed approach	
					Time(sec)	Time(sec)	Time(sec)	$n_1, n_2$
Bicolor	57723	2	5715	9.9%	10265.5	1281.89	27.497	40,20
		3	9631	16.69%	16124.86	2885.9	45.36	40,20
Butterfly	40000	4	5166	12.9%	10245.77	570.19	30.81	20,5
College	139968	5	14672	10.48%	28441.65	2908.99	403.7	12,6
Bungee	175104	6	2129	1.22%	6750.74	118.15	22.44	20,10
Tower	121404	7	6783	5.59%	25510.09	839.25	252.49	10,5
Girl	120000	8	23428	19.52%	72461.059	5916.93	703.46	10,5



(a) (b)

**Fig. 2: (a) Bicolor image with mask in black (9.9%). (b) An inpainting result.**



(a) (b)

**Fig. 3: (a) Bicolor image with mask in black (16.69%). (b) An inpainting result.**



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

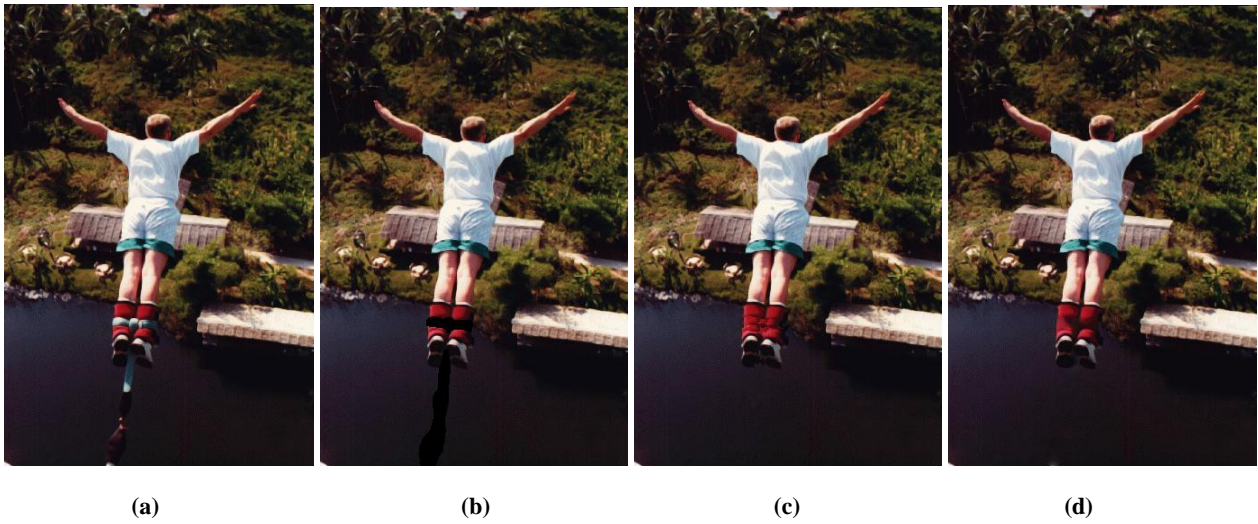
**Fig. 4: (a) Butterfly image with patch in white. (b) Butterfly image with mask in black. (c) An inpainting result using proposed approach. (d) An inpainting result using Bertalmio et al [5] approach.**



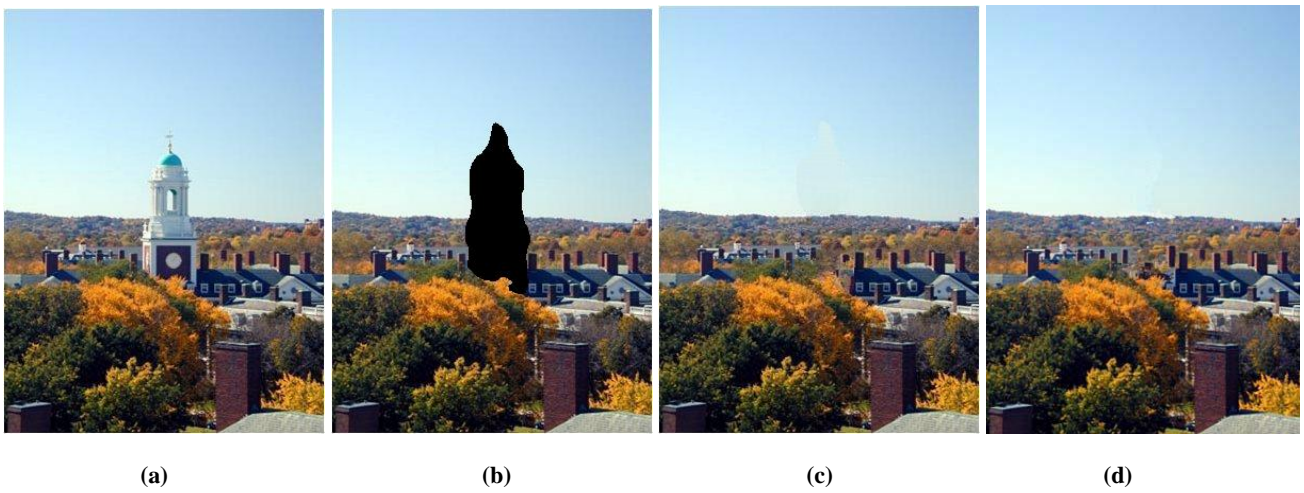
(a) (b) (c) (d)

**Fig. 5: (a) College image (b) College image with mask in black. (c) An inpainting result using proposed approach. (d) An inpainting result using Anupam et al [2] approach.**

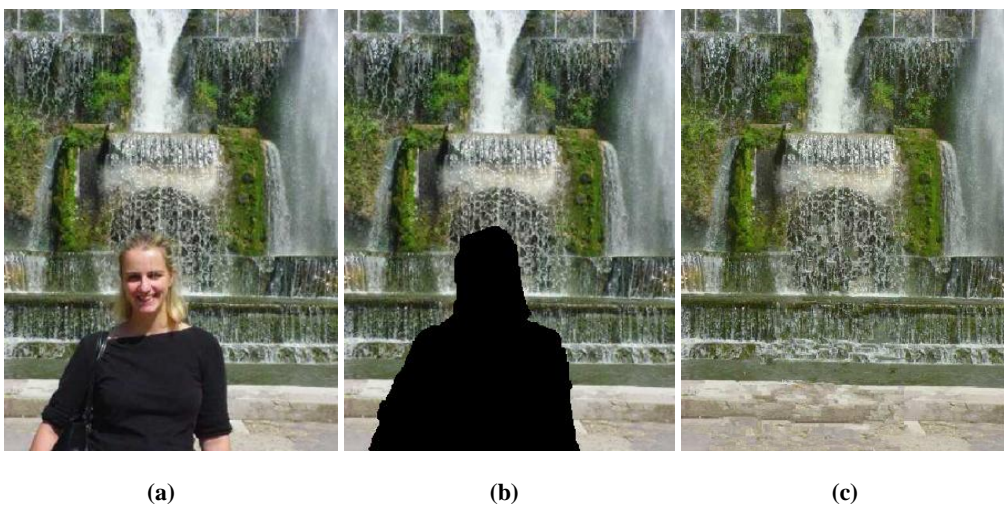




**Fig. 6: (a) Bungee image (b) Bungee image with mask in black. (c) An inpainting result using proposed approach. (d) An inpainting result using Bertalmio et al [5] approach.**



**Fig. 7: (a) Tower image (b) Tower image with mask in black. (c) An inpainting results using proposed approach. (d) An inpainting result using Anupam et al [2] approach.**



**Fig. 8: (a) Girl image (b) Girl image with mask in black. (c) An inpainting results using Bertalmio et al [5] approach.**

## 5. CONCLUSION

Image inpainting can be done using various techniques which is mainly used for object removal. In this paper, exemplar based approach is used which searches for the relevant information from the remaining part of the image. The proposed algorithm has reduced search region to greater extent and hence the computation time is also reduced. So, when the exemplar based image inpainting is applied to an image with lost part or manually selected part to be removed, image with visually plausible recovered part is obtained.

Currently this technique provides better results for only images having objects with linear geometry. The further work can be extended to improve the performance of the current system and to improve the structural quality of the object to be inpainted. Further analysis can be done on improving the current system to work on the images having objects with curved geometry.

## REFERENCES

- [1] A. Criminisi, P. Perez, and K. Toyama, "Region Filling and Object Removal by Exemplar- Based Image Inpainting," In *IEEE Transactions on Image Processing*, 13(9), pp. 1200-1212, 2004.
- [2] Anupam, P. Goyal and S. Diwakar, "Fast and Enhanced Algorithm for Exemplar Based Image Inpainting," In *Fourth Pacific-Rim Symposium on Image and Video Technology (PSIVT)*, pp. 325-330, 2010.
- [3] L. Yin and C. Chang, "An effective exemplar-based image inpainting method," In *IEEE 14th International Conference on Communication Technology (ICCT)*, pp. 739-743, 2012.
- [4] F. Cao, Y. Gousseau, S. Masnou and P. Perez, "Geometrically Guided Exemplar-Based Inpainting," *SIAM Journal on Imaging Sciences*, Volume 4 Issue 4, pp. 1143-1179, October 2011.
- [5] M. Bertalmio, V. Caselles, S. Masnou and G. Sapiro, "Inpainting," A Survey, *Institute Camille Jordan*, "math.univ-lyon1.fr/~masnou/fichiers/publications/survey.pdf".
- [6] S. Li, R. Wang, J. Xie and Y. Dong, "Exemplar Image Inpainting by Means of Curvature-Driven Method," *International Conference on Computer Science and Electronics Engineering*, pp. 326-329, 2012.
- [7] Y. Qin and F. Wang, "A Curvature Constraint Exemplar-Based Image Inpainting," *IEEE*, 2010.
- [8] T. Sarawut and A. Nishihara, "Exemplar-based image inpainting with patch shifting scheme," *17th International Conference on Digital Signal Processing*, 2011.