International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 4, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

Combining Texture Synthesis & Inpainting for unwanted object removal and image completion

Ms. Jakiya Kazi
Department of Electronics and Telecommunication
D.K.T.E's T.E.I., Shivaji University Kolhapur, India
E-mail ID : jakia_qazi@yahoo.com

Prof. Y. M. Patil
Department of Electronics
K.I.T.'s C.O.E. Shivaji University Kolhapur, India
E-mail ID : ymp2002@rediffmail.com

***Abstract* – In this paper, a novel algorithm is proposed for removal of unwanted objects from an image and image reconstruction in such a way that alteration in image is undetectable to human eye. Application of this technique includes restoration of damaged photographs and films, removal of superimposed text and removal/replacement of selected objects etc. After user selects the regions to be removed or restored, the algorithm automatically fills these regions with data sampled from reminder of the image. The proposed algorithm combines the strength of "Inpainting" and "Texture Synthesis" together to restore the Structure and texture information of the damaged image efficiently. Computational efficiency of algorithm is achieved by block based sampling process. The algorithm works well on real and synthetic images and is capable of removing large objects as well as thin scratches.**

***Keywords* – Exemplar, Inpainting, Texture synthesis, Object removal**

I. Introduction

The modification of an image in a way that is undetectable to an observer who is unknown of an original image is known as retouching or inpainting. Fig. 1 shows an example of this task, where objects in the background are manually selected as a target region and are automatically replaced by the information from the surrounding image. The algorithm fills in the holes in image which are created by removal of an object by searching for the similar patches which are also referred as "exemplars" from the nearby source region.

(a) (b)

(c)

Fig.1. (a) Original image. (b) Manual selection of objects (c) modified image.

In the past this problem has been addressed by two classes of algorithms: (i)"texture synthesis" for generating large image regions from sample textures, and (ii) "inpainting" for filling small image gaps. Before going into details let us have a brief overview of these two techniques.

II. Analysis of Texture synthesis and Inpainting

A. Texture Synthesis:

Texture synthesis is the process of algorithmically constructing a large digital image from a small digital sample image taking advantages of its structural contents. Wei and Levoy in [2] presents the texture synthesis algorithm which is based on the Markov Random Field model of texture. The output image is generated pixel by pixel in scanline order. The basic form of texture synthesis algorithm is as follows:

- The output image is initialized to random noise with a histogram equal to that of the input image.
- For each pixel in the output image, in scanline order, do:
  - In the output image , an L-shaped neighborhood of current pixel of a specific size (The size of the neighborhood should be on the scale of the largest regular texture structure to capture its low frequency components)is considered, see fig. 2
  - A search is performed in the input sample for a pixel with a neighborhood most similar to the one identified in the previous step.
  - The current pixel value in the output image is copied from the position in the input sample identified as the most similar by this search.

International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 4, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

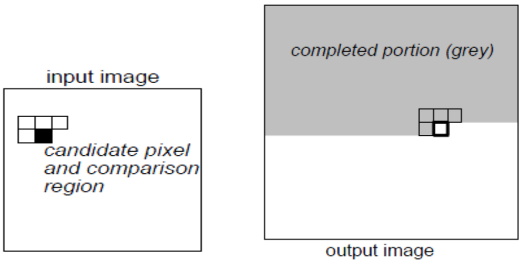
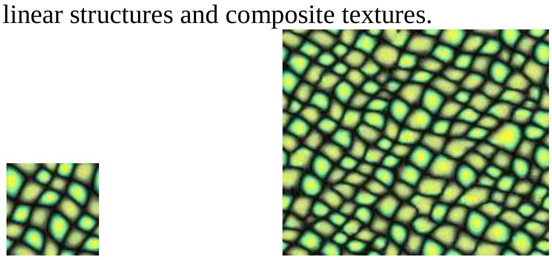


Fig.2. The Wei and Levoy (WL) texture synthesis process. Pixels are generated by scanline order.

Fig.3. shows an output image created by using texture synthesis. The simple texture synthesis algorithm mentioned above can be modified in so many ways. Wei & Levoy proposed multiresolution synthesis which allows to use smaller neighborhoods & Tree Structured Vector Quantization (TVSQ) to accelerate the search for the best matching pixel. These methods increase the speed of synthesis but introduce implementation complexity. M. Ashikhmin proposed a modified patch based algorithm in [5] which rely on visual masking to hide the seams between the patches. Pieces created by this algorithm have irregular shapes and are therefore more suitable for synthesizing natural textures.

Texture synthesis algorithm is also formulated using a statistical model, some methods use parametric models and some use non-parametric models. Though the extensive research have been done in texture synthesis these algorithms found to work well in replicating consistent textures but they have difficulty filling holes in photographs of real world scenes which often consist of linear structures and composite textures.



(a) (b)

Fig.3. Wei and Levoy texture synthesis algorithm: (a)input image (b) output image

B. Inpainting:

Inpainting is a technique used to automatically recover the damaged or missing regions in digital images or used to remove the unwanted objects from images. As mentioned above texture synthesis can fill large regions but they require user to specify what texture to put where. This is the limitation of texture synthesis approach, as the region to be inpainted may have hundreds of different backgrounds, some of them being structure not texture.

The image Inpainting algorithm simultaneously fills region surrounded by different background by propagating linear structures (called as isophotes).They are inspired by partial differential equation of physical heat flow. Inpainting technique presented by Bertalmio et al [3] is analyzed in this section.

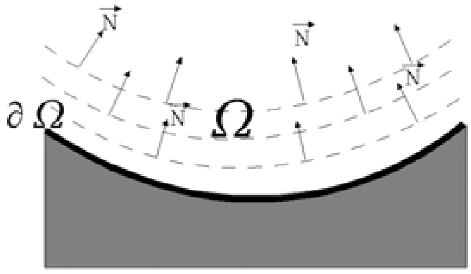


Fig.4. Propagation direction as the normal to the signed distance to the boundary of the region to be inpainted.

Bertalmio used following notations-

Ω- the region to be inpainted
∂Ω- contour/ boundary

The proposed technique prolongs the isophote lines arriving at ∂Ω , while maintaining the angle of arrival. The algorithm proceeds drawing from inward in this way, while curving the prolongation lines progressively to prevent them from crossing each other. The methodology of inpainting is as follows:

(1) The global picture determines how to fill in the gap, the purpose of inpainting being to restore the unity of the work
(2) The structure of the area surrounding Ω is continued into the gap, contour lines are drawn via the prolongation of those arriving at ∂Ω
(3) The different regions inside Ω , as defined by the contour lines, are filled with color, matching those of ∂Ω ; and
(4) The small details are painted (e.g. little white spots on an otherwise uniformly blue sky): in other words, “texture” is added.

The algorithm simultaneously and iteratively performs the steps (2) and (3) above .The gap Ω is progressively shrinked by extending inward, in a smooth way, the lines arriving at the contour ∂Ω. Fig. 5 shows an example of inpainting, where mike is removed and inpainted.

(a) (b)

Fig.5. Result of Bertalmio’s inpainting algorithm: (a) input image (b) output image

International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 4, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

III. Combining Texture Synthesis With Diffusion

Both texture synthesis and diffusion have their own advantages and drawbacks for image inpainting. Diffusion allows the continuity of contours but gives blurred results whereas Texture synthesis permits to conserve the textures but usually fails at preserving the edges and big structures. The proposed algorithm combines the advantages of these two approaches into a single efficient algorithm in exemplar based inpainting.

Previous work:

The algorithm presented here is based on research along similar lines. Bertalmio M, L. Vese and G. Sapiro in [8] presented a work that decomposes the original image into two components, one of which is processed by inpainting and the other by texture synthesis. The output image is sum of two processed components. This approach is limited to the removal of small image gaps because the diffusion process continues to blur the filled regions and also the approach avoids the automatic switching between “pure texture” and “pure structure mode”.

I.Drori, D. Cohen in [9] proposed an algorithm that used exemplar-based detail synthesis for image completion. The algorithm is extremely slow and it also can introduce blur artifacts.

Harrison in [4] proposed the first attempt to use exemplar based synthesis for object removal, where in the order of pixel filling in target region was dictated by level of texturedness of pixels neighborhood. The technique drove the fill order by the local shape of the target region, but did not seek to explicitly propagate linear structures.

Jia et al. [10] has proposed a technique based on texture segmentation step and tensor voting algorithm for the smooth linking of structures across the holes. The algorithm works well to connect curved structures by explicit generation of subjective contours, over which textural structures are propagated, but the algorithm requires expensive segmentation step and a hard decision about what constitutes a boundary between two textures.

Zalesny et al [6] describe an algorithm for parallel synthesis of composite textures. They device a special purpose solution for synthesizing the interface between two knitted textures.

A. Exemplar based Inpainting algorithm:

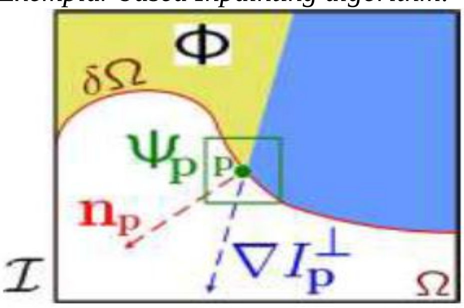


Fig.6. Notation diagram

The proposed algorithm uses isophote driven image sampling process. The algorithm takes the known image patches i.e. exemplars and propagates them into the missing regions. To handle the missing region with composite textures and structures, patch priority is defined to encourage the filling order of patches on the structure.

First, given an input image I, the user selects a target region Ω to be removed and filled. The source region Φ is defined as the entire image minus target region (I- Ω) see fig 6. which remains fixed throughout the algorithm it works as a dilated band around the target region or it may be manually specified by the user. The source region provides samples used in the filling process. The contour of the target region is denoted by δΩ which is also referred as “fill front”. The size of template window Ψ must be specified. A default window size is 9 x 9 pixels but in practice require the user to set it to be slightly larger than the largest “texel” in the source region.

Once these parameters are determined the algorithm proceeds automatically. The heart of this algorithm is priority/ patch ordering mechanism that allows exemplar based approach to handle the structural features of an input image. Priority is composed of two terms- Confidence term denoted by C(p) and the data term D(p)both are defined over pixels.

- $C(p) = (Sum_{\{q \backslash in\ \Psi p \backslash intersect\ (I\text{-}\Omega)\}} C(q)) / (area\ of\ patch)$
- $D(p) = abs(Isophote(p). Normal(p)) /alpha$

Confidence term is a measure of reliable information surrounding the pixel p. Confidence tends to decay as the centre of the fill region is approached. Confidence is used to capture the texture property but it ignores structural information in the image because of which if priority only consisted the confidence term, the patches would be selected in an “onion peel” manner which may lead to visible artifacts such as unrealistically broken structures.

The data term D(p) is the function of how strong an isophote hits the boundary/contour δΩ at each iteration. An isophote is basically the gradient at a pixel rotated by 90 degrees, it captures the "strength of flow" of an edge. It encourages the linear structures to be synthesized first and therefore propagated securely into the target region. Broken lines tend to connect, thus realizing the “connectivity principle” of vision technology. But If only the data term is used in the priority, however, edges end up propagating where they shouldn't.

The proposed algorithm computes the priority by considering both confidence term C(p) and data term D(p) in order to get good results in images containing both texture and structure information.

B. Overview of algorithm:

The algorithm iterates the following three steps until all pixels have been filled.

1) Computation of patch priorities:

The strategy of the proposed algorithm is to perform synthesis task through a best first filling that depends upon the priorities assigned to each patch on the fill front. Patches those are on the continuation of strong edges and those which are surrounded by high confidence values play an important role in priority computation.

International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 4, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

Considering a patch Ψp centered at point $_{\mathbf{P}}$ for some $_{\mathbf{P}}$ Є δΩ as shown in fig.6, the priority P($_{\mathbf{P}}$)is defined as the product of two terms

$$P(_{\mathbf{P}})=C(_{\mathbf{P}})D(_{\mathbf{P}}) \qquad (1)$$

The confidence term C($_{\mathbf{P}}$) and the data term D($_{\mathbf{P}}$) are defined as follows:

$$C(_{\mathbf{P}})= \frac{\sum q \in \Psi\mathbf{p}\cap(\mathbf{I}-\boldsymbol{\Omega})\, C(\mathbf{q})}{|\Psi\mathbf{p}|}$$

and

$$D(\mathbf{p}) = \frac{|\nabla I\, p\perp .\mathbf{n}\, p\,|}{\alpha}$$

Where |Ψp| is the area of the patch,α is the normalization factor (e.g. α=255 for a typical gray level), $\mathbf{n}_p$ is a unit vector orthogonal to front δΩ in the point $_{\mathbf{P}}$ and⊥ denotes the orthogonal operator. The priority P($_{\mathbf{P}}$) is computed for every border patch , with a distinct patches for each pixel on the boundary of the target region. During initialization the function C($_{\mathbf{P}}$) is set to C($_{\mathbf{P}}$)=0 for all pixels belonging to target region **Ω** and C($_{\mathbf{P}}$)=1 for all pixels belonging to source region (I- **Ω**)i.e. Φ.

2) Propagating Texture and Structure information:

Once all the priorities on the fill front are computed, the target patch $\Psi\hat{p}$ with highest priority is found and then it is filled with data sampled from the source region Φ. The algorithm propagates image texture by direct sampling of the source region. Then the patch from the source region which is most similar to the target patch $\Psi\hat{p}$ is searched, which is defined as follows:

$$\Psi\hat{q} = \arg\min_{\Psi_q \in \Phi} d(\Psi\hat{p}\,, \Psi q)$$

Where the distance d(Ψ $_a$, Ψ$_b$) between two generic patches Ψ $_a$ and Ψ$_b$ is simply defined as the sum of squared differences (SSD) of the already filled pixels in two patches. pixel colors are represented in CIE Lab color space.

After finding the exemplar patch $\Psi\hat{q}$, the value of each pixel to be filled is p′ | p′ Є $\Psi\hat{p}$ ∩ **Ω** is copied from its corresponding position inside $\Psi\hat{q}$ this achieves the propagation of both structure and texture information from the source region Φ to the target region **Ω**, one patch at a time.

3) Updating Confidence values:

After the target patch $\Psi\hat{p}$ is filled with new pixel values, the confidence C($_{\mathbf{P}}$) is updated in the area occupied by $\Psi\hat{p}$ as follows:

C($_{\mathbf{P}}$)=C($\hat{p}$) ∀ $_{\mathrm{P}}$ Є $\Psi\hat{p}$ ∩ **Ω**

Above equation allows to measure the relative confidence of patches on the fill front, without image specific parameters. As filling proceeds, confidence values decay, indicating that the color values of pixels near the centre of the target region are less sure.

C. Pseudo code description of region filling algorithm:

Extract manually selected initial front $\delta\Omega^0$
Repeat following steps until the region is completely filled

- Identify the fill front $\delta\Omega^t$. If $\Omega^t = \emptyset$, exit.
- Compute priorities P($_{\mathbf{P}}$) ∀ $_{\mathbf{P}}$ Є $\delta\Omega^t$
- find the patch $\Psi\hat{p}$ with maximum priority, i.e. $\hat{p}$=arg max$_{P \in \delta\Omega t}$ P(P)
- find the exemplar $\Psi\hat{q}$ Є Φ that minimizes $d(\Psi\hat{p}\,, \Psi\hat{q})$.
- Copy image data from $\Psi\hat{q}$ to $\Psi\hat{p}$ ∀ $_{\mathbf{P}}$ Є $\Psi\hat{p}$ ∩ Ω
- Update C($_{\mathbf{P}}$) ∀ $_{\mathbf{P}}$ Є $\Psi\hat{p}$ ∩ Ω

D. Implementation details:

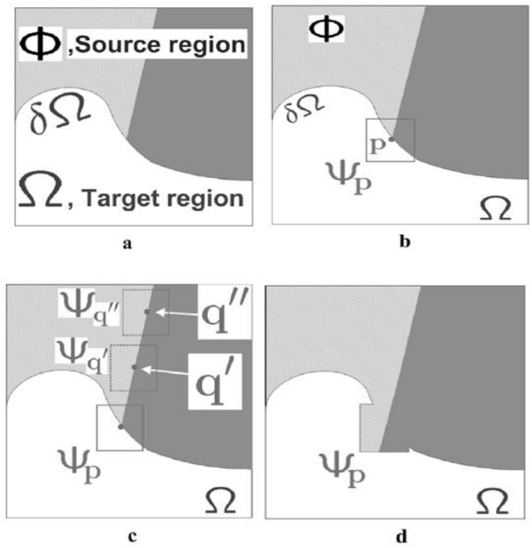


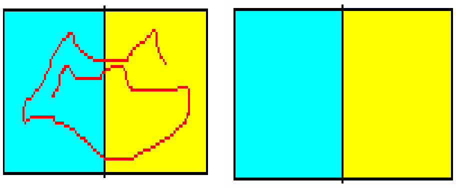
Fig.7. Structure propagation by exemplar-based texture synthesis

Now let us focus on a single iteration of this algorithm to show how structure and texture are adequately handled by exemplar based synthesis. Suppose that the square template Ψ_P ЄΩ .centered at the point P [Fig.7 (b)] is to be filled. The best-match sample from the source region comes from the patch Ψq Є Ω., which is most similar to those parts that are already filled in Ψp. In the example in Fig. 7(b), we see that if Ψp lies on the continuation of an image edge, the most likely best matches will lie along the same (or a similarly colored) edge [e.g., Ψq' and Ψq'' in Fig. 7(c)]. All that is required to propagate the isophote inwards is a simple transfer of the pattern from the best-match source patch [Fig. 7(d)].

IV. Results

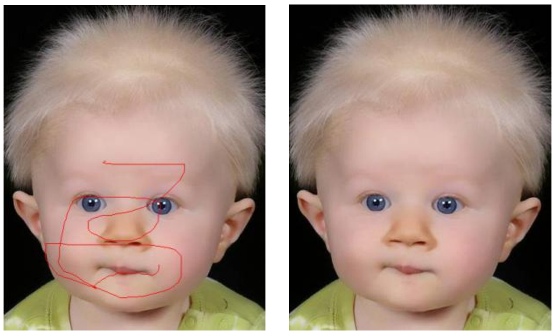
This algorithm is proposed for removing large objects from digital photographs. The result is an image in which the selected objects have been replaced by a visually plausible background that mimics the appearance of the source region. Algorithm produces promising results in images containing both structure and texture information. Algorithm runs on a 2.0 GHz core i3 with 2GB of RAM.

Figures bellow show the results of proposed exemplar based region filling algorithm with synthetic and real images.



(a) (b)

Fig.8. Proposed algorithm applied on synthetic image (a) Input image-Picture1 (b) Output image



(a) (b)

Fig.9. Proposed algorithm applied on real image to remove scratches from an image (a) Input image –Picture2(b) Output image

Fig.10. Proposed Algorithm applied on real image to remove unwanted object from an image(a)input image-picture3 (b) output image

Figures 8-10 above demonstrate the power and versatility of the algorithm i.e. the algorithm is capable of removing thin scratches to large objects from both real and synthetic images efficiently. Values of PSNR in the table bellow give the measure of reconstruction of an image.

Table I: PSNR values and Time taken for above inpainting results

Image	Total Number of pixels in original image	Total Number of pixels to fill	PSNR	Time taken to inpaint the region in seconds
Picture1 (Fig.8)	261120	291	55.929	314.54
Picture2 (Fig.9)	200598	1324	47.95	963.219
Picture3 (Fig.10)	80115	2449	22.516	137.896

V. Comparison with Few Existing Exemplar Based Inpainting Techniques

Jiying Wu and Q. Ruan in[11] proposed a cross isophotes exemplar based inpainting algorithm, in which a cross-isophotes patch priority term was designed based on the analysis of anisotropic diffusion. It can find proper exemplars to fill in the target region and preserves linear structure, it improves the peak signal to noise ratio to a small extent compared to the proposed algorithm in [1] but it still has a problem of reconstructing the curved structure in the occlusion.

Zhaolin Lu[12] proposed a PDE- based image completion algorithm in which the geometrical property of an image structure is preserved. The method introduces better patch matching scheme, which incorporates curvature and color of image. The method works a little well compared to proposed technique but still has limitations with complex textures. Proposed algorithm works almost as well as above exemplar based inpainting techniques.

VI. Conclusion

The proposed algorithm overcomes the issues that characterize the traditional inpainting and texture synthesis approach and achieves the desired properties of 1) correct propagation of linear structures, 2) robustness to changes in shape of the target region, and 3) balanced simultaneous structure and texture propagation, all in a single, efficient algorithm .It performs at least as well as previous techniques designed for the restoration of small scratches, and, in instances in which larger objects are removed, it dramatically outperforms earlier work in terms of both perceptual quality and computational efficiency.

Compared with the other traditional inpainting algorithms, the exemplar-based inpainting algorithm has performed plausible results for inpainting the large missing region. But they work well only if the missing region consists of simple structure and texture. If there are not enough samples in the image, it will be impossible to synthesize the desired image. It shows that there is still a demand for an efficient image reconstruction method.

The digital inpainting problem is still far from being completely solved. Although a large number of algorithms exist that are capable of producing amazing results, they are usually limited to images that portray certain features. Overall, the implemented restoration methods also have some limitations for which more research needs to be done.

References

[1] A. Criminisi, P. Perez, and K. Toyama, “Region filling and object removal by examplar-based image inpainting,” IEEE Trans. Image Process., vol. 13, pp. 1200–1212, 2004.

[2] L.-W. Wey and M. Levoy, “Fast texture synthesis using tree-structured vector quantization,” in *Proc. ACM Conf. Computer Graphics (SIGGRAPH)*,2000.

[3] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, “Image inpainting,” in *Proc. ACM Conf. Comp. Graphics (SIGGRAPH)*,

New Orleans, LA, July 2000, http://mountains.ece.umn.edu/~guille/inpainting. htm, pp. 417–424.

[4] P. Harrison, “A nonhierarchical procedure for re-synthesis of complex texture,” in *Proc. Int. Conf. Central Europe Computer Graphics, Visualization, and computer Vision, Plzen, Czech Republic, Feb.2001*

[5] M. Ashikhmin, “Synthesizing natural textures,” in *Proc. ACMSymp. Interactive 3D Graphics*, Mar. 2001, pp. 217–226

[6] A. Zalesny, V. Ferrari, G. Caenen, and L. van Gool, “Parallel composite texture synthesis,” in *Texture 2002 Workshop—ECCV*, Copenhagen, Denmark, June 2002.

[7] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, “Simultaneous structure and texture image inpainting,” in *Proc. Conf. Comp. Vision Pattern Rec.*,MadisonWI,2003 ,http://mountains.ece.umn.edu/~guille/inpainting.htm

[8] A. Criminisi, P. Pérez, and K. Toyama, “Object removal by exemplar based inpainting” in proc. *conf. computer Vision and Pattern Recognition*, Madison,WI, June2003.

[9] I. Drori, D. Cohen-Or, and H. Yeshurun, “Fragment-based image completion,” in *ACM Trans. Graphics (SIGGRAPH)*, vol. 22, San Diego, CA, 2003, pp. 303–312.

[10] J. Jia and C.-K. Tang, “Image repairing: Robust image synthesis by adaptive and tensor voting,” in *Proc. Conf. Computer Vision and Pattern Recognition*, Madison, WI, 2003.

[11] J.Wu and Q. Ruan, “Object removal by cross isophotes examplar based image inpainting,” in Proc.Int. Conf. Pattern Recognition, 2006 , pp.810-813

[12] Zhaolin Lu, He Huang, Leida Li, Deqiang Cheng “A Novel Hybrid Image Inpainting Model” Presented at the IEEE International Conference on Genetic and evolutionary Computing 2010

AUTHOR’S PROFILE

Jakiya Kazi

Born in Kolhapur ,India in 1983, received the Bachelor of Engineering Degree in Electronics from SUK, Kolhaur, India in the year 2005, pursuing post graduation in Electronics and Telecommunication.

Currently, she is working as Assistant Professor in D.K.T.E.’s T.E.I., Ichalkaranji, India. She has the experience of six years in this field.

Yuvaraj Patil

Born in India, received the Degree in Electronics Engineering from Shivaji University ,Kolhapur, India and Masters of Electronics and computer science Engineering from WCE, SUK, India, registered for Ph.D. in 2007.

Currently he is working as Assistant Professor in KIT, Kolhapur, India. He has teaching experience of 21 years for UG and 6 years for PG.